|  |  |
| --- | --- |
| **Project Case** |  |
| COMP7066 | T0273  Expert Systems |
| **Computer Science** | **E182-COMP7066-IC01-00** |
| ***Valid on*** *Even Semester Year 2017/2018* | **Revision 00** |

1. Seluruh kelompok tidak diperkenankan untuk:

*The whole group is not allowed to:*

* + - Melihat sebagian atau seluruh proyek kelompok lain,

*Seeing a part or the whole project from other groups*

* + - Menyadur sebagian maupun seluruh proyek dari buku,

*Adapted a part or the whole project from the book*

* + - Mendownload sebagian maupun seluruh proyek dari internet,

*Downloading a part or the whole project from the internet,*

* + - Mengerjakan soal yang tidak sesuai dengan tema yang ada di soal proyek,

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

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

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

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

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

1. Jangan lupa untuk melihat kriteria penilaian proyek yang ditempel di papan pengumuman, atau tanya asisten anda.

*Don’t forget to look at the project assessment criteria that posted on the announcement board, or ask your teaching assistant.*

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* |
| Java 8  Eclipse Neon 2 with JESS 71p2 |

## 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* |
| CLP | Folder Project (JAVA, CLP, CLASSPATH, FATJAR, PROJECT, JAR, PREFS, CLASS) |

**Soal**

*Case*

# These following facts must be included when the application is started:

## **Male’s Facts**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Age** | **Hobby** | **Interest** | **Height** | **Income** |
| Calvin | 20 | Game | Female | 170 | 600 |
| Vincent | 18 | Sport | Female | 165 | 450 |
| Bambang | 33 | Music | Female | 175 | 200 |
| David | 69 | Sport | Male | 159 | 1000 |
| Cristian | 27 | Game | Female | 192 | 50 |

## **Female’s Facts**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Age** | **Hobby** | **Interest** | **Height** |
| Cindy | 18 | Music | Male | 150 |
| Jennifer | 50 | Game | Female | 190 |
| Tukiyem | 99 | Sport | Male | 167 |
| Elis | 32 | Music | Male | 170 |
| Yolanda | 28 | Game | Male | 164 |

**TICDER**

**TICDER** is a social search app that allows users to find a date. The customer of **TICDER** is increasing because desperate people, who want to find a lover grows rapidly each day. You as a well-known programmer asked to build an application that can give an expert solution for helping **TICDER** to help users find a matching date. Below are the descriptions of this application:

# The application consists of **6 main menus**. They are:

* + - 1. **View Members**
      2. **Add a New Member**
      3. **Update Member’s Profile**
      4. **Delete Member**
      5. **Search Match**
      6. **Exit**

Then, the application will ask the user to input the **choice**. The choice must be **numeric** and **between 1 and 6**.

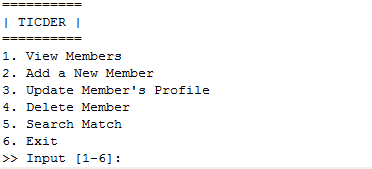


Figure 1. Main Menu

# If the user chooses **menu 1** “**View Members**”, then:

## The application will **show a sub menu** consists of ‘**Male**’ and ‘**Female**’ and ask the user to input the **choice**. The choice must be **numeric** and either **1 or 2**. The user can also input **0** to **back** to **main menu**.



