

Laporan Hasil Praktikum Algoritma Struktur Data
Jobsheet 9



Febryan Akhmad Taajuddin

244107020180

Kelas 1E

Program Studi Teknik Informatika

Jurusan Teknologi Informasi

Politeknik Negeri Malang

2025

Percobaan 1

1. Buat class Mahasiswa10

```
public class Mahasiswa10 {
    String nama;
    String nim;
    String kelas;
    int nilai;

    Mahasiswa10() {

    }

    Mahasiswa10(String nama, String nim, String kelas) {
        this.nama = nama;
        this.nim = nim;
        this.kelas = kelas;
        nilai = -1;
    }

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

2. Buat class StackTugasMahasiswa dengan atribut stack, size, top

```
public class StackTugasMahasiswa10 {
    Mahasiswa10[] stack;
    int top;
    int size;
}
```

3. Buat Stack top untuk mengeset indeks awal

```
public StackTugasMahasiswa10(int size){
    this.size = size;
    stack = new Mahasiswa10[size];
    top = -1;
}
```

4. Buat method isFull bertipe boolean untuk mengecek apakah tumpukan tugas mahasiswa sudah terisi penuh sesuai kapasitas

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

5. buat method isEmpty bertipe boolean untuk mengecek apakah tumpukan tugas masih kosong

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

6. Buat method push. Method ini menerima parameter mhs yang berupa object dari class Mahasiswa

```
public void push (Mahasiswa10 mhs) {
    if (!isEmpty()) {
        top++;
        stack[top] = mhs;
    }else{
        System.out.println("Stack penuh! Tidak bisa menambahkan tugas lagi");
    }
}
```

7. Buat method Pop

```
public Mahasiswa10 pop(){
    if (!isEmpty()) {
        Mahasiswa10 m = stack[top];
        top--;
        return m;
    }else{
        System.out.println("Stack kosong! Tidak ada tugas untuk dinilai");
        return null;
    }
}
```

8. Buat method peek untuk dapat mengecek tumpukan tugas mahasiswa yang berada di posisi paling atas

```
public Mahasiswa10 peek() {
    if (!isEmpty()) {
        return stack[top];
    }else{
        System.out.println("Stack kosong! Tidak ada tugas yang dikumpulkan");
        return null;
    }
}
```

9. Tambahkan method print untuk dapat menampilkan semua daftar tugas mahasiswa pada Stack

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

10. Buat class MahasiswaDemo10 untuk program main

```
import java.util.Scanner;

public class MahasiswaDemo10 {
    public static void main(String[] args) {
        StackTugasMahasiswa10 stack = new StackTugasMahasiswa10(5);
        Scanner scan = 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 = 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();
                    Mahasiswa10 mhs = new Mahasiswa10(nama, nim,
                        kelas);
                    stack.push(mhs);
                    System.out.printf("Tugas %s berhasil
                        dikumpulkan\n", mhs.nama);
                    break;
                case 2:
                    Mahasiswa10 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;
            }
        } while (true);
    }
}
```

```

        case 3:
            Mahasiswa10 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\tNIM\tKelas");
            stack.print();
            break;
        default:
            System.out.println("Pilihan tidak valid.");
            break;
    }
} while (pilih >= 1 && pilih <= 4);
}
}

```

11. Run kode program

```

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: 3
Tugas terakhir dikumpulkan oleh Erik

```

```

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

```

Pertanyaan percobaan 1

1. Lakukan perbaikan pada kode program, sehingga keluaran yang dihasilkan sama dengan verifikasi hasil percobaan! Bagian mana yang perlu diperbaiki?
2. Berapa banyak data tugas mahasiswa yang dapat ditampung di dalam Stack? Tunjukkan potongan kode programnya!
3. Mengapa perlu pengecekan kondisi !isFull() pada method push? Kalau kondisi if-else tersebut dihapus, apa dampaknya?
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!
5. Tambahkan method untuk dapat menghitung berapa banyak tugas yang sudah dikumpulkan saat ini, serta tambahkan operasi menunya!
6. Commit dan push kode program ke Github

Jawaban percobaan 1

1. Modifikasi 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("");
}
```

