



**LAPORAN PRAKTIKUM
ALGORITMA DAN STRUKTUR DATA
JOBSHEET 7**



NAMA = RAKAGALI RESDA KRISANDI PUTRA
KELAS = TI -1B / 19
NIM = 244107020136



JOBSHEET IX

STACK

1. Tujuan Praktikum

Setelah melakukan materi praktikum ini, mahasiswa mampu:

1. Membuat struktur data Stack
2. Menerapkan algoritma Stack ke dalam program Java

2. Praktikum

2.1 Percobaan 1: Mahasiswa Mengumpulkan Tugas

```
Pilih : 1
Nama :Tika
Nim :1003
Kelas:1c
Tugas Tika berhasil dikumpulkan
Pilih Menu
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Lihat Tugas Teratas
4. Melihat Daftar Tugas
Pilih : 4
Daftar semua tugas
Nama    NIM    Kelas
Dila    1001    1A
Erik    1002    1B
Tika    1003    1c
```



```
Pilih Menu
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Lihat Tugas Teratas
4. Melihat Daftar Tugas
Pilih : 2
Menilai Tugas dari Tika
Masukan nilai (0-100) :
87
Nilai Tugas Tika = 87
Pilih Menu
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Lihat Tugas Teratas
4. Melihat Daftar Tugas
Pilih : 4
Daftar semua tugas
Nama    NIM    Kelas
Dila    1001    1A
Erik    1002    1B

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

2.1.1 Pertanyaan

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

```
public void print(){
    for (int i = 0; i <= top; i++) {
```

diubah jadi

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

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

5 stack karna sudah dideklarasikan secara statis

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

3. Mengapa perlu pengecekan kondisi `!isFull()` pada method `push`? Kalau kondisi if-else tersebut



dihapus, apa dampaknya?

Karena Stack menggunakan array berukuran tetap (`Mahasiswa19[] stack = new Mahasiswa19[size];`), kita harus mengecek apakah stack sudah penuh sebelum menambahkan elemen baru. Jika tidak dicek, dan kita coba push data saat `top == size - 1`, maka akan terjadi `ArrayIndexOutOfBoundsException` karena mencoba menulis di luar batas array.

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 Mahasiswa19 bottom() {
    if (!isEmpty()) {
        return stack[0]; // data pertama kali masuk
    } else {
        System.out.println(x:"Stack kosong! Tidak ada tugas yang dikumpulkan.");
        return null;
    }
}
```

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

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

```
5. Melihat Tugas Terbawah
Pilih : 1
Nama :asdas
Nim :234
Kelas:fsd
Tugas asdas berhasil dikumpulkan
Pilih Menu
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Lihat Tugas Teratas
4. Melihat Daftar Tugas
5. Melihat Tugas Terbawah
Pilih : 5
Tugas pertama dikumpulkan oleh asdas
```

- 6.
7. Commit dan push kode program ke Github



2.2 Percobaan 2: Konversi Nilai Tugas ke Biner

2.2.1 Verifikasi Hasil Percobaan

Cocokkan hasil compile kode program Anda dengan gambar berikut ini.

```
fienu:
1. lee gtlmpulka Tugas
2. le+ilai Tugas
3. melihat Tugas Teratas
4. lelihat Daftar
Tugas Pilih: 2
mer ilai tugas dari Tika
Hasukkan 'nilai (0-100): 87
)ilai Tugas Tika adalah 87
Milai Biner Tugas: 1010111
```

```
5. Melihat Tugas Terbawah
Pilih : 1
Nama :tika
Nim :123
Kelas:123
Tugas tika berhasil dikumpulkan
Pilih Menu
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Lihat Tugas Teratas
4. Melihat Daftar Tugas
5. Melihat Tugas Terbawah
Pilih : 2
Menilai Tugas dari tika
Masukan nilai (0-100) :
87
Nilai Tugas tika = 87
Nilai Biner Tugas : 1010111
Pilih Menu
```



2.2.2 Pertanyaan

1. Jelaskan alur kerja dari method konversiDesimalKeBiner!
 - **Inisialisasi stack** untuk menyimpan sisa hasil pembagian (0 atau 1).
 - Lakukan **pembagian nilai dengan 2**, simpan sisa (nilai % 2) ke dalam stack.
 - Ulangi sampai nilai == 0.
 - Setelah semua sisa disimpan, **keluarkan isi stack satu per satu** dan gabungkan ke String biner.
 - Karena stack LIFO, sisa pertama keluar terakhir —> hasil biner akan benar.
2. Pada method konversiDesimalKeBiner, ubah kondisi perulangan menjadi while (kode != 0), bagaimana hasilnya? Jelaskan alasannya!

Bisa menyebabkan infinite loop jika nilai negatif, karena nilai / 2 dari bilangan negatif tetap negatif, tidak pernah mencapai 0.

2.4 Latihan Praktikum

Waktu : 90 Menit

Mahasiswa mengajukan surat izin (karena sakit atau keperluan lain) setiap kali tidak mengikuti perkuliahan. Surat terakhir yang masuk akan diproses atau divalidasi lebih dulu oleh admin Prodi. Perhatikan class diagram berikut.

Surat<NoAbsen>
idsurat: String namaMahasiswa: String kelas: String jenisizin: char durasi: int
Surat<NoAbsen>() Surat<NoAbsen>(idsurat: String, namaMahasiswa: String, kelas: String, jenisizin: char, durasi: int)

Atribut jenisizin digunakan untuk menyimpan keterangan ijin mahasiswa (S: sakit atau I: izin keperluan lain) dan durasi untuk menyimpan lama waktu izin.

Berdasarkan class diagram tersebut, implementasikan class Surat dan tambahkan class



StackSurat untuk mengelola data Surat. Pada class yang memuat method main, buat pilihan menu berikut:

1. Terima Surat Izin untuk memasukkan data surat
2. Proses Surat Izin untuk memproses atau memverifikasi surat
3. Lihat Surat Izin Terakhir untuk melihat surat teratas
4. Cari Surat untuk mencari ada atau tidaknya surat izin berdasarkan **nama mahasiswa**

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

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

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

```
Menu:
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat
5. Keluar
Pilih: 4
Masukkan nama mahasiswa: raka
Surat dari raka ditemukan.
```



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