

LAPORAN HASIL PRAKTIKUM
Algoritma Struktur Data
JOBSHEET 9



Muhammad Fitra Adhim Nurrochman
2441007020089

TI 1E

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

PERCOBAAN 1

1. Hasil kode program
 - a. Mahasiswa19.java

```
public class Mahasiswa19 {  
    String nim;  
    String nama;  
    String kelas;  
    int nilai;  
  
    Mahasiswa19() {  
  
    }  
  
    Mahasiswa19(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. StackTugasMahasiswa19.java

```
public class StackTugasMahasiswa19 {
    Mahasiswa19[] stack;
    int top;
    int size;

    public StackTugasMahasiswa19(int size){
        this.size = size;
        stack = new Mahasiswa19[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(Mahasiswa19 mhs) {
        if (!isFull()) {
            top++;
            stack[top] = mhs;
        } else {
            System.out.println("Stack penuh! Tidak bisa menambahkan tugas lagi");
        }
    }

    public Mahasiswa19 pop() {
        if (!isEmpty()) {
            Mahasiswa19 m = stack[top];
            top--;
            return m;
        } else {
            System.out.println("Stack kosong! Tidak ada tugas untuk dinilai.");
            return null;
        }
    }

    public Mahasiswa19 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("");
    }
}
```

c. MahasiswaDemo19.java

```
import java.util.Scanner;

public class MahasiswaDemo19 {
    public static void main(String[] args) {
        StackTugasMahasiswa19 stack = new StackTugasMahasiswa19(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.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();
                    Mahasiswa19 mhs = new Mahasiswa19(nama, nim, kelas);
                    stack.push(mhs);
                    System.out.printf("Tugas %s berhasil dikumpulkan\n",
mhs.nama);

                    break;
                case 2:
                    Mahasiswa19 dinilai = stack.pop();
                    if (dinilai != null) {
                        System.out.println("Menilai tugas dari " +
dinilai.nama);

                        System.out.println("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:
                    Mahasiswa19 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);
    }
}
```

2. Hasil Run Kode Program

```
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

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

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

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

3. Pertanyaan

- Lakukan perbaikan pada kode program, sehingga keluaran yang dihasilkan sama dengan verifikasi hasil percobaan! Bagian mana yang perlu diperbaiki?
- Berapa banyak data tugas mahasiswa yang dapat ditampung di dalam Stack? Tunjukkan potongan kode programnya!
- Mengapa perlu pengecekan kondisi `!isEmpty()` pada method `peekBawah()`? Kalau kondisi `if-else` tersebut dihapus, apa dampaknya?
- Modifikasi kode program pada class `MahasiswaDemo` dan `StackTugasMahasiswa` sehingga pengguna juga dapat melihat mahasiswa yang pertama kali mengumpulkan tugas melalui operasi lihat tugas terbawah!
- Tambahkan method untuk dapat menghitung berapa banyak tugas yang sudah dikumpulkan saat ini, serta tambahkan operasi menunya!

Jawaban

- Mengubah iterasi dari `top` ke `0` pada method `print`

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

Hasil sebagai berikut

```
Menu:  
1. Mengumpulkan Tugas  
2. Menilai Tugas  
3. Melihat Tugas Teratas  
4. Melihat Daftar Tugas  
Pilih: 4  
Daftar semua tugas  
nama    NIM    Kelas  
tika     234    1c  
erik     1234   1b  
dila     123    1a
```

- Maksimal data yang dapat ditampung adalah 5

```
StackTugasMahasiswa19 stack = new StackTugasMahasiswa19(size:5);
```

- Kondisi pengecekan diperlukan agar mencegah terjadinya overflow, jika kondisi tersebut dihapus maka akan terjadi error pada saat memasukkan data ke 6 (misalkan jika `size` adalah 5)
- Modifikasi pada `stacktugasmahasiswa`

```
public Mahasiswa19 peekBawah() {  
    if (!isEmpty()) {  
        return stack[0];  
    } else {  
        System.out.println(x:"Stack kosong! Tidak ada tugas yang dikumpulkan");  
        return null;  
    }  
}
```

Modifikasi pada `mahasiswademo`

```
case 5:  
    Mahasiswa19 lihatBawah = stack.peekBawah();  
    if (lihatBawah != null) {  
        System.out.println("Tugas terbawah dikumpulkan oleh " + lihatBawah.nama);  
    }  
    break;
```

Sama seperti peek akan tetapi nilai yang dikembalikan adalah stack[0] yaitu indeks ke 0 pada stack yang berarti paling bawah

