

**LAPORAN HASIL PRAKTIKUM**  
**ALGORITMA DAN STRUKTUR DASAR**  
**JOBSHEET 9**



**NAMA : DANDIKA MARTHA C.**  
**NIM : 244107020092**  
**KELAS : 1E**

**Program Studi Teknik Informatika**  
**Jurusan Teknologi Informasi**  
**Praktikum**  
**2025**

## Percobaan 1

### Code

#### - Mahasiswa07.java

```
package jobsheet_9;

public class Mahasiswa07 {
    String nim;
    String nama;
    String kelas;
    int nilai;
    public Mahasiswa07() {
    }
    public Mahasiswa07 (String nama, String nim, String kelas) {
        this.nama = nama;
        this.nim = nim;
        this.kelas = kelas;
        nilai = -1;
    }
    void tugasDinilai (int nilai) {
        this.nilai = nilai;
    }
}
```

#### - StackTugasMahasiswa07.java

```
package jobsheet_9;

public class StackTugasMahasiswa07 {
    Mahasiswa07 [] stack;
    int top;
    int size;
    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;
        }
    }
    public boolean isEmpty() {
        if (top == -1) {
            return true;
        } else {
            return false;
        }
    }
}
```

## Lanjutan StackTugasMahasiswa07.java

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

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

public Mahasiswa07 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);
    }
    System.out.println("");
}
}
```

## - MahasiswaDemo07.java

```
package jobsheet_9;
import java.util.Scanner;
public class MahasiswaDemo07 {
    public static void main(String[] args) {
        StackTugasMahasiswa07 stack = new StackTugasMahasiswa07(5);
        Scanner scan = new Scanner(System.in);
        int pilih;

        do {
            System.out.println("\nMenu:");
            System.out.println("1. Pengumpulan 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();
                    Mahasiswa07 mhs = new Mahasiswa07(nama, nim, kelas);
                    stack.push(mhs);
                    System.out.printf("Tugas %s berhasil dikumpulkan\n",
mhs.nama);
                    break;
```

## Lanjutan MahasiswaDemo07.java

```
        case 2:
            Mahasiswa07 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:
            Mahasiswa07 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);
}
```

## - Hasil

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

## Pertanyaan

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

**Jawab : Mengubah kode pada class StackTugasMahasiswa07.java di method print()**

```
public void print() {  
    for (int i = top; i >= 0; i--) {
```

## - Hasil

```
Menu:  
1. Pengumpulan 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!

**Jawab : ada 5 tugas yang dapat di tampung**

```
StackTugasMahasiswa07 stack = new StackTugasMahasiswa07(5);
```

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

**Jawab : Kondisi !isFull** digunakan untuk membatasi data yang masuk agar sesuai dengan kapasitas indeks yang di sediakan, jika kondisi if-else tersebut di hapus maka akan terjadi error **ArrayIndexOutOfBounds**

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!

**Jawab :**

**- Menambahkan method pada class StackTugasMahasiswa07.java**

```
public Mahasiswa07 peekBottom() {
    if (!isEmpty()) {
        return stack[0];
    } else {
        System.out.println("Stack kosong! Tidak ada tugas yang
dikumpulkan.");
        return null;
    }
}
```

**- Menambah Menu pada MahasiswaDemo07.java**

```
case 5:
    Mahasiswa07 terbawah = stack.peekBottom();
    if (terbawah != null) {
        System.out.println("Tugas pertama dikumpulkan oleh " +
terbawah.nama);
    }
    break;
```

**- Hasil**

```
Menu:
1. Pengumpulan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Melihat Tugas Pertama Dikumpulkan
Pilih: 1
Nama: Erik
NIM: 1002
Kelas: 1B
Tugas Erik berhasil dikumpulkan

Menu:
1. Pengumpulan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Melihat Tugas Pertama Dikumpulkan
Pilih: 5
Tugas pertama dikumpulkan oleh Dila
```



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

**Jawab :**

**- Menambahkan method pada class StackTugasMahasiswa07.java**

```
public int jumlahTugas() {  
    if (!isEmpty()) {  
        return top + 1;  
    } else {  
        return 0;  
    }  
}
```

**- Menambah Menu pada MahasiswaDemo07.java**

```
case 6:  
    System.out.println("Jumlah tugas yang sudah dikumpulkan: " +  
stack.jumlahTugas());  
    break;
```

