

Kuis 1

Mata Kuliah Pemograman Berbasis Objek (PBO)

“Pertemuan 5 – Studi kasus Startup Edutech (Andi & Budi) ”

Dosen Pengajar:

M. Hasyim Ratsanjani S.Kom.,M.kom.



Disusun Oleh

Kelompok 1 TI – 2F :

- | | |
|--------------------------------|------------------|
| 1. Nevita Triya Yuliana | NIM 244107020208 |
| 2. Nur Alfiyanti | NIM 244107020055 |
| 3. Rifo Anggi Barbara Danuarta | NIM 244107020063 |

PROGRAM STUDI D-IV TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

T. A 2025 / 2026

1. STUDI KASUS:

Andi dan Budi, dua sahabat karib sejak kuliah, memutuskan untuk memulai bisnis bersama setelah lulus. Mereka mendirikan sebuah *startup* di bidang *edutech*. Andi, dengan keahliannya dalam pengembangan *software*, bertanggung jawab atas pengembangan platform pembelajaran online. Budi, dengan kemampuan komunikasinya yang baik, mengelola aspek pemasaran dan penjualan.

Startup mereka berkembang pesat, menarik minat banyak pengguna. Namun, mereka menghadapi tantangan baru: bagaimana menjaga kualitas konten pembelajaran dan memastikan bahwa platform mereka tetap relevan dengan perkembangan teknologi. Andi dan Budi memutuskan untuk melibatkan para ahli pendidikan sebagai konsultan untuk membantu mereka mengembangkan kurikulum yang berkualitas.

Selain itu, mereka juga ingin memperluas jangkauan pasar mereka ke luar negeri. Budi bertanggung jawab untuk menjalin kerjasama dengan mitra internasional, sementara Andi fokus pada adaptasi platform mereka untuk memenuhi kebutuhan pasar yang berbeda.

Desainlah sebuah class diagram yang merepresentasikan skenario di atas. Perhatikan entitas-entitas seperti:

- **Kursus:** Mata pelajaran yang ditawarkan oleh *startup*.
- **Konten:** Materi pembelajaran, seperti video, artikel, atau kuis.
- **Instruktur:** Pengajar yang menciptakan dan memberikan konten.
- **Peserta:** Pengguna yang mendaftar dan mengikuti kursus.
- **Pembayaran:** Transaksi pembayaran untuk kursus.

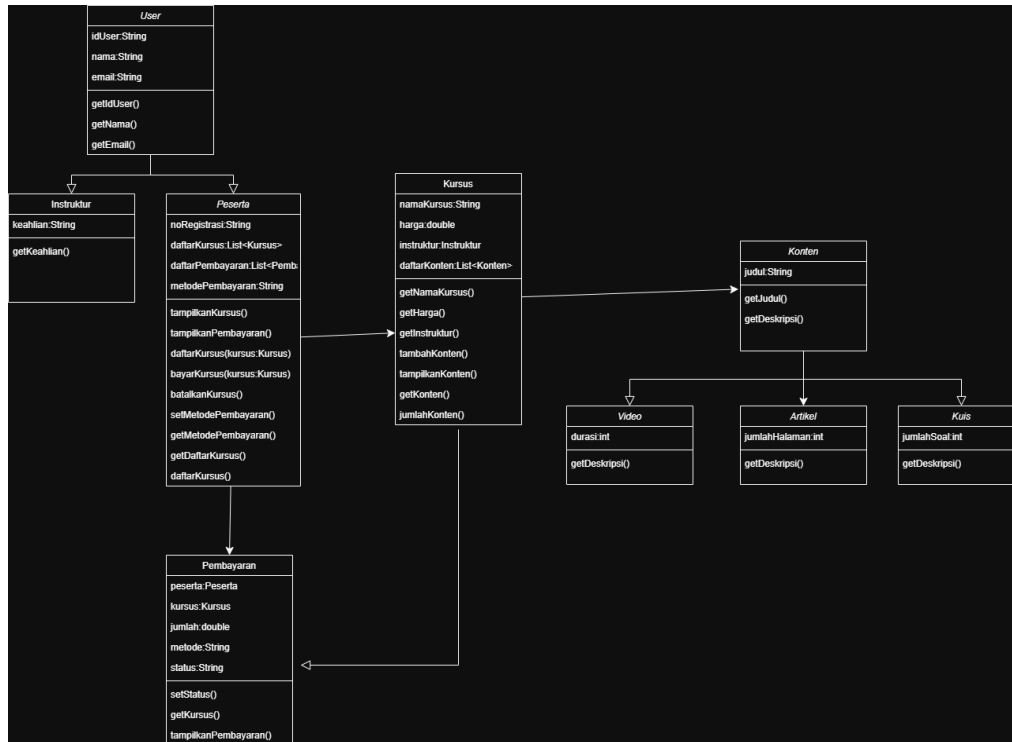
Pertanyaan:

