

UTS PBO

1.

```
import java.util.Scanner;
2 usages
class waktu_awal_percakapan{
    3 usages
    byte jam_awal;
    3 usages
    byte menit_awal;
    3 usages
    byte detik_awal;

    1 usage
    public int total_detik(){
        return (jam_awal * 3600 + menit_awal * 60 + detik_awal);
    }
}
```

Untuk soal no 1, saya menggunakan class waktu_awal_percakapan sebagai inheritance. Saya menggunakan tipe data byte untuk variabel jam_awal, menit_awal, dan detik_awal. Alasan pakai byte agar filenya tidak memakan ukuran file yang besar, terlebih untuk input jam, menit dan detik tidak sampai angka 127.

```
2 usages
class waktu_akhir_percakapan{
    3 usages
    byte jam_akhir;
    3 usages
    byte menit_akhir;
    3 usages
    byte detik_akhir;

    1 usage
    public int total_detik(){ return (jam_akhir * 3600 + menit_akhir * 60 + detik_akhir); }
}
```

Sama seperti di atas, class waktu_akhir_percakapan sebagai inheritance class. Sama seperti di atas, saya menggunakan tipe data byte untuk variabel jam_akhir, menit_akhir, dan detik_akhir.

```
public class UTSNo1{  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
  
        waktu_awal_percakapan awal = new waktu_awal_percakapan();  
        System.out.print("Masukkan jam pada awal percakapan : ");  
        awal.jam_awal = sc.nextByte();  
        System.out.print("Masukkan menit pada awal percakapan : ");  
        awal.minit_awal = sc.nextByte();  
        System.out.print("Masukkan detik pada awal percakapan : ");  
        awal.detik_awal = sc.nextByte();  
        System.out.println("Waktu awal percakapan : (" + awal.jam_awal + ":" + awal.minit_awal +  
            ":" + awal.detik_awal + ")");  
    }  
}
```

Masuk ke main, saya menggunakan objek "awal" sebagai pemanggil inheritance class waktu_awal_percakapan. Saya membuat inputan jam, menit, lalu detik, setelah itu membuat tampilan untuk waktu awalnya (jam_awal:menit_awal:detik_awal).

```
waktu_akhir_percakapan akhir = new waktu_akhir_percakapan();  
  
System.out.print("Masukkan jam pada akhir percakapan : ");  
akhir.jam_akhir = sc.nextByte();  
System.out.print("Masukkan jam pada akhir percakapan : ");  
akhir.minit_akhir = sc.nextByte();  
System.out.print("Masukkan jam pada akhir percakapan : ");  
akhir.detik_akhir = sc.nextByte();  
System.out.println("Waktu akhir percakapan : (" + akhir.jam_akhir + ":" + akhir.minit_akhir +  
    ":" + akhir.detik_akhir + ")");
```

Saya menggunakan objek "akhir" sebagai pemanggil inheritance class waktu_akhir_percakapan. Saya membuat inputan jam, menit, lalu detik, setelah itu membuat tampilan untuk waktu akhir percakapannya (jam_akhir:menit_akhir:detik_akhir).

```
long harga_pulsa = (akhir.total_detik() - awal.total_detik() ) * 30;  
System.out.println("Maka harga pulsa selama anda menelepon adalah : " + harga_pulsa);
```

Masuk ke fungsi harga untuk pulasanya, saya menggunakan tipe data long. Untuk fungsinya saya menggunakan system per detik, di mana harga pulasanya adalah Rp 30 per detik. Jadi saya kurangi saja total waktu dari percakapan akhir yang dijadikan detik dengan waktu dari percakapan awal jika dijadikan detik. Lalu membuat tampilan untuk harga pulasanya.

```
Masukkan jam pada awal percakapan : 8
Masukkan menit pada awal percakapan : 10
Masukkan detik pada awal percakapan : 12
Waktu awal percakapan : (8:10:12)
Masukkan jam pada akhir percakapan : 10
Masukkan jam pada akhir percakapan : 12
Masukkan jam pada akhir percakapan : 14
Waktu akhir percakapan : (10:12:14)
Maka harga pulsa selama anda menelepon adalah : 219660
```

Jika di-run programnya, maka akan ditanyakan untuk jam, menit serta detik. Untuk percakapan awalnya, saya menggunakan waktu (8:10:12) serta untuk percakapan akhirnya, saya menggunakan waktu (10:12:14), maka harga dari percakapan tersebut adalah Rp.219.660

