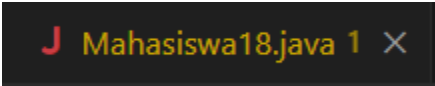


Nama: Rachmad Aprisandhy  
kelas : TI-1B  
no Abs : 18

## JOBSHEET 9 ALGORITMA STRUKTUR DATA

### A. Percobaan 1

1. Buat class mahasiswa noAbs



2. Lengkapi class mahasiswa dengan atribut nama, nim dan nilai

```
public class Mahasiswa18 {  
    String NIM;  
    String nama;  
    String kelas;  
    int nilai;
```

3. Tambahkan konstruktor berparameter

```
Mahasiswa18(String NIM, String nama, String kelas) {  
    this.NIM = NIM;  
    this.nama = nama;  
    this.kelas = kelas;  
    nilai = -1;  
}
```

4. Tambahkan method tugas dinilai() seperti dibawah ini

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

5. Buat class lagi Bernama StackTugasMahasiswa noAbs dengan menambahkan atribut

```
public class StackTugasMahasiswa18 {  
    Mahasiswa18[] stack;  
    int top;  
    int Size;
```

6. Tambahkan konstruktor berparameter pada class StackTugasMahasiswa

```
public StackTugasMahasiswa18(int size) {  
    this.Size = size;  
    stack = new Mahasiswa18[size];  
    top = -1;  
}
```

7. Buat method isFull

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

8. Buat method isEmpty

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

9. Buat method baru Bernama method push dan method ini menerima parameter mhs dari object class mahasiswa

```
public void push(Mahasiswa18 mhs) {  
  
    if (!isFull()) {  
        top++;  
        stack[top] = mhs;  
    } else {  
        System.out.println(x:"Stack penuh ! tidak bisa menambahkan tugas lagi.");  
    }  
}
```

10. Lalu tambahkan method pop untuk mengeluarkan tugas yang akan dinilai

```
public Mahasiswa18 pop() {  
    if (!isEmpty()) {  
        Mahasiswa18 mhs = stack[top];  
        top--;  
        return mhs;  
    } else {  
        System.out.println(x:"Stack kosong ! tidak ada tugas untuk dinilai.");  
        return null;  
    }  
}
```

11. Buat method peek untuk mengecek tumpukan tugas

```
public Mahasiswa18 peek() {
    if (!isEmpty()) {
        return stack[top];
    } else {
        System.out.println(x:"Stack kosong ! tidak ada tugas yang dikumpulkan.");
        return null;
    }
}
```

12. Tambahkan method print untuk menampilkan semua daftar tugas

```
}void print () {
    for (int i = 0; i<= top; i++){
        System.out.println(stack[i].nama + "\t" + stack[i].NIM + "\t" + stack[i].kelas);
    }
    System.err.println(x:"");
}
}
```

13. Buat main class bernama MahasiswaDemo18 asb dengan instansi object StackTugasMahasiswa deklarasikan juga scanner dengan variable scan dan bertipe integer

```
import java.util.Scanner;
public class MahasiswaDemo18 {
    Run main | Debug main | Run | Debug
    public static void main(String[] args) {
        StackTugasMahasiswa18 stack = new StackTugasMahasiswa18(size:5);
        Scanner scan = new Scanner(System.in);
        int pilih;
```

