

Nama : Zaskia Maulidina Mutiara Hati

NIM : 244107060056

Kelas : SIB 1A

JOBSHEET 9 ALGORITMA STRUKTUR DATA

Percobaan 1

- **Class Mahasiswa24**

```
jobsheet_9 > Mahasiswa24.java > Mahasiswa24
1  package jobsheet_9;
2
3  public class Mahasiswa24 {
4      String nim;
5      String nama;
6      String kelas;
7      int nilai;
8
9      Mahasiswa24(){
10
11     }
12
13     Mahasiswa24(String nim, String nama, String kelas) {
14         this.nim = nim;
15         this.nama = nama;
16         this.kelas = kelas;
17         nilai = -1;
18     }
19
20     void tugasDinilai(int nilai) {
21         this.nilai = nilai;
22     }
23 }
```

- **Class StackTugasMahasiswa24**

```
jobsheet_9 > StackTugasMahasiswa24.java > StackTugasMahasiswa24 > print()
1  package jobsheet_9;
2
3  public class StackTugasMahasiswa24 {
4      Mahasiswa24[] stack;
5      int top;
6      int size;
7
8      public StackTugasMahasiswa24(int size) {
9          this.size = size;
10         stack = new Mahasiswa24[size];
11         top = -1;
12     }
13
14     public boolean isFull() {
15         if (top == size - 1) {
16             return true;
17         } else {
18             return false;
19         }
20     }
21
22     public boolean isEmpty() {
23         if (top == -1) {
24             return true;
25         } else {
26             return false;
27         }
28     }
29 }
```

```

30     public void push(Mahasiswa24 mhs) {
31         if (!isFull()) {
32             top++;
33             stack[top] = mhs;
34         } else {
35             System.out.println(x:"Stack penuh! Tidak bisa menambahkan tugas lagi.");
36         }
37     }
38
39     public Mahasiswa24 pop() {
40         if (!isEmpty()) {
41             Mahasiswa24 m = stack[top];
42             top--;
43             return m;
44         } else {
45             System.out.println(x:"Stack kosong! Tidak ada tugas untuk dinilai.");
46             return null;
47         }
48     }
49
50     public Mahasiswa24 peek() {
51         if (!isEmpty()) {
52             return stack[top];
53         } else {
54             System.out.println(x:"Stack kosong! Tidak ada tugas yang dikumpulkan.");
55             return null;
56         }
57     }

```

```

59     public void print() {
60         for (int i = 0; i <= top; i++) {
61             System.out.println(stack[i].nama + "\t" + stack[i].nim + "\t" + stack[i].kelas );
62         }
63         System.out.println(x:"");
64     }
65 }
66

```

- Class MahasiswaDemo24

```
obsheet_9 > MahasiswaDemo24.java > MahasiswaDemo24 > main(String[])
1 package jobsheet_9;
2 import java.util.Scanner;
3 public class MahasiswaDemo24 {
    Run | Debug
4     public static void main(String[] args) {
5         Scanner input24 = new Scanner(System.in);
6         StackTugasMahasiswa24 stack = new StackTugasMahasiswa24(size:5);
7
8         int pilih;
9         do {
10             System.out.println(x:"\nMenu");
11             System.out.println(x:"1. Mengumpulkan Tugas");
12             System.out.println(x:"2. Menilai Tugas");
13             System.out.println(x:"3. Melihat Tugas Teratas");
14             System.out.println(x:"4. Melihat Daftar Tugas");
15             System.out.print(s:"Pilih menu: ");
16             pilih = input24.nextInt();
17             input24.nextLine();
18
19             switch (pilih) {
20                 case 1:
21                     System.out.print(s:"Nama: ");
22                     String nama = input24.nextLine();
23                     System.out.print(s:"NIM: ");
24                     String nim = input24.nextLine();
25                     System.out.print(s:"Kelas: ");
26                     String kelas = input24.nextLine();
27                     Mahasiswa24 mhs = new Mahasiswa24(nim, nama, kelas);
28                     stack.push(mhs);
29                     System.out.printf(format:"Tugas %s berhasil dikumpulkan\n", mhs.nama);
30                     break;
31
32                 case 2:
33                     Mahasiswa24 dinilai = stack.pop();
34                     if (dinilai != null) {
35                         System.out.println("Menilai tugas dari " + dinilai.nama);
36                         System.out.print(s:"Masukkan nilai (0-100): ");
37                         int nilai = input24.nextInt();
38                         dinilai.tugasDinilai(nilai);
39                         System.out.printf(format:"Nilai Tugas %s adalah %d\n", dinilai.nama, nilai);
40                     }
41                     break;
42                 case 3:
43                     Mahasiswa24 lihat = stack.peek();
44                     if (lihat != null) {
45                         System.out.println("Tugas terakhir dikumpulkan oleh " + lihat.nama);
46                     }
47                     break;
48                 case 4:
49                     System.out.println(x:"Daftar semua tugas");
50                     System.out.println(x:"Nama\tNIM\tKelas");
51                     stack.print();
52                     break;
53                 default:
54                     System.out.println(x:"Pilihan tidak valid.");
55             }
56         } while (pilih >= 1 && pilih <= 4);
57     }
}
```

