|  |  |
| --- | --- |
| **Assignment Case** |  |
| COMP6048  Data Structures |
| **Computer Science** | **E213-COMP6048-DZ11-01** |
| ***Valid on*** *Even Semester Year 2021/2022* | **Revision 00** |

1. Seluruh mahasiswa tidak diperkenankan untuk:

*All students are not allowed to:*

* + - Berdiskusi dan/atau bekerja sama dengan mahasiswa lainnya

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

* + - Melihat sebagian atau seluruh jawaban mahasiswa lain

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

* + - 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*

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

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

* + - 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,*

* + - Melakukan tindakan kecurangan lainnya,

*Committing other dishonest actions,*

* + - 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* |
| Dev-C++ 5.11 |

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

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

|  |  |
| --- | --- |
| **Tugas Mandiri**  *Assignment* | **UAP**  *Final Exam* |
| CPP | CPP |

## Soal

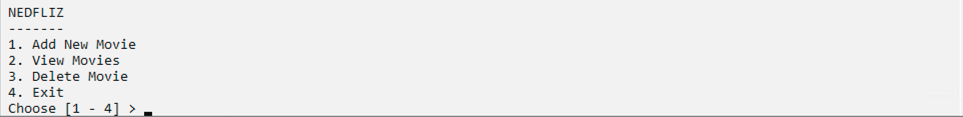
*Case*

**NEDFLIZ**

**NEDFLIZ** is a famous movie shop in Jakarta. **NEDFLIZ** handle a lot of movie orders every day. To increase the efficiency in **NEDFLIZ**, you as a programmer in **NEDFLIZ** are asked to create a program that can maintain customer order data using **C language** and **Chaining Hash Table** data structure. The program that will be created must be following the below requirements.

* The program will have **4 menus**:

1. Add New Movie
2. View Movie
3. Delete Movie
4. Exit

**

*Figure 1. Main Menu*

* If the user choose **menu 1** (**Add New Movie**), then:

1. The program will ask user to input the **movie’s title** and **validate** that the **title** must **between 5 and 20** characters (**inclusive**).
2. The program will ask user to input the **movie’s genre** and **validate** that the **genre** must be “**Adventure**” or “**Romance**” or “**Comedies**” (**case insensitive**).
3. The program will ask user to input the **movie’s rating** and **validate** that the **rating** must be **between 0.0 and 5.0** (**inclusive**).
4. The program will ask user to input the **movie’s year published** and **validate** that the **year** must be **between 1900 and 2020** (**inclusive**).

Graphical user interface, text, application

Description automatically generated

*Figure 2. Input validation*

1. The program will **generate the Movie Id** for the inputted order data. The Movie Id must follow the format below:

**YXXX**

XXX : the last 3 digits of the last movie id added by 1

**Genre:**

* Adventure : Y = “AE”
* Romance : Y = “RE”
* Comedies : Y = “CS”

**Example**:

The last movie id is **AE003**

Then the new generated movie id with genre **Comedies** is **CS004**

*Figure 3. Movie Id Format*

1. The program will **store new order data** to the **next item** of the **last item** of **chaining hash table** with **size** **100** using the following **hash function**.

Key = X[4] + X[2] - X[3] % Y

Key : the hash table index that will store the data

X : ID of the Movie that will be stored

Y : size of the hash table (100)

**Example**:

Movie Id : RE001

Size : 100

Key : (49 + 48 – 48) % 100

: 49

Then the order data will store at index 49 of hash table

*P.s for each character on X is based on ASCII*

*Figure 4. Hash Function*

* If the user choose **menu 2** (**View Movies**), then:

1. The program will **show all the movie list**. If there are no movie yet or the list is empty, then the program **should show no order message** “**No Data !!!**” and **redirect** user back to menu page

A picture containing graphical user interface

Description automatically generated

*Figure 5. No Movie Yet*

1. Otherwise, the program will show all the movie.

A picture containing text

Description automatically generated*Figure 6. Show movie list*

* If the user choose **menu 3** (**Delete Movie**), then:

1. The program will **show all the movie list**. If there are no movie yet or the list is empty, then the program **should show no movie message** and **redirect** user back to menu page

A picture containing graphical user interface

Description automatically generated

*Figure 7. Remove Movie No Data*

1. Otherwise, the program will ask user to input the **movie id to delete** and **validate** that **movie ID must start with “AE”, “RE”, “CS” (case sensitive)**.

Text, letter

Description automatically generated

*Figure 8. Remove Movie ID Validation*

1. The program will **search** for the **specified movie (case sensitive)**.

* If the **movie id** is **not found** the program will **show** the **movie not found** message and user **will be redirect back to menu page**.

Graphical user interface, text

Description automatically generated with medium confidence

*Figure 9. Movie Not Found Message*

* If the **movie id exists**, then:
  + The program will **remove** the **movie data** from the **chaining hash table**
  + The program will **show** the **movie ID and title**
  + The programwill **redirect back to menu page**

A picture containing text

Description automatically generated

*Figure 10. Successfully deleted movie message*

* If the user choose **menu 4** (**Exit**), then the program will **exit**.

**Please run the EXE file to get more detail about the application.**

Here are the rules that you must follow to create your project:

1. Use appropriate software for this subject based on **Sistem Praktikum** that can be downloaded from Binusmaya
2. Use the techniques taught during practicum
3. Collect appropriate files for this subject based on **Sistem Praktikum** that can be downloaded from Binusmaya
4. Include the other files that can support your project, such as:
   * All files in your project
   * Other files (image, audio, video, etc.) used in your project
   * \*.DOC file (documentation of your project) that contains the reference links of additional files (image, audio, video, etc.) used in your project