Nama : Hafizh Naufal Raditya

NIM : H1D024061

SHIFT BARU : E

1. Soal Pertemuan Satu :

   Pengguna.java

```
class Pengguna {

    int umur;
    double berat;
    double tinggi;
    final double faktorAktivitas = 1.2;

    Pengguna(int umur, double berat, double tinggi) {
        this.umur = umur;
        this.berat = berat;
        this.tinggi = tinggi;
    }

    double hitungBMI() {
        double tinggiMeter = this.tinggi / 100.0;
        return this.berat / (tinggiMeter * tinggiMeter);
    }

    void tampilKategori() {
        double bmi = hitungBMI();
        if (bmi < 18.5) {
            System.out.println("Kategori: Kurus");
        } else if (bmi < 25) {
            System.out.println("Kategori: Normal");
        } else {
            System.out.println("Kategori: Obesitas");
        }
    }

    double hitungKalori() {
        return this.berat * this.faktorAktivitas * 10.0;
    }

    void tampilInfo() {
        System.out.println("=== Informasi Kesehatan Pengguna ===");
        System.out.println("Umur: " + this.umur + " tahun");
        System.out.println("Berat: " + this.berat + " kg");
```

```
        System.out.println("Tinggi: " + this.tinggi + " cm");
        System.out.printf("BMI: %.2f\n", hitungBMI());
        tampilKategori();
        System.out.printf("Kebutuhan Kalori: %.0f kal/hari\n", hitungKalori());
        System.out.println("====================================");
    }
}
```

UjiPengguna.java

```
class UjiPengguna {
    public static void main(String[] args) {
        Pengguna p = new Pengguna(25, 60.0, 170.0);
        p.tampilInfo();

        System.out.println();
        System.out.println("--- Demonstrasi Konstanta FINAL ---");
        System.out.println("Nilai faktor aktivitas: " + p.faktorAktivitas);
        System.out.println("Faktor aktivitas bersifat final (konstan)");
        System.out.println("Artinya nilai ini tidak bisa diubah setelah diinisialisasi");
        System.out.println();
        System.out.println("Jika kita mencoba menulis (contoh, dikomentari karena
menyebabkan error pada kompilasi):");
        System.out.println("// p.faktorAktivitas = 1.5; // -> compiler error: cannot assign
a value to final variable faktorAktivitas");
    }
}
```

Alur kerja program:

1. Buat objek Pengguna(25, 60.0, 170.0)
2. Panggil method tampilInfo()
3. Tampilkan data dasar (umur, berat, tinggi)
4. Hitung BMI = berat / (tinggi/100)^2
5. Tentukan kategori BMI (Kurus / Normal / Obesitas)
6. Hitung kalori = berat × faktorAktivitas × 10
7. Tampilkan semua hasil
8. Fungsi yang digunakan (signature + penjelasan singkat — langsung copyable):

Fungsi – fungsi yang digunakan :

1. Pengguna(int umur, double berat, double tinggi)
2. Konstruktor: inisialisasi atribut umur, berat, tinggi.
3. double hitungBMI()
   Mengembalikan nilai BMI = berat / (tinggiMeter * tinggiMeter).
   void tampilKategori()

Menentukan dan menampilkan kategori berdasarkan hasil hitungBMI().

4. double hitungKalori()
   Mengembalikan kebutuhan kalori harian: berat × faktorAktivitas × 10.
5. void tampilInfo()
   Menampilkan umur, berat, tinggi, BMI, kategori, dan kebutuhan kalori.
6. Final double faktorAktivitas = 1.2
   Konstanta yang dipakai pada perhitungan kalori (tidak dapat diubah).

Output :

```
PS E:\Tugas Kuliah\Semester Tiga\Praktikum PBO\Responsi 2 PBO>  & 'C:\Program Files\Eclipse Adoptium\
jdk-17.0.15.6-hotspot\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\MSI_\Ap
pData\Roaming\Code\User\workspaceStorage\782561e0349342cb08c44f6fa006ea90\redhat.java\jdt_ws\Responsi
 2 PBO_99167d3b\bin' 'UjiPengguna'
=== Informasi Kesehatan Pengguna ===
Umur: 25 tahun
Berat: 60.0 kg
Tinggi: 170.0 cm
BMI: 20.76
Kategori: Normal
Kebutuhan Kalori: 720 kal/hari
==================================

--- Demonstrasi Konstanta FINAL ---
Nilai faktor aktivitas: 1.2
Faktor aktivitas bersifat final (konstan)
Artinya nilai ini tidak bisa diubah setelah diinisialisasi

Jika kita mencoba menulis (contoh, dikomentari karena menyebabkan error pada kompilasi):
// p.faktorAktivitas = 1.5; // -> compiler error: cannot assign a value to final variable faktorAktiv
itas
PS E:\Tugas Kuliah\Semester Tiga\Praktikum PBO\Responsi 2 PBO> ^C
PS E:\Tugas Kuliah\Semester Tiga\Praktikum PBO\Responsi 2 PBO>
```

