



PPT Week 7 Praktikum PBO

Latihan



Latihan 1 - Manajemen Stok Toko Buku

Anda diminta untuk membuat program Java untuk mengelola stok buku di sebuah toko. Setiap buku memiliki judul, penulis, harga, dan jumlah stok. Toko tersebut ingin bisa menambahkan buku baru ke sistem, memperbarui stok saat penjualan, dan menampilkan semua buku yang tersedia dengan stok di atas batas tertentu. Ketentuannya sebagai berikut:

Class Buku:

- Buat kelas Buku dengan private attributes: judul, penulis, harga, dan stok.
- Tambahkan constructor untuk inisialisasi atribut.
- Tambahkan getter dan setter untuk masing-masing atribut.



Latihan 1 - Manajemen Stok Toko Buku

Class TokoBuku:

- Kelas ini harus memiliki sebuah ArrayList yang menyimpan objek-objek Buku.
- Metode untuk tambahBuku yang menambahkan buku baru ke ArrayList dan menaikkan jumlah total buku.
- Metode jualBuku yang menerima judul buku dan jumlah yang dijual, dan mengurangi jumlah stok dari buku tersebut.
- Metode tampilkanStokBatas yang menerima sebuah angka sebagai parameter dan menampilkan semua buku dengan stok lebih dari angka tersebut.



Latihan 1 - Manajemen Stok Toko Buku

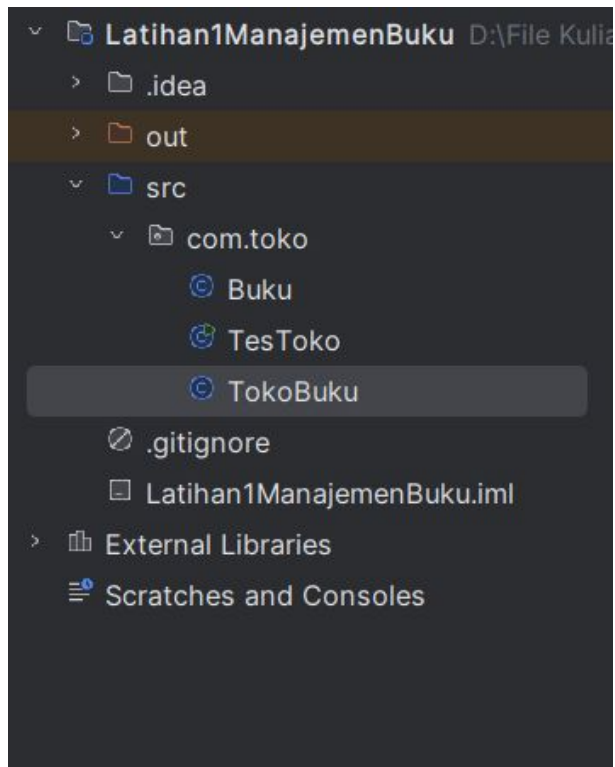
Class TesToko(Main):

- Buat kelas utama yang menggunakan kelas TokoBuku untuk menambahkan beberapa buku, menjual beberapa buku, dan menampilkan buku dengan stok di atas jumlah tertentu.

Persyaratan:

- Taruh semua file class pada folder com.buku
- Gunakan prinsip enkapsulasi, package, ArrayList, StaticKeyword

Jawaban Latihan 1 - Hirarki Folder



Jawaban Latihan 1 - Class Buku (1)

```
package com.toko;

public class Buku { 4 usages
    private String judul; 3 usages
    private String penulis; 3 usages
    private double harga; 3 usages
    private int stok; 3 usages


    public Buku(String judul, String penulis, double harga, int stok) { no usages
        this.judul = judul;
        this.penulis = penulis;
        this.harga = harga;
        this.stok = stok;
    }

    // Getter dan setter untuk atribut
    public String getJudul() { 2 usages
        return judul;
    }

    public void setJudul(String judul) { no usages
        this.judul = judul;
    }

    public String getPenulis() { no usages
        return penulis;
    }
}
```

Jawaban Latihan 1 - Class Buku (2)



```
public void setPenulis(String penulis) { no usages
    this.penulis = penulis;
}

public double getHarga() { no usages
    return harga;
}

public void setHarga(double harga) { no usages
    this.harga = harga;
}

public int getStok() { 4 usages
    return stok;
}

public void setStok(int stok) { 1 usage
    this.stok = stok;
}
}
```


