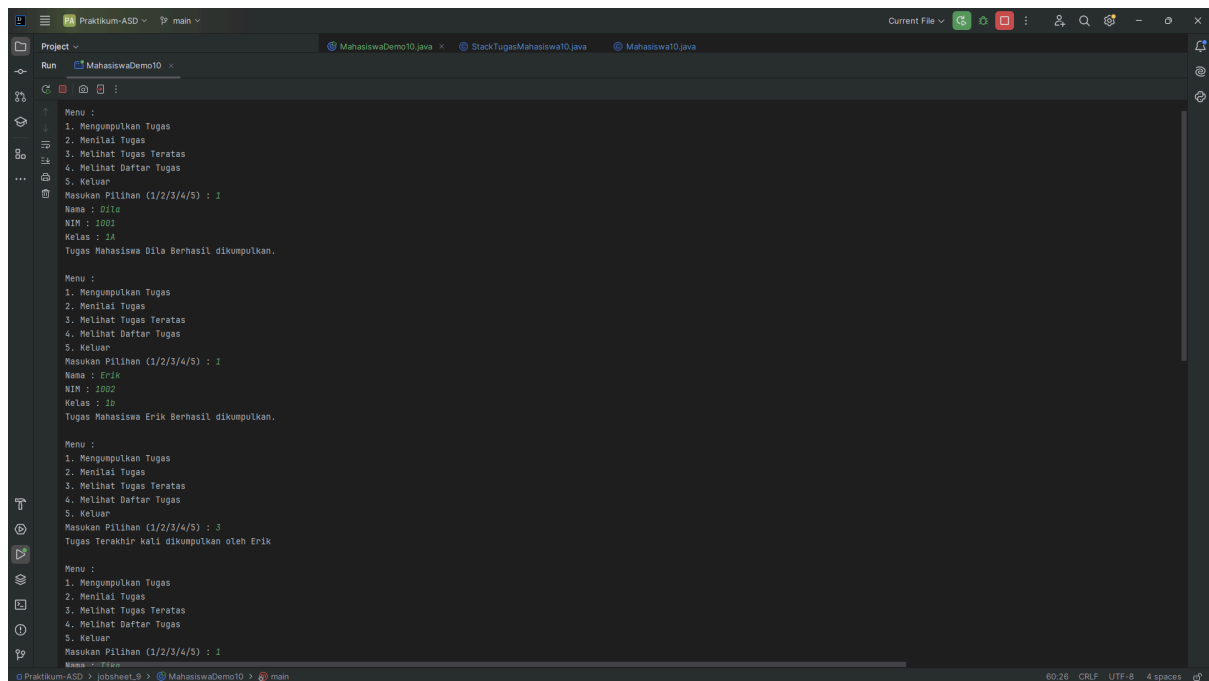


### 2.1.3 Pertanyaan

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 **lisFull()** 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**