2. Soal Pert 2

BarangMusik.java

```
class BarangMusik {
    // Deklarasikan variabel/state yang diperlukan di sini
    String kode;
    String nama;
    double harga;
    int stok;

    // Constructor pertama: hanya kode dan nama
    BarangMusik(String kode, String nama) {
        this.kode = kode;
        this.nama = nama;
        this.harga = 0.0;
```

```
        this.stok = 0;
    }

    // Constructor kedua: kode, nama, dan harga
    BarangMusik(String kode, String nama, double harga) {
        this.kode = kode;
        this.nama = nama;
        this.harga = harga;
        this.stok = 0;
    }

    // Constructor ketiga: kode, nama, harga, dan stok (lengkap)
    BarangMusik(String kode, String nama, double harga, int stok) {
        this.kode = kode;
        this.nama = nama;
        this.harga = harga;
        this.stok = stok;
    }

    void ubahHarga(double hargaBaru) {
        this.harga = hargaBaru;
    }

    void tambahStok(int jumlah) {
        this.stok += jumlah;
    }

    void tampilInfo() {
        System.out.println("Alat musik " + this.nama + " | Kode: " + this.kode + " |
Harga: Rp " + this.harga + " | Stok: " + this.stok + " unit");
    }
}
```

UjiBarang.java

```
class UjiBarang {
    public static void main(String[] args) {
        // Buat objek barang pertama gitar
        BarangMusik gitar = new BarangMusik("GTR-001", "Gitar Akustik Yamaha");

        // Atur harga gitar
        gitar.ubahHarga(1500000.0);

        // Tambah stok gitar
        gitar.tambahStok(5);
```

```
        // Buat objek drum
        BarangMusik drum = new BarangMusik("DRM-001", "Drum Set Pearl",
8500000.0, 3);

        // Tampilkan informasi kedua barang
        System.out.println("=== INVENTARIS TOKO NADAKITA ===");
        gitar.tampilInfo();
        drum.tampilInfo();
    }
}
```

Alur kerja program:

1. Buat beberapa objek BarangMusik menggunakan constructor berbeda (mis. hanya kode+nama, kode+nama+harga, kode+nama+harga+stok).
2. Panggil tampilInfo() untuk tiap objek.
3. Ubah atribut menggunakan setter (mis. ubahHarga) bila diperlukan.
4. Tampilkan ulang info untuk verifikasi.

Fungsi yang digunakan:

- BarangMusik(String kode, String nama)
  Konstruktor: inisialisasi kode dan nama.
- BarangMusik(String kode, String nama, double harga)
  Konstruktor: inisialisasi kode, nama, harga.
- BarangMusik(String kode, String nama, double harga, int stok)
  Konstruktor: inisialisasi semua atribut termasuk stok.
- void ubahHarga(double hargaBaru)
  Setter untuk mengubah harga (dapat berisi validasi).
- void tampilInfo()
  Menampilkan semua atribut barang.

Output :

```
PS E:\Tugas Kuliah\Semester Tiga\Praktikum PBO\Responsi 2 PBO>  & 'C:\Program Files\Eclipse
spot\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\MSI_\AppData\Ro
torage\782561e0349342cb08c44f6fa006ea90\redhat.java\jdt_ws\Responsi 2 PBO_99167d3b\bin' 'Uji
=== INVENTARIS TOKO NADAKITA ===
Alat musik Gitar Akustik Yamaha | Kode: GTR-001 | Harga: Rp 1500000.0 | Stok: 5 unit
Alat musik Drum Set Pearl | Kode: DRM-001 | Harga: Rp 8500000.0 | Stok: 3 unit
PS E:\Tugas Kuliah\Semester Tiga\Praktikum PBO\Responsi 2 PBO>
```

3. Soal Pert 3

Karyawan.java

```
class Karyawan {

    protected String nama;
    protected double gajiPokok;


    Karyawan(String nama, double gajiPokok) {
        this.nama = nama;
        this.gajiPokok = gajiPokok;
    }


    void tampilInfo() {
        System.out.println("Nama: " + this.nama + " | Gaji Pokok: Rp " +
this.gajiPokok);
    }
}
```

Manajer.java

```
class Manajer extends Karyawan {

    private double tunjangan;


    Manajer(String nama, double gajiPokok, double tunjangan) {
        super(nama, gajiPokok);
        this.tunjangan = tunjangan;
    }


    @Override
    void tampilInfo() {
        System.out.println("Nama: " + this.nama + " | Gaji Pokok: Rp " + this.gajiPokok
+ " | Tunjangan: Rp " + this.tunjangan);
        System.out.println("Total Pendapatan: Rp " + (this.gajiPokok + this.tunjangan));
    }
}
```

UjiKaryawan.java

```
class UjiKaryawan {
```

```
    public static void main(String[] args) {
        // Tampilkan Header
        System.out.println("=== DATA KARYAWAN TECHMAJU ===");

        // Buat objek Karyawan biasa (Budi)
        Karyawan budi = new Karyawan("Budi Santoso", 4000000.0);

        // Tampilkan info Karyawan biasa
        System.out.println("Status: Karyawan Biasa");
        budi.tampilInfo();

        System.out.println(); // Baris baru/jarak

        // Buat objek Manajer (Siti)
        Manajer siti = new Manajer("Siti Aminah", 6000000.0, 2500000.0);

        // Tampilkan info Manajer
        System.out.println("Status: Manajer");
        siti.tampilInfo();
    }
}
```

Alur kerja program:

1. Buat objek Karyawan dan Manajer (Manajer extends Karyawan).
2. Panggil method yang diwariskan dan yang di-override (mis. hitungGaji, tampilInfo).
3. Bandingkan hasil antara objek parent dan child.