- Output

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

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

2. Banyak tugas mahasiswa yang dapat ditampung sebanyak 5

- Kodenya:

```
StackTugasMahasiswa10 stack = new StackTugasMahasiswa10(5);
```

3. Pengecekan kondisi `isFull()` untuk memastikan bahwa Stack tidak melebihi kapasitas maksimum. Jika kondisi maksimum dipaksa untuk memasukkan data maka akan terjadi Stack Overflow yang akan terjadi error.
4. Modifikasi kode program
 - Class MahasiswaDemo10

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

- Class StackTugasMahasiswa10

```
public void melihatTugasTerbawah() {
    if (!isEmpty()) {
        System.out.println("Tugas pertama kali dikumpulkan oleh"
            + stack[0].nama);
    } else {
        System.out.println("Stack kosong! Tidak ada tugas yang"
            + "dikumpulkan");
    }
}
```

- Output

```
Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Melihat Tugas Terbawah
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
5. Melihat Tugas Terbawah
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
5. Melihat Tugas Terbawah
Pilih: 1
Nama: Roni
NIM: 1003
Kelas: 1E
Tugas Roni berhasil dikumpulkan
```

```
Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Melihat Tugas Terbawah
Pilih:
5
Tugas pertama kali dikumpulkan oleh Dila
```

5. Modifikasi kode program

- Class MahasiswaDemo10

```
        case 6:
            int banyakTugas = stack.banyakTugas();
            System.out.println("Banyak tugas yang sudah dikumpulkan:
" + banyakTugas);
            break;
```

- Class StackTugasMahasiswa10

```
public int banyakTugas(){
    return top + 1;
}
```

- Output

```
Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Melihat Tugas Terbawah
6. Banyak tugas yang sudah dikumpulkan
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
5. Melihat Tugas Terbawah
6. Banyak tugas yang sudah dikumpulkan
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
5. Melihat Tugas Terbawah
6. Banyak tugas yang sudah dikumpulkan
Pilih: 1
Nama: Roni
NIM: 1003
Kelas: 1E
Tugas Roni berhasil dikumpulkan

Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
5. Melihat Tugas Terbawah
6. Banyak tugas yang sudah dikumpulkan
Pilih: 6
Banyak tugas yang sudah dikumpulkan: 3
```

6. Commit ke Github



Percobaan 2

1. Tambahkan method konversiDesimalKeBiner dengan menerima parameter kode bertipe int

```
public String konversiDesimalKeBiner(int nilai){
    StackKonversi10 stack = new StackKonversi10();
    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;
}
```

2. Tambahkan empat method yaitu isEmpty, isFull, push, dan pull sebagai operasi utama Stack pada class StackKonversi

```
public class StackKonversi10 {
    int [] tumpukanBiner;
    int size;
    int top;

    public StackKonversi10(){
        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("Stack penuh");
        } else {
            top++;
            tumpukanBiner[top] = data;
        }
    }
    public int pop(){
        if (isEmpty()) {
            System.out.println("Stack kosong");
            return -1;
        }else{
            int data = tumpukanBiner[top];
            top--;
            return data;
        }
    }
}
```

3. Agar nilai tugas mahasiswa dikonversi ke dalam bentuk biner setelah dilakukan penilaian, maka tambahkan baris kode program pada method pop di class MahasiswaDemo

```
        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);
            String biner = stack.konversiDesimalKeBiner(nilai);
            System.out.println("Nilai Biner Tugas: "+ biner);
        }
```

4. Run kode program

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

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
Menilai tugas dari Tika
Masukkan nilai (0-100): 87
Nilai Tugas Tika adalah 87
Nilai Biner Tugas: 1010111

