

## Jobsheet 9

### Praktikum 1

1. Buat repository baru di praktikum ASD lalu buat file baru dengan nama Mahasiswa(NoBAseN).java
2. Lengkapi atribut dari class Mahasiswa07 dan buat konstruktor default dan konstruktor berparameter

```
public class Mahasiswa07{
    String nama, nim, kelas;
    int nilai;

    Mahasiswa07(String nama, String nim, String kelas){
        this.nama = nama;
        this.nim = nim;
        this.kelas = kelas;
        nilai = -1;
    }

    public Mahasiswa07() {}
```

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

3. Buat method tugasDinilai()
4. Lalu buat class baru dengan nama StackTugasMahasiswa(NoAbsen).java dan lengkapi

```
public class StackTugasMahasiswa07 {

    Mahasiswa07 stack[];
    int size;
    int top;
```

atributnya

5. Tambahkan konstruktor berparameter dan set nilai awal dari top

```
public StackTugasMahasiswa07(int size) {
    this.size = size;
    stack = new Mahasiswa07[size];
    top = -1;
}
```

```

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

```

6. Lalu buat method isFull

```

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

```

7. Buat method isEmpty

8. Buat method push

```

public void push(Mahasiswa07 mhs) {
    if (!isFull()) {
        top++;
        stack[top] = mhs;
    } else {
        System.out.println(x:"Stack penuh! Tidak bisa menambahkan tugas lagi.");
    }
}

```

9. Buat method pop

```

public Mahasiswa07 pop() {
    if (!isEmpty()) {
        Mahasiswa07 m = stack[top];
        top--;
        return m;
    } else {
        System.out.println(x:"Stack kosong! Tidak ada tugas untuk dinilai.");
        return null;
    }
}

```

10. Buat method peek

```

public Mahasiswa07 peek() {
    if (!isEmpty()) {
        return stack[top];
    } else {
        System.out.println(x:"Stack kosong! Tidak ada tugas yang dikumpulkan");
        return null;
    }
}

```

11. Buat method print

```
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("x:");
}
```

12. Lalu buat class baru sebagai class utama dengan nama MahasiswaDemo(NoAbsen).java dan buat fungsi mainnya sertakan juga scanner dan instansiasi StackTugasMahasiswa07

```
import java.util.Scanner;

public class MahasiswaDemo07 {

    Run main | Debug main | Run | Debug
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        StackTugasMahasiswa07 stack = new StackTugasMahasiswa07(size:5);
        int pilih;
```

13. Buat fungsi do while

```
do {
    System.out.println(x: "\nMenu:");
    System.out.println(x: "1. Mengumpulkan Tugas");
    System.out.println(x: "2. Menilai Tugas");
    System.out.println(x: "3. Melihat Tugas Teratas");
    System.out.println(x: "4. Melihat Daftar Tugas");
    System.out.print(s: "Pilih: ");
    pilih = scan.nextInt();
    scan.nextLine();
```

14. Tambahkan switch case

```
switch (pilih) {
    case 1:
        System.out.print(s:"Nama: ");
        String nama = scan.nextLine();
        System.out.print(s:"NIM: ");
        String nim = scan.nextLine();
        System.out.print(s:"Kelas: ");
        String kelas = scan.nextLine();
        Mahasiswa07 mhs = new Mahasiswa07(nama, nim, kelas);
        stack.push(mhs);
        System.out.printf(format:"Tugas %s berhasil dikumpulkan\n", mhs.nama);
        break;
    case 2:
        Mahasiswa07 dinilai = stack.pop();
        if (dinilai != null) {
            System.out.println("Menilai tugas dari " + dinilai.nama);
            System.out.print(s:"Masukkan nilai(0-100): ");
            int nilai = scan.nextInt();
            dinilai.tugasDinilai(nilai);
            System.out.printf(format:"Nilai Tugas %s adalah %d\n", dinilai.nama, nilai);
        }
        break;
    case 3:
        Mahasiswa07 lihat = stack.peek();
        if (lihat != null) {
            System.out.println("Tugas terakhir dikumpulkan oleh " + lihat.nama);
        }
        break;
    case 4:
        System.out.println(x:"Daftar semua tugas");
        System.out.println(x:"Nama\tNIM\tKelas");
        stack.print();
        break;
    default:
        throw new AssertionError();
}
} while (pilih >= 1 && pilih <= 4);
```

