

LAPORAN HASIL PRAKTIKUM
ALGORITMA dan STRUKTUR DATA
JOBSHEET 9



Alexsa Fitria Ayu Siswoyo.

244107020020

TI - 1E

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
2025

Percobaan 1: Mahasiswa Mengumpulkan Tugas

A. Class Mahasiswa

```
package jobsheet9;

public class Mahasiswa02 {
    String nama;
    String nim;
    String kelas;
    int nilai;

    Mahasiswa02() {

    }

    Mahasiswa02(String nama, String nim, String kelas) {
        this.nama = nama;
        this.nim = nim;
        this.kelas = kelas;
        nilai = -1;
    }

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

B. Class StackTugasMahasiswa

```
package jobsheet9;

public class StackTugasMahasiswa02 {
    Mahasiswa02[] stack;
    int top;
    int size;

    public StackTugasMahasiswa02(int size) {
        this.size = size;
        stack = new Mahasiswa02[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(Mahasiswa02 mhs) {
    if (!isFull()) {
        top++;
        stack[top] = mhs;
    } else {
        System.out.println("Stack penuh! Tidak bisa
menambahkan tugas lagi.");
    }
}

public Mahasiswa02 pop() {
    if (!isEmpty()) {
        Mahasiswa02 m = stack[top];
        top--;
        return m;
    } else {
        System.out.println("Stack kosong! Tidak ada
tugas untuk dinilai.");
        return null;
    }
}

public Mahasiswa02 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("");
}
}

```



```

        case 3:
            Mahasiswa02 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);

    scan.close();
}
}

```

D. Outputnya

```

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 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
Dila      1001      1A
Erik      1002      1B
Tika      1003      1C
```

E. Pertanyaan

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

```
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("");
}
```

```

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!

```
StackTugasMahasiswa02 stack = new StackTugasMahasiswa02(5);
```

3. Mengapa perlu pengecekan kondisi `!isFull()` pada method `push`? Kalau kondisi `if-else` tersebut dihapus, apa dampaknya?
- Karena dengan pengecekan kondisi `!isFull()` pada method `push` sangat penting karena mencegah program eror saat stack penuh
 - Ketika kondisi `if-else` dihapus akan menyebabkan kesalahan saat runtime.
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!
- Class `StackTugasMahasiswa02.java`

```

public Mahasiswa02 peek() {
    if (!isEmpty()) {
        return stack[0];
    } else {
        System.out.println("Stack kosong! Tidak ada
tugas yang dikumpulkan");
        return null;
    }
}

```

- Class `MahasiswaDemo02.java`

```

Mahasiswa02 firstStudent = stack.peek();
if (firstStudent != null) {
    System.out.println("Mahasiswa pertama yang
mengumpulkan tugas:");
    System.out.println(firstStudent.nama + "\t" +
firstStudent.nim + "\t" + firstStudent.kelas);
}
break;

```

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

- Class StackTugasMahasiswa02.java

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

- Class MahasiswaDemo02.java

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

6. Commit dan push kode program ke Github

Percobaan 2: Konversi Nilai Tugas ke Biner

A. Class StackTugasMahasiswa

```
public String konversiDesimalKeBiner(int nilai) {
    StackKonversi02 stack = new StackKonversi02();
    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. Class StackKonversi

```
package jobsheet9;

public class StackKonversi02 {
    int[] tumpukanBiner;
    int size;
    int top;
```



```

public StackKonversi02() {
    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;
    }
}
}

```

C. Class MahasiswaDemo

```

case 2:
    Mahasiswa02 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;

```

D. Outputnya

```
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
Nilai Biner Tugas: 1010111
```

E. Pertanyaan

1. Jelaskan alur kerja dari method konversiDesimalKeBiner!
 - Proses konversi bilangan desimal ke biner dilakukan melalui pembagian berulang dengan angka 2, di mana setiap sisa pembagiannya disimpan dalam sebuah stack.
2. Pada method konversiDesimalKeBiner, ubah kondisi perulangan menjadi while (kode != 0), bagaimana hasilnya? Jelaskan alasannya!