Menu:
1. Mengumpulkan Tugas
2. Menilai Tugas
3. Melihat Tugas Teratas
4. Melihat Daftar Tugas
Pilih: 5
Pilihan tidak valid.
```

Pertanyaan percobaan 2

1. Jelaskan alur kerja dari method konversiDesimalKeBiner!
2. Pada method konversiDesimalKeBiner, ubah kondisi perulangan menjadi while (kode != 0), bagaimana hasilnya?

Jawaban percobaan 2

1. Deklarasi Stack konversi digunakan untuk membuat objek untuk menyimpan sisa bagi nilai / 2, lalu dilakukan perulangan untuk mencari nilai sisa bagi yang akan di push ke stack dalam urutan terbalik (LIFO), lalu membuat tempat untuk menyimpan nilai yang akan dikirim ke stack, lalu memindahkan nilai yang dalam stack ke biner dengan urutan yang benar.
2. Program akan tetap berjalan karena nilai yang dimasukkan merupakan positif. Jika yang dimasukkan negatif, program bisa mengalami infinite loop. Namun, hal ini tidak menjadi masalah dalam konteks program ini karena nilai yang dimasukkan merupakan nilai tugas mahasiswa.

Latihan Praktikum

1. Berikut kode programnya
 - a. Class Surat10

```
public class Surat10 {
    String idSurat;
    String namaMahasiswa;
    String kelas;
    char jenisIzin;
    int durasi;

    Surat10() {

    }
    Surat10(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;
    }
}
```

b. Class StackSurat10

```
public class StackSurat10 {
    Surat10[] stack;
    int top;
    int size;

    public StackSurat10(int size) {
        this.size = size;
        stack = new Surat10[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(Surat10 surat) {
        if (!isFull()) {
            top++;
            stack[top] = surat;
        } else {
            System.out.println("Stack penuh! Tidak bisa menambahkan surat izin lagi");
        }
    }
}
```

```

    public Surat10 pop() {
        if (!isEmpty()) {
            Surat10 surat = stack[top];
            top--;
            return surat;
        } else {
            System.out.println("Stack kosong! Tidak ada surat izin
untuk diproses");
            return null;
        }
    }

    public Surat10 peek() {
        if (!isEmpty()) {
            return stack[top];
        } else {
            System.out.println("Stack kosong! Tidak ada surat izin
yang diajukan");
            return null;
        }
    }

    public void printCari(int hasil) {
        String keterangan = "";
        if (stack[hasil].jenisIzin == 'S') {
            keterangan = "Sakit";
        } else if (stack[hasil].jenisIzin == 'I') {
            keterangan = "Izin";
        } else {

        }

        System.out.printf("%-12s %-9s %-9s %-14s %-10d",
stack[hasil].idSurat,
        stack[hasil].namaMahasiswa, stack[hasil].kelas, keterangan,
stack[hasil].durasi);
    }

    public void print() {
        String keterangan = "";
        for (int i = top; i >= 0; i--) {
            if (stack[i].jenisIzin == 'S') {
                keterangan = "Sakit";
            } else if (stack[i].jenisIzin == 'I') {
                keterangan = "Izin";
            } else {
                System.out.println("Keterangan izin tidak valid!");
            }
            System.out.printf("%-12s %-9s %-9s %-14s %-10d\n",
stack[i].idSurat,
                stack[i].namaMahasiswa, stack[i].kelas, keterangan,
stack[i].durasi);
        }
    }
}

```

c. Class SearchingSurat10

```
public class SearchingSurat10 {
    public int searchingSurat(StackSurat10 stack, String cariNama, int
size) {
        int posisi = -1;
        for (int j = 0; j <= stack.top; j++) {
            if
(stack.stack[j].namaMahasiswa.equalsIgnoreCase(cariNama)) {
                posisi = j;
                break;
            }
        }
        return posisi;
    }
}
```

d. Class SuratDemo10

```
import java.util.Scanner;
public class SuratDemo10 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Masukkan jumlah mahasiswa: ");
        int size = sc.nextInt();
        sc.nextLine();
        StackSurat10 stack = new StackSurat10(size);
        int pilih;
        String terima = "";