Figure 2. View Members Sub Menu

## If the user chooses the **sub menu 1** (“**Male**”), then the application will show **all Male facts**.

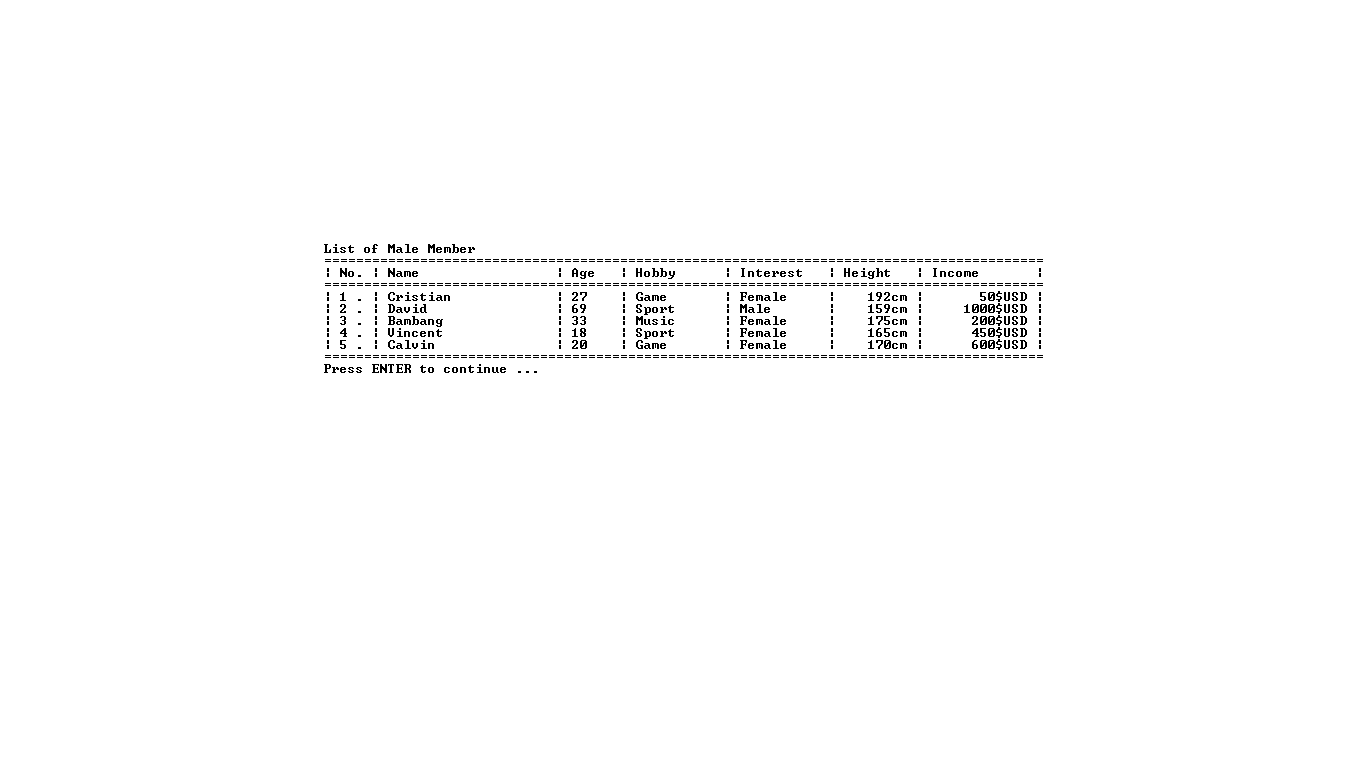


Figure 3. List of Male Facts

## If the user chooses the **sub menu 2** (“**Female**”), then the application will show **all Female facts**.

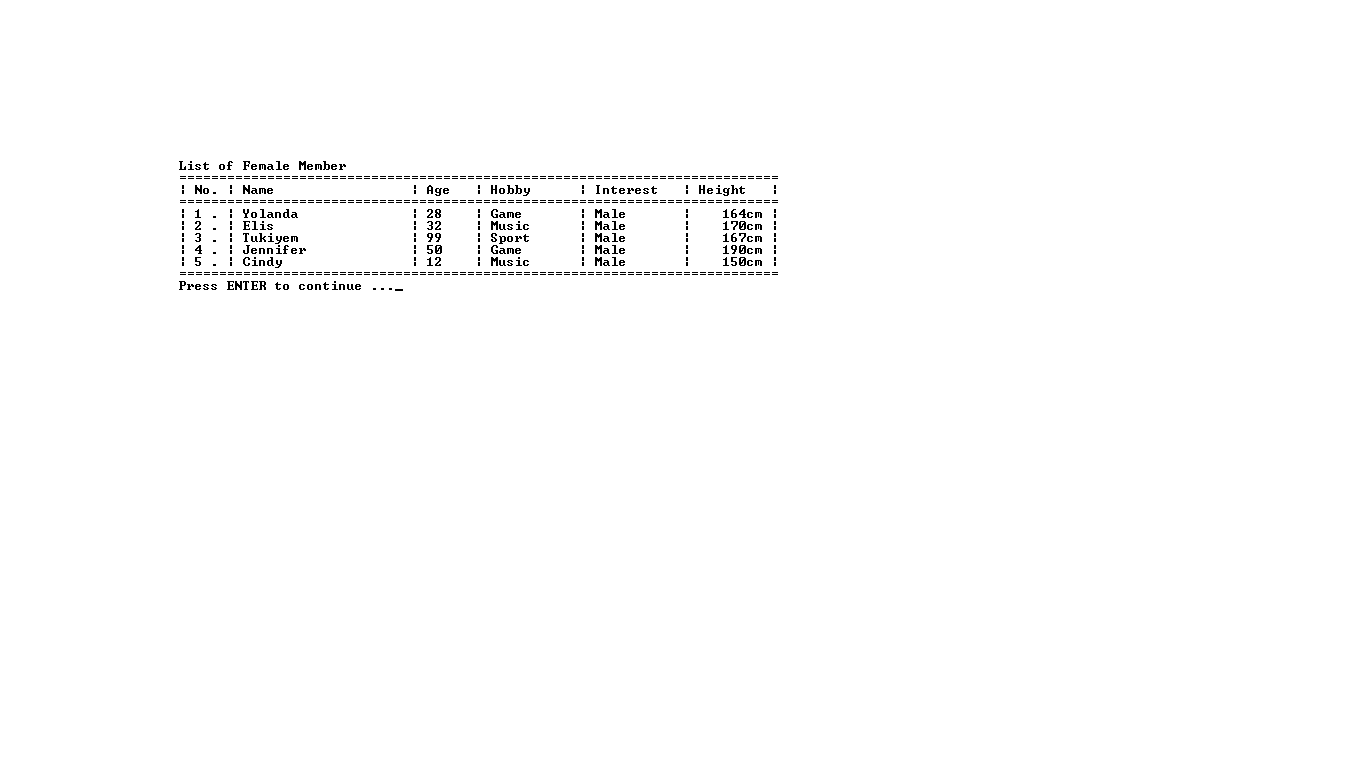


Figure 4. List of Female Facts

# If the user chooses **menu 2** “**Add a New Member**”, then:

## The application will **show a sub menu** consists of ‘**Male**’ and ‘**Female**’ and ask the user to input the **choice**. The choice must be **numeric** and either **1 and 2**. The user can also input **0** to **back** to **main menu**.



Figure 5. Add New Member Sub Menu

## If the user chooses the **sub menu 1** (“**Male**”), then:

### The application will ask the user to input the **name of the member**. The **length** of the **name** must be **between 3 and 20** characters.

### The application will ask the user to input the **member**’**s age**. The **age** must be **numeric and** **between** **16 and 80**.

### The application will ask the user to input the **member**’**s hobby**. The **hobby** must be **either** “**Sport**”, “**Music**” or “**Game**”. (**CASE-SENSITIVE**)

### The application will ask the user to input the **member**’**s relationship preference**. The **preference** must be **either** “**Male**” or “**Female**”. (**CASE-SENSITIVE**)

### The application will ask the user to input the **member**’**s height**. The **height** must be **numeric** and **between 100 and 240**.

