

**JOBSHEET 9**  
**ALGORITMA STRUKTUR**  
**DATA**



**Burhnauddin ihsan**  
**244107020189**  
**TI 1E/06**

**PROGRAM STUDI D-IV TEKNIK INFORMATIKA JURUSAN**  
**TEKNOLOGI INFORMASI**  
**POLITEKNIK NEGERI MALANG**  
**2025**

## PERCOBAAN 1

### 1. Kode program

#### a. Mahasiswa06

```
public class Mahasiswa6 {
    String nim,nama,kelas;
    int nilai;

    Mahasiswa6(){};
    Mahasiswa6(String nama, String nim, String kelas){
        this.nim = nim;
        this.nama = nama;
        this.kelas = kelas;
        nilai = -1;
    }

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

#### b. TugasMahasiswa6

```
public class StackTugasMahasiswa6 {
    Mahasiswa6[] stack;
    int size,top;

    StackTugasMahasiswa6(int size){
        this.size = size;
        stack = new Mahasiswa6[size];
        top = -1;
    }

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

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

    void push (Mahasiswa6 mhs){
        if (!isFull()) stack[++top] = mhs;
        else System.out.println("Stack Penuh! Tidak bisa menambahkan
tugas lagi.");
    }

    Mahasiswa6 pop(){
        if(top != -1){
            Mahasiswa6 m = stack[top];
            top--;
            return m;
        }else {
            System.out.println("Stack kosong! Tidak ada tugas untuk
dinilai.");
            return null;
        }
    }
}
```



```

break;

        case 3:
            Mahasiswa6 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 \t NIM \t Kelas");
            stack.print();
            break;
        default:
            System.out.println("Pilihan tidak valid");
            break;
    }
} while (pilih > 0 && pilih < 7);
}
}

```

## 2. Hasil dari kode program

```

Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
Pilih: 1
Nama: burhan
NIM: 2441
Kelas: 1e
Tugas burhan berhasil dikumpulkan

Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
Pilih: 3
Tugas terakhir dikumpulkan oleh burhan

Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
Pilih: 1
Nama: ihsan
NIM: 2331
Kelas: 1e
Tugas ihsan berhasil dikumpulkan

Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
Pilih: 2
Menilai tugas dari ihsan
Masukkan nilai (0-100): 98
Nilai tugas ihsan adalah 98

```

```

Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
Pilih: 1
Nama: udin
NIM: 1234
Kelas: 1e
Tugas udin 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
udin    1234    1e
burhan  2441    1e

```

## PERTANYAAN

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

```
void print(){
    for (int i = top; i >=0 ; i--) {
        System.out.printf(format:"%s \t %s \t %s",stack[i].nama,stack[i].nim,stack[i].kelas);
        System.out.println();
    }
    System.out.println();
}
```

Hasil dari kode program

```
Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
Pilih: 1
Nama: burhan
NIM: 1231
Kelas: 1e
Tugas burhan berhasil dikumpulkan

Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
Pilih: 1
Nama: udin
NIM: 12341
Kelas: 1e
Tugas udin berhasil dikumpulkan

Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
Pilih: 3
Tugas terakhir dikumpulkan oleh udin

Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
Pilih: 1
Nama: ihsan
NIM: 2314
Kelas: 1e
Tugas ihsan 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
ihsan     2314     1e
udin      12341    1e
burhan    1231     1e

Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
Pilih: 2
Menilai tugas dari ihsan
Masukkan nilai (0-100): 87
Nilai tugas ihsan 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
udin      12341    1e
burhan    1231     1e

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

2. Pada stack tersebut, dapat menampung maksimal 5 data tugas mahasiswa karena pada demo class, stack diinstansiasikan dengan size 5 menggunakan konstruktor berparameter
  - Potongan kode program MahasiswaDemo6

```
public class MahasiswaDemo6 {
    Run | Debug | Run main | Debug main | Windsurf: Refactor | Explain | Generate Javadoc | X
    public static void main(String[] args) {
        StackTugasMahasiswa6 stack = new StackTugasMahasiswa6(size:5);
    }
}
```

3. Karena jika tidak ada kondisi tersebut, akan menyebabkan error (index out of bounds) apabila dalam suatu kasus sebuah array telah terisi penuh dan tetap dipaksa untuk mengisi elemen pada array tsb
4. modifikasi class MahasiswaDemo6 dan StackTugasMahasiswa6 agar pengguna dapat melihat mahasiswa yang pertama kali mengumpulkan tugas melalui menu lihat tugas terbawah
  - modifikasi class StackTugasMahasiswa6

```
Mahasiswa6 peekBottom() {
    if (!isEmpty()) {
        return stack[0];
    } else {
        System.out.println(x:"Stack kosong! Tidak ada tugas yang dikumpulkan");
        return null;
    }
}
```