2.

```
import java.util.Scanner;
2 usages
public class UTSNo2 {
    6 usages
    private double tinggi_badan;
    5 usages
    private short berat_badan;

    public double getTinggi_badan() { return tinggi_badan; }
    public void setTinggi_badan(double tinggi_badan) { this.tinggi_badan = tinggi_badan; }
    public double getBerat_badan() {
        return berat_badan;
    }
    public void setBerat_badan(short berat_badan) { this.berat_badan = berat_badan; }
```

Untuk no 2, saya menggunakan variabel tinggi_badan dengan tipe data double (karena decimal), lalu variabel berat_badan dengan tipe data short.

```
public UTSNo2(){
    Scanner sc = new Scanner(System.in);
    System.out.print("Masukkan berat badan anda ( dalam kg ): ");
    berat_badan = sc.nextShort();
    System.out.print("Masukkan tinggi badan anda ( dalam m ) : ");
    tinggi_badan = sc.nextDouble();
```

Untuk constructornya, saya menginput berat badan serta tinggi badan

```
public double BMI() { return berat_badan / (tinggi_badan * tinggi_badan); }
```

Untuk mencari BMI, saya menggunakan method BMI dengan tipe data double (karena bisa decimal). Rumus mencari BMI adalah $\text{berat_badan} / (\text{tinggi_badan} * \text{tinggi_badan})$

```
public static void main(String[] args) {  
    UTSNo2 BB = new UTSNo2();  
    System.out.println("Berat badan anda ( dalam kg ) : " + BB.berat_badan);  
    System.out.println("Tinggi badan anda ( dalam m ) : " + BB.tinggi_badan);  
    if (BB.BMI() < 18.5){  
        System.out.println("Anda underweight");  
    }  
    else if (BB.BMI() >= 18.5 && BB.BMI() <= 22.9 ){  
        System.out.println("Anda normal");  
    }  
    else if (BB.BMI() > 22.9){  
        System.out.println("Anda overweight");  
    }  
    System.out.println("Indeks berat badan anda adalah : " + BB.BMI());  
}
```

Masuk ke main, saya membuat objek “ BB “ untuk memanggil constructor class. Setelah itu, saya membuat tampilan kembali berat badan dan tinggi badan. Saya membuat sebuah kondisi juga BMI nya itu di antara 18,5 sampai 22,9, maka indeks berat badan normal, kondisi jika BB lebih dari 22,9, maka overweight, kondisi jika BB kurang dari 18.5, maka statusnya underweight. Saya juga membuat tampilan berapa indeks berat badan.

```
Masukkan berat badan anda ( dalam kg ): 78  
Masukkan tinggi badan anda ( dalam m ) : 1,78  
Berat badan anda ( dalam kg ) : 78  
Tinggi badan anda ( dalam m ): 1.78  
Anda overweight  
Indeks berat badan anda adalah : 24.61810377477591
```

Outputnya saya menggunakan berat badan serta tinggi badan saya, berat badan saya adalah 78 kg dan tinggi badan saya adalah 1,78 m. Status saya adalah overweight dan indeks berat badan saya adalah 24.61.

3.

```
import java.util.Scanner;
2 usages
public class UTSNo3 {
    4 usages
    private int jumlah;

    1 usage
    public UTSNo3() {
        Scanner sc = new Scanner(System.in);
        int[] bil_bulat = new int[10]; // variabel bil_bulat sebagai variabel array dalam loop
        jumlah = 0; // variabel untuk menjumlahkan 10 bilangan bulat, saya deklarasikan 0 agar variabel
        // jumlah ini saat ditambahkan dengan array bil_bulat hasilnya akan sama dengan jumlah array
        // bil bulat

        // masuk ke looping for, saya deklarasikan i yaitu 0 karena untuk index ke-0 pada array bil_bulat
        // lalu saya bataskan i kurang dari panjang array bil_bulat sebagai pembatas agar i tidak
        // mengulang berlebihan.
        for (int i = 0; i < bil_bulat.length; i++) {
            System.out.print("Masukkan bilangan bulat : ");
            bil_bulat[i] = sc.nextInt(); // input bilangan bulat
            jumlah = jumlah + bil_bulat[i];
            // mencari nilai jumlah dari 10 nilai yang sudah diinput ke dalam array
        }
    }
}
```

