

**LAPORAN TUGAS PRAKTIKUM
ALGORITMA DAN STRUKTUR DATA WEEK 9**



NAMA : RADITYA RIEFKI
KELAS : TI 1E
ABSEN : 23

2. Praktikum 2.1 Percobaan 1: Mahasiswa Mengumpulkan Tugas

1. Kode Program

Mahasiswa

```
package jobhsheet10;

public class Mahasiswa23 {
    String nim, nama, kelas;
    int nilai;

    Mahasiswa23(String nama, String nim, String kelas){
        this.nama = nama;
        this.nim = nim;
        this.kelas = kelas;
        nilai = -1;
    }
    void tugasDinilai (int nilai){
        this.nilai = nilai;
    }
}
```

StackTugasMahasiswa

```
package jobhsheet10;

public class StackTugasMahasiswa23 {
    Mahasiswa23 [] stack;
    int top, size;
    public StackTugasMahasiswa23(int size){
        this.size = size;
        stack = new Mahasiswa23[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 (Mahasiswa23 mhs){
        if (!isFull()) {
            top++;
        }
    }
}
```

```

        stack[top] = mhs;
    }else{
        System.out.println("Stack penuh! Tidak bisa menambahkan tugas
lagi.");
    }
}
}
public Mahasiswa23 pop(){
    if (!isEmpty()) {
        Mahasiswa23 m = stack [top];
        top--;
        return m;
    }else{
        System.out.println("Stack kosong!, Tidak ada tugas untuk
dinilai.");
        return null;
    }
}
}
public Mahasiswa23 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("");
}
}
}

```

Main

```

package jobhsheet10;

import java.util.Scanner;

public class MahasiswaDemo23 {
    public static void main(String[] args) {
        int pilih;
        Scanner sc = new Scanner(System.in);
        StackTugasMahasiswa23 stack = new StackTugasMahasiswa23(5);

        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.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();
                Mahasiswa23 mhs = new Mahasiswa23(nama, nim, kelas);
                stack.push(mhs);
                System.out.printf("Tugas %s berhasil dikumpulkan\n",
mhs.nama);

                break;
            case 2:
                Mahasiswa23 dinilai = stack.pop();
                if (dinilai != null) {
                    System.out.println("Menilai tugas dari " +
dinilai.nama);

                    System.out.println("Masukka nilai (0-100): ");
                    int nilai = sc.nextInt();
                    dinilai.tugasDinilai(nilai);
                    System.out.printf("Nilai tugas %s adalah %d\n",
dinilai.nama, nilai);
                }
                break;
            case 3:
                Mahasiswa23 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);
}
}

```

OUTPUT

```
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
```

Pilih: 2

Menilai tugas dari Tika

Masukka nilai (0-100):

87

Nilai tugas Tika adalah 87

1. Mengumpulkan Tugas

2. Menilai Tugas

3. Melihat Tugas Teratas

4. Melihat Daftar Tugas

Pilih: 2

Menilai tugas dari Tika

Masukka nilai (0-100):

87

Nilai tugas Tika adalah 87

Pilih: 2

Menilai tugas dari Tika

Masukka nilai (0-100):

87

Nilai tugas Tika adalah 87

Menu

1. Mengumpulkan Tugas

2. Menilai Tugas

Menu

1. Mengumpulkan Tugas

2. Menilai Tugas

Menu

1. Mengumpulkan Tugas

2. Menilai Tugas

2. Menilai Tugas

3. Melihat Tugas Teratas

3. Melihat Tugas Teratas

4. Melihat Daftar Tugas

4. Melihat Daftar Tugas

Pilih: 4

Daftar semua tugas

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

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

```
StackTugasMahasiswa23 stack = new StackTugasMahasiswa23(size:5);
```

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

- Karena jika tidak ada kondisi tersebut akan menyebabkan index out of bounds contoh kasus seperti sebuah array yang sudah terisi penuh tetapi tetap dipaksa untuk mengisi elemen array tsb

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!