1. **Polimorfisme:** Bagaimana Anda menerapkan konsep polimorfisme pada class diagram Anda?
2. **Pewarisan:** Kelas apa saja yang dapat dijadikan kelas dasar (super class)? Kelas apa saja yang dapat dijadikan kelas turunan(subclass)?
3. **Enkapsulasi:** Bagaimana Anda melindungi data anggota (attributes) dari akses langsung?

Kuis praktik, Terapkan dalam kode program.

2. PENYELESAIAN:

2.1 Class Diagram UML



2.2 Kode Program

2.2.1 class User

```
public class User{
    private String idUser;
    private String nama;
    private String email;

    public User(String idUser, String nama, String email) {
        this.idUser = idUser;
        this.nama = nama;
        this.email = email;
    }

    public String getIdUser() {
        return idUser;
    }

    public String getNama() {
        return nama;
    }

    public String getEmail() {
        return email;
    }
}
```

2.2.2 class Instruktur

```
public class Instruktur extends User {
    private String keahlian;

    public Instruktur(String idUser, String nama, String email, String keahlian) {
        super(idUser, nama, email);
        this.keahlian = keahlian;
    }

    public String getKeahlian() {
        return keahlian;
    }
}
```

2.2.3 class Peserta

```
import java.util.ArrayList;
public class Peserta extends User {
    private String noRegistrasi;
    private String metodePembayaran;
    private ArrayList<Kursus> daftarKursus;
    private ArrayList<Pembayaran> daftarPembayaran;

    public Peserta(String idUser, String nama, String email, String noRegistrasi) {
        super(idUser, nama, email);
        this.noRegistrasi = noRegistrasi;
        this.daftarKursus = new ArrayList<>();
        this.daftarPembayaran = new ArrayList<>();
    }

    public void daftarKursus(Kursus k) {
        if(!daftarKursus.contains(k)) {
            daftarKursus.add(k);
            System.out.println(getNama() + " berhasil daftar kursus " + k.getNamaKursus());
        } else {
            System.out.println(getNama() + " sudah terdaftar di kursus " + k.getNamaKursus());
        }
    }

    public void bayarKursus(Kursus k, String metodePembayaran) {
        for (Pembayaran pb : daftarPembayaran) {
            if (pb.getKursus().equals(k)) {
                System.out.println(getNama() + " sudah pernah membayar kursus " + k.getNamaKursus());
                return;
            }
        }
        Pembayaran p = new Pembayaran(this, k, k.getHarga(), metodePembayaran, status:"Lunas");
        daftarPembayaran.add(p);
        System.out.println(getNama() + " membayar kursus " + k.getNamaKursus() +
            " sebesar " + k.getHarga() + " melalui " + metodePembayaran);
    }

    public void tampilkanPembayaran() {
        System.out.println("Riwayat pembayaran " + getNama() + ":");
        if (daftarPembayaran.isEmpty()) {
            System.out.println(x: " (Belum ada pembayaran)");
        } else {
            for (Pembayaran p : daftarPembayaran) {
                p.tampilkan();
            }
        }
    }
}
```

```

    }
    System.out.println();
}

public String getMetodePembayaran() {
    return metodePembayaran;
}

public void setMetodePembayaran(String metodePembayaran) {
    this.metodePembayaran = metodePembayaran;
}

public void batalkanKursus(Kursus k) {
    if (daftarKursus.remove(k)) {
        System.out.println(getNama() + " membatalkan kursus: " + k.getNamaKursus());
        for (Pembayaran pb : daftarPembayaran) {
            if (pb.getKursus().equals(k)) {
                pb.setStatus(status:"Refund");
            }
        }
    } else {
        System.out.println(getNama() + " tidak sedang mengikuti kursus " + k.getNamaKursus());
    }
}

public void tampilkanKursus() {
    System.out.println("Kursus yang diikuti " + getNama() + ":");
    if (daftarKursus.isEmpty()) {
        System.out.println(x:" (Tidak ada kursus yang diikuti)");
    } else {
        for (Kursus k : daftarKursus) {
            System.out.println(" - " + k.getNamaKursus());
        }
    }
    System.out.println();
}

public ArrayList<Kursus> getDaftarKursus() {
    return daftarKursus;
}
}

