|  |  |
| --- | --- |
| **Assignment Case** |  |
| COMP6115  Object Oriented & Analysis Design |
| **Computer Science** | **E213-COMP6115-PH04-01** |
| ***Valid on*** *Even Semester Year 2020/2021* | **Revision 00** |

1. Seluruh mahasiswa tidak diperkenankan untuk:

*All students are not allowed to:*

* + 1. Berdiskusi dan/atau bekerja sama dengan mahasiswa lainnya

*Discuss and/or work together with other student participants*

* + 1. Melihat sebagian atau seluruh jawaban mahasiswa lain

*Seeing a part or the whole answer from another student*

* + 1. Membuka dan menyalin dari **BUKU** atau **CATATAN**, **VIDEO** dari pengajar (recording kelas, VBL, Youtube, dsb) dan **REFERENSI** lainnya

*Open and copy from any resources such as notes, videos (class recording, VBL, Youtube, etc) and other references*

* + 1. Membuka dan menyalin jawaban dari internet (google, stackoverflow, dsb)

*Open and copy answer from the internet (google, stackoverflow, etc)*

* + 1. Mengerjakan soal yang tidak sesuai dengan tema yang ada di soal,

*Working with another theme which is not in accordance with the existing theme in the matter of the case,*

* + 1. Melakukan tindakan kecurangan lainnya,

*Committing other dishonest actions,*

* + 1. Secara sengaja maupun tidak sengaja melakukan segala tindakan kelalaian yang menyebabkan hasil karyanya berhasil dicontek oleh orang lain / kelompok lain.

*Accidentally or intentionally conduct any failure action that cause the results of the project was copied by someone else / other groups.*

1. Jika mahasiswa terbukti melakukan tindakan seperti yang dijelaskan butir 1 di atas, maka **nilai mahasiswa** yang melakukan kecurangan (menyontek maupun dicontek) akan di – **NOL** – kan.

*If the student is proved to the actions described in point 1 above, the score of the student which committed dishonest acts (cheating or being cheated) will be “Zero”*

1. Perhatikan jadwal pengumpulan jawaban, segala jenis pengumpulan jawaban di luar jadwal tidak dilayani.

*Pay attention to the submission schedule, all kinds of submission outside the schedule will not be accepted*

1. Bila Anda tidak membaca peraturan ini, maka Anda dianggap telah membaca dan menyetujuinya

*If you have missed to read these regulations, so you are considered to have read and agreed on it*

1. Persentase penilaiaan untuk matakuliah ini adalah sebagai berikut:

*Marking percentage for this subject is described as follows:*

|  |  |  |
| --- | --- | --- |
| **Tugas Mandiri**  *Assignment* | **Proyek**  *Project* | **UAP**  *Final Exam* |

|  |  |  |
| --- | --- | --- |
| 40% | 60% | - |

1. Software yang digunakan pada matakuliah ini adalah sebagai berikut:

*Software will be used in this subject are described as follows:*

|  |
| --- |
| **Software**  *Software* |

|  |
| --- |
| Visual Paradigm Community Edition 16.1 Java 8  Eclipse 2020.6  XAMPP 7.4.7  MySQL Java Connection Library 5.1.49 |

## Ekstensi file yang harus disertakan dalam pengumpulan tugas mandiri dan proyek untuk matakuliah ini adalah sebagai berikut:

*File extensions should be included in assignment and project collection for this subject are described as follows:*

|  |  |
| --- | --- |
| **Tugas Mandiri**  *Assignment* | **Proyek**  *Project* |
| VPP | JAVA, CLASS, SQL |