### The application will ask the user to input the **member**’**s income**. The **income** must be **numeric** and **between 0 and 10.000**.

### After that, the application will **insert** all data **into** the **Male facts**.

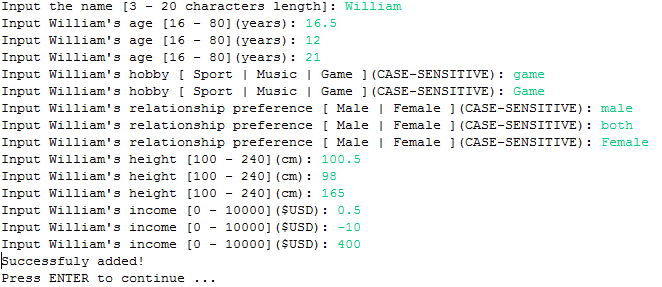


Figure 6. Input new Male Member Data

## If the user chooses the **sub menu 2** (“**Female**”), then:

### The application will ask the user to input the **name of the member**. The **length** of the **name** must be **between 3 and 20** characters.

### The application will ask the user to input the **member**’**s age**. The **age** must be **numeric and** **between** **16 and 80**.

### The application will ask the user to input the **member**’**s hobby**. The **hobby** must be **either** “**Sport**”, “**Music**” or “**Game**”. (**CASE-SENSITIVE**)

### The application will ask the user to input the **member**’**s relationship preference**. The **preference** must be **either** “**Male**” or “**Female**”. (**CASE-SENSITIVE**)

### The application will ask the user to input the **member**’**s height**. The **height** must be **numeric** and **between 100 and 240**.

### After that, the application will **insert** all data **into** the **Female facts**.

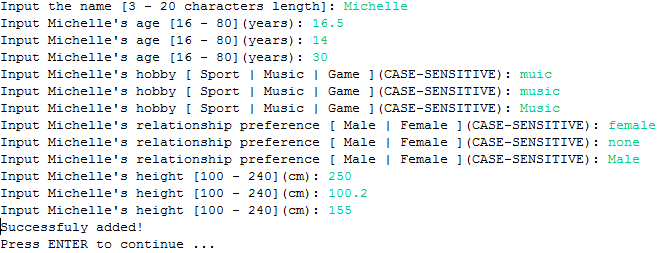


Figure 7. Input new Female Member Data

# If the user chooses **menu 3** “**Update Member’s Profile**”, then:

## The application will **show a sub menu** consists of ‘**Male**’ and ‘**Female**’ and ask the user to input the **choice**. The choice must be **numeric** and either **1 and 2**. The user can also input **0** to **back** to **main menu**.

