**Laporan Jobsheet 4**

**Pemrograman Berorientasi Objek (PBO)**

**Percobaan 1**

**A screenshot of a computer code

AI-generated content may be incorrect.**

1. Buat class sesuai diagram class diatas

**Class Laptop**

public class Laptop {

    private String merk;

    private Processor proc;

    public Laptop() {

    }

    public Laptop(String merk, Processor proc) {

        this.merk = merk;

        this.proc = proc;

    }

    public void setMerk(String merk) {

        this.merk = merk;

    }

    public void setProc(Processor proc) {

        this.proc = proc;

    }

    public String getMerk() {

        return merk;

    }

    public Processor getProc() {

        return proc;

    }

  public void info() {

        System.out.println("Merk Laptop: " + this.merk);

        this.proc.info();

    }

}

**Class Processor**

public class Processor {

    private String merk;

    private double cache;

    public Processor() {

    }

    public Processor(String merk, double cache) {

        this.merk = merk;

        this.cache = cache;

    }

    public void setMerk(String merk) {

        this.merk = merk;

    }

    public void setCache(double cache) {

        this.cache = cache;

    }

    public String getMerk() {

        return merk;

    }

    public double getCache() {

        return cache;

    }

    public void info() {

        System.out.println("Merk Prosesor: " + this.merk);

        System.out.println("Cache Prosesor: " + this.cache + " GB");

    }

}

**Pertanyaan**

1. Di dalam class Processor dan class Laptop , terdapat method setter dan getter untuk masing‑masing atributnya. Apakah gunanya method setter dan getter tersebut ?

= **Method Setter** digunakan untuk mengatur nilai dari sebuah atribut. **Method Getter** digunakan untuk mendapatkan nilai dari sebuah atribut

1. Di dalam class Processor dan class Laptop, masing‑masing terdapat konstruktor default dan konstruktor berparameter. Bagaimanakah beda penggunaan dari kedua jenis konstruktor tersebut ?

* Konstruktor Default : Konstruktor ini tidak memiliki parameter. Gunanya adalah untuk membuat objek tanpa langsung memberikan nilai awal pada atribut-atributnya.

Contoh penggunaan : Laptop L1 = new Laptop();

* Konstruktor Berparameter : Konstruktor ini memiliki parameter yang digunakan untuk menginisialisasi atribut saat objek dibuat. Ini memungkinkan Anda membuat objek dan langsung memberikan nilai pada atribut didalam baris kode.

Contoh penggunaan : Laptop L1 = new Laptop(“Thinkpad”, p);

1. Perhatikan class Laptop, di antara 2 atribut yang dimiliki (merk dan proc), atribut manakah yang bertipe object ?

= pada kelas Laptop, atribut yang bertipe objek adalah proc, tipe datanya adalah Processor, Tipe datanya adalah Processor, yang merupakan kelas lain yang didefinisikan. Atribut merek bertipe data String, yang juga merupakan objek terapi sering dianggap sebagai tipe data dasar

1. Perhatikan class Laptop, pada baris manakah yang menunjukkan bahwa class Laptop memiliki relasi dengan class Processor ?

= private Processor proc;

1. Perhatikan pada class Laptop , Apakah guna dari sintaks proc.info() ?

= Memanggil method info() dari objek Processor yang menjadi atribut di dalam objek Laptop

1. Pada class MainPercobaan1, terdapat baris kode: Laptop l = new Laptop("Thinkpad", p);. Apakah p tersebut ? Dan apakah yang terjadi jika baris kode tersebut diubah menjadi:

= Variabel p adalah objek Processor yang sudah dibuat. Mengubah kode menjadi Laptop l = new Laptop("Thinkpad", new Processor("Intel i5", 3)); akan memberikan hasil yang sama. Bedanya, cara pertama membuat objek Processor di variabel p lalu menggunakannya, sedangkan cara kedua membuat objek Processor baru langsung di dalam konstruktor Laptop tanpa menyimpannya di variabel terpisah.