14. Ketikkan perulangan do-while seperti dibawah ini

```
do {
    System.out.println(x:"\nMenu : ");
    System.out.println(x:"1. mengumpulkan tugas");
    System.out.println(x:"2. menilai tugas");
    System.out.println(x:"3. melihat tugas teratas");
    System.out.println(x:"4. melihat daftar tugas");
    System.out.print(s:"Pilih menu : ");
    pilih = scan.nextInt();

    switch (pilih) {
        case 1:
            System.out.print(s:"Nama : ");
            String nama = scan.nextLine();
            System.out.print(s:"NIM : ");
            String NIM = scan.nextLine();
            System.out.print(s:"Kelas : ");
            String kelas = scan.nextLine();
            Mahasiswa18 mhs = new Mahasiswa18(nama, NIM, kelas);
            stack.push(mhs);
            System.out.printf(format:"Tugas %s berhasil dikumpulkan\n");
            break;
        case 2:
            Mahasiswa18 dinilai = stack.pop();
            if (dinilai != null) {
                System.out.println("menilai tugas dari" + dinilai.nama);
                System.out.print(s:"masukkan nilai (0-100): ");
                int nilai = scan.nextInt();
                dinilai.tugasDinilai(nilai);
                System.out.printf(format:"Nilai tugas %s adalah %d\n", dinilai.nama, nilai);
            }
            break;
        case 3:
            Mahasiswa18 lihat = stack.peek();
            if (lihat != null) {
                System.out.println("tugas terakhir dikumpulkan oleh" + lihat.nama);
            }
            break;
        case 4:
            System.out.println(x:"daftar semua tugas");
            System.out.println(x:"Nama\tNIM\tKelas");
            break;
        default :
            System.err.println(x:"pilihan tidak valid");
            stack.print();
            break;
    }
}while (pilih >= 1 && pilih <= 4);
}
```

15. Commit ke github

## B. Pertanyaan

### 1. Seperti ini untuk perubahan main nya

```
public static void main(String[] args) {
    StackTugasMahasiswa18 stack = new StackTugasMahasiswa18(size:5);
    Scanner scan = new Scanner(System.in);
    int pilih;

    do {
        System.out.println("Menu : ");
        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 menu : ");
        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();
                Mahasiswa mhs = new Mahasiswa(NIM, nama, kelas);
                stack.push(mhs);
                System.out.printf("Tugas %s berhasil dikumpulkan\n", mhs.nama);
                break;
            case 2:
                Mahasiswa 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:
                Mahasiswa 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\NIM\Kelas");
                stack.print();
                break;
            default:
                System.err.println("pilihan tidak valid");
                stack.print();
                break;
        }
    } while (pilih >= 1 && pilih <= 4);
}
```

hasilnya

```
1. mengumpulkan tugas
2. menilai tugas
3. melihat tugas teratas
4. melihat daftar tugas
Pilih menu : 1
Nama : bela
NIM : 223344
Kelas : 3E
Stack penuh ! tidak bisa menambahkan tugas lagi.
Tugas bela berhasil dikumpulkan

Menu :
1. mengumpulkan tugas
2. menilai tugas
3. melihat tugas teratas
4. melihat daftar tugas
Pilih menu : 4
daftar semua tugas
Nama    NIM    Kelas
paijo   2201   1D
desta   22113  1E
dimas   3457   2I
arya    334455 1C
rani    334455 2D
```

### 2. Line yang menampilkan banyak data yang disimpan adalah

```
StackTugasMahasiswa18 stack = new StackTugasMahasiswa18(size:5);
```

yang menunjukkan isinya ada 5

### 3. Pengecekan `isFull()` sangat penting di `push()` untuk menghindari error runtime dan menjaga program tetap aman dan stabil. Jika menghapus pengecekan `if (!isFull())`, maka method `push()` tetap akan menambah `top++`

dan menyimpan data baru tanpa peduli apakah stack sudah penuh atau belum.

