**PRAKTIKUM-4 PDPL**

***STRUCTURAL PATTERN - ADAPTER***

**Mata Kuliah : Pola Desain Perangkat Lunak**

**Semester : 6 (Enam)**

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

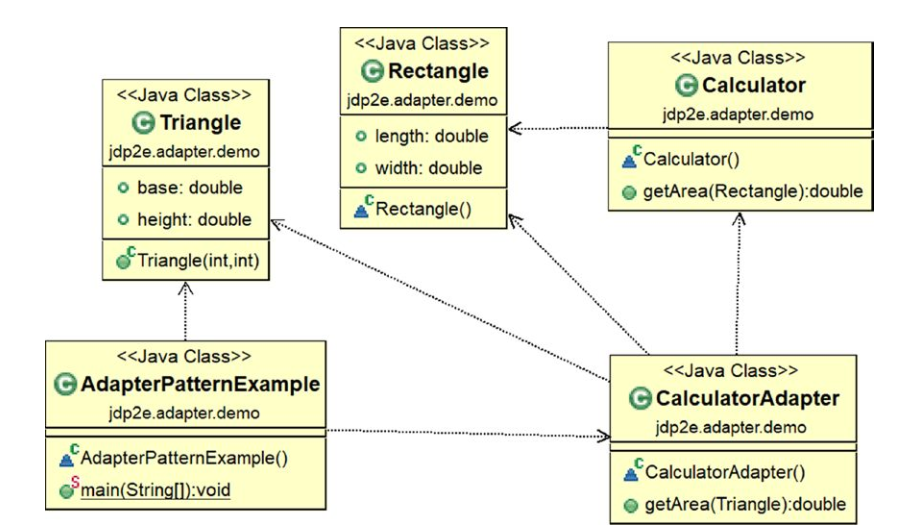
**Nama Mahasiswa : Muhammad Azhar Rasyad**

**NIM : 0110217029**

**Instruksi**

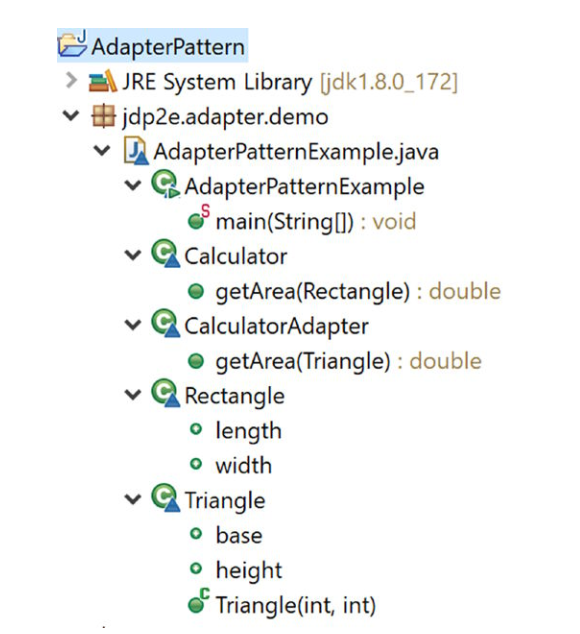
* Kerjakan tugas di bawah ini secara **individu.**
* Kerjakan secara **berurutan.**
* Modifikasi *source code*, dengan cara menambahkan **NIM** Anda pada setiap *caption* yang ditampilkan di *result* yang di*screenshot*.

1. **Perhatikan ilustrasi berikut ini:**



Gambar 1. Ilustrasi Relasi

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



Gambar 2. Package Explorer

1. **Buatlah Kelas Rectangle dan Calculator dengan source sebagai berikut:**

**Rectangle.java**

|  |
| --- |
| **package** jdp2e.adapter.demo;  **public** **class** Rectangle {  **public** **double** length;  **public** **double** width;  } |

**Calculator.java**

|  |
| --- |
| **package** jdp2e.adapter.demo;  **public** **class** Calculator {  **public** **double** getArea(Rectangle rect)  {  **return** rect.length \* rect.width;  }  } |

1. **Lanjutkan dengan membuat kelas Triangle dan CalculatorAdapter dengan masing-masing sourcode sbb.,**

**Triangle.java**

|  |
| --- |
| **package** jdp2e.adapter.demo;  **class** Triangle {  **public** **double** base;//base  **public** **double** height;//height    **public** Triangle(**int** b, **int** h)  {  **this**.base = b;  **this**.height = h;  }  } |