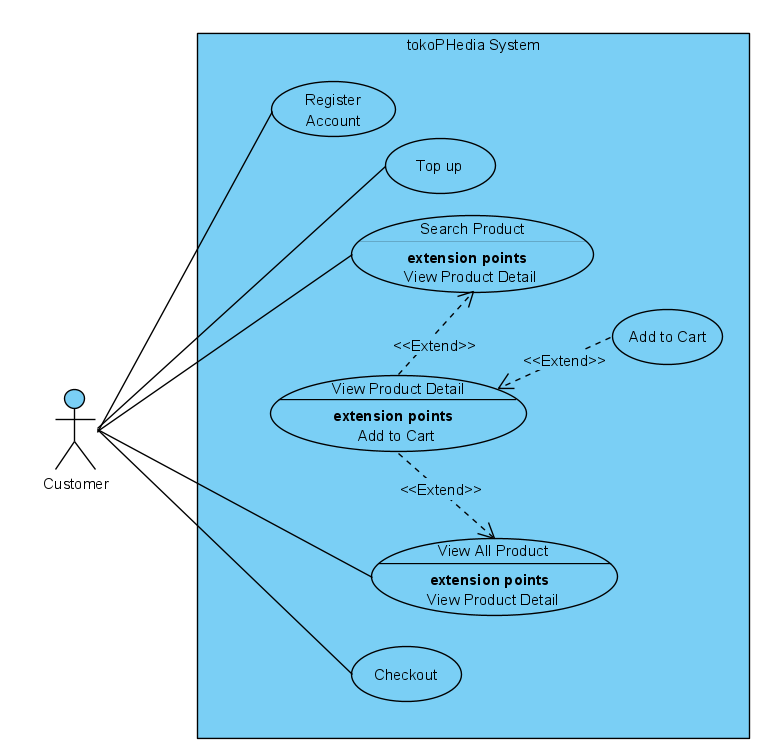
## Soal

*Case*

**tokoPHedia**

**tokoPHedia** is a brand new startup company from Indonesia. **tokoPHedia** provides an online shopping service for their customer. In their system, the customer can **register an account**, **top up their balance**, **view all available product**, **search a product**, **add a product to their cart** and also **checkout.**

As a system analyst, you are requested to create a system design of the **tokoPHedia** system based on the following images by using **Visual Paradigm (\*.vpp file format):**

**Use Case Diagram**

The program must have the **following features**:

* **Register Account**

When the **customer** opens the registration page, the program will show a form for the customer to fill with their data. The customer needs to fill in their **name**, **email**, **password**, **address**, and **phone** **number**. Then after the user submitted the form, the system will **validate** the input and then **generate a unique ID** for the customer.

* **Top Up**

After the **customer** has logged in and open the top up page, the program will show a form where the user can input the **amount** **of** **balance** and the payment **method**. After that, the system will **validate** that if the customer has input a valid **amount** and a valid **payment** **method**. If there is invalid input, show an error message, else **update** the customer balance accordingly.

* **Search Product**

After the **customer** has logged in, the customer can **search** a product based on the **product** **name.** Then, the system will show product with **name** related to the customer’s input.

PS: Customer can **click** on one of those products. If customer clicks one of those products, redirect them to the **product detail** page.

* **View All Product**

After the **customer** has logged in and chooses “**View All Product**”, then the system will show all of the available product.

PS: Customer can **click** on one of those products. If customer clicks one of those products, redirect them to the **product detail** page.

* **View Product Detail**

After the **customer** has logged in and clicked a product, then the system will show the detail of that product such as **description** and **price**.

PS: There will be an add to cart button. If the customer clicks “**Add** **to** **Cart**” button, redirect them to the add to cart page.

* **Add Product to Cart**

After the **customer** has logged in and **presses the add to cart button**, the system will show a form for adding the product to their cart. The customer will input the **quantity** and the system will validate whether the product **stock** is sufficient. Then, the system will validate whether the product is **already** in the customer’s cart or not. If yes, then **update** the cart quantity, if not then **add** the product to their cart.

* **Checkout**

After the **customer** has logged in and presses the checkout button, the system will show all the products in the customer’s cart and also shows the total price. After the customer **confirmed** the checkout, the system will check whether the customer’s balance is **enough**. If yes, then **deduct** the customer balance, update the product’s stock, clear all of the item in the customer’s cart and save the transaction, if not then **show** an error message.

You must create the **following** **diagrams** based on above’s **descriptions**:

* **3 Activity Diagram** (pick **3** from above’s use cases)
* **1 Sequence** **Diagram** (pick **1** from the activity diagrams that you’ve chose before)
* **1 Class** **Diagram** (pick **3** of the **following** **features**):
  + **Register Account**
  + **Top Up**
  + **Search Product**
  + **View All Product**
  + **View Product Detail**
  + **Add Product to Cart**
  + **Checkout**

**Good Luck.**