4. Ini hasil modifikasinya

```
public Mahasiswa18 palingbawah() {  
    if (!isEmpty()) {  
        return stack[0];  
    } else {  
        System.out.println(x:"Stack kosong! Tidak ada tugas.");  
        return null;  
    }  
}  
}  
  
case 5:  
    Mahasiswa18 pertama = stack.palingbawah();  
    if (pertama != null) {  
        System.out.println("Tugas pertama dikumpulkan oleh " + pertama.nama);  
    }  
    break;
```

hasilnya

```
Menu :  
1. mengumpulkan tugas  
2. menilai tugas  
3. melihat tugas teratas  
4. melihat daftar tugas  
5. melihat tugas terbawah  
Pilih menu : 5  
Tugas pertama dikumpulkan oleh joko
```

5. Ini hasilnya

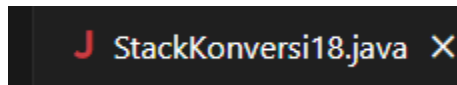
```
Menu :  
1. mengumpulkan tugas  
2. menilai tugas  
3. melihat tugas teratas  
4. melihat daftar tugas  
5. melihat tugas terbawah  
6. melihat jumlah tugas  
Pilih menu : 6  
Jumlah tugas yang sudah dikumpulkan: 2  
PS C:\Users\USER\OneDrive\Algoritma Struktur data\JOBSHEET 9>   
eted. Java: Ready
```

## C. Percobaan 2

1. Tambahkan method konversidesimalkebiner

```
public String KonversiDesimalKeBinner (int nilai){  
    StackKonversi18 stackBiner = new StackKonversi18(size:32);  
    while (nilai > 0) {  
        int sisa = nilai % 2;  
        stackBiner.push(sisa);  
        nilai = nilai / 2;  
    }  
    String biner = new String();  
    while (!stackBiner.isEmpty()) {  
        biner += stackBiner.pop();  
    }  
    return biner;  
}
```

2. Buat file baru Bernama StackKonversiNoabsen



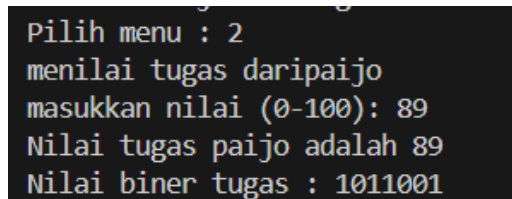
3. Tambahkan kode dibawah ini

```
int [] tumpukanBiner;  
int top;  
int size;  
  
public StackKonversi18(int size) {  
    this.size = 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()) {  
        top++;  
        tumpukanBiner[top] = data;  
    } else {  
        System.out.println(x:"Stack penuh");  
    }  
}  
  
public int pop() {  
    if (isEmpty()) {  
        System.out.println(x:"Stack kosong ! tidak ada tugas untuk dinilai.");  
        return -1;  
    } else {  
        int data = tumpukanBiner[top];  
        top--;  
        return data;  
    }  
}
```

4. Tambahkan kode seperti dibawah ini pada class Mahasiswademo18

```
String biner = stack.KonversiDesimalKeBinner(nilai);  
System.out.printf("Nilai biner tugas : " + biner);
```

5. Hasilnya



```
Pilih menu : 2  
menilai tugas daripaijo  
masukkan nilai (0-100): 89  
Nilai tugas paijo adalah 89  
Nilai biner tugas : 1011001
```

#### D. Pertanyaan

1. Diawali dengan inisialisasi stack terlebih dahulu agar bisa menyimpan hasil bagi perulangan pertama

```
while (nilai > 0) {  
    int sisa = nilai % 2;  
    stackBiner.push(sisa);  
    nilai = nilai / 2;  
}
```

jika nilai lebih besar dari 0 maka akan dibagi 2 dimana nilai sisa akan dimodulo 2 lalu di push kedalam stack lalu nilai dibagi menjadi 2, proses ini akan terus berulang hingga nilai menjadi 0

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

Kode diatas digunakan untuk mengambil nilai satu persatu dari stack menggunakan pop(), nantinya ini akan membantu saat mengembalikan digit dalam urutan biner sehingga bisa tersusun secara benar lalu digabungkan ke string biner lalu String hasil konversi dikembalikan ke pemanggil method.

