**FAKULTAS TEKNOLOGI INFORMASI DAN ELEKTRO**

**UNIVERSITAS TEKNOLOGI YOGYAKARTA**

**UJIAN TENGAH SEMESTER GANJIL TAHUN AKADEMIK 2020/2021**

PROGRAM STUDI : S1 Informatika (semua kelas)

Mata Ujian : Pemrograman Berorientasi Objek

Hari/Tanggal : Sesuai jadwal kuliah masing-masing

Dosen : Tim dosen

Sifat : ***Take home*, buka buku dan referensi di internet**

**PERATURAN**

* Anda harus menjawab **SENDIRI** semua pertanyaan di bawah ini secara jelas dan lengkap. **DILARANG KERAS** untuk berkomunikasi, bekerjasama, dan meminta bantuan siapapun. Mahasiswa yang terbukti plagiat, nilainya **langsung E**.
* Jumlah soal ada dua, masing-masing memiliki bobot 50 poin.
* Gunakan lembar jawaban yang tersedia pada halaman terakhir. Jawaban diketik di komputer, lalu dikonversi ke dalam format PDF.
* Lembar pernyataan **harus ditandatangani** (*scan*) oleh setiap mahasiswa yang bersangkutan. Jawaban yang tidak ditandatangani tidak akan dikoreksi (nilainya = 0).
* Lembar Jawaban yang **sudah** **ditandatangani** dikumpulkan melalui *link* dan jadwal yang ditentukan oleh dosen pengampu masing-masing kelas.**SOAL**

1. **BOBOT: 50 poin**

**Kompetensi M1: Mampu memahami urgensi dan konsep dasar PBO.**

**Kompetensi M2: Mampu memahami konsep class, objek, konstruktor, dan enkapsulasi.**

Perhatikan class diagram di bawah ini:

Diagram

Description automatically generated

1. Implementasikanlah *class* Produk, Transaksi, dan Pelanggan pada *class diagram* di atas menggunakan Bahasa Pemrograman Java! Pastikan setiap class memiliki *constructor* yang sekaligus berfungsi sebagai tempat inisialisasi (pengisian) nilai dari atribut class tersebut. Terapkan juga konsep *encapsulation* dengan memprivatisasi semua atribut serta menyediakan *method* **set** dan **get** pada tiap class agar class lain dapat mengakses atribut dari class tersebut. Sebagai catatan, atribut totalBayar pada class Transaksi dihitung menggunakan *method* hitungTotalBayar() dengan cara mengalikan jumlahProduk dengan harga produk tersebut.
2. Implementasikan juga *class* utama (*main class*) yakni *class* ECommerce. Kemudian di dalam *class* utama ini, lakukanlah instansiasi objek dari *class* Produk dan *class* Pelanggan serta sekaligus menginisialisasi nilai-nilai atributnya, lalu beri nama objek dari masing-masing class tersebut dengan nama **Produk01** dan **Pelanggan01**. Setelah itu, lakukan instansiasi objek dari class Transaksi dan beri nama objeknya **Transaksi01**. Pada saat Anda menginstansiasi objek Transaksi01, lakukan inisialisasi terhadap atribut idTransaksi, idPelanggan, idProduk, taanggalTransaksi, dan jumlahProduk. Isikan nilai pada idProduk dengan idProduk dari objek Produk01 dan nilai pada idPelanggan dengan idPelanggan dari objek Pelanggan01. Gunakan *method* get untuk membaca nilai atribut idProduk dan idPelanggan. Atribut totalBayar dihitung menggunakan method hitungTotalBayar() yang dipanggil di dalam constructor class Transaksi.
3. **BOBOT: 50 poin**

**Kompetensi M3: Mampu memahami konsep pewarisan dan polimorfisme.**

Perhatikan *class diagram* di bawah ini!

Diagram

Description automatically generated

1. Implementasikanlah konsep pewarisan pada *class diagram* di atas menggunakan Bahasa Pemrograman Java! Pastikan setiap *class* memiliki constructor yang sekaligus berfungsi untuk menginisalisasi nilai atribut pada *class* tersebut. Method lihatProduk() pada *class* Produk berfungsi untuk menampilkan data produk yang meliputi: merk, harga, stok, dan diskonMember. Nilai atribut diskonMember diperoleh dari *method* hitungDiskon() dengan ketentuan jika harga produk >= 50.000, maka diskonMember sebesar 2.5%, sedangkan jika harga produk >= 100.000, maka diskonMember sebesar 5%. *Method* isKedaluarsa pada *class* Makanan akan mengembalikan nilai **true** jika tglKedaluarsa > tanggal saat aplikasi dijalankan. Sebaliknya, akan mengembalikan nilai **false** jika tglKedaluarsa < tanggal saat aplikasi dijalankan. Untuk dapat membaca data tanggal pada saat aplikasi dijalankan, Anda membutuhkan library tambahan.
2. Pada *class* Makanan dan Pakaian, lakukan **overriding** terhadap *method* lihatProduk() dari *class* Produk untuk menampilkan informasi lengkap mengenai data produk. Pada *class* Makanan, selain menampilkan merk, harga, stok, dan diskonMember, *method* lihatProduk() juga menampilkan tglKedaluarsa dan netto. Sedangkan pada *class* Pakaian, *method* lihatProduk() juga menampilkan data ukuran dan warna.