**Percobaan 2**

**A screenshot of a computer code

AI-generated content may be incorrect.**

1. Buat class sesuai diagram class diatas

**Class Mobil**

public class Mobil {

private String nama;

private int biaya;

public Mobil(){

}

public void setNama(String nama){

this.nama = nama;

}

public String getNama(){

return nama;

}

public void setBiaya(int biaya){

this.biaya = biaya;

}

public int getBiaya() {

return biaya;

}

public int hitungBiayaMobil(int hari){

return biaya \* hari;

}

}

**Class Sopir**

public class Sopir {

    private String nama;

    private int biaya;

    public Sopir(){

    }

    public void setNama(String nama){

        this.nama = nama;

    }

    public String getNama(){

        return nama;

    }

    public void setBiaya(int biaya){

        this.biaya = biaya;

    }

    public int getBiaya(){

        return biaya;

    }

    public int hitungBiayaSopir(int hari){

        return biaya \* hari;

    }

}

**Class Pelanggan**

public class Pelanggan {

    private String nama;

    private Mobil mobil;

    private Sopir sopir;

    private int hari;



    public Pelanggan(){

    }

    public void setNama(String nama){

        this.nama = nama;

    }

    public String getNama(){

        return nama;

    }

    public void setMobil(Mobil mobil){

        this.mobil = mobil;

    }

    public Mobil getMobil(){

        return mobil;

    }

    public void setSopir(Sopir sopir){

        this.sopir = sopir;

    }

    public Sopir getSopir(){

        return sopir;

    }

    public void setHari(int hari){

        this.hari = hari;

    }

    public int getHari(){

        return hari;

    }

    public int hitungBiayaTotal(){

        return mobil.hitungBiayaMobil(hari) +

        sopir.hitungBiayaSopir(hari);

    }

}

**Pertanyaan**

1. Perhatikan class Pelanggan. Pada baris program manakah yang menunjukan bahwa class Pelanggan memiliki relasi dengan class Mobil dan class Sopir ?

* private Mobil mobil;
* private Sopir sopir;

1. Perhatikan method hitungBiayaSopir pada class Sopir, serta method hitungBiayaMobil pada class Mobil. Mengapa menurut Anda method tersebut harus memiliki argument hari ?

= biaya sewa atau jasa dihitung **berdasarkan jumlah hari**

1. Perhatikan kode dari class Pelanggan. Untuk apakah perintah mobil.hitungBiayaMobil(hari) dan sopir.hitungBiayaSopir(hari) ?

= untuk menghitung total biaya yang harus dibayar pelanggan

1. Perhatikan class MainPercobaan2. Untuk apakah sintaks p.setMobil(m) dan p.setSopir(s) ?

= **menghubungkan objek** Mobil (m) dan Sopir (s) dengan objek Pelanggan (p)

1. Perhatikan class MainPercobaan2. Untuk apakah proses p.hitungBiayaTotal() tersebut ?

= **menghitung dan mendapatkan total biaya** dari seluruh transaksi sewa

1. Perhatikan class MainPercobaan2, coba tambahkan pada baris terakhir dari method main dan amati perubahan saat di‑run!

System.out.println(p.getMobil().getMerk());

Jadi untuk apakah sintaks p.getMobil().getMerk() yang ada di dalam method *main* tersebut?

* **p.getMobil()**: Perintah ini mengambil objek Mobil yang terhubung dengan objek Pelanggan (p).
* **getMerk()**: Setelah mendapatkan objek Mobil, sintaks ini mencoba memanggil method getMerk(). Namun, berdasarkan kode Mobil.java yang Anda berikan, tidak ada method bernama getMerk(). Method yang ada adalah getNama().

**Percobaan 3**

**Class Pegawai**

public class Pegawai {

    private String nip;

    private String nama;

    public Pegawai(String nip, String nama){

        this.nip = nip;

        this.nama = nama;

    }

    public void setNip(String nip){

        this.nip = nip;

    }

    public String getNip(){

        return nip;

    }

    public void setNama(String nama){

        this.nama = nama;

    }

    public String getNama(){

        return nama;

    }

    public String info(){

        String info = "";

        info += "NIP\t: " + this.nip + "\n";

        info += "Nama\t: " + this.nama + "\n";

        return info;

    }

}

**Class KeretaApi**

public class KeretaApi {

    private String nama;

    private String kelas;

    private Pegawai masinis;

    private Pegawai asisten;

    /\*

    public KeretaApi(String nama, String kelas, Pegawai masinis) {

        this.nama = nama;

        this.kelas = kelas;

        this.masinis = masinis;

    }

    \*/

    public KeretaApi(String nama, String kelas, Pegawai masinis, Pegawai asisten) {

        this.nama = nama;

        this.kelas = kelas;

        this.masinis = masinis;

        this.asisten = asisten;

    }

public void setNama(String nama){

        this.nama = nama;

    }

    public String getNama(){

        return nama;

    }

    public void setKelas(String kelas){

        this.kelas = kelas;