Fungsi yang digunakan:

- Karyawan(String nama, double gajiPokok)
  Konstruktor parent.
- double hitungGaji() (di Karyawan)
  Menghitung gaji dasar.
- Manajer(String nama, double gajiPokok, double tunjangan)
  Konstruktor child memanggil super(...).
- @Override double hitungGaji() (di Manajer)
  Override menghitung gaji + tunjangan.
- void tampilInfo() / @Override tampilInfo()
  Menampilkan informasi, child dapat memanggil super.tampilInfo() lalu menambahkan info.

Output :

```
=== DATA KARYAWAN TECHMAJU ===
Status: Karyawan Biasa
Nama: Budi Santoso | Gaji Pokok: Rp 4000000.0

Status: Manajer
Nama: Siti Aminah | Gaji Pokok: Rp 6000000.0 | Tunjangan: Rp 2500000.0
Total Pendapatan: Rp 8500000.0
PS E:\Tugas Kuliah\Semester Tiga\Praktikum PBO\Responsi 2 PBO>
```

4. Soal Pert 4

Customer.java

```
class Customer {
    // Buatkan atribut
    protected String nama;
    protected String idCustomer;
    protected int totalBelanja;

    // Sediakan constructor
    Customer(String nama, String idCustomer, int totalBelanja) {
        this.nama = nama;
        this.idCustomer = idCustomer;
        this.totalBelanja = totalBelanja;
    }

    void tampilkanInfo() {
        // Tampilkan data customer
        System.out.println("Nama: " + this.nama + " | ID: " + this.idCustomer + " | Total
Belanja: Rp " + this.totalBelanja);
    }
}
```

Member.java

```
class Member extends Customer {
    // Tambahkan atribut tambahan
    private int poinReward;
    private String level;

    // Buat constructor dengan super
    Member(String nama, String idCustomer, int totalBelanja, int poinReward, String
level) {
        super(nama, idCustomer, totalBelanja);
```

```
        this.poinReward = poinReward;
        this.level = level;
    }

    @Override
    void tampilkanInfo() {
        // Panggil super, lalu tampilkan data tambahan
        super.tampilkanInfo();
        System.out.println("Poin Reward: " + this.poinReward + " | Level: " + this.level);
    }
}
```

UjiCustomer.java

```
public class UjiCustomer {
    public static void main(String[] args) {

        // Buat object Customer
        Customer customer = new Customer("Budi Santoso", "CST-001", 500000);

        // Buat object Member
        Member member = new Member("Siti Aminah", "MBR-110", 1250000, 240,
"Gold");

        // Tampilkan judul data pelanggan
        System.out.println("=== DATA CUSTOMER BELANJAKU ===");

        // Tampilkan info untuk Customer biasa
        System.out.println("Status: Customer Biasa");
        customer.tampilkanInfo();

        System.out.println();

        // Tampilkan info untuk pelanggan Member
        System.out.println("Status: Member");
        member.tampilkanInfo();
    }
}
```

Alur kerja program:

1. Buat objek Customer dan Member (Member extends Customer).
2. Member memanggil super(...) di constructor untuk inisialisasi parent.
3. Override method tampilkanInfo() di Member dan panggil super.tampilkanInfo() lalu tambahkan atribut member (poin, level).
4. Panggil method yang mengkalkulasi diskon atau benefit member.

   Fungsi yang digunakan:

- Customer(String nama, String idCustomer, double totalBelanja)
  Konstruktor dasar.
- void tampilkanInfo()
  Menampilkan info customer.
- Member(String nama, String id, double total, int poin, String level)
  Konstruktor child memanggil super(...).
- @Override void tampilkanInfo() (Member)
  Memperluas output parent dengan info poin dan level.
- double hitungDiskon()
  Mengembalikan persen diskon berdasarkan level member.

   Output :

```
=== DATA CUSTOMER BELANJAKU ===
Status: Customer Biasa
Nama: Budi Santoso | ID: CST-001 | Total Belanja: Rp 500000

Status: Member
Nama: Siti Aminah | ID: MBR-110 | Total Belanja: Rp 1250000
Poin Reward: 240 | Level: Gold
PS E:\Tugas Kuliah\Semester Tiga\Praktikum PBO\Responsi 2 PBO>
```