- **Output**

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

```
Menu
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
Pilih menu: 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 menu: 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 menu: 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 menu: 4
Daftar semua tugas
```

Nama	NIM	Kelas
Dila	1001	1A
Erik	1002	1B

Pertanyaan 1

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

```
public void print() {
    if (top == -1) {
        System.out.println(x:"Stack kosong! Tidak ada tugas yang dikumpulkan.");
        return;
    } else {
        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:"");
    }
}
```

```
Menu
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
Pilih menu: 4
Daftar semua tugas
Nama    NIM    Kelas
Tika    1003    1C
Erik    1002    1B
Dina    1001    1A
```

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

- 5

```
StackTugasMahasiswa24 stack = new StackTugasMahasiswa24(size:5);
```

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

- Karena stack punya kapasitas terbatas, maka perlu pengecekan !isFull() sebelum menambahkan elemen. Kalau kondisi if-else dihapus, program bisa menambahkan data saat stack sudah penuh, yang menyebabkan error seperti `ArrayIndexOutOfBoundsException`. Jadi, pengecekan ini penting untuk mencegah error dan menjaga program tetap aman.

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!

```
72 public void peekBottom() {
73     if (!isEmpty()) {
74         System.out.println("Mahasiswa yang pertama mengumpulkan tugas adalah " + stack[0].nama);
75     } else {
76         System.out.println(x:"Stack kosong! Tidak ada tugas yang dikumpulkan.");
77     }
78 }
79 }
```

```
case 5:
    stack.peekBottom();
    break;
```

```
Menu
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Mahasiswa Yang Pertama Mengumpulkan Tugas
Pilih menu: 5
Mahasiswa yang pertama mengumpulkan tugas adalah Dina
```

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

```
81 public int getSize() {  
82     return top + 1;  
83 }
```

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

```
Menu  
1. Mengumpulkan Tugas  
2. Menilai Tugas  
3. Melihat Tugas Teratas  
4. Melihat Daftar Tugas  
5. Mahasiswa Yang Pertama Mengumpulkan Tugas  
6. Jumlah Tugas Yang Sudah Dikumpulkan  
Pilih menu: 6  
Jumlah tugas yang sudah dikumpulkan: 1
```

6. Commit dan push kode program ke Github
- <https://github.com/iakmorales/ASD.git>

Percobaan 2

- Output

```
Menu
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Mahasiswa Yang Pertama Mengumpulkan Tugas
6. Jumlah Tugas Yang Sudah Dikumpulkan
Pilih menu: 2
Menilai tugas dari Tika
Masukkan nilai (0-100): 87
Nilai Tugas Tika adalah 87
Nilai Biner Tugas: 1010111
```

Pertanyaan 2

1. Jelaskan alur kerja dari method konversiDesimalKeBiner!

- a)

```
StackKonversi24 stack = new StackKonversi24();
```
- Inisialisasi Stack

- b)

```
while (nilai > 0) {
    int sisa = nilai % 2;
    stack.push(sisa);
    nilai = nilai / 2;
}
```
- Selama nilai > 0, dilakukan pembagian nilai dengan 2.
 - Sisa pembagian (nilai % 2) akan menjadi digit biner (0 atau 1).
 - Sisa ini dimasukkan ke dalam stack menggunakan push().
 - Nilai terus dibagi 2 sampai habis (nilai menjadi 0).