### Jawaban no 1

Tidak ada yang perlu diganti. Semua output sudah sama



```
Run: MahasiswaDemo10
Menu :
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Keluar
Masukan Pilihan (1/2/3/4/5) : 1
Nama : Dila
NIM : 1001
Kelas : 1A
Tugas Mahasiswa Dila Berhasil dikumpulkan.

Menu :
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Keluar
Masukan Pilihan (1/2/3/4/5) : 1
Nama : Erik
NIM : 1002
Kelas : 1B
Tugas Mahasiswa Erik Berhasil dikumpulkan.

Menu :
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Keluar
Masukan Pilihan (1/2/3/4/5) : 3
Tugas Terakhir kali dikumpulkan oleh Erik

Menu :
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Keluar
Masukan Pilihan (1/2/3/4/5) : 1
Nama : Dila
```

```
Project: Praktikum-ASD - main
Run: MahasiswaDemo10

Masukkan Pilihan (1/2/3/4/5) : 1
Nama : Tika
NIM : 1003
Kelas : 1c
Tugas Mahasiswa Tika Berhasil dikumpulkan.

Menu :
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Keluar
Masukkan Pilihan (1/2/3/4/5) : 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
5. Keluar
Masukkan Pilihan (1/2/3/4/5) : 2
Menilai tugas dari Tika
Masukkan nilai (1-100) : 87
Nilai Tugas Tika adalah 87

Menu :
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Keluar
Masukkan Pilihan (1/2/3/4/5) : 4
Daftar semua tugas
Nama      NIM      Kelas
Dila      1001      1A
Erik      1002      1b
Tika      1003      1c
```

```
Project: Praktikum-ASD - main
Run: MahasiswaDemo10

5. Keluar
Masukkan Pilihan (1/2/3/4/5) : 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
5. Keluar
Masukkan Pilihan (1/2/3/4/5) : 2
Menilai tugas dari Tika
Masukkan nilai (1-100) : 87
Nilai Tugas Tika adalah 87

Menu :
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Keluar
Masukkan Pilihan (1/2/3/4/5) : 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
5. Keluar
Masukkan Pilihan (1/2/3/4/5) :
```

## Jawaban no 2

Maksimal dapat menampung 5

```
StackTugasMahasiswa10 stack = new StackTugasMahasiswa10( size: 5);
```

```

public class StackTugasMahasiswa10 { 2 usages  geraldikmal2380 *
    Mahasiswa10[] stack; 7 usages
    int top; 9 usages
    int size; 2 usages

    public StackTugasMahasiswa10(int size){ 1 usage  new *
        this.size = size;
        stack = new Mahasiswa10[size];
        top = -1;
    }
}

```

### Jawaban no 3

Akan terjadi stackoverflow.

Berikut contohnya

```

public void push(Mahasiswa10 mhs){ 1 usage  new *
    if(true){
        top++;
        stack[top] = mhs;
        System.out.println("Tugas Mahasiswa "+mhs.nama+" Berhasil dikumpulkan.");
    }
    else {
        System.out.println("Tugas Mahasiswa "+mhs.nama+" Gagal dikumpulkan.");
        System.out.println("Stack penuh! tidak bisa menambahkan tugas lagi");
    }
}
}

```

Menu :

1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Keluar

Masukan Pilihan (1/2/3/4/5) : 1

Nama : 6

NIM : 6

Kelas : 6

Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index 5 out of bounds for length 5  
 at jobsheet\_9.StackTugasMahasiswa10.push(StackTugasMahasiswa10.java:35)  
 at jobsheet\_9.MahasiswaDemo10.main(MahasiswaDemo10.java:33)

Process finished with exit code 1

### Jawaban no 4

```
do {
    System.out.println("\nMenu : ");
    System.out.println("1. Mengumpulkan Tugas");
    System.out.println("2. Menilai Tugas");
    System.out.println("3. Melihat Tugas Terbawah");
    System.out.println("4. Melihat Tugas Teratas");
    System.out.println("5. Melihat Daftar Tugas");
    System.out.println("6. Keluar");
    System.out.print("Masukan Pilihan (1/2/3/4/5) : ");
    pilihan = mlebu.nextInt();
    mlebu.nextLine();
}
```

```
case 3:
    Mahasiswa10 LihatTerbawah = stack.peekterbawah();
    System.out.println("Mahasiswa Rajin yang mengumpulkan tugas pertama adalah "+LihatTerbawah.nama);
    break;
```

```
public Mahasiswa10 peekterbawah(){ no usages new *
    if(!IsEmpty()){
        return stack[0];
    }
    else {
        System.out.println("Stack kosong tidak ada tugas yang dikumpulkan");
        return null;
    }
}
```

```

C:\Program Files\Java\jdk-23\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.3.5\lib\idea_rt.jar=58510* -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8
Menu :
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Terbawah
4. Melihat Tugas Teratas
5. Melihat Daftar Tugas
6. Keluar
Masukan Pilihan (1/2/3/4/5) : 1
Nama : Siswa1
NIM : 1001
Kelas : 1B
Tugas Mahasiswa Siswa1 Berhasil dikumpulkan.

Menu :
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Terbawah
4. Melihat Tugas Teratas
5. Melihat Daftar Tugas
6. Keluar
Masukan Pilihan (1/2/3/4/5) : 1
Nama : Siswa2
NIM : 1002
Kelas : 1C
Tugas Mahasiswa Siswa2 Berhasil dikumpulkan.

Menu :
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Terbawah
4. Melihat Tugas Teratas
5. Melihat Daftar Tugas
6. Keluar
Masukan Pilihan (1/2/3/4/5) : 3
Mahasiswa Rajin yang mengumpulkan tugas pertama adalah Siswa1

Menu :
1. Mengumpulkan Tugas
2. Menilai Tugas