Jawaban Latihan 1 - Class TokoBuku (1)

```
package com.toko;

import java.util.ArrayList;

public class TokoBuku { no usages
    private ArrayList<Buku> buku; 4 usages
    private static int totalBuku = 0; 2 usages

    public TokoBuku() { no usages
        this.buku = new ArrayList<>();
    }

    public void tambahBuku(Buku bukuBaru) { no usages
        buku.add(bukuBaru);
        totalBuku++;
    }

    public void jualBuku(String judul, int jumlah) { no usages
        for (Buku b : buku) {
            if (b.getJudul().equals(judul) && b.getStok() >= jumlah) {
                b.setStok(b.getStok() - jumlah);
                break;
            }
        }
    }
}
```

Jawaban Latihan 1 - Class TokoBuku (2)

```
public void tampilkanStokBatas(int batas) { no usages
    System.out.println("Daftar Buku dengan Stok di Atas " + batas + ":");
    for (Buku b : buku) {
        if (b.getStok() > batas) {
            System.out.println(b.getJudul() + " - Stok: " + b.getStok());
        }
    }
}

public static int getTotalBuku() { no usages
    return totalBuku;
}
}
```

Jawaban Latihan 1 - Class TesBuku (1)

```
package com.toko;

public class TesToko {
    public static void main(String[] args) {
        TokoBuku toko = new TokoBuku();
        toko.tambahBuku(new Buku( judul: "Java Pemula", penulis: "John Doe", harga: 150000, stok: 50));
        toko.tambahBuku(new Buku( judul: "Advanced Java", penulis: "Jane Roe", harga: 200000, stok: 20));

        System.out.println("Total buku saat ini: " + TokoBuku.getTotalBuku());
        toko.jualBuku( judul: "Java Pemula", jumlah: 10);
        toko.tampilkanStokBatas(15);
    }
}
```



Latihan 2 - Manajemen Pesawat

Anda diminta untuk membuat program Java untuk mengelola keberangkatan dan kedatangan penerbangan di sebuah bandara. Sistem ini harus mampu menambahkan penerbangan baru, mencari penerbangan berdasarkan kode penerbangan, dan memfilter penerbangan berdasarkan status (misal: "scheduled", "delayed", "departed", "arrived"). Sistem juga harus menyimpan total jumlah penerbangan yang telah ditambahkan ke dalam sistem.



Latihan 2 - Manajemen Pesawat

Class Pesawat:

- Kelas ini harus memiliki atribut privat: kodePenerbangan, kotaAsal, kotaTujuan, waktuBerangkat, dan status.
- Konstruktor untuk menginisialisasi atribut tersebut.
- Getter dan setter untuk setiap atribut.

Class ManajemenPesawat

- Kelas ini harus memiliki ArrayList yang menyimpan objek Penerbangan.
- Metode tambahPenerbangan untuk menambahkan penerbangan baru ke ArrayList dan menaikkan jumlah total penerbangan.
- Metode cariPenerbangan yang menerima kode penerbangan dan mengembalikan detail penerbangan tersebut.
- Metode filterPenerbangan yang menerima status dan mengembalikan daftar penerbangan dengan status tersebut.



Latihan 2 - Manajemen Pesawat

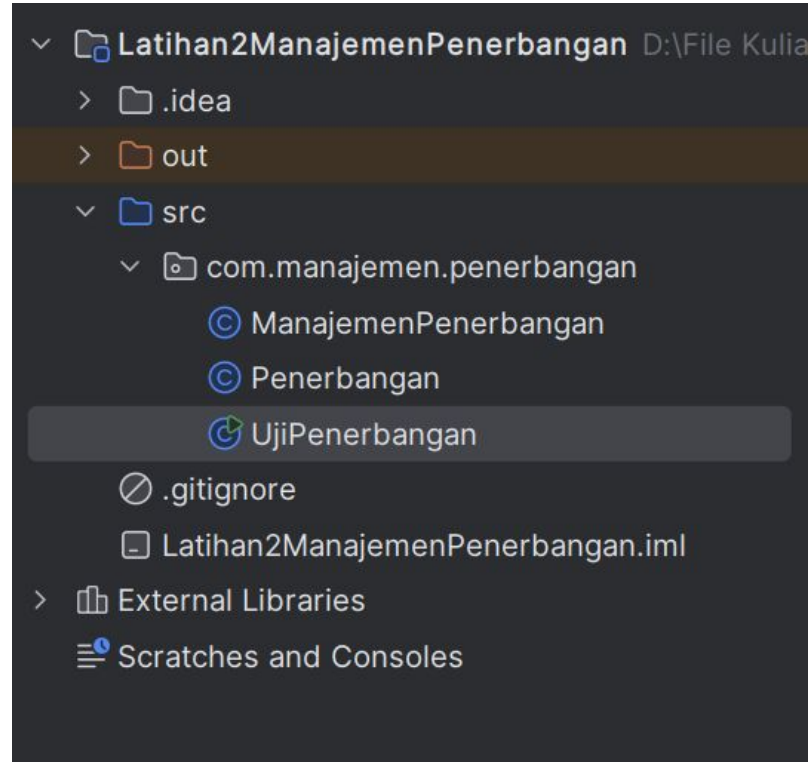
Class UjiPesawat:

