1. **DEFINISI MASALAH**
2. Perusahaan NV. Meneer memiliki koperasi karyawan yang memungkinkan karyawannya berbelanja di koperasi tersebut. Tentunya, karyawan tersebut bisa membayar belanjaanya tersebut di akhir bulan melalui pemotongan gaji. Ada 2 kelas yang terlibat disini, Invoice dan Employee. Kedua class tadi mengimplementasikan interface Payable yang mana ia hanya memiliki satu method yang harus diimplementasikan di kedua class, yaitu getPayableAmount(). Program harus bisa mengolah gaji karyawan di akhir bulan beserta invoice belanjaan karyawan yang nantinya gaji karyawan perbulannya dikurang total harga belanjaanya secara polimorfis. Tampilkan informasi dari karyawan tersebut beserta total gaji setelah dipotong hutang belanjaan di koperasi dan tampilkan pula detail belanjaanya secara polimorfis pula.

Attribut dari Invoice: String productName, Integer quantity, Integer pricePerItem Attribut dari Employee: Integer registrationNumber, String name, Integer salaryPerMonth, Invoice[] invoices

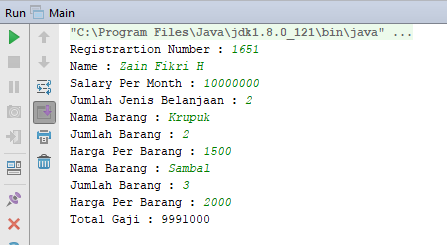
1. **SOURCE CODE**

|  |  |
| --- | --- |
|  | Payable.java |
| 1  2  3  4  5  6 | */\*\*  \* Created by Zain Fikri H on 13/05/2017.  \*/* public interface Payable {  public Integer getPayableAmount(); } |
|  | Invoice.java |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37 | */\*\*  \* Created by Zain Fikri H on 13/05/2017.  \*/* public class Invoice implements Payable {  public String productName;  public Integer quantity;  public Integer pricePerItem;   public String getProductName() {  return productName;  }   public Integer getQuantity() {  return quantity;  }   public Integer getPricePerItem() {  return pricePerItem;  }   public void setProductName(String productName) {  this.productName = productName;  }   public void setQuantity(Integer quantity) {  this.quantity = quantity;  }   public void setPricePerItem(Integer pricePerItem) {  this.pricePerItem = pricePerItem;  }   @Override  public Integer getPayableAmount() {  return getPricePerItem()\*getQuantity();  } } |
|  | Employee.java |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36 | import java.util.ArrayList;  */\*\*  \* Created by Zain Fikri H on 13/05/2017.  \*/* public class Employee implements Payable{  public Integer registrationNumber;  public String name;  public Integer salaryPerMonth;  public static Integer *payable*=0;  public ArrayList<Invoice> invoiceArrayLis= new ArrayList<>();  Invoice invoice = new Invoice();   public Employee(Integer registrationNumber, String name, Integer salaryPerMonth) {  this.registrationNumber = registrationNumber;  this.name = name;  this.salaryPerMonth = salaryPerMonth;  }   public void setInvoice(String nama, Integer quantity, Integer price){  invoice.setProductName(nama);  invoice.setPricePerItem(price);  invoice.setQuantity(quantity);  invoiceArrayLis.add(invoice);  *payable* += invoice.getPayableAmount();  }   @Override  public Integer getPayableAmount() {  System.*out*.print("Total Gaji : ");  return salaryPerMonth-*payable*;  } } |
|  | Main.java |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39 | import java.util.ArrayList; import java.util.Scanner;  */\*\*  \* Created by Zain Fikri H on 14/05/2017.  \*/* public class Main {  public static void main(String[] args) {  Scanner in = new Scanner(System.*in*);  ArrayList<Employee> employees = new ArrayList<>();  System.*out*.print("Registrartion Number : ");  Integer no = in.nextInt();  in.nextLine();  System.*out*.print("Name : ");  String name = in.nextLine();  System.*out*.print("Salary Per Month : ");  Integer salary = in.nextInt();  employees.add(new Employee(no,name,salary));  System.*out*.print("Jumlah Jenis Belanjaan : ");  Integer jenis = in.nextInt();  String nama[]= new String[jenis];  Integer quantity[]= new Integer[jenis];  Integer price[]= new Integer[jenis];  for(int i=0; i<jenis; i++){  System.*out*.print("Nama Barang : ");  nama[i]=in.next();  System.*out*.print("Jumlah Barang : ");  quantity[i]=in.nextInt();  System.*out*.print("Harga Per Barang : ");  price[i]=in.nextInt();  }  for(int i=0;i<nama.length;i++){  employees.get(0).setInvoice(nama[i],quantity[i],price[i]);  }  System.*out*.println(employees.get(0).getPayableAmount());  } } |

1. **PEMBAHASAN**

|  |  |
| --- | --- |
|  | MainPerpus.java |
|  |  |
|  | Kategori.java |
|  |  |
|  | Buku.java |
|  |  |

1. **SCREENSHOT PROGRAM**

****

1. **PRAKTIKUM**
2. **Constructor**

|  |  |
| --- | --- |
|  | MainPerpus.java |
|  |  |

1. Lakukan percobaan diatas dan benahi jika menemukan kesalahan serta jelaskan!

*Jawaban:*

1. Apakah class yang berbentuk Interface bisa diinstansiasi menjadi sebuah objek? Jelaskan alasannya!

*Jawaban:*

1. Apakah suatu class dapat mengimplementasi class interface yang jumlahnya lebih dari satu? Jelaskan alasannya!

*Jawaban:*

1. Ubah source code diatas menjadi proses meminta inputan dari user dan buat menjadi interaktif!

*Jawaban:*

1. Buat objek selain objek diatas dengan menggunakan method yang berbeda dengan yang diatas! (min.1 contoh)

*Jawaban:*

1. **KESIMPULAN**

* Konstruktor