2. Hasilnya sama saja, dikarenakan di kode sebelumnya kode akan dijalankan jika nilai lebih besar dari 0, dalam bilangan decimal (bukan rill(tanpa koma)) batas terbawahnya adalah 0 jadi otomatis bila tidak sama dengan 0 maka variable nilai memiliki value diatas 0



## E. Latihan

### 1. Code

#### a) Surat18

```
1 public class Surat18 {
2     String idSurat;
3     String namaMahasiswa;
4     String kelas;
5     char jenisIzin;
6     int durasi;
7
8     public Surat18() {
9     }
10
11     public Surat18(String idSurat, String namaMahasiswa, String kelas, char jenisIzin, int durasi) {
12         this.idSurat = idSurat;
13         this.namaMahasiswa = namaMahasiswa;
14         this.kelas = kelas;
15         this.jenisIzin = jenisIzin;
16         this.durasi = durasi;
17     }
18
19     public String getNamaMahasiswa() {
20         return namaMahasiswa;
21     }
22
23     public void tampilkanSurat() {
24         System.out.println("ID Surat: " + idSurat);
25         System.out.println("Nama Mahasiswa: " + namaMahasiswa);
26         System.out.println("Kelas: " + kelas);
27         System.out.println("Jenis Izin: " + jenisIzin);
28         System.out.println("Durasi: " + durasi + " hari");
29         System.out.println(x:"-----");
30     }
31 }
32 }
```

#### b) Stack surat 18

```
1 public class StackSurat18 {
2     int top;
3     int size;
4
5     public StackSurat18(int size) {
6         this.size = size;
7         stack = new Surat18[size];
8         top = -1;
9     }
10
11     public boolean isFull() {
12         if (top == size - 1) {
13             return true;
14         } else {
15             return false;
16         }
17     }
18
19     public boolean isEmpty() {
20         if (top == -1) {
21             return true;
22         } else {
23             return false;
24         }
25     }
26
27     public void push(Surat18 surat) {
28         if (!isFull()) {
29             top++;
30             stack[top] = surat;
31         } else {
32             System.out.println(x:"Stack penuh ! tidak bisa menambahkan surat izin lagi.");
33         }
34     }
35 }
```

```

public SuratIs pop() {
    if (!isEmpty()) {
        SuratIs surat = stack[top];
        top--;
        return surat;
    } else {
        System.out.println("Stack kosong ! tidak ada surat izin untuk diambil.");
        return null;
    }
}

public SuratIs peek() {
    if (!isEmpty()) {
        return stack[top];
    } else {
        System.out.println("Stack kosong ! tidak ada surat izin yang dikumpulkan.");
        return null;
    }
}

public void terimaturat() {
    for (int i = 0; i <= top; i++) {
        stack[i].tampilkanSurat();
    }
}

public void prosesSurat() {
    SuratIs surat = pop();
    if (surat != null) {
        System.out.println("Memproses surat izin berikut.");
        surat.tampilkanSurat();
    }
}

```

```

public void prosesSurat() {
    SuratIs surat = pop();
    if (surat != null) {
        System.out.println("Memproses surat izin berikut.");
        surat.tampilkanSurat();
    }
}

public void lihatSuratTerakhir() {
    SuratIs surat = peek();
    if (surat != null) {
        System.out.println("surat izin terakhir.");
        surat.tampilkanSurat();
    }
}

public void cariSurat(String namaMahaasiswa) {
    boolean ditemukan = false;
    for (int i = 0; i <= top; i++) {
        if (stack[i].namaMahaasiswa.equalsIgnoreCase(namaMahaasiswa)) {
            System.out.println("Surat ditemukan.");
            stack[i].tampilkanSurat();
            ditemukan = true;
        }
    }
    if (!ditemukan) {
        System.out.println("Surat dengan nama mahasiswa " + namaMahaasiswa + " tidak ditemukan.");
    }
}
}

