

MODUL 6

LATIHAN SOAL UTS

Latihan

(CPMK 1,2, dan 3) [Soal UTS PBO: Genap 2023/2024-CS]

INSTRUKSI: Buat jawaban berupa kode Java di kertas jawaban yang disediakan!

1. (15 poin) Buatlah sebuah kelas **Dosen** dalam package **dosen**.
 - a. Buatlah attribute private **nama** dan **NIDN**. **NIDN** adalah Nomor Induk Dosen Nasional.
 - b. Buatlah konstruktor untuk inisialisasi **nama** dan **NIDN**.
 - c. Buatlah method getter untuk seluruh attribute.
2. (15 poin) Buatlah sebuah kelas **Mahasiswa** dalam package **mahasiswa**.
 - a. Buatlah attribute private **nama** dan **NIM**. **NIM** adalah Nomor Induk Mahasiswa.
 - b. Buatlah konstruktor untuk inisialisasi **nama** dan **NIM**.
 - c. Buatlah method getter untuk seluruh attribute.
3. (30 poin) Buatlah sebuah kelas **MataKuliah** dalam package **mataKuliah**.
 - a. Buatlah attribute private **nama**, **KodeMK** adalah kode mata kuliah, **ArrayList<Dosen> pengampu** dan **ArrayList<Mahasiswa> daftarMahasiswa**.
 - b. Buatlah konstruktor untuk inisialisasi **nama** dan **kodeMK**.
 - c. Buatlah method setter untuk attribute **daftarMahasiswa** dan **pengampu**.
 - d. Buatlah method getter untuk seluruh attribute.
4. (40 poin) Buatlah kelas **Utama** dalam package **utama**.
 - a. Dalam method **main**, buat objek dan set pengampu dan daftar mahasiswa untuk tiap mata kuliah seperti tabel berikut:

| Mata Kuliah | Kode Mata Kuliah | Pengampu | Mahasiswa |
|-------------|------------------|----------|--------------|
| PBO | 123 | Cahya | Tiara, Diana |
| Kompar | 124 | Budi | Tiara, Diana |
| Metnum | 125 | Budi | Tiara |

- b. Buatlah method static **tampilkanData (MataKuliah matakuliah)** untuk menampilkan data pengampu dan daftar mahasiswa untuk tiap mata kuliah beserta kode mata kuliahnya dan panggil method tersebut di main.

Posttest

Instruksi:

Pilih salah satu studi kasus di bawah ini untuk dikerjakan.

Bonus: Anda diperbolehkan mengerjakan kedua studi kasus untuk memperoleh nilai tambahan. Nilai bonus tersebut akan dimasukkan ke dalam komponen penilaian lain yang masih kurang.

Studi Kasus 1: Manajemen Data Mahasiswa

Sebuah universitas ingin membuat sistem sederhana untuk mengelola data mahasiswa. Setiap mahasiswa memiliki data sebagai berikut:

- NIM (String)
- Nama (String)
- Program Studi (String)
- IPK (double)

Data mahasiswa akan dikelola dalam bentuk daftar menggunakan ArrayList. Anda diminta untuk mengimplementasikan program menggunakan **konsep OOP Java**. Gunakan struktur class, package, dan prinsip OOP yang baik.

Instruksi Pengerjaan:

1. Buat package dengan nama **universitas**.
2. Di dalam package tersebut, buat **class Mahasiswa** yang memiliki:
 - Atribut privat untuk menyimpan NIM, nama, prodi, dan IPK
 - Constructor untuk inisialisasi semua atribut
 - Getter dan setter untuk semua atribut
 - Method `tampilkanData()` untuk menampilkan informasi mahasiswa dalam format:

```
---Data Mahasiswa ---  
NIM   : ...  
Nama  : ...  
Prodi  : ...  
IPK   : ...  
-----
```