5. Soal Pert 5

   analisisMesin.java

```
class analisisMesin {
    public static void main(String[] args) {

        // Buat array untuk menyimpan berbagai jenis mesin
        defaultMesin[] mesinArray = new defaultMesin[5];

        // Isi array dengan objek mesinMotor
        mesinArray[0] = new mesinMotor("Honda Supra X", 125, "Bebek");
```

```
mesinArray[1] = new mesinMotor("Yamaha R25", 250, "Sport");

// Isi array dengan objek mesinTraktor
mesinArray[2] = new mesinTraktor("Kubota MX5200", 520, 4.5);

// Isi array dengan objek mesinTraktorListrik
mesinArray[3] = new mesinTraktorListrik("EcoTrac Z900", 300, 4.2, 70);
mesinArray[4] = new mesinTraktorListrik("Volta FarmX", 300, 3.5, 80);

System.out.println("=== DATA MESIN MEGATECH ===");

// Loop untuk menampilkan info masing-masing mesin
for (int i = 0; i < mesinArray.length; i++) {
   mesinArray[i].tampilInfo();
   System.out.println("Kategori: " + mesinArray[i].kategoriMesin());
   System.out.printf("Performa: %.1f\n", mesinArray[i].nilaiPerforma());
   System.out.println();
}

System.out.println("=== SUARA MESIN ===");

// Loop untuk menghasilkan suara tiap mesin (instanceof)
for (int i = 0; i < mesinArray.length; i++) {
   if (mesinArray[i] instanceof mesinMotor) {
      ((mesinMotor) mesinArray[i]).suaraMesin();
   } else if (mesinArray[i] instanceof mesinTraktorListrik) {
      ((mesinTraktorListrik) mesinArray[i]).suaraMesin();
   } else if (mesinArray[i] instanceof mesinTraktor) {
      ((mesinTraktor) mesinArray[i]).suaraMesin();
   }
}

System.out.println("\n=== MESIN PERFORMA TERTINGGI ===");

// Logika menemukan mesin dengan performa tertinggi
int indexMax = 0;
double performaMax = mesinArray[0].nilaiPerforma();

for (int i = 1; i < mesinArray.length; i++) {
   if (mesinArray[i].nilaiPerforma() > performaMax) {
      performaMax = mesinArray[i].nilaiPerforma();
      indexMax = i;
   }
}
```

```
        System.out.println(mesinArray[indexMax].namaMesin + " → " +
performaMax);

        System.out.println("\n=== TOP 3 MESIN TERBAIK ===");

        // Logika sorting 3 performa tertinggi
        // Bubble sort untuk mengurutkan
        for (int i = 0; i < mesinArray.length - 1; i++) {
            for (int j = 0; j < mesinArray.length - i - 1; j++) {
                if (mesinArray[j].nilaiPerforma() < mesinArray[j + 1].nilaiPerforma()) {
                    // Swap
                    defaultMesin temp = mesinArray[j];
                    mesinArray[j] = mesinArray[j + 1];
                    mesinArray[j + 1] = temp;
                }
            }
        }

        // Tampilkan 3 terbaik
        for (int i = 0; i < 3; i++) {
            System.out.println((i + 1) + ". " + mesinArray[i].namaMesin + " → " +
mesinArray[i].nilaiPerforma());
        }
    }
}

defaultMesin.java
class defaultMesin {
    // Variabel umum mesin (namaMesin, tenagaHP)
    protected String namaMesin;
    protected int tenagaHP;

    // Constructor
    defaultMesin(String nama, int hp) {
        this.namaMesin = nama;
        this.tenagaHP = hp;
    }

    void tampilInfo() {
        // Tampilkan info dasar mesin
        System.out.println("Mesin " + this.namaMesin + " | Tenaga: " + this.tenagaHP +
" HP");
    }

    double nilaiPerforma() {
        // Hitung performa dasar
```

```
        return this.tenagaHP * 1.0;
    }

    String kategoriMesin() {
        // Kategori default
        return "Mesin Umum";
    }
}

mesinMotor.java
class mesinMotor extends defaultMesin {
    // Variabel khusus mesin motor (tipeMotor)
    private String tipeMotor;

    // Constructor
    mesinMotor(String nama, int hp, String tipe) {
        super(nama, hp);
        this.tipeMotor = tipe;
    }

    @Override
    void tampilInfo() {
        // Override info mesin motor
        System.out.println("Mesin Motor " + this.namaMesin + " | Tipe: " +
this.tipeMotor + " | Tenaga: " + this.tenagaHP + " HP");
    }

    @Override
    double nilaiPerforma() {
        // Override performa mesin motor
        return this.tenagaHP * 1.2;
    }

    @Override
    String kategoriMesin() {
        // Override kategori
        return "Mesin Motor";
    }

    void suaraMesin() {
        // Suara mesin motor
        System.out.println(this.namaMesin + " → Brummm! Mesin motor menyala!");
    }
}
```

mesinTraktor.java

```
class mesinTraktor extends defaultMesin {
    // Variabel khusus mesin traktor (kapasitasTarik)
    protected double kapasitasTarik;

    // Constructor
    mesinTraktor(String nama, int hp, double tarik) {
        super(nama, hp);
        this.kapasitasTarik = tarik;
    }

    @Override
    void tampilInfo() {
        // Override info mesin traktor
        System.out.println("Mesin Traktor " + this.namaMesin + " | Tarik: " +
this.kapasitasTarik + " ton | Tenaga: " + this.tenagaHP + " HP");
    }

    @Override
    double nilaiPerforma() {
        // Override performa traktor
        return (this.tenagaHP * 1.1) + (this.kapasitasTarik * 5);
    }

    @Override
    String kategoriMesin() {
        // Override kategori traktor
        return "Mesin Traktor";
    }

    void suaraMesin() {
        // Suara traktor
        System.out.println(this.namaMesin + " → GGGRRRR! Hidup mesinnn!");
    }
}
```