```

2.2.4 class Pembayaran

```
public class Pembayaran {
    private Peserta peserta;
    private Kursus kursus;
    private double jumlah;
    private String metode;
    private String status;

    public Pembayaran(Peserta peserta, Kursus kursus, double jumlah, String metode, String status) {
        this.peserta = peserta;
        this.kursus = kursus;
        this.jumlah = jumlah;
        this.metode = metode;
        this.status = status;
    }

    public void setStatus(String status) {
        this.status = status;
    }

    public Kursus getKursus() {
        return kursus;
    }

    public void tampilkan() {
        System.out.println(peserta.getNama() + " membayar kursus " + kursus.getNamaKursus() +
            " sebesar Rp" + jumlah + " via " + metode + " [" + status + "]");
    }
}
```

2.2.5 class Konten

```
public abstract class Konten {
    private String judul;

    public Konten(String judul) {
        this.judul = judul;
    }

    public String getJudul() {
        return judul;
    }

    public abstract String getDeskripsi();
}
```

2.2.6 class Kursus

```
import java.util.ArrayList;

public class Kursus {
    private String namaKursus;
    private double harga;
    private Instruktur instruktur;
    private ArrayList<Konten> daftarKonten;

    public Kursus(String namaKursus, double harga, Instruktur instruktur) {
        this.namaKursus = namaKursus;
        this.harga = harga;
        this.instruktur = instruktur;
        this.daftarKonten = new ArrayList<>();
    }

    public String getNamaKursus() {
        return namaKursus;
    }

    public double getHarga() {
        return harga;
    }

    public Instruktur getInstruktur() {
        return instruktur;
    }

    public void tambahKonten(Konten k) {
        daftarKonten.add(k);
    }

    public void tampilkanKonten() {
        System.out.println("Konten dalam kursus " + namaKursus + ":");
        for (int i = 0; i < daftarKonten.size(); i++) {
            System.out.println("  " + (i + 1) + ". " + daftarKonten.get(i).getDeskripsi());
        }
        System.out.println();
    }

    public Konten getKonten(int index) {
        if (index >= 0 && index < daftarKonten.size()) {
            return daftarKonten.get(index);
        } else {
            System.out.println(x:"Konten tidak valid!");
            return null;
        }
    }

    public int jumlahKonten() {
        return daftarKonten.size();
    }
}
```

2.2.7 class Video

```
public class Video extends Konten {
    private int durasi; // dalam menit

    public Video(String judul, int durasi) {
        super(judul);
        this.durasi = durasi;
    }

    @Override
    public String getDeskripsi() {
        return "Video : " + getJudul() + " (" + durasi + " menit)";
    }
}
```

2.2.8 class Artikel

```
public class Artikel extends Konten{
    private int jumlahHalaman;

    public Artikel(String judul, int jumlahHalaman) {
        super(judul);
        this.jumlahHalaman = jumlahHalaman;
    }

    @Override
    public String getDeskripsi() {
        return "Artikel : " + getJudul() + " (" + jumlahHalaman + " halaman)";
    }
}
```