```

\*Metode perulangan juga saya ganti karena metode yang ini lebih efisien di soal :

```
        default:
            System.out.println(x:"Pilihan tidak valid.");
    }
} while (pilih >= 1 && pilih <= 4);
```

saya modif menjadi :

```
public class MahasiswaDemo10 { new *
    public static void main(String[] args){ new *
        Scanner mlebu = new Scanner(System.in);
        StackTugasMahasiswa10 stack = new StackTugasMahasiswa10( size: 5);
        boolean perulangan =true;
        int pilihan;
```

```
        break,
    case 7 ://keluar dari program
        System.out.println("bye");
        perulangan = false;
        break;
```

```
    }while (perulangan);
```

Jawaban no 5

```
do {
    System.out.println("\nMenu : ");
    System.out.println("1. Mengumpulkan Tugas");
    System.out.println("2. Menilai Tugas");
    System.out.println("3. Melihat Tugas Terbawah");
    System.out.println("4. Melihat Tugas Teratas");
    System.out.println("5. Melihat Daftar Tugas");
    System.out.println("6. Berapa banyak tugas yg dikumpulkan?");
    System.out.println("7. Keluar");
    System.out.print("Masukan Pilihan (1/2/3/4/5) : ");
    pilihan = mlebu.nextInt();
    mlebu.nextLine();
```

```
case 6:
    int berapa = stack.BerapaBanyakTugasYangSudahDikumpulkan();
    System.out.println("Banyak tugas yang sudah dikumpulin adalah "+berapa);
    break;
```

```

public int BerapaBanyakTugasYangSudahDikumpulkan(){ 1 usage new *
    System.out.println();//jangan di HAPUS NGEBUG NANTI
    int hitungan=0;
    for (int i = 0; i <= top ; i++) {

        if(stack[i].nama == null && stack[i].kelas == null && stack[i].nim == null){//default value string adalah null

        }else{
            hitungan++;
        }
    }
    System.out.println();
    return hitungan;
}

```

### 2.2.3 Pertanyaan

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

#### Jawaban no 1

```

public String KonversiDesimalKeBiner(int nilai){ 1 usage new *
    StackKonversi10 stack = new StackKonversi10();
    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;
}

```

ini adalah fungsi yang mengkonversikan desimal ke biner. Cara kerjanya adalah memasukan nilai dengan parameter nilai.

```

public void push(int data){ 1 usage new *
    if(isFull()){
        System.out.println("Stack telah penuh");
    }
    else {
        top++;
        tumpukanbenir[top] = data;
    }
}

```

Cara kerjanya adalah dengan menggunakan perulangan while dan membuat variabel sisa bagi yang digunakan untuk menemukan sisa bagi dengan modulus. Setelah itu, sisa bagi akan ketemu dan akan di push ke class StackKonversi. lalu akan dimasukan ke array tumpukanbenir dan disimpan.

kembali lagi ke fungsi KonversiDesimalKeBiner, lanjut akan membuat String kosong dan melakukan perulangan untuk mengumpulkan biner agar tersusun dan di return ke DemoMahasiswa.