- c)

```
String biner = new String();
while (!stack.isEmpty()) {
    biner += stack.pop();
}
```
- Karena digit paling belakang dari bilangan biner adalah hasil pembagian terakhir, maka perlu dikeluarkan dari stack (LIFO - Last In First Out).
 - Stack di-pop satu per satu, dan setiap digit ditambahkan ke string biner.

```
return biner;
```

- d)
- Setelah semua digit diambil dari stack, string biner berisi representasi biner dari angka desimal awal.
 - Hasil ini dikembalikan sebagai output method.
2. Pada method konversiDesimalKeBiner, ubah kondisi perulangan menjadi while (kode != 0), bagaimana hasilnya? Jelaskan alasannya!
- Hasilnya akan error, karena variable kode tidak didefinisikan terlebih dahulu.

Tugas

a. Class Surat24

```
jobsheet_9 > Surat24.java > Surat24
1  package jobsheet_9;
2
3  public class Surat24 {
4      String idSurat;
5      String namaMahasiswa;
6      String kelas;
7      char jenisIzin;
8      int durasi;
9
10     Surat24() {
11
12     }
13
14     Surat24(String idSurat, String namaMahasiswa, String kelas, char jenisIzin, int durasi) {
15         this.idSurat = idSurat;
16         this.namaMahasiswa = namaMahasiswa;
17         this.kelas = kelas;
18         this.jenisIzin = jenisIzin;
19         this.durasi = durasi;
20     }
21 }
```

b. Class StackSurat24

```
jobsheet_9 > StackSurat24.java > StackSurat24
1  package jobsheet_9;
2
3  public class StackSurat24 {
4      Surat24[] stack;
5      int top;
6      int size;
7
8      public StackSurat24(int size) {
9          this.size = size;
10         stack = new Surat24[size];
11         top = -1;
12     }
13
14     public boolean isFull() {
15         if (top == size - 1) {
16             return true;
17         } else {
18             return false;
19         }
20     }
21
22     public boolean isEmpty() {
23         if (top == -1) {
24             return true;
25         } else {
26             return false;
27         }
28     }
29
30     public void push(Surat24 srt) {
31         if (!isFull()) {
32             top++;
33             stack[top] = srt;
34         } else {
35             System.out.println(x:"Stack penuh! Tidak bisa menambahkan surat lagi.");
36         }
37     }
38
39     public Surat24 pop() {
40         if (!isEmpty()) {
41             Surat24 s = stack[top];
42             top--;
43             return s;
44         } else {
45             System.out.println(x:"Stack kosong! Tidak ada surat untuk diambil.");
46             return null;
47         }
48     }
49
50     public Surat24 peek() {
51         if (!isEmpty()) {
52             return stack[top];
53         } else {
54             System.out.println(x:"Stack kosong! Tidak ada surat yang tersedia.");
55             return null;
56         }
57     }
58 }
```

```

59 public void print() {
60     if (top == -1) {
61         System.out.println(x:"Stack kosong! Tidak ada surat yang tersedia.");
62     } else {
63         System.out.println(x:"Daftar surat dalam stack:");
64         for (int i = top; i >= 0; i--) {
65             System.out.println(stack[i].toString());
66         }
67     }
68 }
69
70

```

c. Class SuratDemo24

```

jobsheet_9 > SuratDemo24.java > SuratDemo24 > main(String[])
1  package jobsheet_9;
2
3  import java.util.Scanner;
4  public class SuratDemo24 {
5      Run | Debug
6      public static void main(String[] args) {
7          Scanner input24 = new Scanner(System.in);
8
9          StackSurat24 stack = new StackSurat24(size:5);
10
11         int pilihan;
12         do {
13             System.out.println(x:"\n==== Menu ===");
14             System.out.println(x:"1. Masukkan Data Surat");
15             System.out.println(x:"2. Verifikasi Surat");
16             System.out.println(x:"3. Melihat Surat Yang Terakhir Dikirim");
17             System.out.println(x:"4. Cari Surat Berdasarkan Nama Mahasiswa");
18             System.out.println(x:"5. Keluar");
19             System.out.print(s:"Pilih menu: ");
20             pilihan = input24.nextInt();
21             input24.nextLine();
22
23             switch (pilihan) {
24                 case 1:
25                     System.out.print(s:"ID Surat: ");
26                     String idSurat = input24.nextLine();
27                     System.out.print(s:"Nama Mahasiswa: ");
28                     String namaMahasiswa = input24.nextLine();
29                     System.out.print(s:"Kelas: ");
30                     String kelas = input24.nextLine();
31                     System.out.print(s:"Jenis Izin (I untuk Izin, S untuk Sakit): ");
32                     char jenisIzin = input24.next().charAt(index:0);
33                     System.out.print(s:"Durasi (dalam hari): ");
34                     int durasi = input24.nextInt();
35                     Surat24 surat = new Surat24(idSurat, namaMahasiswa, kelas, jenisIzin, durasi);
36                     stack.push(surat);
37                     System.out.println(x:"\nData surat berhasil dimasukkan.");
38                     break;
39                 case 2:
40                     Surat24 suratDiverifikasi = stack.pop();
41                     if (suratDiverifikasi != null) {
42                         System.out.println("\nSurat dari " + suratDiverifikasi.namaMahasiswa + " Dengan ID Surat "
43                             + suratDiverifikasi.idSurat + " telah diverifikasi.");
44                     }
45                     break;
46                 case 3:
47                     Surat24 suratTerakhir = stack.peek();
48                     if (suratTerakhir != null) {
49                         System.out.println("\nSurat terakhir dikirim oleh " + suratTerakhir.namaMahasiswa + " Dengan ID Surat "
50                             + suratTerakhir.idSurat);
51                     }
52                     break;
53             }
54         } while (pilihan != 5);
55     }
56 }