- modifikasi class MahasiswaDemo6

```
case 5:
    Mahasiswa6 lihatTerbawah = stack.peekBottom();
    if (lihatTerbawah != null) {
        System.out.println("Tugas pertama dikumpulkan oleh " + lihatTerbawah.nama);
    }
    break;
default:
```

Hasil dari kode program

```
Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Melihat Tugas Terbawah
Pilih: 1
Nama: burhan
NIM: 1321
Kelas: 1e
Tugas burhan berhasil dikumpulkan

Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Melihat Tugas Terbawah
Pilih: 1
Nama: udin
NIM: 1241
Kelas: 1e
Tugas udin berhasil dikumpulkan

Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Melihat Tugas Terbawah
Pilih: 1
Nama: ihsan
NIM: 12341
Kelas: 1e
Tugas ihsan berhasil dikumpulkan

Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Melihat Tugas Terbawah
Pilih: 5
Tugas pertama dikumpulkan oleh burhan
```

5. Tambah method menghitung jumlah tugas

- Class StackMahasiswa

```
Windsurf: Refactor | Explain | Generate Java
int jumlahTugas() {
    if (!isEmpty()) {
        return top + 1;
    } else {
        return 0;
    }
}
```

- Class MahasiswaDemo6

```
        break;
    case 6:
        int jumlahTugas = stack.jumlahTugas();
        System.out.println("jumlah tugas yang telah dikumpulkan : " + jumlahTugas);
        break;
    default:
        System.out.println(x:"Pilihan tidak valid");
        break;
```

Hasil dari kode program

```
Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Melihat Tugas Terbawah
6. Jumlah tugas yang dikumpulkan
Pilih: 1
Nama: burhan
NIM: 121
Kelas: 1e
Tugas burhan berhasil dikumpulkan

Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Melihat Tugas Terbawah
6. Jumlah tugas yang dikumpulkan
Pilih: 1
Nama: udin
NIM: 1231
Kelas: 1e
Tugas udin berhasil dikumpulkan

Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Melihat Tugas Terbawah
6. Jumlah tugas yang dikumpulkan
Pilih: 6
jumlah tugas yang telah dikumpulkan : 2

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

```
Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Melihat Tugas Terbawah
6. Jumlah tugas yang dikumpulkan
Pilih: 6
jumlah tugas yang telah dikumpulkan : 1
```

```
Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Melihat Tugas Terbawah
6. Jumlah tugas yang dikumpulkan
Pilih: 2
Menilai tugas dari burhan
Masukkan nilai (0-100): 98
Masukkan nilai (0-100): 98
Nilai tugas burhan adalah 98
```

```
Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Melihat Tugas Terbawah
6. Jumlah tugas yang dikumpulkan
Pilih: 6
jumlah tugas yang telah dikumpulkan : 0
```

```
Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Melihat Tugas Terbawah
6. Jumlah tugas yang dikumpulkan
Pilih:
```

Nilai tugas burhan adalah 98

```
Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Melihat Tugas Terbawah
6. Jumlah tugas yang dikumpulkan
Pilih: 6
jumlah tugas yang telah dikumpulkan : 0
```



## PERCOBAAN 2

1. Menambahkan method konversiDesimalKeBiner dengan menerima parameter dengan tipe int

```
String konversiDesimalKeBiner(int nilai) {
    StackKonversi6 stack = new StackKonversi6();
    while (nilai > 0) {
        int sisa = nilai % 2;
        stack.push(sisa);
        nilai /= 2;
    }

    String biner = new String();
    while (!stack.isEmpty()) {
        biner += stack.pop();
    }
    return biner;
}
```

2. Membuat file baru dengan nama StackKonversi11.java untuk memisahkan stack mahasiswa

```
public class StackKonversi6 {
    int[] tumpukanBiner;
    int size;
    int top;

    public StackKonversi6() {
        this.size = 32;
        tumpukanBiner = new int[size];
        top = -1;
    }