Kode StackTugasMahasiswa 23

```
Mahasiswa23 peekBottom(){
    if (!isEmpty()) {
        return stack[0];
    }else{
        System.out.println(x:"Stack kosong! Tidak ada tugas yang dikumpulkan");
        return null;
    }
}
```

Kode MahasiswaDemo

```
case 5:
    Mahasiswa23 lihatTerbawah = stack.peekBottom();
    if (lihatTerbawah != null) {
        System.out.println("Tugas pertama dikumpulkan oleh " + lihatTerbawah.nama);
    }
}
```

Output

```

Menu
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Melihat Daftar Tugas Terbawah (pertama)
Pilih: 1
Nama: doni
NIM: 1
Kelas: 1e
Tugas doni berhasil dikumpulkan

Menu
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Melihat Daftar Tugas Terbawah (pertama)
Pilih: 5
Tugas pertama dikumpulkan oleh doni
PS D:\CollegeFile\SMT 2\ALSD>

```

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

Kode function

```

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

```

Kode MahasiswaDemo

```

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

```

Output

```

Pilih: 1
Nama: doni
NIM: 1
Kelas: 1e
Tugas doni berhasil dikumpulkan

Menu
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Melihat Daftar Tugas Terbawah (pertama)
6. Jumlah Tugas Yang Dikumpulkan
Pilih: 1
Nama: ero
NIM: 2
Kelas: 1e
Tugas ero berhasil dikumpulkan

Menu
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Melihat Daftar Tugas Terbawah (pertama)
6. Jumlah Tugas Yang Dikumpulkan
Pilih: 6
Jumlah tugas yang telah dikumpulkan: 2
PS D:\CollegeFile\SMT 2\ALSD>

```


2.2 Percobaan 2: Konversi Nilai Tugas ke Biner

Kode StackKonversi

```
package jobhsheet10;

public class StackKonversi23 {
    int[] tumpukanBiner;
    int size;
    int top;

    public StackKonversi23(){
        this.size = 32; //asumsi 32 bit
        tumpukanBiner = new int[size];
        top = -1;
    }
    boolean isEmpty(){
        return top == -1;
    }
    boolean isFull(){
        return top == size - 1;
    }
    void push (int data){
        if (isFull()) {
            System.out.println("Stack penuh");
        }else{
            top++;
            tumpukanBiner[top] = data;
        }
    }
    int pop(){
        if (isEmpty()) {
            System.out.println("Stack kosong");
            return -1;
        }else {
            int data = tumpukanBiner[top];
            top--;
            return data;
        }
    }
}
```

Kode StackTugasMahasiswa

```
package jobhsheet10;

public class StackTugasMahasiswa23 {
    Mahasiswa23 [] stack;
    int top, size;
    public StackTugasMahasiswa23(int size){
        this.size = size;
        stack = new Mahasiswa23[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 (Mahasiswa23 mhs){
        if (!isFull()) {
            top++;
            stack[top] = mhs;
        }else{
            System.out.println("Stack penuh! Tidak bisa menambahkan tugas lagi.");
        }
    }
    public Mahasiswa23 pop(){
        if (!isEmpty()) {
            Mahasiswa23 m = stack [top];
            top--;
            return m;
        }else{
            System.out.println("Stack kosong!, Tidak ada tugas untuk dinilai.");
            return null;
        }
    }
    public Mahasiswa23 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].nama + "\t" + stack[i].nim + "\t"
+stack[i].kelas);
    }
    System.out.println("");
}
Mahasiswa23 peekBottom(){
    if (!isEmpty()) {
        return stack[0];
    }else{
        System.out.println("Stack kosong! Tidak ada tugas yang
dikumpulkan");
        return null;
    }
}
int jumlahTugas(){
    if (!isEmpty()) {
        return top + 1;
    }else{
        return 0;
    }
}
String konversiDesimalKeBiner(int nilai){
    StackKonversi23 stack = new StackKonversi23();
    while (nilai >0){
        int sisa = nilai % 2;
        stack.push(sisa);
        nilai /= 2;
    }
    String biner = new String();
    while (!stack.isEmpty()){
        biner += stack.pop();
    }
    return biner;
}
}

