**PRAKTIKKUM-2 PDPL**

***CREATIONAL PATTERN - PROTOTYPE***

**Mata Kuliah : Pola Desain Perangkat Lunak**

**Semester : 6 (Enam)**

**Dosen : Tifanny Nabarian, S.Kom., M.T.I.**

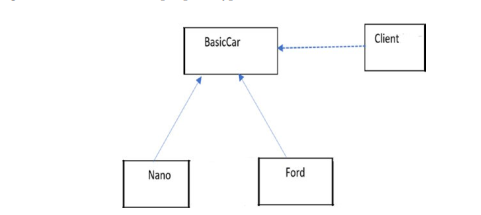
**Nama Mahasiswa : Muhammad Azhar Rasyad**

**NIM : 0110217029**

**Instruksi**

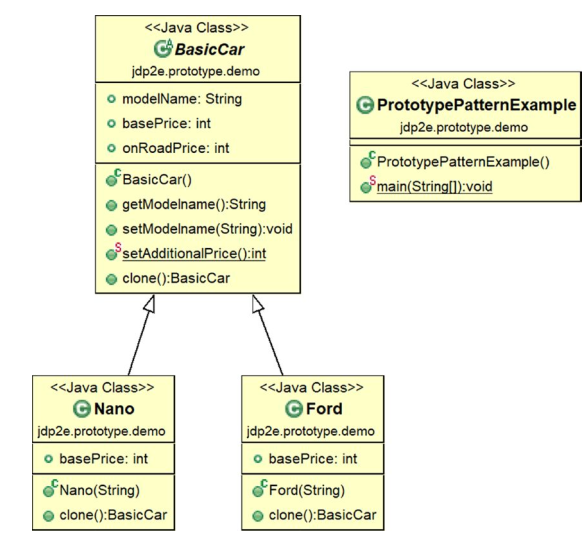
* Kerjakan tugas di bawah ini secara **individu.**
* Kerjakan secara **berurutan.**
* Modifikasi *source code*, dengan cara menambahkan **NIM** Anda pada setiap *caption* yang ditampilkan di *result* yang di*screenshot*.

1. **Perhatikan ilustrasi berikut ini:**



Gambar 1. Ilustrasi Relasi

1. **Pahamilah struktur kelas yang akan dibuat.**



Gambar 2. Class Diagram

1. **Buatlah kelas BasicCar dengan source sebagai berikut:**

**BasicCar.java**

|  |
| --- |
| **import** java.util.Random;  **public** **abstract** **class** BasicCar **implements** Cloneable {  **public** String modelName;  **public** **int** basePrice,onRoadPrice;  **public** String getModelname() {  **return** modelName;  }  **public** **void** setModelname(String modelname) {  **this**.modelName = modelname;  }  **public** **static** **int** setAdditionalPrice()  {  **int** price = 0;  Random r = **new** Random();  //We will get an integer value in the range 0 to 100000  **int** p = r.nextInt(100000);  price = p;  **return** price;  }  **public** BasicCar clone() **throws** CloneNotSupportedException  {  **return** (BasicCar)**super**.clone();  }  } |

1. **Lanjutkan dengan membuat kelas Nano dan Ford, dengan source sebagai berikut:**

**Nano.java**

|  |
| --- |
| **public** **class** Nano **extends** BasicCar{  //A base price for Nano  **public** **int** basePrice=100000;  **public** Nano(String m)  {  modelName = m;  }  @Override  **public** BasicCar clone() **throws** CloneNotSupportedException  {  **return** (Nano)**super**.clone();  //return this.clone();  }  } |

**Ford.java**

|  |
| --- |
| **public** **class** Ford **extends** BasicCar{  //A base price for Ford  **public** **int** basePrice=100000;  **public** Ford(String m)  {  modelName = m;  }  @Override  **public** BasicCar clone() **throws** CloneNotSupportedException  {  **return** (Ford)**super**.clone();  }  } |

1. **Lanjutkan dengan membuat kelas PrototypePatternExample, dengan source sbb.:**

**PrototypePatternExample.java**

|  |
| --- |
| **public** **class** PrototypePatternExample {  **public** **static** **void** main(String[] args) **throws**  CloneNotSupportedException  {  System.***out***.println("\*\*\*Prototype Pattern Demo\*\*\*\n");  BasicCar nano = **new** Nano("Green Nano") ;  nano.basePrice=100000;  BasicCar ford = **new** Ford("Ford Yellow");  ford.basePrice=500000;  BasicCar bc1;  //Nano  bc1 =nano.clone();  //Price will be more than 100000 for sure  bc1.onRoadPrice = nano.basePrice+BasicCar.*setAdditionalPrice*();  System.***out***.println("Car is: "+ bc1.modelName+" and it's price is Rp."+bc1.onRoadPrice);  //Ford  bc1 =ford.clone();  //Price will be more than 500000 for sure  bc1.onRoadPrice = ford.basePrice+BasicCar.*setAdditionalPrice*();  System.***out***.println("Car is: "+ bc1.modelName+" and it's price is Rp."+bc1.onRoadPrice);  }  } |