2.2.9 class Kuis

```
public class Kuis extends Konten {
    private int jumlahSoal;

    public Kuis(String judul, int jumlahSoal) {
        super(judul);
        this.jumlahSoal = jumlahSoal;
    }

    @Override
    public String getDeskripsi() {
        return "Kuis : " + getJudul() + " (" + jumlahSoal + " soal)";
    }
}
```


2.2.10 class Main

```
import java.util.ArrayList;
import java.util.Scanner;
public class Main {
    Run | Debug | Run main | Debug main
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        // Membuat instruktur dan kursus awal
        Instruktur instruktur1 = new Instruktur(idUser:"I001", nama:"Pak Budi", email:"budi@gmail.com", keahlian:"Java");
        Instruktur instruktur2 = new Instruktur(idUser:"I002", nama:"Bu Sari", email:"sari@gmail.com", keahlian:"Database");

        Kursus kursus1 = new Kursus(namaKursus:"Pemrograman Java", harga:500000, instruktur1);
        kursus1.tambahKonten(new Video(judul:"Intro Java", durasi:15));
        kursus1.tambahKonten(new Artikel(judul:"Konsep OOP", jumlahHalaman:5));
        kursus1.tambahKonten(new Kuis(judul:"Latihan OOP", jumlahSoal:10));

        Kursus kursus2 = new Kursus(namaKursus:"Basis Data", harga:400000, instruktur2);
        kursus2.tambahKonten(new Video(judul:"ERD", durasi:20));
        kursus2.tambahKonten(new Artikel(judul:"Normalisasi", jumlahHalaman:6));
        kursus2.tambahKonten(new Kuis(judul:"Latihan SQL", jumlahSoal:8));

        ArrayList<Kursus> semuaKursus = new ArrayList<>();
        semuaKursus.add(kursus1);
        semuaKursus.add(kursus2);

        // Input peserta secara dinamis
        System.out.print(s:"Masukkan jumlah peserta: ");
        int jumlahPeserta = sc.nextInt();
        sc.nextLine();

        ArrayList<Peserta> pesertaList = new ArrayList<>();

        for (int i = 0; i < jumlahPeserta; i++) {
            System.out.println(x:"\n=====");
            System.out.println("Data Peserta ke-" + (i + 1));
            System.out.println(x:"=====");
            System.out.print(s:"ID Peserta      : ");
            String id = sc.nextLine();
            System.out.print(s:"Nama Peserta   : ");
            String nama = sc.nextLine();
            System.out.print(s:"Email Peserta  : ");
            String email = sc.nextLine();
            System.out.print(s:"No Registrasi  : ");
            String noReg = sc.nextLine();
        }
    }
}
```

```

Peserta peserta = new Peserta(id, nama, email, noReg);

// Input metode pembayaran sekali saja per peserta
System.out.print(s:"Metode Pembayaran (Transfer Bank / E-Wallet): ");
String metode = sc.nextLine();
peserta.setMetodePembayaran(metode);

pesertaList.add(peserta);

// Pemilihan kursus
boolean tambahKursus = true;
while (tambahKursus) {
    System.out.println(x:"\nDaftar Kursus Tersedia:");
    for (int j = 0; j < semuaKursus.size(); j++) {
        Kursus k = semuaKursus.get(j);
        System.out.println((j + 1) + ". " + k.getNamaKursus() + " - Rp" +
            k.getHarga() + " (Instruktur: " + k.getInstruktur().getNama() + ")");
    }
    System.out.print("Pilih kursus (1-" + semuaKursus.size() + "): ");
    int pilihKursus = sc.nextInt();
    sc.nextLine();

    if (pilihKursus >= 1 && pilihKursus <= semuaKursus.size()) {
        Kursus kDipilih = semuaKursus.get(pilihKursus - 1);
        peserta.daftarKursus(kDipilih);
        peserta.bayarKursus(kDipilih, peserta.getMetodePembayaran());

        // Pemilihan apakah ingin menambah kursus lain?
        System.out.print(s:"Tambah kursus lain? (y/n): ");
        String jawab = sc.nextLine();
        if (!jawab.equalsIgnoreCase(anotherString:"y")) {
            tambahKursus = false;
        }
    } else {
        System.out.println(x:"Pilihan kursus tidak valid!");
    }
}
}