Saya menggunakan variabel jumlah dengan tipe data int, variabel jumlah ini berfungsi untuk menjumlahkan hasil dari penjumlahan semua hasil dari nilai masing-masing index array. Masuk ke constructor, saya menggunakan variabel bil_bulat dengan tipe data int. Variabel bil_bulat sebagai variabel array dalam loop. Saya deklarasikan variabel jumlah adalah 0 agar hasilnya tidak berubah jika ditambahkan dengan hasil jumlah total masing-masing nilai index. Saya menggunakan perulangan for, dengan deklarasi variabel i = 0, lalu i kurang dari panjang array variabel bil_bulat, lalu i increment.

```
public static void main(String[] args) {
    UTSNo3 x = new UTSNo3();
    System.out.printf("\nJumlah dari array bil_bulat yang sudah diinput adalah : " + x.jumlah);
}
}
```

Pada main, saya menggunakan objek "x" untuk memanggil constructor class. Saya membuat tampilan jumlah dari array bil_bulat.

```
Masukkan bilangan bulat : 4
Masukkan bilangan bulat : 5
Masukkan bilangan bulat : 6
Masukkan bilangan bulat : 7
Masukkan bilangan bulat : 8
Masukkan bilangan bulat : 9
Masukkan bilangan bulat : 10
Masukkan bilangan bulat : 13
Masukkan bilangan bulat : 12
Masukkan bilangan bulat : 11

Jumlah dari array bil_bulat yang sudah diinput adalah : 85
```

Untuk outputnya, saya menggunakan array [4, 5, 6, 7, 8, 9, 10, 13, 12, 11], lalu saya input satu-satu. Lalu untuk hasilnya adalah 85.

4.

```
import java.util.Scanner;

2 usages
class luas_lingkaran {
    3 usages
    double jari_jari;
    1 usage
    double pi = 3.14;

    1 usage
    public double luas_lingkaran() { return pi * jari_jari * jari_jari; }
```

Saya menggunakan inheritance class dengan nama class luas_lingkaran. Pada class ini terdapat variabel jari-jari, lalu variabel pi, saya deklarasi pi adalah 3.14. Untuk fungsi luas_lingkaran, saya menggunakan tipe data double. Rumusnya adalah $\pi * \text{jari-jari} * \text{jari-jari}$.

```
2 usages
class luas_segitiga {
    2 usages
    double alas;
    2 usages
    double tinggi;

    1 usage
    double luas_segitiga() { return alas * tinggi / 2; }
```

Sama seperti di atas, saya menggunakan inheritance class dengan nama class luas_segitiga. Pada class ini terdapat variabel alas, lalu variabel tinggi. Untuk fungsi luas_segitiga, saya menggunakan tipe data double. Rumusnya adalah $\text{alas} \times \text{tinggi} / 2$.

```
}
public class UTSNo4{
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        luas_lingkaran L_Lingkaran = new luas_lingkaran();

        System.out.print("Masukkan jari-jari lingkaran : ");
        L_Lingkaran.jari_jari = sc.nextDouble();

        System.out.println("Luas lingkaran adalah : " + L_Lingkaran.luas_lingkaran());

        luas_segitiga L_Segitiga = new luas_segitiga();

        System.out.print("Masukkan alas segitiga : ");
        L_Segitiga.alas = sc.nextDouble();

        System.out.print("Masukkan tinggi segitiga : ");
        L_Segitiga.tinggi = sc.nextDouble();

        System.out.println("Luas segitiga adalah : " + L_Segitiga.luas_segitiga());
    }
}
```

Lalu masuk ke main, saya menggunakan objek "L_Lingkaran" untuk memanggil inheritance class luas_lingkaran. Saya membuat input jari-jari, lalu membuat tampilan luas lingkaran dengan fungsi yang ada di class.

Saya menggunakan objek "L_Segitiga" untuk memanggil inheritance class luas_segitiga. Saya membuat input alas, tinggi, lalu membuat tampilan luas segitiga dengan fungsi yang ada di class.

```
Masukkan jari-jari lingkaran : 10
Luas lingkaran adalah : 314.0
Masukkan alas segitiga : 10
Masukkan tinggi segitiga : 20
Luas segitiga adalah : 100.0

Process finished with exit code 0
|
```

Untuk outputnya, saya menginput jari-jari 10, maka luasnya adalah 314. Untuk segitiga, saya menginput alas adalah 10, dan tinggi adalah 20, maka luasnya adalah 100.