1. **Update source pada langkah 5, dengan menampilkan NIM Anda. Kemudian jalankan *main* program tersebut.**

|  |
| --- |
| *Screenshot Hasil Running Main Program* |

1. Berikan penjelasan terkait proses implementasi *prototype* pada source 3,4 dan 5.

|  |
| --- |
| **Hint:**   * perhatikan penggunaan **clone()** pada *source code* * apakah implementasi *prototype* pada *source code*, menggunakan shallow copy atau deep copy?Jelaskan   Pada kelas BasicCar.java berisi field-field untuk membuat sebuah objek BasicCar yaitu model mobilnya, harga tambahannya secara random, dan fungsi paling penting yaitu clone() untuk menduplikasi interface dengan menambahkan pada kelas BasicCar implemens Cloneable.  Pada kelas Nano.java berisi variable harga awal nano, model nano, dan fungsi untuk menduplikasi dari kelas BasicCar.java  Pada kelas Ford.java berisi variable harga awal ford, model ford, dan fungsi untuk menduplikasi dari kelas BasicCar.java  Pada kelas PrototypePatternExample.java berisi untuk membuat objek baru yaitu nano dan ford serta menduplikasi BasicCar ke kelas nano dan ford  Implementasi prototype pada source code tersebut menggunakan *deep copy* karena pada kelas PrototypePatternExample.java hanya menduplikasi 1 buah cloningan yaitu bc1 yang digunakan pada 2 objek, nano dan ford. Serta didalam setiap kelas BasicCar, Nano, Ford terdapat fungsi clone() agar kelas yang lain dapat diduplikasi dengan adanya fungsi berikut : |

1. Buat kelas baru terpisah dari package sebelumnya, dengan nama kelas **Box.java dan TestPrototypePattern.java,** dengan *source code* sebagai berikut:

**Box.java**

|  |
| --- |
| **public** **class** Box **implements** Cloneable{  **private** String color;    **public** Box(String col){  setColor(col);  }    **public** Box clone(){  Box b = **null**;  **try**{  b = (Box)**super**.clone();  }**catch**(Exception e){    }  **return** b;  }    **public** String getColor() {  **return** color;  }    **public** **void** setColor(String color){  **this**.color = color;  }  } |

**TestPrototypePattern.java**

|  |
| --- |
| **public** **class** TestPrototypePattern {  **public** **static** **void** main(String[] args){  Box box1 = **new** Box("GREEN");  Box box2 = **new** Box("RED");    System.***out***.println("Box 1 dan Box 2 yang telah di isi");  System.***out***.println("Box 1 : "+box1.getColor());  System.***out***.println("Box 2 : "+box2.getColor());  System.***out***.println();      System.***out***.println("Box 3 di clone dengan Box 1");  Box box3 = box1.clone();  System.***out***.println("Box 3 : "+box3.getColor());    System.***out***.println("Box 4 di clone dengan Box 2");  Box box4 = box2.clone();  System.***out***.println("Box 4 : "+box4.getColor());    }  } |

1. Modifikasi *source code* 8 dengan mencetak NIM Anda, kemudian *Running main program* pada *source code* langkah 8.

|  |
| --- |
| *Screenshot Hasil Running Main Program* |

1. Berikan penjelasan terkait proses implementasi *prototype* pada **source 8**.

|  |
| --- |
| **Hint:**   * perhatikan penggunaan **clone()** pada *source code* * apakah implementasi *prototype* pada *source code*, menggunakan *shallow* *copy* atau *deep copy*?Jelaskan!   Pada kelas Box.java berisi source code fungsi untuk mengatur warna dan menduplikasi box  Pada kelas TestPrototypePattern.java berisi source code untuk membuat objek-objek baru yang akan dicloning  Implementasi prototype ini menggunakan *shallow copy* karena setiap perubahan pada box yang akan diclone maka terjadi perubahan pada box yang dicloning    Contoh diatas merupakan salah satu contoh *shallow copy* karena setiap perubahan yang terjadi pada box1 maka box3 ikut berubah pula |

**\*\*\***