```

Kode MahasiswaDemo

```
package jobhsheet10;

import java.util.Scanner;

public class MahasiswaDemo23 {
    public static void main(String[] args) {
        int pilih;
        Scanner sc = new Scanner(System.in);
        StackTugasMahasiswa23 stack = new StackTugasMahasiswa23(5);

        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 Daftar Tugas Terbawah
(pertama)");
            System.out.println("6. Jumlah Tugas Yang Dikumpulkan");
            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();
                    Mahasiswa23 mhs = new Mahasiswa23(nama, nim, kelas);
                    stack.push(mhs);
                    System.out.printf("Tugas %s berhasil dikumpulkan\n",
mhs.nama);

                    break;
                case 2:
                    Mahasiswa23 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:
                    Mahasiswa23 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();
    case 5:
        Mahasiswa23 lihatTerbawah = stack.peekBottom();
        if (lihatTerbawah != null) {
            System.out.println("Tugas pertama dikumpulkan oleh " +
lihatTerbawah.nama);
        }
        break;
    case 6:
        int jumlahTugas = stack.jumlahTugas();
        System.out.println("Jumlah tugas yang telah dikumpulkan: " +
jumlahTugas);
        break;

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

    }
    }while(pilih >=1 && pilih <= 4);
}
}

```

OUTPUT

```

0. Jumlah tugas yang dikumpulkan
Pilih: 2
Menilai tugas dari Tika
Masukkan nilai (0-100): 87
Nilai tugas Tika adalah 87
Nilai Biner Tugas: 1010111

```

2.2.3 Pertanyaan

1. Jelaskan alur kerja dari method konversiDesimalKeBiner!

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. Pada method konversiDesimalKeBiner, ubah kondisi perulangan menjadi while (kode != 0), bagaimana hasilnya? Jelaskan alasannya!

- 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

TUGAS PRAKTIKUM

Kode Program surat

```
package jobhsheet10;

public class Surat23 {
    String idSurat, namaMhs, kelas;
    char jenisIzin;
    int durasi;
    boolean isVerificated;

    Surat23(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;
    }
}
```

Kode Program StackSurat

```
package jobhsheet10;

public class StackSurat23 {
    Surat23[] stack;
    int size;
    int top;

    StackSurat23(int size) {
        this.size = size;
        stack = new Surat23[size];
        top = -1;
    }

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

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

    void push(Surat23 s) {
        if (!isFull()) {
            top++;
            stack[top] = s;
        } else {
            System.out.println("stack penuh!");
        }
    }

    Surat23 pop() {
        if(!isEmpty()) {
            Surat23 s = stack[top];
            top--;
            return s;
        } else {
            System.out.println("stack kosong!");
            return null;
        }
    }

    Surat23 peek() {
        if (!isEmpty()) {
            return stack[0];
        } else {
            System.out.println("Stack kosong!");
            return null;
        }
    }
}
```

```

    }
}

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

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

```

Stack SuratDemo

```

package jobhsheet10;

import java.util.Scanner;

public class MahasiswaDemo23 {
    public static void main(String[] args) {
        int pilih;
        Scanner sc = new Scanner(System.in);
        StackTugasMahasiswa23 stack = new StackTugasMahasiswa23(5);

        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 Daftar Tugas Terbawah
(pertama)");
            System.out.println("6. Jumlah Tugas Yang Dikumpulkan");
            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();
                    Mahasiswa23 mhs = new Mahasiswa23(nama, nim, kelas);
                    stack.push(mhs);

```



```

        System.out.printf("Tugas %s berhasil dikumpulkan\n",
mhs.nama);
        break;
    case 2:
        Mahasiswa23 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:
        Mahasiswa23 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();
    case 5:
        Mahasiswa23 lihatTerbawah = stack.peekBottom();
        if (lihatTerbawah != null) {
            System.out.println("Tugas pertama dikumpulkan oleh " +
lihatTerbawah.nama);
        }
        break;
    case 6:
        int jumlahTugas = stack.jumlahTugas();
        System.out.println("Jumlah tugas yang telah dikumpulkan: " +
jumlahTugas);
        break;

    default:
        System.out.println("Pilihan tidak valid.");
    }
}while(pilih >=1 && pilih <= 4);
}
}

```

OUTPUT

```
Menu:
1. Menerima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat berdasarkan nama mahasiswa
Pilih: 1
ID Surat: 1
nama mahasiswa: eko
Kelas: 1E
Jenis Izin (S atau I): S
Durasi: 100
Surat eko berhasil dikumpulkan
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 : eko
Menu:
1. Menerima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat berdasarkan nama mahasiswa
Pilih: 4
Masukkan nama surat dengan nama mahasiswa yang dicari : eko
Surat milik eko ditemukan pada tumpukan ke 1
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 eko
Status Surat milik eko : true
Menu:
1. Menerima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat berdasarkan nama mahasiswa
Pilih: 
```