mesinTraktorListrik.java

```
class mesinTraktorListrik extends mesinTraktor {
    // Variabel khusus mesin traktor listrik (kapasitasBaterai)
    private double kapasitasBaterai;

    // Constructor
    mesinTraktorListrik(String nama, int hp, double tarik, double baterai) {
        super(nama, hp, tarik);
        this.kapasitasBaterai = baterai;
```

```
    }

    @Override
    void tampilInfo() {
        // Override info traktor listrik
        System.out.println("Mesin Traktor Listrik " + this.namaMesin + " | Tarik: " +
this.kapasitasTarik + " ton | Baterai: " + this.kapasitasBaterai + " kWh | Tenaga: " +
this.tenagaHP + " HP");
    }

    @Override
    double nilaiPerforma() {
        // Override performa traktor listrik
        return (this.tenagaHP * 0.9) + (this.kapasitasTarik * 10);
    }

    @Override
    String kategoriMesin() {
        // Override kategori listrik
        return "Mesin Traktor Listrik";
    }

    @Override
    void suaraMesin() {
        // Suara traktor listrik
        System.out.println(this.namaMesin + " → Bzzzzz! Mesin traktor listrik aktif!");
    }
}
```

Alur kerja program:

1. Buat interface/kelas parent (mis. DefaultMesin) dengan method start() dan status().
2. Implementasikan subclass berbeda (MesinMotor, MesinTraktor) yang override start()/status().
3. Simpan objek-polymorphic ke array DefaultMesin[], isi dengan MesinMotor dan MesinTraktor.
4. Iterasi array dan panggil start() untuk menunjukkan dynamic dispatch.

Fungsi yang digunakan:

- class DefaultMesin { void start(); void status(); }
  Definisi kontrak/parent.
- class MesinMotor extends DefaultMesin { @Override void start() ... }
  Implementasi untuk motor.
- class MesinTraktor extends DefaultMesin { @Override void start() ... }

Implementasi untuk traktor.

- void tesPolimorfisme()
  Membuat array DefaultMesin[] dan memanggil metode pada tiap elemen.

Output :

```
=== DATA MESIN MEGATECH ===
Mesin Motor Honda Supra X | Tipe: Bebek | Tenaga: 125 HP
Kategori: Mesin Motor
Performa: 150.0

Mesin Motor Yamaha R25 | Tipe: Sport | Tenaga: 250 HP
Kategori: Mesin Motor
Performa: 300.0

Mesin Traktor Kubota MX5200 | Tarik: 4.5 ton | Tenaga: 520 HP
Kategori: Mesin Traktor
Performa: 594.5

Mesin Traktor Listrik EcoTrac Z900 | Tarik: 4.2 ton | Baterai: 70.0 kWh | Tenaga: 300 HP
Kategori: Mesin Traktor Listrik
Performa: 312.0

Mesin Traktor Listrik Volta FarmX | Tarik: 3.5 ton | Baterai: 80.0 kWh | Tenaga: 300 HP
Kategori: Mesin Traktor Listrik
Performa: 305.0

=== SUARA MESIN ===
Honda Supra X ? Brummm! Mesin motor menyala!
Yamaha R25 ? Brummm! Mesin motor menyala!
Kubota MX5200 ? GGGRRRR! Hidup mesinnn!
EcoTrac Z900 ? Bzzzzz! Mesin traktor listrik aktif!
Volta FarmX ? Bzzzzz! Mesin traktor listrik aktif!

=== MESIN PERFORMA TERTINGGI ===
Kubota MX5200 ? 594.5

=== TOP 3 MESIN TERBAIK ===
1. Kubota MX5200 ? 594.5
2. EcoTrac Z900 ? 312.0
3. Volta FarmX ? 305.0
PS E:\Tugas Kuliah\Semester Tiga\Praktikum PBO\Responsi 2 PBO>
```

6. Soal Pert 6

KapalEksplorasi.java

```
public class KapalEksplorasi extends KendaraanGalaksi {

    private int modulScan;

    public KapalEksplorasi(String namaKendaraan, int kapasitasPenumpang, int
modulScan) {
```

```
        super(namaKendaraan, kapasitasPenumpang);
        this.modulScan = modulScan;
    }


    @Override
    public void aktifkanMesin() {
        if (getLevelEnergi() < 15) {
            System.out.println("Energi tidak mencukupi untuk memulai ekspedisi!");
        } else {
            System.out.println("Kapal eksplorasi siap berangkat!");
        }
    }

    @Override
    public void jelajah(int jarak) {
        int energiDibutuhkan = jarak * 2; // Konsumsi energi: 2% per km
        if (getLevelEnergi() < energiDibutuhkan) {
            System.out.println("Energi tidak cukup untuk menjelajah!");
        } else {
            kurangiEnergi(energiDibutuhkan);
            System.out.println("Kapal eksplorasi menjelajah sejauh " + jarak + " km.");
        }
    }

    @Override
    public void isiEnergi(int jumlah) {
        tambahEnergi(jumlah);
        System.out.println("Energi kapal eksplorasi ditambah. Energi sekarang: " +
getLevelEnergi() + "%");
    }


    public void scanPlanet(String namaPlanet) {
        System.out.println("Melakukan scan pada planet " + namaPlanet + " dengan
modul level " + modulScan + ".");
    }
}
```