```

```

// Memilih konten untuk setiap kursus peserta
for (Kursus k : peserta.getDaftarKursus()) {
    k.tampilkanKonten();
    boolean pilihKontenValid = false;
    while (!pilihKontenValid) {
        System.out.print("Pilih konten yang ingin dibuka (1-" + k.jumlahKonten() + "): ");
        int pilihKonten = sc.nextInt();
        sc.nextLine();
        Konten kontenDipilih = k.getKonten(pilihKonten - 1);
        if (kontenDipilih != null) {
            System.out.println(">>> " + peserta.getNama() + " membuka " + kontenDipilih.getDeskripsi());
            pilihKontenValid = true;
        } else {
            System.out.println(x:"Konten tidak valid, coba lagi.");
        }
    }
}

// Membatalkan kursus jika peserta ingin membatalkan
System.out.println(x:"\n=== Batalkan Kursus ===");
for (Peserta p : pesertaList) {
    for (Kursus k : new ArrayList<>(p.getDaftarKursus())) {
        System.out.print("Apakah " + p.getNama() + " ingin membatalkan kursus " + k.getNamaKursus() + "? (y/n): ");
        String jawaban = sc.nextLine();
        if (jawaban.equalsIgnoreCase(anotherString:"y")) {
            p.batalkanKursus(k);
        }
    }
}

// Menampilkan daftar kursus dan riwayat pembayaran setiap peserta
System.out.println(x:"\n=== Daftar Kursus dan Riwayat Pembayaran ===");
for (Peserta p : pesertaList) {
    p.tampilkanKursus();
    p.tampilkanPembayaran();
    System.out.println(x:"-----");
}
sc.close();
}
}

```

3. OUTPUT

Masukkan jumlah peserta: 2

=====

Data Peserta ke-1

=====

ID Peserta : 001
Nama Peserta : Dafa
Email Peserta : Dafa@gmail.com
No Registrasi : 011
Metode Pembayaran (Transfer Bank / E-Wallet): Bank

Daftar Kursus Tersedia:

1. Pemrograman Java - Rp500000.0 (Instruktur: Pak Budi)
2. Basis Data - Rp400000.0 (Instruktur: Bu Sari)

Pilih kursus (1-2): 1

Dafa berhasil daftar kursus Pemrograman Java

Dafa membayar kursus Pemrograman Java sebesar 500000.0 melalui Bank

Tambah kursus lain? (y/n): n

Konten dalam kursus Pemrograman Java:

1. Video : Intro Java (15 menit)
2. Artikel : Konsep OOP (5 halaman)
3. Kuis : Latihan OOP (10 soal)

Pilih konten yang ingin dibuka (1-3): 1

>>> Dafa membuka Video : Intro Java (15 menit)

=====

Data Peserta ke-2

=====

ID Peserta : 002
Nama Peserta : Andi
Email Peserta : Andi@gmail.com
No Registrasi : 022
Metode Pembayaran (Transfer Bank / E-Wallet): E-Wallet

Daftar Kursus Tersedia:

