

**LAPORAN HASIL PRAKTIKUM
ALGORITMA DAN STRUKTUR DATA
JOBSHEET 9 STACK**



RANI MIFTAHUS SAÁDAH

244107020057

TI_1E

**PROGRAM STUDI D_IV TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
PRAKTIKUM 2025**

1.1 Percobaan 1 : Mahasiswa Mengumpulkan Tugas

- Mahasiswa25.java

```
public class Mahasiswa25 {
    String nim;
    String nama;
    String kelas;
    int nilai;

    public Mahasiswa25() {
        this.nilai = -1;
    }

    public Mahasiswa25 (String nim, String nama, String kelas) {
        this.nim = nim;
        this.nama = nama;
        this.kelas = kelas;
        this.nilai = -1;
    }
    public void tugasDinilai(int nilai) {
        this.nilai = nilai;
    }
}
```

- StackTugasMahasiswa25.java

```
public class StackTugasMahasiswa03 {

    Mahasiswa25[] stack;
    int top;
    int size;

    public StackTugasMahasiswa03(int size) {
        this.size = size;
        stack = new Mahasiswa25[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(Mahasiswa25 mhs) {
        if (!isFull()) {
            top++;
            stack[top] = mhs;
        }
    }
}
```

```

        }else {
            System.out.println("Stack penuh! tidak bisa menambahkan tugas
lagi. ");
        }
    }

    public Mahasiswa25 pop() {
        if (!isEmpty()) {
            Mahasiswa25 m = stack[top];
            top --;
            return m;
        }else {
            System.out.println("stack kosong! tidak ada tugas untuk
dinilai");
            return null;
        }
    }

    public Mahasiswa25 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("");
    }
}

```

- **MahasiswaDemo25.java**

```

import java.util.Scanner;
public class MahasiswaDemo25 {

    public static void main(String[] args) {
        StackTugasMahasiswa03 stack = new StackTugasMahasiswa03(5);

        Scanner scan = new Scanner(System.in);
        int pilih;

        do {
            System.out.println("\nMenu:");
            System.out.println("1. Mengumpulkan Tugas");
            System.out.println("2. Menilai Tugas");
            System.out.println("3. Melihat Tugas Teratas");
            System.out.println("4. Melihat Daftar Tugas");
            System.out.println("0. Keluar");
            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();
                Mahasiswa25 mhs = new Mahasiswa25(nama, nim, kelas);
                stack.push(mhs);
                System.out.printf("Tugas %s berhasil dikumpulkan\n",
mhs.nama);
                break;

            case 2:
                Mahasiswa25 dinilai = stack.pop();
                if (!stack.isEmpty()) {
                    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:
                Mahasiswa25 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);
}

```

1.2 Verifikasi Hasil Percobaan

```
Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
0. Keluar
Pilih: 1
Nama: arifaa
NIM: 24412
Kelas: 1E
Tugas arifaa berhasil dikumpulkan

Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
0. Keluar
Pilih: 1
Nama: angell
NIM: 24413
Kelas: 1E
Tugas angell berhasil dikumpulkan

Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
0. Keluar
Pilih: 3
Tugas terakhir dikumpulkan oleh angell
```

```
Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
0. Keluar
Pilih: 1
Nama: lidyaa
NIM: 24414
Kelas: 1E
Tugas lidyaa berhasil dikumpulkan
```

```
Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
0. Keluar
Pilih: 4
Daftar semua tugas:
Nama    NIM    Kelas
arifaa  24412  1E
angell  24413  1E
lidyaa  24414  1E
```

```
Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
0. Keluar
Pilih: 2
Menilai tugas dari lidyaa
Masukkan nilai (0-100): 90
Nilai Tugas lidyaa adalah 90
```

```

Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
0. Keluar
Pilih: 4
Daftar semua tugas:
Nama    NIM    Kelas
arifaa  24412  1E
angell  24413  1E

```

```

Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
0. Keluar
Pilih: 0
Pilihan tidak valid.