        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");
            System.out.print("Pilih: ");
            pilih = sc.nextInt();
            sc.nextLine();

            switch (pilih) {
                case 1:
                    System.out.print("Id Surat: ");
                    String idSurat = sc.nextLine();
                    System.out.print("Nama: ");
                    String namaMahasiwa = sc.nextLine();
                    System.out.print("Kelas: ");
                    String kelas = sc.nextLine();
                    System.out.print("Jenis Izin: ");
                    char jenisIzin = sc.next().charAt(0);
                    System.out.print("Durasi Izin: ");
                    int durasi = sc.nextInt();
                    sc.nextLine();
                    Surat10 srt = new Surat10 (idSurat, namaMahasiwa,
kelas, jenisIzin, durasi);
                    stack.push(srt);
                    System.out.printf("Surat %s berhasil
dikumpulkan\n", srt.namaMahasiswa);
                    break;
```

```

        case 2:
            if (!stack.isEmpty()) {
                Surat10 trm = new Surat10();
                System.out.println("Daftar semua Surat:");
                System.out.println("ID Surat      Nama
Kelas      Keterangan      Durasi (Hari)");
                stack.print();
                System.out.println("");
                System.out.printf("Apakah surat %s diterima?
(y/n)", stack.peek().namaMahasiswa);
                terima = sc.nextLine();
                if (terima.equalsIgnoreCase("y")) {
                    trm = stack.pop();
                    System.out.printf("Surat dari %s
diterima.\n", trm.namaMahasiswa);
                } else if (terima.equalsIgnoreCase("n")) {
                    System.out.printf("Surat dari %s tidak
diterima.", stack.peek().namaMahasiswa);
                } else {
                    System.out.println("Keterangan yang anda
masukkan tidak vali!");
                }
            } else {
                System.out.println("Tidak ada surat untuk
diproses.");
            }
            break;

        case 3:
            Surat10 lihat = stack.peek();
            if (lihat != null) {
                System.out.println("Surat terakhir dikumpulkan
oleh " + lihat.namaMahasiswa);
                System.out.println("");
            }
            break;

        case 4:
            SearchingSurat10 pencari = new SearchingSurat10();
            System.out.print("Masukkan Nama Mahasiswa: ");
            String cariNama = sc.nextLine();
            int hasil = pencari.searchingSurat(stack,
cariNama, size);
            if (hasil != -1) {
                System.out.println("Detail Surat: ");
                System.out.println("ID Surat      Nama
Kelas      Keterangan      Durasi (Hari)");
                stack.printCari(hasil);
                System.out.println("");
            } else {
                System.out.println("Surat tidak
ditemukan.\n");
            }
            break;

        default:
            System.out.println("Pilihan tidak valid.\n");
            break;
    }

    } while (pilih >= 1 && pilih <= 4);
}
}

```

- Output

```
Masukkan jumlah mahasiswa: 3

Menu:
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat Izin
Pilih: 1
Id Surat: 1
Nama: Agus
Kelas: 1E
Jenis Izin: S
Durasi Izin: 2
Surat Agus berhasil dikumpulkan

Menu:
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat Izin
Pilih: 1
Id Surat: 2
Nama: Roni
Kelas: 1E
Jenis Izin: I
Durasi Izin: 3
Surat Roni berhasil dikumpulkan

Menu:
1. Terima Surat Izin
2. Proses Surat Izin
3. Lihat Surat Izin Terakhir
4. Cari Surat Izin
Pilih: 1
Id Surat: 3
Nama: Dirman
Kelas: 1E
Jenis Izin: K
Durasi Izin: 2
Surat Dirman berhasil dikumpulkan
```


Menu:

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

Pilih: 2

Daftar semua Surat:

ID Surat	Nama	Kelas	Keterangan	Durasi (Hari)
Keterangan izin tidak valid!				
3	Dirman	1E		2
2	Roni	1E	Izin	3
1	Agus	1E	Sakit	2

Apakah surat Dirman diterima? (y/n)y

Surat dari Dirman diterima.

Menu:

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

Pilih: 3

Surat terakhir dikumpulkan oleh Roni

Menu:

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

Pilih: 4

Masukkan Nama Mahasiswa: Agus

Detail Surat:

ID Surat	Nama	Kelas	Keterangan	Durasi (Hari)
1	Agus	1E	Sakit	2