### 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 **!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**

Jawaban

1. Perubahannya ada pada kondisi, update dan inisialisasi di method print class StackTugasMahasiswa07. Berikut kode dan hasil run dari kode

```
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(x:"");
}
```

```
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
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
Nama  NIM    Kelas
Erik  1002    1B
Dila  1001    1A
```

2. 5

```
import java.util.Scanner;

public class MahasiswaDemo07 {

    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        StackTugasMahasiswa07 stack = new StackTugasMahasiswa07(size:5);
```

3. Karena kalau stack tidak ada pengecekan penuh atau tidak maka akan eror saat dijalankan dan kita tetap memasukkan data ke array yang sudah penuh

```
public Mahasiswa07 peek() {
    if (!isEmpty()) {
        return stack[top];
    } else {
        System.out.println(x:"Stack kosong! Tidak ada tugas yang dikumpulkan");
        return null;
    }
}

public Mahasiswa07 peekBott() {
    if (!isEmpty()) {
        return stack[bottom];
    } else {
        System.out.println(x:"Stack kosong! Tidak ada tugas yang dikumpulkan");
        return null;
    }
}
```

4.

```
case 3:
    Mahasiswa07 lihat = stack.peek();
    Mahasiswa07 lihatBott = stack.peekBott();
    if (lihat != null) {
        System.out.println("Tugas pertama dikumpulkan oleh " + lihatBott.nama);
        System.out.println("Tugas terakhir dikumpulkan oleh " + lihat.nama);
    }
    break;
```

```
public int HitungTugas() {
    int count = 0;
    if (!isEmpty()) {
        for (int i = top; i >= 0; i--) {
            count++;
        }
    } else {
        System.out.println(x:"Stack kosong! Tidak ada tugas yang dikumpulkan");
    }
    return count;
}
```

5.

```
case 5:
    System.out.println("Jumlah tugas terkumpul: " + stack.HitungTugas());
    break;
```

## Praktikum 2

1. Buat class baru yaitu StackKonversi07 lalu tambahkan atribut dan method-method

```
public class StackKonversi07 {  
  
    int tumpukanBiner[];  
    int size, top;  
  
    public StackKonversi07() {  
        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(x:"Stack penuh!");  
        } else {  
            top++;  
            tumpukanBiner[top] = data;  
        }  
    }  
  
    public int pop() {  
        if (isEmpty()) {  
            System.out.println(x:"Stack kosong!");  
            return -1;  
        } else {  
            int data = tumpukanBiner[top];  
            top--;  
            return data;  
        }  
    }  
}
```

2. Lalu pada class StackTugasMahasiswa07 tambahkan method untuk konversi bilangan desimal

```
public String konversiDesimalKeBiner(int nilai) {  
    StackKonversi07 stack = new StackKonversi07();  
    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;  
}
```

ke biner

3. Pada MahasiswaDemo07, pada case 2 panggil method `konversiDesimalKeBiner`

```
String biner = stack.konversiDesimalKeBiner(nilai);  
System.out.println("Nilai Biner Tugas: " + biner);
```

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

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

4.

### 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

1. Pertama menginstansiasikan class `StackKonversi07` dengan nama variabel `stack`, lalu ada perulangan `while` yang mana akan melakukan push nilai sisa modulus 2 ke `stack` dan nilai tersebut akan dibagi, proses `while` dilakukan hingga nilai  $> 0$ . Lalu ada instansiasi String `biner`, dilanjutkan dengan `while` yang akan melakukan pop nilai dari `stack` ke String `biner`, `while` ini akan berhenti jika dalam `stack` sudah kosong.
2. Hasilnya sama saja seperti nilai  $> 0$ , hal ini bisa terjadi karena tipe data dari nilai adalah `int` yang mana jika sudah menyentuh nilai  $= 1$  dan dibagi menjadi 2 akan menghasilkan 0

Latihan



```

1 public class Surat07 {
2     String idSurat, namaMahasiswa, kelas;
3     char jenisIzin;
4     int durasi;
5
6     public Surat07() {
7     }
8
9     public Surat07(String idSurat, String namaMahasiswa, String kelas, char jenisIzin, int durasi) {
10         this.idSurat = idSurat;
11         this.namaMahasiswa = namaMahasiswa;
12         this.kelas = kelas;
13         this.jenisIzin = jenisIzin;
14         this.durasi = durasi;
15     }
16 }
17