3. Buat **class ManajemenMahasiswa** di dalam package **universitas** yang memiliki:

- Atribut `ArrayList<Mahasiswa>` untuk menyimpan daftar mahasiswa
- Method `tambahMahasiswa (Mahasiswa m)` untuk menambahkan data ke dalam list
- Method `tampilkanSemua ()` untuk menampilkan seluruh data mahasiswa
- Method `tampilkanIPKTertinggi ()` untuk menampilkan mahasiswa dengan IPK tertinggi

4. Buat **class Main** di luar package universitas, yang:

- Memiliki sistem menu berbasis console sebagai berikut:

```

==== MENU ====
1. Tambah Mahasiswa
2. Tampilkan Semua Mahasiswa
3. Tampilkan Mahasiswa dengan IPK Tertinggi
4. Keluar
Pilih:

```

- Gunakan Scanner untuk input dari user.
- Gunakan konsep objek, enkapsulasi, dan struktur method dengan baik.
- Contoh tampilan output:

```

==== MENU ====
1. Tambah Mahasiswa
2. Tampilkan Semua Mahasiswa
3. Tampilkan Mahasiswa dengan IPK Tertinggi
4. Keluar
Pilih: 1
Masukkan NIM : 12345
Masukkan Nama : Dimas
Masukkan Prodi : Ilmu Komputer
Masukkan IPK : 3.85
Data berhasil ditambahkan!

Pilih: 2
DAFTAR MAHASISWA:
--Data Mahasiswa--
NIM : 12345
Nama : Dimas
Prodi : Informatika
IPK : 3.85
-----

```

Note: Jika studi kasus dirasa kurang jelas, Anda dapat mengembangkan program sesuai dengan imajinasi dan kreativitas masing-masing.

Studi kasus 2: Sistem Manajemen Kursus Online

Membuat sebuah platform kursus online aplikasi Java untuk mengelola:

- Instruktur
- Kursus
- Materi
- Peserta
- Sertifikat

Struktur Package:

```
kursusonline/  
├── model/  
│   ├── Instruktur.java  
│   ├── Kursus.java  
│   ├── Peserta.java  
│   ├── Materi.java  
│   └── Sertifikat.java  
├── service/  
│   └── KursusService.java  
└── Main.java
```

Class dan Validasi:

- **Instruktur:** id, nama, spesialisasi
- **Peserta:** id, nama, email, noHp, sertifikat
Email harus berakhiran @gmail.com, jika tidak maka peserta tidak berhasil ditambahkan Hint :(endsWith("@gmail.com"))
- **Kursus:** kode, nama, instruktur, daftarPeserta, materi
Method: tambahPeserta, tambahMateri, tampilkanDetailKursus
- **Materi:** judul, deskripsi, durasi (menit)

Kursus

- Atribut: kode, nama, instruktur, daftarPeserta, daftarMateri

Method:

- tambahMateri(Materi m)
- tambahPeserta(Peserta p) dengan validasi:
 - Email peserta harus valid
 - Tidak boleh ada peserta dengan ID yang sama
 - tampilkanDetailKursus() untuk menampilkan seluruh data kursus.

Sertifikat

- Atribut: idSertifikat, namaKursus, tanggalTerbit

KursusService

- Atribut: ArrayList<Kursus>

Method:

- tambahKursus(Kursus k)
- tampilkanSemuaKursus()
- cariKursusBerdasarkanKode(String kode)
- tambahPesertaKeKursus(String kodeKursus, Peserta peserta)

Program Utama

Main.java:

- Buat 2 instruktur
- Buat 2 kursus dan hubungkan dengan masing-masing instruktur
- Buat 3 peserta (satu dengan email tidak valid)
- Buat 3 materi dan tambahkan ke kursus
- Tambahkan peserta ke kursus (satu akan gagal karena email tidak valid)

Tampilkan seluruh detail kursus menggunakan tampilkanSemuaKursus()

Note: Jika studi kasus dirasa kurang jelas, Anda dapat mengembangkan program sesuai dengan imajinasi dan kreativitas masing-masing.