

POLITEKNIK NEGERI MALANG
TEKNOLOGI INFORMASI
TEKNIK INFORMATIKA



Nama: Abdul Rahman Hanif Darmawan

NIM: 244107020232

Kelas: TI-1A

Prodi: D4-TEKNIK INFORMATIKA

Jobsheet 9

2.1 Percobaan 1 : Mahasiswa Mengumpulkan Tugas

2.1.2 Verifikasi Hasil Percobaan

```
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 : 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
```

2.1.3 Pertanyaan

1. Lakukan perbaikan pada kode program, sehingga keluaran yang dihasilkan sama dengan verifikasi hasil percobaan! Bagian mana yang perlu diperbaiki?
 - Keluaran sudah sama dengan hasil verifikasi hasil percobaan
2. Berapa banyak data tugas mahasiswa yang dapat ditampung di dalam Stack? Tunjukkan potongan kode programnya!
 - Stack dapat menampung 5 data mahasiswa
3. Mengapa perlu pengecekan kondisi `!isFull()` pada method `push`? Kalau kondisi `if-else` tersebut dihapus, apa dampaknya?
 - Untuk mencegah menambahkan data saat stack sudah penuh, jika dihapus maka bisa menyebabkan error.
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 Mahasiswa02 peekBottom() {  
    if (!isEmpty()) {  
        return stack[0];  
    } else {  
        System.out.println("Stack kosong! Tidak ada tugas yang  
dikumpulkan");  
        return null;  
    }  
}
```

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

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

2.2 Percobaan 2: Konversi Nilai Tugas ke Biner

2.2.2 Verifikasi Hasil Percobaan

```
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
Nilai Biner Tugas: 1010111
```

2.2.3 Pertanyaan

1. Jelaskan alur kerja dari method konversiDesimalKeBiner!
 - Membagi nilai dengan 2 berulang kali lalu menyimpan sisa pembagian ke stack dan setelah itu mengambil isi stack untuk membentuk angka biner dari hasil pembagian.
2. Pada method konversiDesimalKeBiner, ubah kondisi perulangan menjadi while (kode != 0), bagaimana hasilnya? Jelaskan alasannya!
 - Jika kondisi diubah menjadi while (kode != 0), program akan error karena variabel kode tidak ada. jika kode diganti nilai, hasilnya tetap sama karena nilai != 0 dan nilai > 0 setara untuk nilai positif.

2.4 Latihan Praktikum

- Surat.java

```
package Tugas;
public class Surat {
    String idSurat;
    String namaMahasiswa;
    String kelas;
    char jenisIzin; // 'S' = Sakit, 'I' = Izin
    int durasi;

    public Surat(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;
    }
}
```

- StackSurat.java

```
package Tugas;
public class StackSurat {
    Surat[] stack;
    int top;
    int size;

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

    public void push(Surat surat) {
        if (top < size - 1) {
            stack[++top] = surat;
        } else {
            System.out.println("Stack penuh, tidak bisa menambahkan surat.");
        }
    }

    public Surat pop() {
        if (top >= 0) {
            return stack[top--];
        } else {
            System.out.println("Stack kosong.");
            return null;
        }
    }

    public Surat peek() {
        if (top >= 0) {
            return stack[top];
        } else {
            return null;
        }
    }

    public void cariSurat(String nama) {
        boolean ditemukan = false;
        for (int i = top; i >= 0; i--) {
            if (stack[i].namaMahasiswa.equalsIgnoreCase(nama)) {
                System.out.printf("Ditemukan: %s - Izin %c selama %d hari\n",
                                   stack[i].namaMahasiswa, stack[i].jenisIzin,
                                   stack[i].durasi);
                ditemukan = true;
            }
        }
        if (!ditemukan) {
            System.out.println("Surat tidak ditemukan.");
        }
    }
}
```

- SuratMain.java

```
package Tugas;
import java.util.Scanner;
public class SuratMain {
    public static void main(String[] args) {
        StackSurat stack = new StackSurat(10);
        Scanner scan = new Scanner(System.in);
        int pilih;
        do {
            System.out.println("\nMenu:");
            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 berdasarkan Nama");
            System.out.println("0. Keluar");
            System.out.print("Pilih: ");
            pilih = scan.nextInt(); scan.nextLine();
            switch (pilih) {
                case 1:
                    System.out.print("ID Surat: ");
                    String id = scan.nextLine();
                    System.out.print("Nama Mahasiswa: ");
                    String nama = scan.nextLine();
                    System.out.print("Kelas: ");
                    String kelas = scan.nextLine();
                    System.out.print("Jenis Izin (S/I): ");
                    char jenis = scan.nextLine().toUpperCase().charAt(0);
                    System.out.print("Durasi (hari): ");
                    int durasi = scan.nextInt(); scan.nextLine();
                    stack.push(new Surat(id, nama, kelas, jenis, durasi));
                    System.out.println("Surat berhasil ditambahkan.");
                    break;
                case 2:
                    Surat diproses = stack.pop();
                    if (diproses != null) {
                        System.out.printf("Memverifikasi surat dari %s (%c selama %d hari)\n",
                            diproses.namaMahasiswa, diproses.jenisIzin,
                            diproses.durasi);
                    }
                    break;
                case 3:
                    Surat lihat = stack.peek();
                    if (lihat != null) {
                        System.out.printf("Surat terakhir dari %s (%c selama %d hari)\n",
                            lihat.namaMahasiswa, lihat.jenisIzin, lihat.durasi);
                    } else {
                        System.out.println("Tidak ada surat dalam stack.");
                    }
                    break;
                case 4:
                    System.out.print("Masukkan nama mahasiswa: ");
                    String cariNama = scan.nextLine();
                    stack.cariSurat(cariNama);
                    break;
                case 0:
                    System.out.println("Program selesai.");
                    break;
                default:
                    System.out.println("Pilihan tidak valid.");
            }
        } while (pilih != 0);
    }
}
```

Hasil

```
Menu:
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat Izin berdasarkan Nama
0. Keluar
Pilih: 1
ID Surat: 234
Nama Mahasiswa: asdf
Kelas: 2a
Jenis Izin (S/I): s
Durasi (hari): 2
Surat berhasil ditambahkan.

Menu:
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat Izin berdasarkan Nama
0. Keluar
Pilih: 1
ID Surat: 928
Nama Mahasiswa: sdf
Kelas: 1b
Jenis Izin (S/I): i
Durasi (hari): 3
Surat berhasil ditambahkan.

Menu:
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat Izin berdasarkan Nama
0. Keluar
Pilih: 1
ID Surat: 8774
Nama Mahasiswa: lksjd
Kelas: 3a
Jenis Izin (S/I): i
Durasi (hari): 2
Surat berhasil ditambahkan.

Menu:
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat Izin berdasarkan Nama
0. Keluar
Pilih: 2
Memverifikasi surat dari lksjd (I selama 2 hari)

Menu:
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat Izin berdasarkan Nama
0. Keluar
Pilih: 3
Surat terakhir dari sdf (I selama 3 hari)

Menu:
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat Izin berdasarkan Nama
0. Keluar
Pilih: 4
Masukkan nama mahasiswa: asdf
Ditemukan: asdf - Izin S selama 2 hari
```

Link Github

[Pertemuan 10](#)