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

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

    void push(int biner) {
        if (!isFull()) {
            top++;
            tumpukanBiner[top] = biner;
        } else {
            System.out.println("stack penuh! tidak bisa menambahkan tugas lagi!");
        }
    }

    int pop() {
        if (isEmpty()) {
            System.out.println("stack kosong!");
            return -1;
        } else {
            int biner = tumpukanBiner[top];
            top--;
            return biner;
        }
    }
}
```

3. Menambahkan method konversiDesimalKeBiner untuk mengonversi nilai setelah melakukan penilaian tugas pada method pop di class MahasiswaDemo11

```
case 2:
    Mahasiswa6 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;
```

4. Hasil dari kode program

```
Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Melihat Tugas Terbawah
6. Jumlah tugas yang dikumpulkan
Pilih: 1
Nama: burhan
NIM: 1231
Kelas: 1e
Tugas burhan berhasil dikumpulkan

Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Melihat Tugas Terbawah
6. Jumlah tugas yang dikumpulkan
Pilih: 1
Nama: udin
NIM: 1231
Kelas: 1e
Tugas udin berhasil dikumpulkan

Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Melihat Tugas Terbawah
6. Jumlah tugas yang dikumpulkan
Pilih: 1
Nama: ihsan
NIM: 1231
Kelas: 1e
Tugas ihsan berhasil dikumpulkan

Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Melihat Tugas Terbawah
6. Jumlah tugas yang dikumpulkan
Pilih: 2
Menilai tugas dari ihsan
Masukkan nilai (0-100): 89
Nilai tugas ihsan adalah89
Nilai Biner Tugas: 1011001
```

## PERTANYAAN

1. Pertama, sebuah objek StackKonversi dibuat untuk menyimpan sisa hasil pembagian bilangan desimal dengan 2. Selama nilai lebih besar dari 0, program akan menghitung sisa pembagian nilai % 2, lalu menyimpannya ke dalam stack menggunakan stack.push(sisa). Nilai kemudian dibagi 2 menggunakan nilai /= 2 untuk melanjutkan proses konversi. Setelah seluruh sisa pembagian disimpan dalam stack, program akan membentuk string biner dari data yang ada di stack. Ini dilakukan dengan cara mengambil satu per satu elemen dari stack menggunakan stack.pop() dan menambahkannya ke variabel biner. Karena stack bersifat LIFO (Last In First Out), hasil konversi yang terbentuk memiliki urutan biner yang benar. Setelah stack kosong, string biner yang terbentuk dikembalikan sebagai hasil konversi.
2. Mengubah kondisi perulangan menjadi while(nilai !=0)

```
String konversiDesimalKeBiner(int nilai) {  
    StackKonversi6 stack = new StackKonversi6();  
    while (nilai > 0) {  
        int sisa = nilai % 2;  
        stack.push(sisa);  
        nilai /= 2;  
    }  
  
    String biner = new String();  
    while (!stack.isEmpty()) {  
        biner += stack.pop();  
    }  
    return biner;  
}
```

Hasil dari kode program

```
Menu:  
1. Mengumpulkan Tugas  
2. Menilai Tugas  
3. Melihat Tugas Teratas  
4. Melihat Daftar Tugas  
5. Melihat Tugas Terbawah  
6. Jumlah tugas yang dikumpulkan  
Pilih: 1  
Nama: burhan  
NIM: 12321  
Kelas: 1e  
Tugas burhan berhasil dikumpulkan  
  
Menu:  
1. Mengumpulkan Tugas  
2. Menilai Tugas  
3. Melihat Tugas Teratas  
4. Melihat Daftar Tugas  
5. Melihat Tugas Terbawah  
6. Jumlah tugas yang dikumpulkan  
Pilih: 1  
Nama: udin  
NIM: 1231  
Kelas: 1e  
Tugas udin berhasil dikumpulkan  
  
Menu:  
1. Mengumpulkan Tugas  
2. Menilai Tugas  
3. Melihat Tugas Teratas  
4. Melihat Daftar Tugas  
5. Melihat Tugas Terbawah  
6. Jumlah tugas yang dikumpulkan  
Pilih: 2  
Menilai tugas dari udin  
Masukkan nilai (0-100): 84  
Nilai tugas udin adalah 84  
Nilai Biner Tugas: 1010100
```

input masih bisa dijalankan dan tetap sesuai logika, tetapi jika nilai yang diinputkan adalah negatif maka akan terjadi kemungkinan infinite loop atau hasil yang tidak valid

## LATIHAN

### 1. Kode program

#### - Surat6

```
public class Surat6 {
    String idSurat, namaMhs, kelas;
    char jenisIzin;
    int durasi;
    boolean isVerificated;