**CalculatorAdapter.java**

|  |
| --- |
| **package** jdp2e.adapter.demo;  **public** **class** CalculatorAdapter {  **public** **double** getArea(Triangle triangle)  {  Calculator c = **new** Calculator();  Rectangle rect = **new** Rectangle();    //Area of Triangle=0.5\*base\*height  rect.length = triangle.base;  rect.width = 0.5 \* triangle.height;  **return** c.getArea(rect);  }  } |

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

**AdapterPatternExample.java**

|  |
| --- |
| **package** jdp2e.adapter.demo;  **public** **class** AdapterPatternExample {  **public** **static** **void** main(String[] args) {  System.***out***.println("\*\*\*Adapter Pattern Demo\*\*\*\n");  CalculatorAdapter calculatorAdapter = **new** CalculatorAdapter();  Triangle t = **new** Triangle(20,10);  System.***out***.println("Area of Triangle is " + calculatorAdapter.getArea(t) + " Square unit");  }  } |

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 *Adapter* pada source 3,4 dan 5.

|  |
| --- |
| Proses awal yang dilakukan yaitu membuat sebuah kelas Calculator untuk menghitung luas namun hanya untuk bentuk Rectangle dengan variable length dan width yang dikalikan. Akan tetapi, ada bentuk lainnya yang juga ingin dihitung luasnya yaitu Triangle dengan variable base dan height. Namun karena kelas Calculator hanya dibuat untuk menghitung Rectangle maka Triangle tidak dapat dihitung luasnya.  Oleh karena itu dibuatlah sebuah kelas CalculatorAdapter yang berfungsi agar Triangle dapat dihitung luasnya juga dengan menggunakan kelas Calculator. Pertanyaannya bagaimana Calculator yang sebelumnya hanya untuk Rectangle dapat berfungsi juga untuk Triangle, karena proses kedua bentuk tersebut berbeda yaitu Rectangle dengan length \* width sementara Triangle dengan (base \* height) / 2.  Jawabannya yaitu dengan mengubah proses Rectangle agar menjadi Triangle yang dilakukan dalam kelas CalculatorAdapter. Dalam kelas tersebut yang pertama dilakukan yaitu memanggil kelas Calculator dan Rectangle, selanjutnya mengubah proses Rectangle menjadi Triangle dengan cara setiap variable Rectangle dijadikan variable Triangle seperti berikut :   * length menjadi base * width menjadi height   Akan tetapi dalam kelas Triangle tidak hanya mengkalikan kedua variable tersebut namun setelah dikalikan kemudian dibagi 1/2 atau dikalikan 0.5. Lalu harus diletakkan dimana sisa proses Trianglenya? Sisa prosesnya dapat diletakkan pada variable base ataupun height seperti berikut :   * 0.5 \* base atau 0.5 \* height   Dengan cara tersebut sama halnya dengan (base \* height) / 2 pada proses Triangle, langkah terakhir yaitu menghitungnya menggunakan proses yang ada di kelas Calculator. Perlu diingat kembali kelas Calculator sebelumnya hanya mempunyai perhitungan Rectangle sehingga prosesnya yaitu length \* width berbeda dengan Triangle. Namun karena sebelumnya pada kelas CalculatorAdapter telah diubah variable length dan width maka prosesnya ikut berubah menjadi 0.5 \* base \* heigth atau base \* 0.5 \* height sehingga mendapatkan hasil perhitungan Triangle  Salah satu contohnya sebagai berikut :    Pada contoh diatas kita ingin menghitung luas Triangle dengan base = 20 dan height = 10 akan tetapi hanya ada kelas Calculator yang menghitung Rectangle, untuk itulah pada contoh diatas menggunakan CalculatorAdapter agar Triangle dapat dihitung juga menggunakan kelas Calculator tersebut dan hasilnya seperti berikut :    Hasilnya tepat 100 dan bukan proses length \* width yang seharusnya 200 namun karena sudah diubah menjadi proses Triangle maka ditambahkan proses dikali 0.5 atau dibagi 1/2 sehingga 200 / 2 menjadi 100. |

1. Menurut pendapat Anda setelah mengimplementasikan *sourcode* di atas, apa perbedaan mendasar dari metode pada **Creational Pattern** dan **Structural Pattern**?

|  |
| --- |
| Menurut saya perbedaan mendasar dari metode Creational Pattern dan Structural Pattern terletak pada **pola penyusunnya**, jika Creational Pattern digunakan untuk membuat suatu objek dengan menggunakan pola tertentu sementara Structural Pattern digunakan dalam menyusun suatu objek dan kelas agar membentuk suatu pola yang terstruktur. |

**\*\*\***