```

### c) Surat izin main

```

import java.util.Scanner;

public class SuratIzinMain {
    RunMain(DaftarMain)
    public static void main(String[] args) {
        Run(DaftarMain)
        StackSuratIs stack = new StackSuratIs(5);
        Scanner scan = new Scanner(System.in);
        int pilih;

        do {
            System.out.println("=====");
            System.out.println("Menu : ");
            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 izin");
            System.out.println("5. keluar");
            System.out.print("Pilih menu : ");
            pilih = scan.nextInt();
            scan.nextLine();

            switch (pilih) {
                case 1:
                    System.out.print("IdSurat : ");
                    String IdSurat = scan.nextLine();
                    System.out.print("Nama Mahasiswa : ");
                    String namaMahaasiswa = scan.nextLine();
                    System.out.print("Kelas : ");
                    String kelas = scan.nextLine();
                    System.out.print("Jenis Izin (I/S/A) : ");
                    char jenisIzin = scan.nextLine().charAt(0);
                    System.out.print("Durasi (hari) : ");
                    int durasi = scan.nextInt();
                    SuratIs surat = new SuratIs(IdSurat, namaMahaasiswa, kelas, jenisIzin, durasi);
                    stack.push(surat);
                    System.out.printf("Surat izin %s berhasil diterima\n", surat.namaMahaasiswa);

                case 2:
                    SuratIs proses = stack.pop();
                    if (proses != null) {
                        System.out.println("Proses surat izin dari " + proses.namaMahaasiswa);
                        System.out.printf("Surat izin %s telah diproses\n", proses.namaMahaasiswa);
                        break;
                    }
                case 3:
                    SuratIs lihat = stack.peek();
                    if (lihat != null) {
                        System.out.println("Surat izin terakhir diterima dari " + lihat.namaMahaasiswa);
                        lihat.tampilkanSurat();
                    }
                    break;
                case 4:
                    System.out.print("Masukkan NIM yang dicari : ");
                    String carinama = scan.nextLine();
                    boolean ditemukan = false;
                    for (int i = 0; i <= stack.top; i++) {
                        if (stack.stack[i].namaMahaasiswa.equals(carinama)) {
                            System.out.println("Surat izin ditemukan.");
                            stack.stack[i].tampilkanSurat();
                            ditemukan = true;
                            break;
                        }
                    }
                    if (!ditemukan) {
                        System.out.println("Surat izin dengan NIM " + carinama + " tidak ditemukan.");
                    }
                    break;
                case 5:
                    System.out.println("Keluar dari program.");
                    break;
            }
        } while (pilih != 5);
        scan.close();
    }
}

```

d) Hasil

```
=====
Menu :
=====
1. terima surat izin
2. proses surat izin
3. lihat surat izin terakhir
4. cari surat izin
5. keluar
Pilih menu : 1
idSurat : 224
Nama Mahasiswa : Arga
Kelas : 1b
Jenis Izin (I/S/A) : s
Durasi (hari) : 2
Surat izin Arga berhasil diterima
```

```
=====
1. terima surat izin
2. proses surat izin
3. lihat surat izin terakhir
4. cari surat izin
5. keluar
Pilih menu : 1
idSurat : 5564
Nama Mahasiswa : deka
Kelas : 2b
Jenis Izin (I/S/A) : A
Durasi (hari) : 1
Surat izin deka berhasil diterima
```

```
=====
Menu :
=====
1. terima surat izin
2. proses surat izin
3. lihat surat izin terakhir
4. cari surat izin
5. keluar
Pilih menu : 3
Surat izin terakhir diterima dari deka
ID Surat: 5564
Nama Mahasiswa: deka
```

```
=====
1. terima surat izin
2. proses surat izin
3. lihat surat izin terakhir
4. cari surat izin
5. keluar
Pilih menu : 4
Masukkan NIM yang dicari: deka
Surat izin ditemukan:
ID Surat: 5564
Nama Mahasiswa: deka
Kelas: 2b
Jenis Izin: A
Durasi: 1 hari
```