## - Hasil

Menu:

1. Pengumpulan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Melihat Tugas Pertama Dikumpulkan
6. Lihat Jumlah Tugas

Pilih: 6

Jumlah tugas yang sudah dikumpulkan: 0

Menu:

1. Pengumpulan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Melihat Tugas Pertama Dikumpulkan
6. Lihat Jumlah Tugas

Pilih: 1

Nama: Dila

NIM: 1001

Kelas: 1A

Tugas Dila berhasil dikumpulkan

Menu:

1. Pengumpulan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Melihat Tugas Pertama Dikumpulkan
6. Lihat Jumlah Tugas

Pilih: 1

Nama: Erik

NIM: 1002

Kelas: 1B

Tugas Erik berhasil dikumpulkan

Menu:

1. Pengumpulan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Melihat Tugas Pertama Dikumpulkan
6. Lihat Jumlah Tugas

Pilih: 6

Jumlah tugas yang sudah dikumpulkan: 2

## Percobaan 2

### Code

#### - Tambahkan code pada class **StackTugasMahasiswa07.java**

```
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;
}
```

#### - Buat class **StackKonversi07.java**

```
package jobsheet_9;
public class StackKonversi07 {
    int tumpukanBiner [];
    int size;
    int top;

    public StackKonversi07() {
        this.size = 32; //asumsi 32 bit
        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;
        }
    }
}
```

**- Tambahkan code pada class MahasiswaDemo07.java**

```
case 2:
    Mahasiswa07 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.konversiDesimalkeBiner(nilai);
        System.out.println("Nilai Biner Tugas: " + biner);
    }
    break;
```

**- Hasil**

```
Menu:
1. Pengumpulan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Melihat Tugas Pertama Dikumpulkan
6. Lihat Jumlah Tugas
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`!**

**Jawab :**

1. Melakukan perulangan selama nilai lebih dari 0,
  2. Di dalam perulangan  $\text{sis} = \text{nilai} \% 2$
  3. Sisa di push ke stack
  4.  $\text{Nilai} = \text{nilai} / 2$  untuk mengurangi nilai sekaligus menentukan nilai yang akan di moduluskan.
  5. Perulangan kedua dilakukan selama stack tidak kosong
  6. Melakukan pop di stack untuk mengisi nilai biner
- 2. Pada method `konversiDesimalKeBiner`, ubah kondisi perulangan menjadi `while (kode != 0)`, bagaimana hasilnya? Jelaskan alasannya!**

**Jawab : Hasilnya tetap sama**, hal ini terjadi karena perulangan akan terus terjadi sampai **`nilai == 0`**

## Latihan

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

**Jawab :**

**- Surat07.java**

```
package jobsheet_9;

public class Surat07 {
    String idSurat;
    String namaMahasiswa;
    String kelas;
    char jenisIzin;
    int durasi;
    public Surat07() {
    }
    public Surat07(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;
    }
}
```

## - StackSurat07.java

```
package jobsheet_9;

public class StackSurat07 {
    Surat07 [] stack;
    int top;
    int size;
    public StackSurat07 (int size) {
        this.size = size;
        stack = new Surat07[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(Surat07 surat) {
        if (!isFull()) {
            top++;
            stack[top] = surat;
        } else {
            System.out.println("Stack penuh! Tidak bisa menambahkan tugas lagi.");
        }
    }
}
```

## - Lanjutan StackSurat07.java

```
public Surat07 pop() {
    if (!isEmpty()) {
        Surat07 s = stack[top];
        top--;
        return s;
    } else {
        System.out.println("Stack kosong! Tidak ada tugas untuk
dinilai.");
        return null;
    }
}

public Surat07 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 = top; i >= 0; i--) {
        System.out.println(stack[i].idSurat + "\t" +
stack[i].namaMahasiswa + "\t" + stack[i].kelas + "\t" +
stack[i].jenisIzin + "\t" + stack[i].durasi);
    }
    System.out.println("");
}
}
```



**- SuratDemo07.java**