KendaraanGalaksi.java

```
public abstract class KendaraanGalaksi {

    private String namaKendaraan;
    private int levelEnergi;        // 0–100
    private int kapasitasPenumpang;
```

```
    public KendaraanGalaksi(String namaKendaraan, int kapasitasPenumpang) {
        this.namaKendaraan = namaKendaraan;
        this.kapasitasPenumpang = kapasitasPenumpang;
        this.levelEnergi = 100; // default saat dibuat pasti 100%
    }


    public String getNamaKendaraan() {
        return namaKendaraan;
    }

    public int getLevelEnergi() {
        return levelEnergi;
    }

    public int getKapasitasPenumpang() {
        return kapasitasPenumpang;
    }


    protected void kurangiEnergi(int jumlah) {
        this.levelEnergi -= jumlah;
        if (this.levelEnergi < 0) this.levelEnergi = 0; // Jaga agar tidak minus
    }

    protected void tambahEnergi(int jumlah) {
        this.levelEnergi += jumlah;
        if (this.levelEnergi > 100) this.levelEnergi = 100; // Jaga agar tidak lebih dari
100
    }


    public final void tampilStatus() {
        System.out.println(namaKendaraan + " | Energi: " + levelEnergi + "% |
Kapasitas: " + kapasitasPenumpang + " awak");
    }


    public abstract void aktifkanMesin();
    public abstract void jelajah(int jarak);
    public abstract void isiEnergi(int jumlah);
}
```

PesawatTempur.java

```
public class PesawatTempur extends KendaraanGalaksi {


    private int jumlahRudal;


    public PesawatTempur(String namaKendaraan, int kapasitasPenumpang, int
jumlahRudal) {
        super(namaKendaraan, kapasitasPenumpang);
        this.jumlahRudal = jumlahRudal;
    }


    @Override
    public void aktifkanMesin() {
        if (getLevelEnergi() < 20) {
            System.out.println("Energi terlalu rendah! Mesin tidak dapat diaktifkan.");
        } else {
            System.out.println("Mesin pesawat tempur diaktifkan.");
        }
    }

    @Override
    public void jelajah(int jarak) {
        int energiDibutuhkan = jarak * 3; // Konsumsi energi: 3% per km
        if (getLevelEnergi() < energiDibutuhkan) {
            System.out.println("Energi tidak mencukupi untuk menjelajah sejauh " + jarak
+ " km.");
        } else {
            kurangiEnergi(energiDibutuhkan);
            System.out.println("Pesawat tempur menjelajah sejauh " + jarak + " km.");
        }
    }

    @Override
    public void isiEnergi(int jumlah) {
        tambahEnergi(jumlah);
        System.out.println("Energi pesawat tempur ditambah. Energi sekarang: " +
getLevelEnergi() + "%");
    }


    public void tembakRudal(int jumlah) {
        if (jumlahRudal >= jumlah) {
            jumlahRudal -= jumlah;
            System.out.println("Menembakkan " + jumlah + " rudal!");
```

```
        } else {
            System.out.println("Rudal tidak cukup!");
        }
    }
}
```

UjiGalaksi.java

```
public class UjiGalaksi {
    public static void main(String[] args) {


        PesawatTempur pesawat = new PesawatTempur("Astra-Fury", 2, 8);
        KapalEksplorasi kapal = new KapalEksplorasi("Voyager X", 10, 4);



        System.out.println("--- PESAWAT TEMPUR ---");
        pesawat.aktifkanMesin();
        pesawat.jelajah(10);
        pesawat.jelajah(30); // Cek energi
        pesawat.tembakRudal(3);
        pesawat.tampilStatus();

        System.out.println("\n--- KAPAL EKSPLORASI ---");
        kapal.aktifkanMesin();
        kapal.jelajah(15);
        kapal.scanPlanet("Kepler-442b");
        kapal.tampilStatus();
    }
}
```

Alur kerja program:

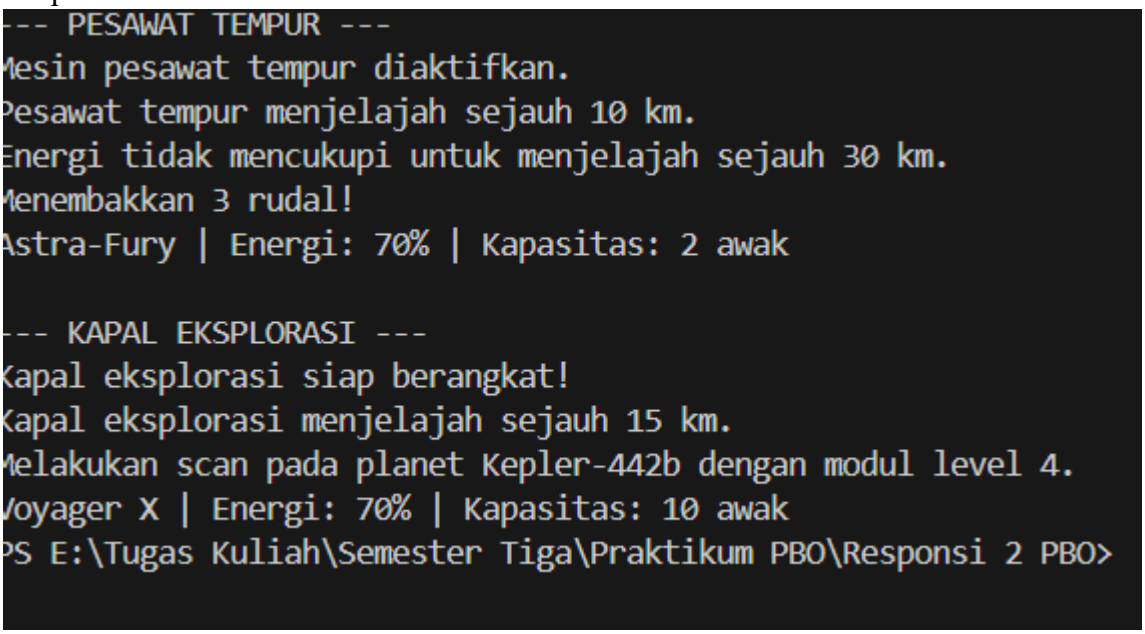
1. Definisikan abstract class KendaraanGalaksi dengan abstract methods aktifkanMesin(), jelajah(int), isiEnergi(int).
2. Buat subclass PesawatTempur dan KapalEksplorasi yang mengimplementasi abstract method.
3. Buat objek tiap subclass, panggil aktifkanMesin(), jelajah(), isiEnergi(), tampilStatus().
4.