    Surat6() {

    }

    Surat6(String idSurat, String namaMhs, String kelas, char jenisIzin,
int durasi) {
        this.idSurat = idSurat;
        this.namaMhs = namaMhs;
        this.kelas = kelas;
        this.jenisIzin = jenisIzin;
        this.durasi = durasi;
        this.isVerificated = false;
    }
}
```

#### - StackSurat6

```
public class StackSurat6 {
    Surat6[] stack;
    int size;
    int top;

    StackSurat6(int size) {
        this.size = size;
        stack = new Surat6[size];
        top = -1;
    }

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

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

    void push(Surat6 s) {
        if (!isFull()) {
            top++;
            stack[top] = s;
        } else {
            System.out.println("stack penuh!");
        }
    }
}
```

```

Surat6 pop() {
    if(!isEmpty()) {
        Surat6 s = stack[top];
        top--;
        return s;
    } else {
        System.out.println("stack kosong!");
        return null;
    }
}

Surat6 peek() {
    if (!isEmpty()) {
        return stack[top];
    } else {
        System.out.println("Stack kosong!");
        return null;
    }
}

int cariSurat(String target) {
    for(int i = 0; i <= top; i++) {
        if (stack[i].namaMhs.equalsIgnoreCase(target)) {
            return i;
        }
    }
    return -1;
}

void tampilkanSurat(int i) {
    System.out.println("Surat milik " + stack[i].namaMhs + " ditemukan
pada tumpukan ke " + (top+1-i));
}
}

```

- SuratDemo6

```

import java.util.Scanner;

public class SuratDemo6 {
    public static void main(String[] args) {
        StackSurat6 stack = new StackSurat6(5);
        Scanner sc = new Scanner(System.in);
        int pilih;

        do {
            System.out.println("Menu:");
            System.out.println("1. Menerima 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.print("Pilih: ");
            pilih = sc.nextInt();
            char jenisIzin;
            sc.nextLine();
            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();
do {
    System.out.print("Jenis Izin (S atau I): ");
    jenisIzin = sc.next().charAt(0);
} while (jenisIzin != 'S' && jenisIzin != 'I');
System.out.print("Durasi: ");
int durasi = sc.nextInt();
Surat6 surat = new Surat6(idSurat, nama, kelas, jenisIzin,
durasi);
    stack.push(surat);
    System.out.printf("Surat %s berhasil dikumpulkan\n",
surat.namaMhs);
    break;
case 2:
    Surat6 suratVerif = stack.pop();
    if (suratVerif != null) {
        System.out.println("Verifikasi Surat milik " +
suratVerif.namaMhs);
        suratVerif.isVerificated = true;
        System.out.printf("Status Surat milik %s : %b\n",
suratVerif.namaMhs, suratVerif.isVerificated);
    }
    break;
case 3:
    Surat6 suratTeratas = stack.peek();
    if (suratTeratas != null) {
        System.out.println("Surat teratas milik : " +
suratTeratas.namaMhs);
    }
    break;
case 4:
    System.out.print("Masukkan nama surat dengan nama mahasiswa yang
dicari : ");
    String namaMhs = sc.next();
    int index = stack.cariSurat(namaMhs);
    stack.tampilkanSurat(index);
    break;
case 6:
    break;
default:
    System.out.println("Pilihan tidak valid.");
}
} while (pilih >= 1 && pilih <= 4);

}

}

```

2. Hasil dari kode program

```
Menu:
1. Menerima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat berdasarkan nama mahasiswa
Pilih: 1
ID Surat: 123
nama mahasiswa: burhan
Kelas: 1e
Jenis Izin (S atau I): S
Durasi: 90
Surat burhan berhasil dikumpulkan
Menu:
1. Menerima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat berdasarkan nama mahasiswa
Pilih: 1
ID Surat: 121
nama mahasiswa: udin
Kelas: 1e
Jenis Izin (S atau I): S
Durasi: 60
Surat udin berhasil dikumpulkan
Menu:
1. Menerima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat berdasarkan nama mahasiswa
Pilih: 1
ID Surat: 124
nama mahasiswa: ihsan
Kelas: 1e
Jenis Izin (S atau I): I
Durasi: 120
Surat ihsan berhasil dikumpulkan
Menu:
1. Menerima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat berdasarkan nama mahasiswa
Pilih: 2
Verifikasi Surat milik ihsan
Status Surat milik ihsan : true
Menu:
1. Menerima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat berdasarkan nama mahasiswa
Pilih: 3
Surat teratas milik : udin
Menu:
```