- e. Menambahkan method totalTugas

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

```
case 6:  
    System.out.println("Jumlah tugas yang dikumpulkan: " + (stack.top + 1));  
    break;
```

Hasil run

```
Menu:  
1. Mengumpulkan Tugas  
2. Menilai Tugas  
3. Melihat Tugas Teratas  
4. Melihat Daftar Tugas  
5. Melihat Tugas Terbawah  
6. Hitung Jumlah Tugas  
Pilih: 1  
Nama: qwe  
NIM: 123  
kelas: 1  
Tugas qwe berhasil dikumpulkan  
  
Menu:  
1. Mengumpulkan Tugas  
2. Menilai Tugas  
3. Melihat Tugas Teratas  
4. Melihat Daftar Tugas  
5. Melihat Tugas Terbawah  
6. Hitung Jumlah Tugas  
Pilih: 1  
Nama: sdf  
NIM: 123  
kelas: 2  
Tugas sdf berhasil dikumpulkan  
  
Menu:  
1. Mengumpulkan Tugas  
2. Menilai Tugas  
3. Melihat Tugas Teratas  
4. Melihat Daftar Tugas  
5. Melihat Tugas Terbawah  
6. Hitung Jumlah Tugas  
Pilih: 1  
Nama: qwere  
NIM: 23234  
kelas: 3  
Tugas qwere berhasil dikumpulkan  
  
Menu:  
1. Mengumpulkan Tugas  
2. Menilai Tugas  
3. Melihat Tugas Teratas  
4. Melihat Daftar Tugas  
5. Melihat Tugas Terbawah  
6. Hitung Jumlah Tugas  
Pilih: 6  
Jumlah tugas yang dikumpulkan: 3
```

PERCOBAAN 2

1. Hasil kode program
 - a. Menambahkan method konversidesimalkebiner

```
public String konversiDesimalKeBiner(int nilai) {
    StackKonversi19 stack = new StackKonversi19();
    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;
}

Praktikum-ASD > Jobsheet 9 > StackKonversi19.java > StackKonversi19 >
1 public class StackKonversi19 {
2     int[] tumpukanBiner;
3     int size;
4     int top;
5
6     public StackKonversi19() {
7         this.size = 32; //asumsi 32 bit
8         tumpukanBiner = new int[size];
9         top = -1;
10    }
11
12    public boolean isEmpty() {
13        return top == -1;
14    }
15
16    public boolean isFull() {
17        return top == size - 1;
18    }
19
20    public void push(int data) {
21        if (isFull()) {
22            System.out.println("Stack penuh");
23        } else {
24            top++;
25            tumpukanBiner[top] = data;
26        }
27    }
28
29    public int pop() {
30        if (isEmpty()) {
31            System.out.println("Stack Kosong");
32            return -1;
33        } else {
34            int data = tumpukanBiner[top];
35            top--;
36            return data;
37        }
38    }
39 }
```


2. Hasil run kode program

```
Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Melihat Tugas Terbawah
6. Hitung Jumlah Tugas
Pilih: 1
Nama: tika
NIM: 123
kelas: 1c
Tugas tika berhasil dikumpulkan

Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Melihat Tugas Terbawah
6. Hitung Jumlah Tugas
Pilih: 2
Menilai tugas dari tika
Masukkan nilai (0-100):
87
Nilai tugas tika adalah 87
Nilai Biner Tugas: 1010111
```

3. Pertanyaan

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

Jawaban

- Pertama menginisialisasi stack lalu menghitung nilai sisa dengan cara memodulus nilai dengan modulus 2, lalu hasil modulus yaitu sisa di push ke stack, setelah mendapatkan sisa, nilai desimal dibagi 2, nilai / 2, perulangan while berlanjut sampai nilai sama dengan 0, setelah semua sisa pembagian dimasukkan ke dalam stack, bilangan biner disusun dengan cara mengambil elemen dari stack satu per satu menggunakan pop, karena stack memiliki sifat yang terakhir dimasukkan itu yang pertama kali dikeluarkan maka elemen yang terakhir dimasukkan akan diambil pertama, hasilnya bilangan biner tersusun dengan benar
- Hasil modifikasi

```

public String konversiDesimalKeBiner(int nilai) {
    StackKonversi19 stack = new StackKonversi19();
    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;
}

```

Hasilnya akan tetap sama saja, dalam kasus ini menghitung nilai tugas mahasiswa, jika beda kasus bisa saja berbeda dan mengharuskan menggunakan nilai > 0, karena kondisi tersebut memastikan bahwa perulangan hanya berjalan jika nilai lebih besar dari 0, ketika nilai menjadi 0 maka perulangan berhenti sedangkan kondisi nilai != 0 bisa saja melakukan perulangan selama nilai tidak sama dengan 0 atau negatif, misal jika dalam kasus lain nilai inputnya adalah negatif maka saat dibagi 2 tidak akan pernah menghasilkan 0, sehingga perulangan tidak akan pernah berhenti