**Lembar Jawaban UTS**

**Pemrograman Berorientasi Objek (20330618)**

**Nama : Ardhika Restu Yoviyanto**

**NIM : 5190411312**

**Kelas : G**

**Pernyataan :** *“Semua jawaban* ***SAYA******KERJAKAN SENDIRI*** *tanpa bekerjasama maupun meminta bantuan siapapun. Saya bersedia diberi* ***nilai E*** *jika terbukti bekerjasama ataupun meminta bantuan dari orang lain”.*

|  |
| --- |
| **(Ardhika Restu Yoviyanto)** |

**Tanda Tangan :**

1) a. Produk.java

|  |
| --- |
| class Produk{  //Atribut  private String idProduk, merk;  private int stok;  private double harga;  //Constructor  public Produk(String \_idProduk, String \_merk, int \_stok, double \_harga){  this.idProduk = \_idProduk;  this.merk = \_merk;  this.stok = \_stok;  this.harga = \_harga;  }  //Start method set  public void set\_idProduk(String \_idProduk){  this.idProduk = \_idProduk;  }  public void set\_merk(String \_merk){  this.merk = \_merk;  }  public void set\_stok(int \_stok){  this.stok = \_stok;  }  public void set\_harga(double \_harga){  this.harga = \_harga;  }  //End method set  //Start method get  public String get\_idProduk(){  return this.idProduk;  }  public String get\_merk(){  return this.merk;  }  public int get\_stok(){  return this.stok;  }  public double get\_harga(){  return this.harga;  }  //End method get  //Method in class Produk  public void tambahStok(int TambahStok){  this.stok = this.stok + TambahStok;  }  public void kurangiStok(int KurangiStok){  this.stok = this.stok - KurangiStok;  }  } |

b. Transaksi.java

|  |
| --- |
| class Transaksi {  //Atribut  private String idTransaksi, idPelanggan, idProduk, tanggalTransaksi;  private double totalBayar;  private int jumlahProduk;  //Constructor  public Transaksi(String \_idTransaksi, String \_idPelanggan, String \_idProduk, String \_tanggalTransaksi, double \_totalBayar, int \_jumlahProduk){  this.idTransaksi = \_idTransaksi;  this.idPelanggan = \_idPelanggan;  this.idProduk = \_idProduk;  this.tanggalTransaksi = \_tanggalTransaksi;  this.totalBayar = \_totalBayar;  this.jumlahProduk = \_jumlahProduk;  //panggil method hitungTotalBayar()  this.hitungTotalBayar();  }  //Start Method Set  public void set\_idTransaksi(String \_idTransaksi){  this.idTransaksi = \_idTransaksi;  }  public void set\_idPelanggan(String \_idPelanggan){  this.idPelanggan = \_idPelanggan;  }  public void set\_idProduk(String \_idProduk){  this.idProduk = \_idProduk;  }  public void set\_tanggalTransaksi(String \_tanggalTransaksi){  this.tanggalTransaksi = \_tanggalTransaksi;  }  public void set\_totalBayar(double \_totalBayar){  this.totalBayar = \_totalBayar;  }  public void set\_jumlahProduk(int \_jumlahProduk){  this.jumlahProduk = \_jumlahProduk;  }  //End method set  //Start method get  public String get\_idTransaksi(){  return this.idTransaksi;  }  public String get\_idPelanggan(){  return this.idPelanggan;  }  public String get\_idProduk(){  return this.idProduk;  }  public String get\_tanggalTransaksi(){  return this.tanggalTransaksi;  }  public double get\_totalBayar(){  return this.totalBayar;  }  public int get\_jumlahProduk(){  return this.jumlahProduk;  }  //End method get  //Method in class Transaksi  public void hitungTotalBayar(){  //buat objek baru Produk02 dari class Produk untuk memanggil method get\_harga(), pada produk.  Produk Produk02 = new Produk("1190411312", "ASUS", 10, 3.450000);  this.totalBayar = this.jumlahProduk \* Produk02.get\_harga();  }  public void tampilTransaksi(){  System.out.println("id Transaksi : "+this.idTransaksi);  System.out.println("id Pelanggan : "+this.idPelanggan);  System.out.println("id Produk : "+this.idProduk);  System.out.println("Tanggal Transaksi : "+this.tanggalTransaksi);  System.out.println("Jumlah Produk : "+this.jumlahProduk);  System.out.println("Total Bayar : "+this.totalBayar);  }  } |