```

Pertanyaan

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

```

public class Mahasiswa25 {
    String nim, nama, kelas;
    int nilai;

    Mahasiswa25() {
        this.nim = "";
        this.nama = "";
        this.kelas = "";
        this.nilai = -1;
    }

    Mahasiswa25(String nim, String nama, String kelas) {
        this.nim = nim;
        this.nama = nama;
        this.kelas = kelas;
        this.nilai = -1;
    }

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

```

```

public class StackTugasMahasiswa25 {
    Mahasiswa25[] stack;
    int size, top;

    StackTugasMahasiswa25(int size) {
        this.size = size;
        this.stack = new Mahasiswa25[size];
        this.top = -1;
    }

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

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

```

```

    }

    void push(Mahasiswa25 mhs) {
        if (!isFull()) {
            stack[++top] = mhs;
        } else {
            System.out.println("Stack penuh. Tidak dapat
menambahkan tugas.");
        }
    }

    Mahasiswa25 pop() {
        if (!isEmpty()) {
            return stack[top--];
        } else {
            System.out.println("Stack kosong. Tidak ada
tugas yang bisa dinilai.");
            return null;
        }
    }

    Mahasiswa25 peek() {
        if (!isEmpty()) {
            return stack[top];
        } else {
            System.out.println("Stack kosong.");
            return null;
        }
    }

    void print() {
        if (!isEmpty()) {
            for (int i = top; i >= 0; i--) {
                System.out.println(stack[i].nama + "\t" +
stack[i].nim + "\t" + stack[i].kelas);
            }
        } else {
            System.out.println("Stack kosong.");
        }
    }
}

```

```

import java.util.Scanner;

public class MahasiswaDemo25 {
    public static void main(String[] args) {
        StackTugasMahasiswa25 stack = new
StackTugasMahasiswa25(5);
        Scanner sc = new Scanner(System.in);
        int pilih;

        do {
            System.out.println("\nMenu:");
            System.out.println("1. Mengumpulkan Tugas");
            System.out.println("2. Menilai Tugas");
            System.out.println("3. Melihat Tugas Teratas");
            System.out.println("4. Melihat Daftar Tugas");
            System.out.println("0. Keluar");
            System.out.print("Pilih: ");

```

```

        pilih = sc.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();
                Mahasiswa25 mhs = new Mahasiswa25(nim,
nama, kelas);
                stack.push(mhs);
                System.out.printf("Tugas %s berhasil
dikumpulkan\n", mhs.nama);
                break;

            case 2:
                Mahasiswa25 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:
                Mahasiswa25 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;

            case 0:
                System.out.println("Terima kasih.");
                break;

            default:
                System.out.println("Pilihan tidak
valid.");
        }
    } while (pilih != 0);
}

```



```
Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
0. Keluar
Pilih: 1
Nama: arifaa
NIM: 24413
Kelas: 1E
Tugas arifaa berhasil dikumpulkan
```

```
Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
0. Keluar
Pilih: 1
Nama: raniiii
NIM: 24415
Kelas: 1E
Tugas raniiii berhasil dikumpulkan
```

```
Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
0. Keluar
Pilih: 3
Tugas terakhir dikumpulkan oleh raniiii
```

```
Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
0. Keluar
Pilih: 1
Nama: syeeri
NIM: 24416
Kelas: 1E
Tugas syeeri berhasil dikumpulkan
```

```
Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
0. Keluar
Pilih: 4
Daftar semua tugas:
Nama    NIM    Kelas
syeeri  24416  1E
raniiii 24415  1E
arifaa  24413  1E
```

```
Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
0. Keluar
Pilih: 2
Menilai tugas dari syeeri
Masukkan nilai (0-100): 89
Nilai Tugas syeeri adalah 89
```

```
Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
0. Keluar
Pilih: 4
Daftar semua tugas:
Nama    NIM    Kelas
raniiii 24415  1E
arifaa  24413  1E
```

2. Berapa banyak data tugas mahasiswa yang dapat ditampung di dalam Stack? Tunjukkan potongan kode programnya!

Ada 5

```
StackTugasMahasiswa25 stack = new StackTugasMahasiswa25(size:5);
```

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

Untuk mencegah stack overflow, yaitu kondisi di mana data dimasukkan melebihi kapasitas stack, Jika if-else dihapus akan menyebabkan error atau program crash ketika menambahkan data saat stack sudah penuh

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!

```
public Mahasiswa25 bottom() {
    if (!isEmpty()) {
        return stack[0];
    } else {
        System.out.println(x:"Stack kosong.");
        return null;
    }
}

}

case 5:
    Mahasiswa25 bawah = stack.bottom();
    if (bawah != null) {
        System.out.println("Mahasiswa pertama mengumpulkan: " + bawah.nama);
    } else {
        System.out.println(x:"Stack kosong.");
    }
    break;
```

```

Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Melihat Tugas Pertama yang Dikumpulkan
0. Keluar
Pilih: 1
Nama: cimeng
NIM: 24415
Kelas: 1E
Tugas cimeng berhasil dikumpulkan

Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Melihat Tugas Pertama yang Dikumpulkan
0. Keluar
Pilih: 1
Nama: musrifah
NIM: 24416
Kelas: 1E
Tugas musrifah berhasil dikumpulkan

Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Melihat Tugas Pertama yang Dikumpulkan
0. Keluar
Pilih: 5
Mahasiswa pertama mengumpulkan: cimeng