TUGAS

1. Hasil kode program

```

public class Surat19 {
    String idSurat;
    String namaMahasiswa;
    String kelas;
    char jenisIzin;
    int durasi;

    public Surat19() {

    }
    public Surat19(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;
    }
}

```

```

import java.util.Scanner;

public class MainSurat19 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        StackSurat stack = new StackSurat(10);
        int pilih;

        do {
            System.out.println("\nMenu Aplikasi Surat Izin");
            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("5. Keluar");
            System.out.print("Pilih menu: ");
            pilih = sc.nextInt();
            sc.nextLine();

            switch (pilih) {
                case 1:
                    System.out.print("ID Surat19: ");
                    String id = sc.nextLine();
                    System.out.print("Nama Mahasiswa: ");
                    String nama = sc.nextLine();
                    System.out.print("Kelas: ");
                    String kelas = sc.nextLine();
                    System.out.print("Jenis Izin (S/I): ");
                    char jenisIzin = sc.next().charAt(0);
                    System.out.print("Durasi (hari): ");
                    int durasi = sc.nextInt();

                    Surat19 srtBaru = new Surat19(id, nama, kelas, jenisIzin, durasi);
                    stack.push(srtBaru);
                    break;

                case 2:
                    Surat19 srtProses = stack.pop();
                    if (srtProses != null) {
                        System.out.println("Memproses surat dari: " + srtProses.namaMahasiswa);
                    }
                    break;

                case 3:
                    Surat19 srtAtas = stack.peek();
                    if (srtAtas != null) {
                        System.out.println("Surat19 terakhir dari: " + srtAtas.namaMahasiswa);
                    }
                    break;

                case 4:
                    System.out.print("Masukkan nama mahasiswa: ");
                    String cariNama = sc.nextLine();
                    Surat19 hasilCari = stack.cariSurat(cariNama);
                    if (hasilCari != null) {
                        System.out.println("Surat ditemukan:");
                        System.out.println("ID: " + hasilCari.idSurat);
                        System.out.println("Nama: " + hasilCari.namaMahasiswa);
                        System.out.println("Kelas: " + hasilCari.kelas);
                        System.out.println("Jenis Izin: " + hasilCari.jenisIzin);
                        System.out.println("Durasi: " + hasilCari.durasi);
                    } else {
                        System.out.println("Surat tidak ditemukan!");
                    }
                    break;

                case 5:
                    System.out.println("Terima kasih!");
                    break;

                default:
                    System.out.println("Pilihan tidak valid!");
            }
        } while (pilih != 5);
    }
}

```

```
public class StackSurat {
    private Surat19[] data;
    private int size;
    private int top;

    public StackSurat(int size) {
        this.size = size;
        data = new Surat19[size];
        top = -1;
    }

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

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

    public void push(Surat19 srt) {
        if (!isFull()) {
            top++;
            data[top] = srt;
        } else {
            System.out.println("Stack penuh!");
        }
    }

    public Surat19 pop() {
        if (!isEmpty()) {
            Surat19 temp = data[top];
            top--;
            return temp;
        } else {
            System.out.println("Stack kosong!");
            return null;
        }
    }

    public Surat19 peek() {
        if (!isEmpty()) {
            return data[top];
        } else {
            System.out.println("Stack kosong!");
            return null;
        }
    }
}
```

2. Hasil Run Kode Program

```
Menu Aplikasi Surat Izin
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat
5. Keluar
Pilih menu: 1
ID Surat19: 123
Nama Mahasiswa: adh
Kelas: 1a
Jenis Izin (S/I): s
Durasi (hari): 2
```

```
Menu Aplikasi Surat Izin
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat
5. Keluar
Pilih menu: 3
Surat19 terakhir dari: adh
```

```
Menu Aplikasi Surat Izin
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat
5. Keluar
Pilih menu: 4
Masukkan nama mahasiswa: adh
Surat ditemukan:
ID: 123
Nama: adh
Kelas: 1a
Jenis Izin: s
Durasi: 2
```

```
Menu Aplikasi Surat Izin
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat
5. Keluar
Pilih menu: 2
Memproses surat dari: adh
```

```
Menu Aplikasi Surat Izin
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat
5. Keluar
Pilih menu: 3
Stack kosong!
```

```
Menu Aplikasi Surat Izin
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat
5. Keluar
Pilih menu: █
```