c. Pelanggan.java

|  |
| --- |
| class Pelanggan {  //Atribut  private String idPelanggan, noHp;  //Constructor  public Pelanggan(String \_idPelanggan, String \_noHp){  this.idPelanggan = \_idPelanggan;  this.noHp = \_noHp;  }  //Start Method Set  public void set\_idPelanggan(String \_idPelanggan){  this.idPelanggan = \_idPelanggan;  }  public void set\_noHp(String \_noHp){  this.noHp = \_noHp;  }  //End Method Set  //Start Method get  public String get\_idPelanggan(){  return this.idPelanggan;  }  public String get\_noHp(){  return this.noHp;  }  //End Method get  //Method in class Pelanggan  public void lihatProfil(){  System.out.println("id Pelanggan : "+this.idPelanggan);  System.out.println("No HP : "+this.noHp);  }  } |

d. Ecommerce.java

|  |
| --- |
| class ECommerce {  public static void main(String[] args) {  Produk Produk01 = new Produk("1190411312", "ASUS", 10, 3.450000);  Pelanggan Pelanggan01 = new Pelanggan("5190411312", "082313104589");  Transaksi Transaksi01 = new Transaksi("001", Pelanggan01.get\_idPelanggan(), Produk01.get\_idProduk(), "17-12-2020", 0, 2);  }  } |

2) a. Produk.java

|  |
| --- |
| class Produk{  //Atribut  protected String merk;  protected double harga, diskonMember;  protected int stok;  //Constructor  public Produk(String \_merk, double \_harga, double \_diskonMember, int \_stok){  this.merk = \_merk;  this.harga = \_harga;  this.diskonMember = \_diskonMember;  this.stok = \_stok;  }  //Method  public void lihatProduk(){  System.out.println("Merk : "+this.merk);  System.out.println("Harga : "+this.harga);  System.out.println("Stok : "+this.stok);  System.out.println("Diskon Member : "+this.diskonMember);  }  public void hitungDiskon(){  //Total Diskon dalam rasio harga 50000 - 99999  if(this.harga >= 50000 && this.harga < 100000){  this.diskonMember = (2.5/100) \* this.harga;  //Total Diskon dengan harga lebih dari sama dengan 100000  }else if(this.harga >= 100000){  this.diskonMember = (5.0/100) \* this.harga;  }  }} |

b. Makanan.java

|  |
| --- |
| //library tambahan  import java.text.SimpleDateFormat;  import java.util.Date;  import java.text.ParseException;  class Makanan extends Produk{  //Atribut  private String tglKadaluarsa;  private double netto;  //Constructor  public Makanan(String \_tglKadaluarsa, double \_netto){  super("Keripik Apel Malang", 60000, 0, 10);  this.tglKadaluarsa = \_tglKadaluarsa;  this.netto = \_netto;  }  //Method  public boolean isKedaluarsa() throws ParseException {  SimpleDateFormat dateFormat = new SimpleDateFormat("dd-MM-yyyy");  Date date = new Date();  String \_TanggalSekarang = dateFormat.format(date);  //Conversion String to Date  Date TanggalKadaluarsa = dateFormat.parse(this.tglKadaluarsa);  Date TanggalSekarang = dateFormat.parse(\_TanggalSekarang);  if(TanggalSekarang.before(TanggalKadaluarsa)){  return true;  }else{  return false;  }  }  //overiding  public void lihatProduk(){  System.out.println("Merk : "+super.merk);  System.out.println("Harga : "+super.harga);  System.out.println("Stok : "+super.stok);  System.out.println("Diskon Member : "+super.diskonMember);  System.out.println("Tanggal Kadaluarsa : "+this.tglKadaluarsa);  System.out.println("Netto : "+this.netto);  }  } |

c. Pakaian.java

|  |
| --- |
| class Pakaian extends Produk{  //Atribut  private String ukuran, warna;  //Constructor  public Pakaian(String \_ukuran, String \_warna){  super("Kemeja Batik", 130000, 0, 8);  this.ukuran = \_ukuran;  this.warna = \_warna;  }  //overriding  public void lihatProduk(){  System.out.println("Merk : "+super.merk);  System.out.println("Harga : "+super.harga);  System.out.println("Stok : "+super.stok);  System.out.println("Diskon Member : "+super.diskonMember);  System.out.println("Ukuran : "+this.ukuran);  System.out.println("Warna : "+this.warna);  }  } |