SOAL PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK
PERIODE X

Laboratorium Rekayasa Perangkat Lunak, ITATS

PERTEMUAN 4 RPL-MF8T9-A

MEKANISME PRAKTIKUM

1) Buat Sebuah **Project Java Baru** pada IntelliJ IDEA dengan nama Project:
   "*PertemuanX_NPM AKHIR*"
   Ganti "X" menjadi Pertemuan yang sedang berlangsung.
2) Pada saat Praktikum, Jawabanlah Soal Pertanyaan yang memiliki Label **WAJIB** terlebih dahulu Pada Lembar **"Laporan Praktikum".**
3) Segala Bentuk **Soal yang memiliki Jawaban** berupa **Kode Program**, maka kode program tersebut harus disimpan pada **File java Project** yang telah dibuat.
4) Setiap **File Java** yang dibuat harus mencantumkan Pertanyaan pada bagian atas (baris pertama)
5) Simpan **File Laporan Praktikum** yang berupa **DOCX** menjadi **FILE PDF** kemudian ubah nama file PDF menjadi:
   "PertemuanX_NPM AKHIR.pdf"
6) Upload File **Laporan Praktikum [PDF]** pada form yang sudah disediakan.

TUGAS PRAKTIKUM

1. Jelaskan apa bedanya **Abstraction** dan **Interfaces**! [wajib]
2. Jelaskan apa yang dimaksud **Exception Handling**! [wajib]
3. Budi Arye membuat sedang belajar tentang abstraction dan exception handling
   Ia membuat program sebagai berikut :

```
abstract class Animal {
    string nama;

    Animal(String naMa) {
        this.nama = nama;
    }

    abstract void bersuara();

    void sleep() {
        System.out.println(nama + " lagi tidur.");
    }
}
```

```
class Wolf implements Animal {
    Wolf(String name) {
        super(String name);
    }

    @Override
    void bersuara() {
        System.out.println(nama + " says: Howl Howl");
    }
}
```

```
class Dog extends Wolf {

    Dog(String name) {

        super(name);

    }

    @Override

    void bersuara() {

        System.out.println(nama + " says: Woof Woof");}}

public class Main {
    public static void main(String[] args) {
        try {
            Wolf seringila = new Wolf("Budyeah");
            seringila.bersuara();
            seringila.sleep();
            Dog anjing = new Dog("Aryeah");
            anjing.bersuara();
            anjing.sleep();
            int[] numbers = {1, 2, 3};
            System.out.println(numbers[5]);
        } catch (ArrayBoundsException e) {
            System.out.println("Error: " + e.getMessage());
        } finally {
            System.out.println("end.");
        }
    }
}
```

Bantu Budi menyelesaikan kodingannya agar mendapatkan output kurang lebih sebagai berikut: [wajib]

```
Budyeah says: Woof Woof
Budyeah mending tidur.
Aryeah says: Howl Howl
Aryeah mending tidur.
Error: Index 5 out of bounds for length 3
End.
```

4. Dari soal praktikum sebelumnya kalian telah membuat class **Paket** yang di extend atau diturunkan ke kelas **PaketDomestik** dan **PaketInternasional**.Buatlah interface **Pembayaran** dengan metode **prosesPembayaran(double jumlah)**. Kemudian buatlah class **PembayaranTunai** dan **PembayaranKartu** yang mengimplementasikan interface **Pembayaran**,dengan syarat sebagai berikut:
    - Pada class **PembayaranKartu**, tambahkan atribut nomorkartu dan namapemilik.
    - Tambahkan metode bayar pada class **PaketDomestik** yang menerima objek Pembayaran sebagai parameter dan memanggil metode prosesPembayaran.
    - Buatlah objek **PembayaranTunai** dan **PembayaranKartu**, lalu gunakan metode bayar pada objek PaketDomestik untuk melakukan pembayaran.

5. Berdasarkan soal No.4 Buatlah interface **ValidasiPembayaran** dengan metode **validasi (double jumlah)**. Buatlah class **ValidasiSaldo** dan **ValidasiKartu** yang mengimplementasikan interface **ValidasiPembayaran**, dengan ketentuan sebagai berikut:

- Pada class **ValidasiSaldo**, tambahkan atribut saldo dan metode validasi untuk memeriksa apakah saldo mencukupi.
- Pada class **ValidasiKartu**, tambahkan atribut nomorkartu dan metode validasi untuk memeriksa validnya nomor kartu.
- Tambahkan metode **validasiPembayaran** pada class PaketDomestik yang menerima objek **ValidasiPembayaran** sebagai parameter dan memanggil metode validasi.
- Buat objek ValidasiSaldo dan ValidasiKartu, lalu gunakan metode validasiPembayaran pada objek PaketDomestik untuk melakukan validasi sebelum pembayaran.