- Buat kelas ini untuk memanipulasi dan menampilkan data menggunakan kelas ManajemenPenerbangan, termasuk menambahkan beberapa penerbangan, mencari penerbangan tertentu, dan menampilkan penerbangan berdasarkan status.

Persyaratan:

- Taruh semua file class pada folder package com.manajemen.penerbangan
- Gunakan prinsip enkapsulasi, package, ArrayList, StaticKeyword

Jawaban Latihan 2 - Hirarki Folder



Jawaban Latihan 2 - Class Penerbangan (1)

```
package com.manajemen.penerbangan;

public class Penerbangan { 12 usages
    private String kodePenerbangan; 3 usages
    private String kotaAsal; 3 usages
    private String kotaTujuan; 3 usages
    private String waktuBerangkat; 3 usages
    private String status; 3 usages


    public Penerbangan(String kodePenerbangan, String kotaAsal, String kotaTujuan, String waktuBerangkat, String status) { 2 usages
        this.kodePenerbangan = kodePenerbangan;
        this.kotaAsal = kotaAsal;
        this.kotaTujuan = kotaTujuan;
        this.waktuBerangkat = waktuBerangkat;
        this.status = status;
    }

    // Getter dan Setter

    public String getKodePenerbangan() { 3 usages
        return kodePenerbangan;
    }

    public void setKodePenerbangan(String kodePenerbangan) { no usages
        this.kodePenerbangan = kodePenerbangan;
    }
}
```


Jawaban Latihan 2 - Class Penerbangan (2)



```
public String getKotaAsal() { 1 usage
    return kotaAsal;
}

public void setKotaAsal(String kotaAsal) { no usages
    this.kotaAsal = kotaAsal;
}

public String getKotaTujuan() { 1 usage
    return kotaTujuan;
}

public void setKotaTujuan(String kotaTujuan) { no usages
    this.kotaTujuan = kotaTujuan;
}

public String getWaktuBerangkat() { no usages
    return waktuBerangkat;
}

public void setWaktuBerangkat(String waktuBerangkat) { no usages
    this.waktuBerangkat = waktuBerangkat;
}

public String getStatus() { 2 usages
    return status;
}

public void setStatus(String status) { no usages
    this.status = status;
}
}
```

Jawaban Latihan 2 - Class ManajemenPenerbangan (1)

```
package com.manajemen.penerbangan;

import java.util.ArrayList;

public class ManajemenPenerbangan { no usages
    private ArrayList<Penerbangan> penerbangan; 4 usages
    private static int totalPenerbangan; 2 usages

    public ManajemenPenerbangan() { no usages
        this.penerbangan = new ArrayList<>();
    }

    public void tambahPenerbangan(Penerbangan baru) { no usages
        penerbangan.add(barau);
        totalPenerbangan++;
    }

    public Penerbangan cariPenerbangan(String kode) { no usages
        for (Penerbangan p : penerbangan) {
            if (p.getKodePenerbangan().equals(kode)) {
                return p;
            }
        }
        return null;
    }
}
```

Jawaban Latihan 2 - Class ManajemenPenerbangan (2)

```
public ArrayList<Penerbangan> filterPenerbangan(String status) { no usages
    ArrayList<Penerbangan> hasil = new ArrayList<>();
    for (Penerbangan p : penerbangan) {
        if (p.getStatus().equals(status)) {
            hasil.add(p);
        }
    }
    return hasil;
}

public static int getTotalPenerbangan() { no usages
    return totalPenerbangan;
}
}
```