```
public String konversiDesimalKeBiner(int nilai) {
    StackKonversi02 stack = new StackKonversi02();
    while (nilai != 0) {
        int sisa = nilai % 2;
        stack.push(sisa);
        nilai = nilai / 2;
    }
}
```

```
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
Nilai Biner Tugas: 1010111
```

- Hasilnya akan tetap sama.
- Jika kode diupdate dengan benar di dalam perulangan, maka hasilnya tetap sama dengan menggunakan (while > 0). Namun, jika kode tidak didefinisikan dengan benar, maka perulangan bisa menjadi perulangan yang tak berujung.

Latihan Praktikum

A. Class Surat

```
package jobsheet9;

public class Surat02 {
    String idSurat;
    String namaMahasiswa;
    String kelas;
    char jenisIzin;
    int durasi;

    Surat02() {
    }

    Surat02(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;
    }
}
```

B. Class StackSurat

```
package jobsheet9;

public class StackSurat02 {
    Surat02[] stack;
    int size;
    int top;

    public StackSurat02(int size) {
        this.size = size;
        stack = new Surat02[size];
        top = -1;
    }

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

```

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

public void push(Surat02 surat) {
    if (!isFull()) {
        top++;
        stack[top] = surat;
    } else {
        System.out.println("Stack penuh! Tidak bisa
menambahkan surat izin lagi.");
    }
}

public Surat02 pop() {
    if (!isEmpty()) {
        Surat02 surat = stack[top];
        top--;
        return surat;
    } else {
        System.out.println("Stack kosong! Tidak ada
surat izin yang dapat diproses.");
        return null;
    }
}

public Surat02 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 = 0; i <= top; i++) {
        if
(stack[i].namaMahasiswa.equalsIgnoreCase(namaMahasiswa)) {
            return true;
        }
    }
    return false;
}
}

```

C. Class SuratDemo

```
package jobsheet9;

import java.util.Scanner;

public class SuratDemo02 {
    public static void main(String[] args) {
        StackSurat02 stack = new StackSurat02(5);
        Scanner scan = 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");
            System.out.print("Pilih : ");
            pilih = scan.nextInt();
            scan.nextLine();
            switch (pilih) {
                case 1:
                    if (stack.isFull()) {
                        System.out.println("Stack sudah
penuh! Silakan nilai atau hapus surat sebelum menambahkan
surat baru.");
                    } else {
                        System.out.print("ID Surat\t : ");
                        String idSurat = scan.nextLine();
                        System.out.print("Nama Mahasiswa\t :
");
                        String namaMahasiswa =
scan.nextLine();
                        System.out.print("Kelas\t\t : ");
                        String kelas = scan.nextLine();
                        System.out.print("Jenis Izin (S/I) :
");
                        char jenisIzin =
scan.nextLine().charAt(0);
                        System.out.print("Durasi Izin\t :
");
                        int durasi = scan.nextInt();
                        Surat02 srt = new Surat02(idSurat,
namaMahasiswa, kelas, jenisIzin, durasi);
                        stack.push(srt);
                        System.out.printf("Surat izin dari
%s berhasil diterima\n", namaMahasiswa);
                    }
                    break;
            }
        } while (true);
    }
}
```

```

        case 2:
            Surat02 prosesSurat = stack.pop();
            if (prosesSurat != null) {
                System.out.println("Surat izin dari
" + prosesSurat.namaMahasiswa + " berhasil diproses");
            }
            break;

        case 3:
            Surat02 suratTerakhir = stack.peek();
            if (suratTerakhir != null) {
                System.out.println("Surat izin
terakhir: ");
                System.out.println("Id Surat\t : " +
suratTerakhir.idSurat);
                System.out.println("Nama Mahasiswa\t
: " + suratTerakhir.namaMahasiswa);
                System.out.println("Keterangan\t : "
+ (suratTerakhir.jenisIzin == 'S' ? "Sakit" : "Izin
Keperluan Lain"));
                System.out.println("Durasi\t : " +
suratTerakhir.durasi + " hari");
            }
            break;

        case 4:
            System.out.print("Masukkan nama
mahasiswa untuk mencari surat : ");
            String cariNama = scan.nextLine();
            boolean ditemukan =
stack.search(cariNama);
            if (ditemukan) {
                System.out.printf("Surat izin untuk
%s ditemukan.\n", cariNama);
            } else {
                System.out.printf("Surat izin untuk
%s tidak ditemukan.\n", cariNama);
            }
            break;
            default:
                System.out.println("Pilihan tidak
valid.");
            }
        } while (pilih >= 1 && pilih <= 5);
        scan.close();
    }
}

```

D. Outputnya

```
Menu:
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat
Pilih : 1
ID Surat      : TI123
Nama Mahasiswa : Alexsa
Kelas        : TI-IE
Jenis Izin (S/I) : S
Durasi Izin   : 2
Surat izin dari Alexsa berhasil diterima
```

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

```
Menu:
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat
Pilih : 3
Surat izin terakhir:
Id Surat      : TI123
Nama Mahasiswa : Alexsa
Keterangan    : Sakit
Durasi       : 2 hari
```

```
Menu:
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat
Pilih : 4
Masukkan nama mahasiswa untuk mencari surat : Alexsa
Surat izin untuk Alexsa ditemukan.
```