```

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

```

    public int jumlahTugas() {
        return top +1;
    }
}

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

```

```

Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Melihat Tugas Pertama yang Dikumpulkan
6. Jumlah Tugas yang Dikumpulkan
0. Keluar
Pilih: 1
Nama: nadya
NIM: 24417
Kelas: 1E
Tugas nadya berhasil dikumpulkan

Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Melihat Tugas Pertama yang Dikumpulkan
6. Jumlah Tugas yang Dikumpulkan
0. Keluar
Pilih: 1
Nama: wahyu
NIM: 24419
Kelas: 1E
Tugas wahyu berhasil dikumpulkan

Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Melihat Tugas Pertama yang Dikumpulkan
6. Jumlah Tugas yang Dikumpulkan
0. Keluar
Pilih: 6
Jumlah tugas yang sudah dikumpulkan: 2

```

2.1 Percobaan 2 : Konversi Nilai Tugas ke Biner

Mahasiswa25.java

```

public class Mahasiswa25 {
    String nim, nama, kelas;
    int nilai;

    Mahasiswa25() {
        this.nim = "";
        this.nama = "";
        this.kelas = "";
        this.nilai = -1;
    }

    Mahasiswa25(String nim, String nama, String kelas) {
        this.nim = nim;
        this.nama = nama;
        this.kelas = kelas;
        this.nilai = -1;
    }

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

```

StackKonversi25.java

```
public class StackKonversi25 {
    int[] tumpukanBiner;
    int size;
    int top;

    public StackKonversi25() {
        this.size = 32;
        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;
        }
    }
}
```

StackTugasMahasiswa25.java

```
public class StackTugasMahasiswa25 {
    Mahasiswa25[] stack;
    int size, top;

    StackTugasMahasiswa25(int size) {
        this.size = size;
        this.stack = new Mahasiswa25[size];
        this.top = -1;
    }
}
```

```

    }

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

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

    void push(Mahasiswa25 mhs) {
        if (!isFull()) {
            stack[++top] = mhs;
        } else {
            System.out.println("Stack penuh. Tidak dapat menambahkan
tugas.");
        }
    }

    Mahasiswa25 pop() {
        if (!isEmpty()) {
            return stack[top--];
        } else {
            System.out.println("Stack kosong. Tidak ada tugas yang bisa
dinilai.");
            return null;
        }
    }

    Mahasiswa25 peek() {
        if (!isEmpty()) {
            return stack[top];
        } else {
            System.out.println("Stack kosong.");
            return null;
        }
    }

    void print() {
        if (!isEmpty()) {
            for (int i = top; i >= 0; i--) {
                System.out.println(stack[i].nama + "\t" + stack[i].nim +
"\t" + stack[i].kelas);
            }
        } else {
            System.out.println("Stack kosong.");
        }
    }

    Mahasiswa25 bottom() {
        if (!isEmpty()) {
            return stack[0];
        } else {
            System.out.println("Stack kosong.");
            return null;
        }
    }

    public int jumlahTugas() {
        return top + 1;
    }

```

```

    public String konversiDesimalkeBiner(int nilai) {
        StackKonversi25 stack = new StackKonversi25();
        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;
    }
}

```

MahasiswaDemo25.java

```

import java.util.Scanner;

public class MahasiswaDemo25 {
    public static void main(String[] args) {
        StackTugasMahasiswa25 stack = new StackTugasMahasiswa25(5);
        Scanner sc = new Scanner (System.in);
        int pilih;

        do {
            System.out.println("\nMenu:");
            System.out.println("1. Mengumpulkan Tugas");
            System.out.println("2. Menilai Tugas");
            System.out.println("3. Melihat Tugas Teratas");
            System.out.println("4. Melihat Daftar Tugas");
            System.out.println("5. Melihat Tugas Pertama yang
Dikumpulkan");
            System.out.println("6. Jumlah tugas yang dikumpulkan");
            System.out.println("0. Keluar");
            System.out.print("Pilih: ");
            pilih = sc.nextInt();
            sc.nextLine();

            switch (pilih) {
                case 1:
                    System.out.print("Nama: ");
                    String nama = sc.nextLine();
                    System.out.print("NIM: ");
                    String nim = sc.nextLine();
                    System.out.print("Kelas: ");
                    String kelas = sc.nextLine();
                    Mahasiswa25 mhs = new Mahasiswa25(nim, nama, kelas);
                    stack.push(mhs);
                    System.out.printf("Tugas %s berhasil dikumpulkan\n",
mhs.nama);

                    break;

                case 2:
                    Mahasiswa25 dinilai = stack.pop();
                    if (dinilai != null) {
                        System.out.println("Menilai tugas dari " +
dinilai.nama);
                    }