Jawaban Latihan 2 - Class ManajemenPenerbangan (2)

```
public ArrayList<Penerbangan> filterPenerbangan(String status) { no usages
    ArrayList<Penerbangan> hasil = new ArrayList<>();
    for (Penerbangan p : penerbangan) {
        if (p.getStatus().equals(status)) {
            hasil.add(p);
        }
    }
    return hasil;
}

public static int getTotalPenerbangan() { no usages
    return totalPenerbangan;
}
}
```

Jawaban Latihan 2 - Class UjiPenerbangan (1)

```
package com.manajemen.penerbangan;

import java.util.ArrayList;

public class UjiPenerbangan {
    public static void main(String[] args) {
        ManajemenPenerbangan manajer = new ManajemenPenerbangan();
        manajer.tambahPenerbangan(new Penerbangan( kodePenerbangan: "QZ302", kotaAsal: "Jakarta", kotaTujuan: "Bali", waktuBerangkat: "06:00", status: "scheduled"));
        manajer.tambahPenerbangan(new Penerbangan( kodePenerbangan: "GA223", kotaAsal: "Jakarta", kotaTujuan: "Singapore", waktuBerangkat: "09:30", status: "delayed"));

        System.out.println("Total penerbangan: " + ManajemenPenerbangan.getTotalPenerbangan());

        Penerbangan p = manajer.cariPenerbangan( kode: "GA223");
        if (p != null) {
            System.out.println("Penerbangan ditemukan: " + p.getKodePenerbangan() + " - " + p.getStatus());
        } else {
            System.out.println("Penerbangan tidak ditemukan.");
        }

        ArrayList<Penerbangan> delayed = manajer.filterPenerbangan( status: "delayed");
        System.out.println("Penerbangan yang Delayed:");
        for (Penerbangan d : delayed) {
            System.out.println(d.getKodePenerbangan() + " - " + d.getKotaAsal() + " ke " + d.getKotaTujuan());
        }
    }
}
```

Post Test



Sistem Manajemen Nilai Mahasiswa

Buatlah sebuah program untuk menghitung nilai rata-rata mahasiswa dari setiap kelas dan seluruh nilai rata-rata mahasiswa dari semua kelas dengan ketentuan sebagai berikut:

Class Mahasiswa:

- Terdapat 3 field yaitu nim, nama, dan nilai;
- terdapat konsep enkapsulasi pada class ini;
- User dapat membuat objek dari kelas ini dengan memasukkan 2 jenis parameter, yang pertama nim, nama, dan nilai sekaligus. Atau bisa hanya dengan nim dan nama dengan isi default field nilai dengan 0.



Sistem Manajemen Nilai Mahasiswa

Class Kelas:

- Terdapat 4 field pada class Kelas, yaitu namaKelas, daftarMahasiswa yang bertipe arrayList dari class Mahasiswa, kemudian jumlahMahasiswa yang bertipe static, dan mhs yang bertipe class Mahasiswa;
- Pada class Kelas terdapat constructor yang memiliki jumlah parameter 1, yaitu namaKelas. Pada constructor ini terdapat inisiasi object dari arrayList daftarMahasiswa
- Terdapat fungsi tambah mahasiswa ke dalam kelas
- terdapat fungsi hapus mahasiswa dalam kelas
- terdapat fungsi hitung rata-rata untuk menghitung rata-rata nilai mahasiswa dalam kelas tersebut
- Menerapkan konsep enkapsulasi pada field namaKelas, daftarMahasiswa, dan jumlahMahasiswa



Sistem Manajemen Nilai Mahasiswa

Class Main:

- Buatlah beberapa object mahasiswa (minimal 10)
- Buatlah beberapa object kelas (minimal 3)
- Bagi object mahasiswa yang telah kamu buat ke dalam masing-masing kelas (bebas pembagiannya)
- Inisiasi arrayList dengan nama daftarKelas untuk mengumpulkan kelas-kelas tersebut ke dalam satu wadah
- Masukkan semua object kelas yang telah dibuat ke dalam variabel daftarKelas
- Terdapat fungsi hitungRataRataSemuaKelas untuk menghitung rata-rata nilai mahasiswa dari semua kelas
- terdapat fungsi displatRataRata untuk menampilkan nama kelas dengan nilai rata-ratanya
- terdapat fungsi displayRataRataSemuaKelas untuk menampilkan jumlah mahasiswa dari seluruh kelas dan rata-rata nilai mahasiswa dari seluruh kelas.

Terima Kasih