```

1.

```

1 public class StackSurat07 {
2
3     Surat07 stack[];
4     int size;
5     int top;
6
7     public StackSurat07(int size) {
8         this.size = size;
9         stack = new Surat07[size];
10        top = -1;
11    }
12
13    public boolean isFull() {
14        if (top == size - 1) {
15            return true;
16        } else {
17            return false;
18        }
19    }
20
21    public boolean isEmpty() {
22        if (top == -1) {
23            return true;
24        } else {
25            return false;
26        }
27    }
28
29    public void push(Surat07 surat) {
30        if (!isFull()) {
31            top++;
32            stack[top] = surat;
33        } else {
34            System.out.println(x:"Stack penuh! Tidak bisa menambahkan surat lagi.");
35        }
36    }
37

```

```

37
38     public Surat07 pop() {
39         if (!isEmpty()) {
40             Surat07 s = stack[top];
41             top--;
42             return s;
43         } else {
44             System.out.println(x:"Stack kosong! Tidak ada surat untuk diverifikasi.");
45             return null;
46         }
47     }
48
49     public Surat07 peek() {
50         if (!isEmpty()) {
51             return stack[top];
52         } else {
53             System.out.println(x:"Stack kosong! Tidak ada surat yang dikumpulkan");
54             return null;
55         }
56     }
57
58     public Surat07 cariSurat(String namaMahasiswa) {
59         Surat07 balikan = null;
60         for (int i = top; i >= 0; i--) {
61             if(namaMahasiswa.equalsIgnoreCase(stack[i].namaMahasiswa)){
62                 balikan = stack[i];
63             } else {
64                 balikan = null;
65             }
66         }
67         return balikan;
68     }
69
70 }
71

```

```

1
2     import java.util.Scanner;
3
4     public class SuratDemo07 {
5
6         Run main | Debug main | Run | Debug
7         public static void main(String[] args) {
8             Scanner sc = new Scanner(System.in);
9             StackSurat07 stack = new StackSurat07(size:5);
10            int pilih;
11            do {
12                System.out.println(x:"\nMenu:");
13                System.out.println(x:"1. Terima Surat Izin");
14                System.out.println(x:"2. Proses Surat Izin");
15                System.out.println(x:"3. Lihat Surat Izin Terakhir");
16                System.out.println(x:"4. Cari Surat");
17                System.out.print(s:"Pilih: ");
18                pilih = sc.nextInt();
19                sc.nextLine();
20                switch (pilih) {
21

```

```

        sc.nextLine();
        switch (pilih) {
            case 1:
                System.out.print(s:"ID Surat: ");
                String idSurat = sc.nextLine();
                System.out.print(s:"Nama: ");
                String nama = sc.nextLine();
                System.out.print(s:"Kelas: ");
                String kelas = sc.nextLine();
                System.out.print(s:"Jenis Izin: ");
                char jenisIzin = sc.nextLine().charAt(index:0);
                System.out.print(s:"durasi(jam): ");
                int durasi = sc.nextInt();
                sc.nextLine();
                Surat07 surat = new Surat07(idSurat, nama, kelas, jenisIzin, durasi);
                stack.push(surat);
                System.out.printf(format:"Surat %s berhasil didata\n", surat.namaMahasiswa);
                break;
            case 2:
                Surat07 diverifikasi = stack.peek();
                if (diverifikasi != null) {
                    System.out.println("Surat izin " + diverifikasi.namaMahasiswa);
                    System.out.println(x:"ID Surat\tNama\tKelas\tJenis Surat\tDurasi");
                    System.out.println(diverifikasi.idSurat + "\t\t" + diverifikasi.namaMahasiswa + "\t" + diverifikasi.kelas
                        + "\t" + diverifikasi.jenisIzin + "\t\t" + diverifikasi.durasi);
                    System.out.print(s:"Apakah surat diverifikasi(y/n): ");
                    String verifikasi = sc.nextLine();
                    if(verifikasi.equalsIgnoreCase(anotherString:"y")){
                        diverifikasi = stack.pop();
                    } else{
                        break;
                    }
                }
                break;
            case 3:
                Surat07 lihat = stack.peek();
                if (lihat != null) {
                    System.out.println("Surat terakhir dikumpulkan oleh " + lihat.namaMahasiswa);
                }
                break;
            case 4:
                System.out.print(s:"Masukkan nama Mahasiswa yang ingin dicari: ");
                String cariNama = sc.nextLine();
                Surat07 cari = stack.cariSurat(cariNama);
                if(cari == null){
                    System.out.println("Mahasiswa " + cariNama + " tidak ditemukan");
                }else{
                    System.out.println("Surat dengan nama " + cariNama + " ditemukan");
                }
                break;
            default:
                throw new AssertionError();
        }
    }
    le (pilih >= 1 && pilih <= 4);

```