```

```

        System.out.print("Masukkan nilai (0-100): ");
        int nilai = sc.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;

    case 3:
        Mahasiswa25 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;

    case 5:
        Mahasiswa25 terbawah = stack.bottom();
        if (terbawah != null) {
            System.out.println("Tugas pertama dikumpulkan
oleh " + terbawah.nama);
        }
        break;

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

    case 0:
        System.out.println("Terima kasih.");
        break;

    default:
        System.out.println("Pilihan tidak valid.");
    }
} while (pilih != 0);
}

```

2.2 Verifikasi

Menu:

1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Melihat Tugas Pertama yang Dikumpulkan
6. Jumlah tugas yang dikumpulkan
0. Keluar

Pilih: 2

Menilai tugas dari Fika

Masukkan nilai (0-100): 89

Nilai Tugas Fika adalah 89

Nilai Biner Tugas: 1011001

Pertanyaan

1. Jelaskan alur kerja dari method konversiDesimalKeBiner!
Mengonversi nilai desimal ke biner menggunakan prinsip stack:
 - Sisa hasil bagi 2 di-push ke stack
 - Setelah desimal = 0, dilakukan pop dan cetak dari stack, menghasilkan biner secara terbalik
2. Pada method konversiDesimalKeBiner, ubah kondisi perulangan menjadi while (kode != 0), bagaimana hasilnya? Jelaskan alasannya!

```
public String konversiDesimalkeBiner(int nilai) {  
    StackKonversi25 stack = new StackKonversi25();  
    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;  
}
```

```

Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Melihat Tugas Pertama yang Dikumpulkan
6. Jumlah tugas yang dikumpulkan
0. Keluar
Pilih: 2
Stack kosong. Tidak ada tugas yang bisa dinilai.

Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Melihat Tugas Pertama yang Dikumpulkan
6. Jumlah tugas yang dikumpulkan
0. Keluar
Pilih: 1
Nama: Yoga
NIM: 2441025
Kelas: 1E
Tugas Yoga berhasil dikumpulkan

Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Melihat Tugas Pertama yang Dikumpulkan
6. Jumlah tugas yang dikumpulkan
0. Keluar
Pilih: 2
Menilai tugas dari Yoga
Masukkan nilai (0-100): 92
Nilai Tugas Yoga adalah 92
Nilai Biner Tugas: 1011100

```

- Hasil sama seperti `while(kode > 0)`
- Tapi jika nilai `kode = 0` maka tidak akan ada bit yang dipush, biner tetap kosong
- Lebih tepat tetap menggunakan `kode > 0` agar menghasilkan 0

LATIHAN PRAKTIKUM

Surat25.java

```

public class Surat25 {
    String idSurat;
    String namaMahasiswa;
    String kelas;
    char jenisIzin;
    int durasi;

    public Surat25() {}

    public Surat25(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;
    }

    public void tampilSurat() {

```

```
        System.out.println("ID: " + idSurat + ", Nama: " + namaMahasiswa + ",  
Kelas: " + kelas + ", Jenis Izin: " + jenisIzin + ", Durasi: " + durasi +  
" hari");  
    }  
}
```

StackSurat25.java

```
public class StackSurat25 {  
    Surat25[] stack;  
    int size, top;  
  
    public StackSurat25(int size) {  
        this.size = size;  
        stack = new Surat25[size];  
        top = -1;  
    }  
  
    public boolean isEmpty() {  
        return top == -1;  
    }  
  
    public boolean isFull() {  
        return top == size - 1;  
    }  
  
    public void push(Surat25 srt) {  
        if (!isFull()) {  
            stack[++top] = srt;  
        } else {  
            System.out.println("Tumpukan surat penuh.");  
        }  
    }  
  
    public Surat25 pop() {  
        if (!isEmpty()) {  
            return stack[top--];  
        } else {  
            System.out.println("Tidak ada surat untuk diproses.");  
            return null;  
        }  
    }  
  
    public Surat25 peek() {  
        if (!isEmpty()) {  
            return stack[top];  
        } else {  
            return null;  
        }  
    }  
  
    public boolean cariSurat(String nama) {  
        boolean ditemukan = false;  
        for (int i = top; i >= 0; i--) {  
            if (stack[i].namaMahasiswa.equalsIgnoreCase(nama)) {  
                System.out.println("Surat ditemukan:");  
                stack[i].tampilSurat();  
                ditemukan = true;  
            }  
        }  
    }  
}
```

```

        return ditemukan;
    }

    public void printAll() {
        if (isEmpty()) {
            System.out.println("Tidak ada surat yang masuk.");
        } else {
            for (int i = top; i >= 0; i--) {
                stack[i].tampilSurat();
            }
        }
    }
}