1. Pemrograman Java - Rp500000.0 (Instruktur: Pak Budi)
2. Basis Data - Rp400000.0 (Instruktur: Bu Sari)

Pilih kursus (1-2): 2

```

Pilih kursus (1-2): 2
Andi berhasil daftar kursus Basis Data
Andi membayar kursus Basis Data sebesar 400000.0 melalui E-Wallet
Tambah kursus lain? (y/n): y

Daftar Kursus Tersedia:
1. Pemrograman Java - Rp500000.0 (Instruktur: Pak Budi)
2. Basis Data - Rp400000.0 (Instruktur: Bu Sari)
Pilih kursus (1-2): 1
Andi berhasil daftar kursus Pemrograman Java
Andi membayar kursus Pemrograman Java sebesar 500000.0 melalui E-Wallet
Tambah kursus lain? (y/n): n
Konten dalam kursus Basis Data:
  1. Video   : ERD (20 menit)
  2. Artikel : Normalisasi (6 halaman)
  3. Kuis    : Latihan SQL (8 soal)

Pilih konten yang ingin dibuka (1-3): 3
>>> Andi membuka Kuis      : Latihan SQL (8 soal)
Konten dalam kursus Pemrograman Java:
  1. Video   : Intro Java (15 menit)
  2. Artikel : Konsep OOP (5 halaman)
  3. Kuis    : Latihan OOP (10 soal)

Pilih konten yang ingin dibuka (1-3): 1
>>> Andi membuka Video    : Intro Java (15 menit)

=== Batalkan Kursus ===
Apakah Dafa ingin membatalkan kursus Pemrograman Java? (y/n): y
Dafa membatalkan kursus: Pemrograman Java
Apakah Andi ingin membatalkan kursus Basis Data? (y/n): n
Apakah Andi ingin membatalkan kursus Pemrograman Java? (y/n): n

=== Daftar Kursus dan Riwayat Pembayaran ===
Kursus yang diikuti Dafa:
  (Tidak ada kursus yang diikuti)

Riwayat pembayaran Dafa:
Dafa membayar kursus Pemrograman Java sebesar Rp500000.0 via Bank [Refund]

Riwayat pembayaran Dafa:
Dafa membayar kursus Pemrograman Java sebesar Rp500000.0 via Bank [Refund]

-----
Kursus yang diikuti Andi:
- Basis Data
- Pemrograman Java

Riwayat pembayaran Andi:
Andi membayar kursus Basis Data sebesar Rp400000.0 via E-Wallet [Lunas]
Andi membayar kursus Pemrograman Java sebesar Rp500000.0 via E-Wallet [Lunas]

```

4. PERTANYAAN

1. Polimorfisme: Bagaimana Anda menerapkan konsep polimorfisme pada class diagram Anda?

➤ Jawab: Polimorfisme diterapkan pada kelas Konten dan turunannya (Video, Artikel, Kuis)

- Konten didefinisikan sebagai kelas abstrak dengan method `getDeskripsi()`.
- Setiap subclass (Video, Artikel, Kuis) mengimplementasikan method `getDeskripsi()` dengan cara berbeda sesuai jenis kontennya.
- Pada saat program berjalan, ketika peserta memilih konten, objek dipanggil melalui referensi Konten, tetapi method yang dieksekusi sesuai dengan objek aslinya.
Contoh: `kontenDipilih.getDeskripsi()` bisa mengembalikan "Video : Intro Java (15 menit)" atau "Artikel : Konsep OOP (5 halaman)" tergantung objek aslinya.

2. Pewarisan: Kelas apa saja yang dapat dijadikan kelas dasar (super class)? Kelas apa saja yang dapat dijadikan kelas turunan(subclass)

➤ Jawab:

- Superclass (kelas dasar):
 - ➔ User: menyimpan atribut umum (id, nama, email).
 - ➔ Konten: sebagai abstrak konten kursus.
- Subclass (kelas turunan):
 - ➔ Instruktur dan Peserta: turunan dari User.
 - ➔ Video, Artikel, Kuis: turunan dari Konten.

3. Enkapsulasi: Bagaimana Anda melindungi data anggota (attributes) dari akses langsung?

➤ Jawab:

Enkapsulasi dilakukan dengan cara:

- Semua atribut di kelas (User, Peserta, Instruktur, Kursus, Pembayaran, Konten) diset sebagai private (atau protected jika hanya boleh diakses subclass).
- Untuk mengakses/mengubah nilai atribut, digunakan getter & setter method.
Contoh: `getNama()`, `setMetodePembayaran()`.
- Dengan begitu, data tidak bisa diakses langsung dari luar kelas.
Hal ini menjaga keamanan data dan memastikan perubahan hanya melalui method yang sudah dikontrol.