

LAPORAN
DASAR PEMROGRAMAN PRAKTIKUM
JOBSHEET 9



1E

Irfan Pandu Pratama (244107020193)

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
TAHUN 2025

kode program Mahasiswa13

```
public class Mahasiswa13 {
    String nim, nama, kelas;
    int nilai;

    Mahasiswa13() {
    }

    Mahasiswa13(String nama, String nim, String
kelas) {
        this.nama = nama;
        this.nim = nim;
        this.kelas = kelas;
        nilai = -1;
    }

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

kode program MahasiswaDemo13

```
import java.util.Scanner;
public class MahasiswaDemo13 {
    public static void main(String[] args) {
        StackTugasMahasiswa13 stack = new
StackTugasMahasiswa13(5);
        Scanner sc = 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 = sc.nextInt();
            sc.nextLine();
            switch (pilih) {
                case 1:
                    System.out.print("Nama: ");
                    String nama = sc.nextLine();
                    System.out.print("NIM: ");
                    String nim = sc.nextLine();
                    System.out.print("Kelas: ");
                    String kelas =
sc.nextLine();
                    Mahasiswa13 mhs = new
Mahasiswa13(nama, nim, kelas);
                    stack.push(mhs);
                    System.out.printf("Tugas %s
berhasil dikumpulkan\n", mhs.nama);
                    break;
                case 2:
                    Mahasiswa13 dinilai =
stack.pop();
                    if (dinilai != null) {
                        System.out.println("Meni
lai tugas dari " + dinilai.nama);
                        System.out.print("Masukk
an nilai (0-100): ");
                        int nilai =
sc.nextInt();
                        dinilai.tugasDinilai(nil
ai);
                        System.out.printf("Nilai
Tugas %s adalah %d\n", dinilai.nama, nilai);
                    }
            }
        }
    }
}
```

kode program StackTugasMahasiswa13

```
public class StackTugasMahasiswa13{
    Mahasiswa13
    [] stack;
    int size;
    int top;

    public StackTugasMahasiswa13
    (int size) {
        this.size = size;
        stack = new Mahasiswa13
        [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(Mahasiswa13
    mhs) {
        if (!isFull()) {
            top++;
            stack[top] = mhs;
        } else {
            System.out.println("Stack penuh!
tidak bisa menambahkan tugas lagi!");
        }
    }

    public Mahasiswa13
    pop() {
        if(!isEmpty()) {
            Mahasiswa13
            m = stack[top];
            top--;
            return m;
        } else {
            System.out.println("Stack kosong!
tidak ada tugas untuk dinilai");
        }
    }
}
```

Hasil dari kode program diatas :

```
Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
Pilih: 1
Nama: irfan
NIM: 12
Kelas: 1e
Tugas irfan berhasil dikumpulkan
```

```
Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
Pilih: 1
Nama: pandu
NIM: 123
Kelas: 2e
Tugas pandu berhasil dikumpulkan
```

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

Menu:

1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas

Pilih: 1

Nama: tama

NIM: 1234

Kelas: 3e

Tugas tama 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
tama	1234	3e

Pertanyaan Percobaan 1:

1. Perubahan kode program

```
public void print() {  
    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:"");  
}
```

2. Data disimpan pada kode berikut

```
StackTugasMahasiswa13 stack = new StackTugasMahasiswa13(size:5);
```

3. Agar tidak terjadi stack overflow atau data yang diinputkan melebihi kapasitas
4. Penambahan kode program untuk melihat tugas terbawah

```
public Mahasiswa13  
    peekBottom() {  
        if (!isEmpty()) {  
            return stack[0];  
        } else {  
            System.out.println(x:"Stack kosong! Tidak ada tugas yang dikumpulkan");  
            return null;  
        }  
    }  
}
```

5. Penambahan kode program untuk melihat banyak tugas

```
public int jumlahTugas() {  
    if (!isEmpty()) {  
        return top + 1;  
    } else {  
        return 0;  
    }  
}
```

Percobaan 2 (penambahan kode program)

Kode program yang ditambahkan sebagai berikut

```
public String konversiDesimalKeBiner(int nilai) {  
    StackKonversi13 stack = new StackKonversi13();  
    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;  
}
```