Fungsi yang digunakan:

- KendaraanGalaksi(String nama, int kapasitas)
- Konstruktor parent, menyimpan levelEnergi default 100.
- abstract void aktifkanMesin()

- Implementasi berbeda di tiap subclass.
  abstract void jelajah(int jarak)
- Konsumsi energi berbeda per subclass (mis. pesawat 3%/km, kapal 2%/km).
  abstract void isiEnergi(int jumlah)
- Menambah energi hingga maksimum 100.
  protected void kurangiEnergi(int jumlah), tambahEnergi(int jumlah)
- Helper method di parent.
  Final void tampilStatus()
  Menampilkan nama | Energi | Kapasitas.

  Output :



7. Soal Pert 7

   EwalletPayment.java

   public class EWalletPayment implements PaymentMethod {

```
    private String providerName;
    private double balance;
    private double transactionAmount;

    public EWalletPayment(String providerName, double initialBalance, double
transactionAmount) {
        this.providerName = providerName;
        this.balance = initialBalance;
        this.transactionAmount = transactionAmount;
    }
```

```
    @Override
    public double getBalance() {
        return this.balance;
    }

    @Override
    public double getTransactionFee() {
        return 2000.0;
    }

    @Override
    public void processPayment() {
        double totalCost = this.transactionAmount + getTransactionFee();

        System.out.println("Memproses pembayaran sebesar Rp " +
this.transactionAmount + "...");

        if (this.balance >= totalCost) {
            this.balance -= totalCost;
            System.out.println("✓ Pembayaran berhasil!");
            System.out.println("  Nominal: Rp " + this.transactionAmount);
            System.out.println("  Biaya transaksi: Rp " + getTransactionFee());
            System.out.println("  Total biaya: Rp " + totalCost);
        } else {
            System.out.println("✗ Pembayaran gagal! Saldo tidak cukup.");
            System.out.println("  Saldo Anda: Rp " + this.balance);
            System.out.println("  Total yang dibutuhkan: Rp " + totalCost);
        }
    }

    @Override
    public String getPaymentDetails() {
        return "Detail Transaksi: Pembayaran dilakukan melalui " + this.providerName +
" | Nominal: Rp " + this.transactionAmount + " | Biaya: Rp " + getTransactionFee();
    }
}
```

PaymentMethod.java

```
public interface PaymentMethod {

    void processPayment();
```

```
    String getPaymentDetails();

    double getTransactionFee();

    double getBalance();


}
```

PaymentTest.java

```
public class PaymentTest {
    public static void main(String[] args) {
        System.out.println("=== PROGRAM SISTEM PEMBAYARAN (E-WALLET)
===\n");


        EWalletPayment myWallet = new EWalletPayment("OVO", 150000, 50000);


        System.out.println("--- SEBELUM TRANSAKSI ---");
        System.out.println("Saldo awal: Rp " + myWallet.getBalance());


        System.out.println("\n--- PROSES PEMBAYARAN ---");
        myWallet.processPayment();


        System.out.println("\n--- SETELAH TRANSAKSI ---");
        System.out.println("Sisa saldo: Rp " + myWallet.getBalance());
        System.out.println(myWallet.getPaymentDetails());
    }
}
```

Alur kerja program:

1. Definisikan interface PaymentMethod (processPayment, getPaymentDetails, getTransactionFee, getBalance).
2. Implementasikan EWalletPayment yang menyimpan providerName, balance, transactionAmount.
3. Buat objek EWalletPayment, tampilkan saldo awal, panggil processPayment(), tampilkan saldo akhir dan detail transaksi.

Fungsi yang digunakan:

- interface PaymentMethod { void processPayment(); String getPaymentDetails(); double getTransactionFee(); double getBalance(); }
- EWalletPayment(String providerName, double initialBalance, double transactionAmount)
- Konstruktor inisialisasi.
- @Override void processPayment()
- Cek kecukupan saldo, kurangi saldo jika cukup, tampilkan hasil.
- @Override double getBalance()
- Mengembalikan saldo saat ini.
- @Override double getTransactionFee()
- Mengembalikan biaya transaksi (mis. 2000.0).
- @Override String getPaymentDetails()
- Mengembalikan deskripsi transaksi.