```

```

54         case 4:
55             System.out.print(s:"Masukkan nama mahasiswa yang dicari: ");
56             String namaDicari = input24.nextLine();
57             boolean ditemukan = false;
58             for (int i = 0; i <= stack.top; i++) {
59                 if (stack.stack[i].namaMahasiswa.equalsIgnoreCase(namaDicari)) {
60                     System.out.println("\nSurat ditemukan: ID Surat " + stack.stack[i].idSurat + ", Kelas " + stack.stack[i].kelas
61                                     + ", Jenis Izin " + stack.stack[i].jenisIzin + ", Durasi " + stack.stack[i].durasi);
62                     ditemukan = true;
63                     break;
64                 }
65             }
66             if (!ditemukan) {
67                 System.out.println("\nSurat tidak ditemukan untuk nama mahasiswa " + namaDicari);
68             }
69             break;
70         case 5:
71             System.out.println(x:"\nTerimakasih telah menggunakan program.");
72             break;
73         default:
74             System.out.println(x:"\nPilihan tidak valid!");
75     }
76 } while (pilihan != 5);
77 }
78 }

```

d. Output

```

==== Menu ====
1. Masukkan Data Surat
2. Verifikasi Surat
3. Melihat Surat Yang Terakhir Dikirim
4. Cari Surat Berdasarkan Nama Mahasiswa
5. Keluar
Pilih menu: 1

ID Surat: 1001
Nama Mahasiswa: Dina
Kelas: 1A
Jenis Izin (I untuk Izin, S untuk Sakit): I
Durasi (dalam hari): 2

Data surat berhasil dimasukkan.

==== Menu ====
1. Masukkan Data Surat
2. Verifikasi Surat
3. Melihat Surat Yang Terakhir Dikirim
4. Cari Surat Berdasarkan Nama Mahasiswa
5. Keluar
Pilih menu: 1

ID Surat: 1002
Nama Mahasiswa: Erik
Kelas: 1B
Jenis Izin (I untuk Izin, S untuk Sakit): S
Durasi (dalam hari): 1

Data surat berhasil dimasukkan.

```

```

==== Menu ====
1. Masukkan Data Surat
2. Verifikasi Surat
3. Melihat Surat Yang Terakhir Dikirim
4. Cari Surat Berdasarkan Nama Mahasiswa
5. Keluar
Pilih menu: 2

Surat dari Erik Dengan ID Surat 1002 telah diverifikasi.

==== Menu ====
1. Masukkan Data Surat
2. Verifikasi Surat
3. Melihat Surat Yang Terakhir Dikirim
4. Cari Surat Berdasarkan Nama Mahasiswa
5. Keluar
Pilih menu: 3

Surat terakhir dikirim oleh Dina Dengan ID Surat 1001

==== Menu ====
1. Masukkan Data Surat
2. Verifikasi Surat
3. Melihat Surat Yang Terakhir Dikirim
4. Cari Surat Berdasarkan Nama Mahasiswa
5. Keluar
Pilih menu: 4

Masukkan nama mahasiswa yang dicari: Dina

Surat ditemukan: ID Surat 1001, Kelas 1A, Jenis Izin I, Durasi 2

```

```

==== Menu ====
1. Masukkan Data Surat
2. Verifikasi Surat
3. Melihat Surat Yang Terakhir Dikirim
4. Cari Surat Berdasarkan Nama Mahasiswa
5. Keluar
Pilih menu: 5

Terimakasih telah menggunakan program.

```