```
public class StackKonversi13 {  
    int[] tumpukanBiner;  
    int size;  
    int top;  
  
    public StackKonversi13() {  
        this.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()) {  
            System.out.println(x: "Stack penuh");  
        } else {  
            top++;  
            tumpukanBiner[top] = data;  
        }  
    }  
  
    public int pop() {  
        if (isEmpty()) {  
            System.out.println(x: "Stack kosong");  
            return -1;  
        } else {  
            int data = tumpukanBiner[top];  
            top--;  
            return data;  
        }  
    }  
}
```

Pertanyaan Percobaan 2

1. alur kerja method konversi decimal ke biner adalah mengambil bilangan desimal (misalnya kode) dan mengubahnya menjadi biner menggunakan operasi modulus dan pembagian. Setiap sisa hasil kode $\% 2$ disimpan ke dalam stack, lalu kode dibagi 2 terus-menerus hingga 0. Setelah semua sisa disimpan, isi stack dikeluarkan satu per satu (dengan pop) sehingga menghasilkan representasi biner dari bawah ke atas.
2. Jika kondisi perulangan diubah menjadi while (kode $\neq 0$), hasilnya akan tetap benar selama nilai awal kode lebih besar dari 0. Perubahan ini tetap membuat loop berhenti saat nilai kode mencapai 0, sehingga logika konversi biner masih berjalan. Namun, jika kode = 0 sejak awal, maka tidak akan ada sisa yang dimasukkan ke stack, sehingga hasilnya bisa kosong atau salah — untuk itu, perlu pengecekan khusus jika input awal adalah nol.

kode program Surat13

```
public class Surat13{
    String idSurat,namaMahasiswa,kelas;
    char jenisIzin;
    int durasi;
    boolean statusCek;

    Surat13
    () {
    }

    Surat13
    (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;
        this.statusCek = false;
    }

    void suratDicek() {
        this.statusCek = true;
    }
}
```

kode program MainSurat13

```
public class StackSurat13 {
    Surat13[] stack;
    int top;
    int size;

    public StackSurat13(int size) {
        this.size = size;
        stack = new Surat13[size];
        top = -1;
    }
    public boolean isFull() {
        return top == size - 1;
    }
    public boolean isEmpty() {
        return top == -1;
    }

    public void push(Surat13 srt) {
        if (!isFull()) {
            top++;
            stack[top] = srt;
        } else {
            System.out.println("Stack penuh!
Tidak bisa menambahkan surat lagi.");
        }
    }
    public Surat13 pop() {
        if (!isEmpty()) {
            Surat13 s = stack[top];
            top--;
            return s;
        } else {
            System.out.println("Stack kosong!
Tidak ada surat untuk dicek.");
            return null;
        }
    }

    public Surat13 peek() {
        if (!isEmpty()) {
            return stack[top];
        } else {
            System.out.println("Stack kosong!
Tidak ada surat yang dikumpulkan.");
            return null;
        }
    }
}
```

kode program StackSurat13

```
public class StackSurat13 {
    Surat13[] stack;
    int top;
    int size;

    public StackSurat13(int size) {
        this.size = size;
        stack = new Surat13[size];
        top = -1;
    }
    public boolean isFull() {
        return top == size - 1;
    }
    public boolean isEmpty() {
        return top == -1;
    }

    public void push(Surat13 srt) {
        if (!isFull()) {
            top++;
            stack[top] = srt;
        } else {
            System.out.println("Stack penuh!
Tidak bisa menambahkan surat lagi.");
        }
    }
    public Surat13 pop() {
        if (!isEmpty()) {
            Surat13 s = stack[top];
            top--;
            return s;
        } else {
            System.out.println("Stack kosong!
Tidak ada surat untuk dicek.");
            return null;
        }
    }

    public Surat13 peek() {
        if (!isEmpty()) {
            return stack[top];
        } else {
            System.out.println("Stack kosong!
Tidak ada surat yang dikumpulkan.");
            return null;
        }
    }
}
```

Hasil dari kode program

```
===== MENU =====
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat
5. Keluar Program
Pilih : 1

ID Surat : 1
Nama Mahasiswa : irfan
Kelas : 1e
Jenis perizinan (I/S) : s
Lama waktu izin : 2
Surat irfan berhasil dikumpulkan

===== MENU =====
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat
5. Keluar Program
Pilih : 1

ID Surat : 2
Nama Mahasiswa : pandu
Kelas : 2e
Jenis perizinan (I/S) : i
Lama waktu izin : 3
Surat pandu berhasil dikumpulkan

===== MENU =====
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat
5. Keluar Program
Pilih : 3

Surat terakhir dikumpulkan oleh : pandu
```

===== MENU =====

1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat
5. Keluar Program

Pilih : 2

Memverifikasi surat dari pandu

Surat dari mahasiswa pandu sudah di verifikasi.

===== MENU =====

1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat
5. Keluar Program

Pilih : 4

Masukkan nama mahasiswa yang ingin dicari : pandu

Surat Ditemukan :

ID Surat : 2

Nama Mahasiswa : pandu

Kelas : 2e

Jenis Perizinan : i

Lama Waktu Izin : 3

===== MENU =====

1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat
5. Keluar Program

Pilih : 5