    }

    public String getKelas(){

        return kelas;

    }

    public void setMasinis(Pegawai masinis){

        this.masinis = masinis;

    }

    public Pegawai getMasinis(){

        return masinis;

    }

    public void setAsisten(Pegawai asisten){

        this.asisten = asisten;

    }

public String info() {

        String info = "";

        info += "Nama\t: " + this.nama + "\n";

        info += "Kelas\t: " + this.kelas + "\n";

        info += "-----------------------------\n";

        info += "=========--Masinis--=========\n" + this.masinis.info();

        info += "=========--Asisten--=========\n" + this.asisten.info();

        info += "-----------------------------";

        return info;

    }

}

**Class MainPercobaan3**

package Prasojo.relasiclass.percobaan3;

public class MainPercobaan3 {

    public static void main(String[] args) {

        Pegawai masinis = new Pegawai("1234", "Spongebob Squarepants");

        Pegawai asisten = new Pegawai("4567", "Patrick Star");

        KeretaApi keretaApi = new KeretaApi("Gaya Baru", "Bisnis", masinis, asisten);

        System.out.println(keretaApi.info());

    }

}

**Pertanyaan**

1. Di dalam method info() pada class KeretaApi, baris this.masinis.info() dan this.asisten.info() digunakan untuk apa ?

= untuk memanggil metode **info() dari objek Pegawai**

1. Buatlah main program baru dengan nama class MainPertanyaan pada package yang sama. Tambahkan kode berikut pada method main() !

Pegawai masinis = new Pegawai("1234", "Spongebob Squarepants");

KeretaApi keretaApi = new KeretaApi("Gaya Baru","Bisnis", masinis);

System.out.println(keretaApi.info());

=

A screen shot of a computer program

AI-generated content may be incorrect.

1. Apa hasil output dari main program tersebut ? Mengapa hal tersebut dapat terjadi ?

A screen shot of a computer screen

AI-generated content may be incorrect.

Penyebabnya adalah **konstruktor** yang dipanggil di kelas MainPertanyaan (new KeretaApi("Gaya Baru", "Bisnis", masinis)) tidak cocok dengan konstruktor yang tersedia di kelas KeretaApi.java.

* Di kelas MainPertanyaan, kita mencoba memanggil konstruktor dengan **tiga argumen**: nama (String), kelas (String), dan masinis (objek Pegawai).
* Namun, di kelas KeretaApi.java, konstruktor yang ada **membutuhkan empat argumen**: nama (String), kelas (String), masinis (objek Pegawai), dan **asisten (objek Pegawai)**.

Oleh karena itu, Java tidak dapat menemukan konstruktor yang sesuai untuk membuat objek keretaApi, sehingga terjadi error.

1. Perbaiki class KeretaApi sehingga program dapat berjalan !

public class MainPertanyaan {

    public static void main(String[] args) {

        Pegawai masinis = new Pegawai("1234", "Spongebob Squarepants");

        Pegawai asisten = new Pegawai("4567", "Prasojo");

        KeretaApi keretaApi = new KeretaApi("Gaya Baru", "Bisnis", masinis, asisten);

   System.out.println(keretaApi.info());

    }

}

**Percobaan 4**

**Class Penumpang**

public class Penumpang {

    private String ktp;

    private String nama;

    public Penumpang(String ktp, String nama) {

        this.ktp = ktp;

        this.nama = nama;

    }

    public void setKtp(String ktp) {

        this.ktp = ktp;

    }

    public String getKtp() {

        return ktp;

    }

    public void setNama(String nama) {

        this.nama = nama;

    }

    public String getNama() {

        return nama;

    }

    public String info() {

        String info = "";

        info += "Ktp: " + ktp + "\n";

        info += "Nama: " + nama + "\n";

        return info;

    }

}

**Class Kursi**

public class Kursi {

    private String nomor;

    private Penumpang penumpang;

    public Kursi(String nomor) {

        this.nomor = nomor;

    }

    public void setNomor(String nomor) {

        this.nomor = nomor;

    }

    public String getNomor() {

        return nomor;

    }

    public void setPenumpang(Penumpang penumpang) {

        this.penumpang = penumpang;

    }

    public Penumpang getPenumpang() {

        return penumpang;

    }

    public String info() {

        String info = "";

        info += "Nomor: " + nomor + "\n";

        if (this.penumpang != null) {

            info += "Penumpang: " + penumpang.info() + "\n";

        }

        return info;