```

StackKonversi25.java

```

public class StackKonversi25 {
    int[] tumpukanBiner;
    int size;
    int top;

    public StackKonversi25() {
        this.size = 32;
        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;
        }
    }
}

```

SuratDemo25.java

```
import java.util.Scanner;

public class SuratDemo25 {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        StackSurat25 stack = new StackSurat25(10);
        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 Terakhir");
            System.out.println("4. Cari Surat berdasarkan Nama");
            System.out.println("5. Tampilkan Semua Surat");
            System.out.println("0. Keluar");
            System.out.print("Pilih: ");
            pilih = scan.nextInt(); scan.nextLine();

            switch (pilih) {
                case 1:
                    System.out.print("ID Surat: ");
                    String id = scan.nextLine();
                    System.out.print("Nama Mahasiswa: ");
                    String nama = scan.nextLine();
                    System.out.print("Kelas: ");
                    String kelas = scan.nextLine();
                    System.out.print("Jenis Izin (S/I): ");
                    char jenis = scan.nextLine().toUpperCase().charAt(0);
                    System.out.print("Durasi (hari): ");
                    int durasi = scan.nextInt();
                    Surat25 surat = new Surat25(id, nama, kelas, jenis,
durasi);

                    stack.push(surat);
                    break;

                case 2:
                    Surat25 proses = stack.pop();
                    if (proses != null) {
                        System.out.println("Memproses surat:");
                        proses.tampilSurat();
                    }
                    break;

                case 3:
                    Surat25 lihat = stack.peek();
                    if (lihat != null) {
                        System.out.println("Surat terakhir:");
                        lihat.tampilSurat();
                    } else {
                        System.out.println("Tidak ada surat.");
                    }
                    break;

                case 4:
                    System.out.print("Masukkan nama mahasiswa: ");
                    String cari = scan.nextLine();
                    boolean ditemukan = stack.cariSurat(cari);
                    if (!ditemukan) {
```

```

        System.out.println("Surat tidak ditemukan.");
    }
    break;

    case 5:
        stack.printAll();
        break;

    case 0:
        System.out.println("Keluar...");
        break;

    default:
        System.out.println("Pilihan tidak valid.");
    }

    } while (pilih != 0);
}
}

```

```

Menu:
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Terakhir
4. Cari Surat berdasarkan Nama
5. Tampilkan Semua Surat
0. Keluar
Pilih: 1
ID Surat: 1112
Nama Mahasiswa: angel
Kelas: 1E
Jenis Izin (S/I): S
Durasi (hari): 2

```

```

Menu:
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Terakhir
4. Cari Surat berdasarkan Nama
5. Tampilkan Semua Surat
0. Keluar
Pilih: 1
ID Surat: 1113
Nama Mahasiswa: rani
Kelas: 1E
Jenis Izin (S/I): I
Durasi (hari): 1

```

```

Menu:
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Terakhir
4. Cari Surat berdasarkan Nama
5. Tampilkan Semua Surat
0. Keluar
Pilih: 2
Memproses surat:
ID: 1113, Nama: rani, Kelas: 1E, Jenis Izin: I, Durasi: 1 hari

```

```

Menu:
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Terakhir
4. Cari Surat berdasarkan Nama
5. Tampilkan Semua Surat
0. Keluar
Pilih: 3
Surat terakhir:
ID: 1112, Nama: angel, Kelas: 1E, Jenis Izin: S, Durasi: 2 hari

```

```

Menu:
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Terakhir
4. Cari Surat berdasarkan Nama
5. Tampilkan Semua Surat
0. Keluar
Pilih: 4
Masukkan nama mahasiswa: angel
Surat ditemukan:
ID: 1112, Nama: angel, Kelas: 1E, Jenis Izin: S, Durasi: 2 hari

```

```

Menu:
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Terakhir
4. Cari Surat berdasarkan Nama
5. Tampilkan Semua Surat
0. Keluar
Pilih: 5
ID: 1112, Nama: angel, Kelas: 1E, Jenis Izin: S, Durasi: 2 hari

```