```

package jobsheet_9;
import java.util.Scanner;
public class SuratDemo07 {
    public static void main(String[] args) {
        StackSurat07 stack = new StackSurat07(3);
        Scanner sc = new Scanner(System.in);
        int pilih;

        do {
            System.out.println("=====");
            System.out.println("                MENU UTAMA                ");
            System.out.println("=====");
            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");
            System.out.println("0. Keluar");
            System.out.println("=====");
            System.out.print("Pilih: ");
            pilih = sc.nextInt();
            sc.nextLine();
            System.out.println();
            switch (pilih) {
                case 1:
                    System.out.print("ID Surat          : ");
                    String idSurat = sc.nextLine();
                    System.out.print("Nama Mahasiswa   : ");
                    String nama = sc.nextLine();
                    System.out.print("Kelas           : ");
                    String kelas = sc.nextLine();
                    System.out.print("Jenis Izin (I/S) : ");
                    char izin = sc.next().charAt(0);
                    System.out.print("Durasi           : ");
                    int durasi = sc.nextInt();
                    sc.nextLine();
                    Surat07 surat = new Surat07(idSurat, nama, kelas,
izin, durasi);

                    stack.push(surat);
                    System.out.printf("Surat %s berhasil dikumpulkan\n",
surat.namaMahasiswa);
                    break;
                case 2:
                    Surat07 verifikasi = stack.peek();
                    if (verifikasi != null) {
                        System.out.println("Memverifikasi surat dari " +
verifikasi.namaMahasiswa);
                        System.out.print("Verifikasi? (Ya/Tidak) : ");
                        String verif = sc.nextLine();
                        if (verif.equalsIgnoreCase("ya")) {
                            System.out.printf("Surat dari %s telah
diverifikasi.\n", verifikasi.namaMahasiswa);
                            stack.pop();
                        }
                    }
                    break;
            }
        }
    }
}

```

## - Lanjutan SuratDemo07.java

```
        case 3:
            Surat07 lihat = stack.peek();
            if (lihat != null) {
                System.out.println("Surat Izin terakhir
dikumpulkan oleh " + lihat.namaMahasiswa);
            }
            break;
        case 4:
            System.out.print("Masukkan Nama Mahasiswa: ");
            String cariMhs = sc.nextLine();
            boolean ditemukan = false;
            for (int i = stack.top; i >= 0; i--) {
                if
(stack.stack[i].namaMahasiswa.equalsIgnoreCase(cariMhs)) {
                    System.out.println("Surat Ditemukan");
                    System.out.println("ID Surat\tNama
Mahasiswa\tKelas\tJenis Izin\tDurasi");
                    System.out.println(stack.stack[i].idSurat +
"\t\t" + stack.stack[i].namaMahasiswa + "\t\t" + stack.stack[i].kelas +
"\t" + stack.stack[i].jenisIzin + "\t\t" + stack.stack[i].durasi);
                    ditemukan = true;
                    break;
                }
            }
            if (!ditemukan) {
                System.out.println("Surat tidak ditemukan!");
            }
            break;
        case 0:
            System.out.println("Terimakasih!");
            break;
        default:
            System.out.println("Pilihan tidak valid");
    }
} while (pilih != 0);
sc.close();
}
```

## Hasil Latihan

```
=====
MENU UTAMA
=====
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat
0. Keluar
=====
Pilih: 1

ID Surat      : 1003
Nama Mahasiswa : Tika
Kelas        : 1C
Jenis Izin (I/S) : S
Durasi        : 2
Surat Tika berhasil dikumpulkan

=====
MENU UTAMA
=====
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat
0. Keluar
=====
Pilih: 2

Memverifikasi surat dari Tika
Verifikasi? (Ya/Tidak) : Ya
Surat dari Tika telah diverifikasi.

=====
MENU UTAMA
=====
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat
0. Keluar
=====
Pilih: 3

Surat Izin terakhir dikumpulkan oleh Erik
```

```
=====
MENU UTAMA
=====
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat
0. Keluar
=====
Pilih: 4

Masukkan Nama Mahasiswa: Erik
Surat Ditemukan
ID Surat      Nama Mahasiswa Kelas Jenis Izin Durasi
1002          Erik          1B      S          4

=====
MENU UTAMA
=====
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat
0. Keluar
=====
Pilih: 0

Terimakasih!
PS G:\DATA DIKO\POLINEMA TI\SEMESTER 2\PRAK ASD\praktikumASD>
```