    }

}

**Class Gerbong**

public class Gerbong {

    private String kode;

    private Kursi[] arrayKursi;

    public Gerbong(String kode, int jumlah) {

        this.kode = kode;

        this.arrayKursi = new Kursi[jumlah];

        this.initKursi();

    }

    private void initKursi() {

        for (int i = 0; i < arrayKursi.length; i++) {

            this.arrayKursi[i] = new Kursi(String.valueOf(i + 1));

        }

    }

    public void setKode(String kode) {

        this.kode = kode;

    }

    public String getKode() {

        return kode;

    }

    public void setPenumpang(Penumpang penumpang, int nomor) {

        this.arrayKursi[nomor - 1].setPenumpang(penumpang);

    }

    public Kursi[] getArrayKursi() {

        return arrayKursi;

    }

    public String info() {

        String info = "";

        info += "Kode: " + kode + "\n";

        for (Kursi kursi : arrayKursi) {

            info += kursi.info();

        }

        return info;

    }

}

**Pertanyaan**

1. Pada main program dalam class MainPercobaan4, berapakah jumlah kursi dalam Gerbong A ?

= 10 kursi

1. Perhatikan potongan kode pada method info() dalam class Kursi. Apa maksud kode tersebut ?

...

if (this.penumpang != null) {

info += "Penumpang: " + penumpang.info() + "\n";

} ...

= maksud dari kode ini adalah melakukan pengecekan apakah sebuah kursi sudah memiliki penumpang atau belum.

* Jika this.penumpang tidak null, artinya kursi tersebut sudah diduduki, maka informasi penumpang akan ditambahkan ke dalam string info dengan memanggil method penumpang.info().
* Jika this.penumpang adalah null, artinya kursi tersebut kosong, maka blok if akan dilewati dan informasi penumpang tidak akan dicetak. Ini mencegah terjadinya NullPointerException saat mencoba memanggil method dari objek yang belum ada (null).

1. Mengapa pada method setPenumpang() dalam class Gerbong, nilai nomor dikurangi dengan angka 1 ?

= karena indeks awal array nya adalah 0, maka harus dikurangi 1 untuk mendapatkan nilai array yang pertama

1. Instansiasi objek baru budi dengan tipe Penumpang, kemudian masukkan objek baru tersebut pada gerbong dengan gerbong.setPenumpang(budi, 1). Apakah yang Page 9 of 10 terjadi ?

A screen shot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

= Yang terjadi adalah **data penumpang pada kursi nomor 1 akan ditimpa**. Awalnya, kursi nomor 1 diduduki oleh "Mr. Krab". Ketika gerbong.setPenumpang(budi, 1) ditambahkan, objek penumpang di dalam objek Kursi pada indeks 0 akan diganti dari objek p ("Mr. Krab") menjadi objek budi. Hasil akhirnya, kursi nomor 1 akan ditempati oleh "Budi"

1. Modifikasi program sehingga tidak diperkenankan untuk menduduki kursi yang sudah ada penumpang lain !

A screen shot of a computer code

AI-generated content may be incorrect.

A black screen with white text

AI-generated content may be incorrect.

**Tugas**

Sistem ini akan mengelola buku, anggota, dan proses peminjaman di sebuah perpustakaan.

* Buku: Kelas untuk merepresentasikan data sebuah buku.
  + Atribut: judul, penulis, penerbit.
  + Method: Constructor, setter, getter, info().
* Anggota: Kelas untuk merepresentasikan data anggota perpustakaan.
  + Atribut: idAnggota, nama.
  + Method: Constructor, setter, getter, info().
* Peminjaman: Kelas yang merepresentasikan satu transaksi peminjaman. Kelas ini memiliki relasi *has-a* dengan Anggota dan Buku.
  + Atribut: anggota (tipe Anggota), buku (tipe Buku), tglPinjam, tglKembali.
  + Method: Constructor, setter, getter, info().
* Perpustakaan: Kelas utama yang mengelola daftar buku, anggota, dan transaksi peminjaman.
  + Atribut: listBuku (array Buku), listAnggota (array Anggota), listPeminjaman (array Peminjaman).
  + Method: tambahBuku(), tambahAnggota(), prosesPeminjaman(), tampilkanDataPeminjaman().

**Kode Program**

**A screen shot of a computer program

AI-generated content may be incorrect.**

**A screen shot of a computer program

AI-generated content may be incorrect.**

**A screen shot of a computer program

AI-generated content may be incorrect.**

**A screen shot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**