Output :

```
spot\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\MSI_\AppData\Roamin
torage\782561e0349342cb08c44f6fa006ea90\redhat.java\jdt_ws\Responsi 2 PBO_99167d3b\bin' 'Payment
=== PROGRAM SISTEM PEMBAYARAN (E-WALLET) ===

--- SEBELUM TRANSAKSI ---
Saldo awal: Rp 150000.0

--- PROSES PEMBAYARAN ---
Memproses pembayaran sebesar Rp 50000.0...
? Pembayaran berhasil!
  Nominal: Rp 50000.0
  Biaya transaksi: Rp 2000.0
  Total biaya: Rp 52000.0

--- SETELAH TRANSAKSI ---
Sisa saldo: Rp 98000.0
Detail Transaksi: Pembayaran dilakukan melalui OVO | Nominal: Rp 50000.0 | Biaya: Rp 2000.0
PS E:\Tugas Kuliah\Semester Tiga\Praktikum PBO\Responsi 2 PBO>
```

8. Soal Pert 8

   AksesSistem.java
   public interface AksesSistem {

   void login(String pin);

   void logout();

```
    default String getRoleAkses() {
        return "Staff Biasa";
    }
}
```

KaryawanKontrak.java

```
public interface KaryawanKontrak {



    double hitungGaji(int jamKerja);

    void perpanjangKontrak(int durasiBulan);



    default String getStatusCuti() {
        return "Tersedia 12 hari";
    }
}
```

ProgrammingMagang.java

```
public class ProgrammerMagang implements KaryawanKontrak, AksesSistem {



    private String nama;
    private double gajiPerJam;
    private String pinRahasia;
    private boolean sedangLogin;



    public ProgrammerMagang(String nama, double gajiPerJam, String pinRahasia) {
        this.nama = nama;
        this.gajiPerJam = gajiPerJam;
        this.pinRahasia = pinRahasia;
        this.sedangLogin = false;
    }
```

```
    @Override
    public double hitungGaji(int jamKerja) {
        double totalGaji = jamKerja * this.gajiPerJam;
        System.out.println("Nama: " + this.nama);
        System.out.println("Jam Kerja: " + jamKerja + " jam");
        System.out.println("Gaji per Jam: Rp " + this.gajiPerJam);
        System.out.println("Total Gaji: Rp " + totalGaji);
        return totalGaji;
    }

    @Override
    public void perpanjangKontrak(int durasiBulan) {
        System.out.println("Kontrak " + this.nama + " diperpanjang selama " +
durasiBulan + " bulan.");
    }

    @Override
    public String getStatusCuti() {
        return "Tersedia 5 hari";
    }



    @Override
    public void login(String pin) {
        if (pin.equals(this.pinRahasia)) {
            this.sedangLogin = true;
            System.out.println("Login berhasil! Selamat datang, " + this.nama);
        } else {
            System.out.println("Login gagal! PIN salah.");
        }
    }

    @Override
    public void logout() {
        this.sedangLogin = false;
        System.out.println(this.nama + " telah logout.");
    }

    @Override
    public String getRoleAkses() {
        return "Magang IT";
    }
}
```

Alur kerja program:

1. Definisikan interface KaryawanKontrak (hitungGaji, perpanjangKontrak, default getStatusCuti).
2. Definisikan interface AksesSistem (login, logout, default getRoleAkses).
3. Buat class ProgrammerMagang implements kedua interface, implement semua method.
4. Uji: hitungGaji(160), cek getStatusCuti(), tes login dengan PIN salah dan benar, perpanjang kontrak, logout.

Fungsi yang digunakan:

- KaryawanKontrak: double hitungGaji(int jamKerja); void perpanjangKontrak(int durasiBulan); default String getStatusCuti()
- AksesSistem: void login(String pin); void logout(); default String getRoleAkses()
- ProgrammerMagang(String nama, double gajiPerJam, String pinRahasia)
- Konstruktor inisialisasi.
- @Override double hitungGaji(int jamKerja)
- Menghitung dan menampilkan rincian gaji.
- @Override void login(String pin), void logout()
- Mengelola status login dan menampilkan pesan.
- @Override void perpanjangKontrak(int durasiBulan)
- Menampilkan konfirmasi perpanjangan kontrak.

Output :

```
PS E:\Tugas Kuliah\Semester Tiga\Praktikum PBO\Responsi 2 PBO> & 'C:\Program Files\Eclipse Adoptium\jdk-21.0.6.7-hotspot\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\MSI_\AppData\Roaming\Code\User\workspaceStorage\782561e0349342cb08c44f6fa006ea90\redhat.java\jdt_ws\Responsi 2 PBO_99167d3b\bin'
=== PENGUJIAN SISTEM KARYAWAN MAGANG ===

--- Perhitungan Gaji ---
Nama: Andi
Jam Kerja: 160 jam
Gaji per Jam: Rp 50000.0
Total Gaji: Rp 8000000.0

--- Status Cuti ---
Tersedia 5 hari

--- Tes Login Gagal ---
Login gagal! PIN salah.

--- Tes Login Berhasil ---
Login berhasil! Selamat datang, Andi

--- Role Akses ---
Role: Magang IT

--- Perpanjangan Kontrak ---
Kontrak Andi diperpanjang selama 6 bulan.

--- Logout ---
Andi telah logout.
PS E:\Tugas Kuliah\Semester Tiga\Praktikum PBO\Responsi 2 PBO>
```