|  |  |
| --- | --- |
| **Assignment Case** |  |
| COMP6175 | COMP6708001  Object Oriented Programming |
| **Computer Science** | **O223-COMP6708-LC03901-01** |
| ***Valid on*** *Odd Semester Year 2021/2022* | **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* | **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* |
| Java 8  Eclipse 2020.6 |

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

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

|  |  |
| --- | --- |
| **Tugas Mandiri**  *Assignment* | **UAP**  *Final Exam* |
| JAVA, CLASS | JAVA, CLASS |

## Soal

*Case*

**BlueJack Station**

**BlueJack Station** is the biggest train station in Jakarta. Many people come here to board on trains to various destinations. Because the train station grows bigger each year, the station manager needs a program to keep track of all the train routes from **BlueJack Station** to other cities. Unfortunately, he can’t code. You as a good programmer, have been asked to **create a program** that can **list all the train routes** starting from **BlueJack Station**, based on the following criteria using **Java Programming Language:**

At the start of the application, it will show a **menu**. The menu consists of the following specifications :

1. **Insert Route**
2. **View Route**
3. **Delete Route**
4. **Exit**

Background pattern

Description automatically generated with low confidence

**Figure 1. home menu**

* If the user chooses **menu 1 (Insert Route):**
  + Ask the user to input:
    - **Destination**, which must be either **‘Jakarta’, ‘Bogor’, ‘Depok’, ‘Tangerang’, or ‘Bekasi’ (case sensitive)**
    - **Distance,** which must be a **number** between **1 and 1000**
  + After the user **successfully** inputted all the data, **generate** an **ID** for the route based on the following **formula** :

|  |
| --- |
| **ID = ‘BJXXX’ | X = random between 0-9**  **Example: BJ345** |

* + **Add** the data to **Array / Vector / ArrayList.**
  + Finally, **show success message**: **“Successfully added new route!”**

Chart

Description automatically generated with medium confidence

**Figure 2. insert route**

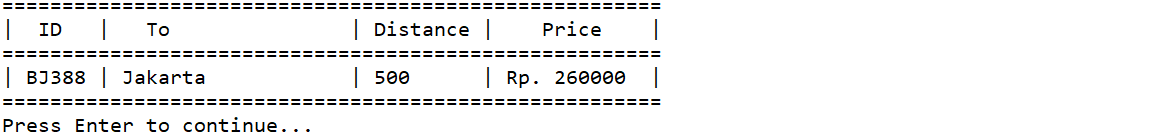
* If the user chooses **menu 2 (View Routes):**
  + If **there is no route data**, show the message **“There are no routes available.”,** then the program will return to **main menu.**



**Figure 3. no data message**

* + Otherwise, **show** the **list of all routes,** along with their **price,** which is calculated based on the following formula:

|  |
| --- |
| **Price = ((distance / 2 \* 1000) + 10000)** |



**Figure 4. view routes**

* If the user chooses **menu 3 (Update Route):**
  + If **there is no route data**, show the message **“There are no routes available.”,** then the program will return to the **main menu.**



**Figure 5. no data message**

* + Otherwise:
    - **Show** the **list of all routes,** along with their **price.**
    - Ask the user to input **Route ID**, which must be **a valid ID**.
    - If the user inputs an **invalid ID,** the program will **display error message** and return to the **main menu.**

Diagram

Description automatically generated with medium confidence

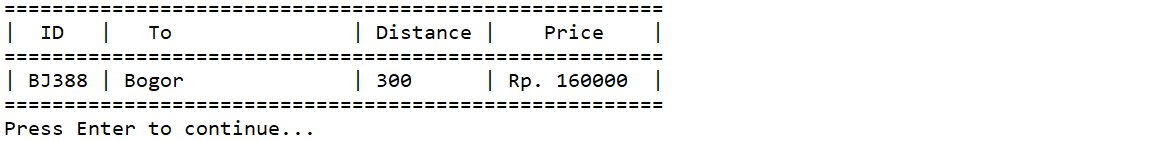
**Figure 6. invalid ID error message**

* + - Otherwise, the program will **ask** the user to **input the route details:**
      * **Destination**, which must be either **‘Jakarta’, ‘Bogor’, ‘Depok’, ‘Tangerang’, or ‘Bekasi’ (case sensitive)**
      * **Distance,** which must be a **number** between **1 and 1000**
    - After the user **successfully** inputted all the data, **update the route data.**
    - Finally, **show success message: “Successfully updated Route ” + the Route ID which was just updated.**

A picture containing diagram

Description automatically generated

**Figure 7. update route**



**Figure 8. view route (after update)**

* If the user chooses **menu 4 (Exit),** then the program will be closed.