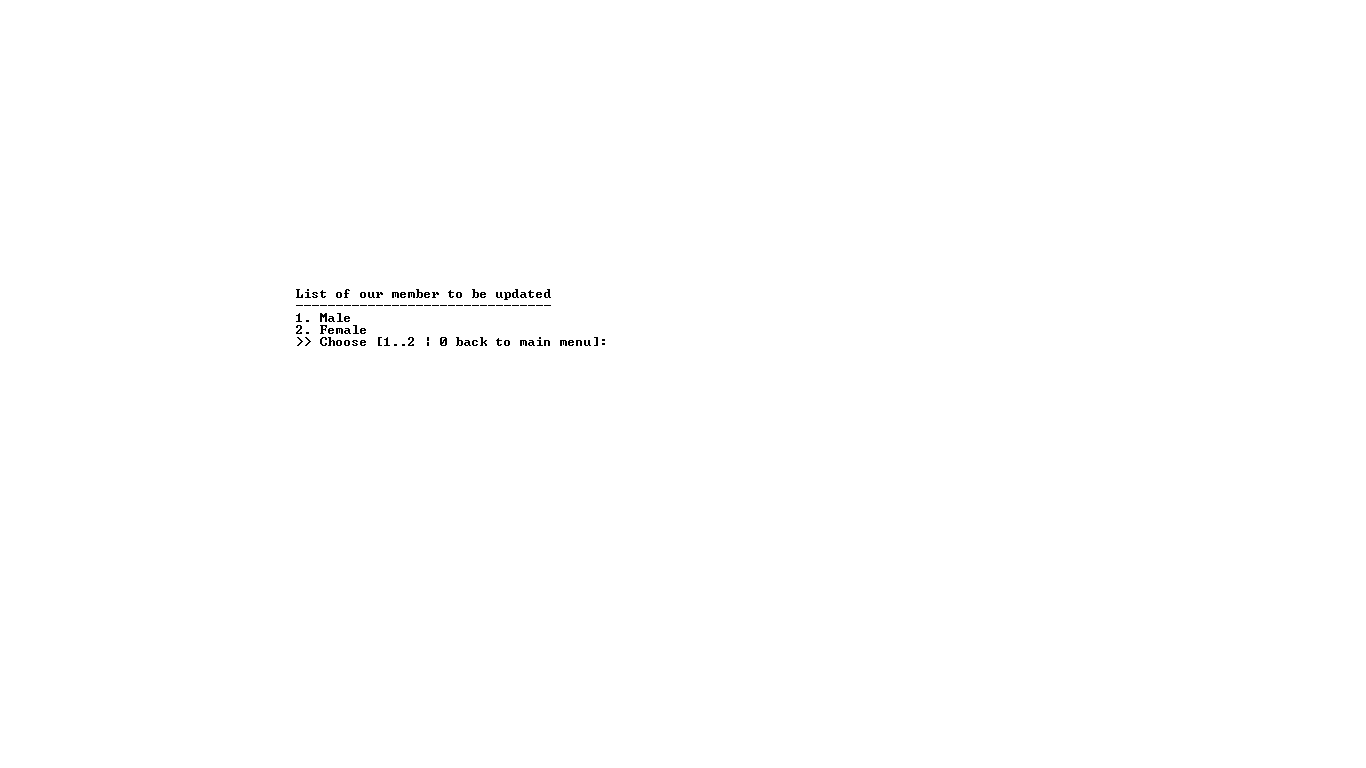


Figure 8. Update Member's Profile Data

## If the user chooses the **sub menu 1** (“**Male**”), then:

### The application will **show** **all** **Male** **facts** and ask the user to **input the number of the male member** that need **to be updated.** The input must be **integer** and **between 1 and the latest number of facts**. The user can also input **0** to **back** to **main menu.**

### The application will ask the user to input the **name of the member**. The **length** of the **name** must be **between 3 and 20** characters.

### The application will ask the user to input the **member**’**s age**. The **age** must be **numeric and** **between** **16 and 80**.

### The application will ask the user to input the **member**’**s hobby**. The **hobby** must be **either** “**Sport**”, “**Music**” or “**Game**”. (**CASE-SENSITIVE**)

### The application will ask the user to input the **member**’**s relationship preference**. The **preference** must be **either** “**Male**” or “**Female**”. (**CASE-SENSITIVE**)

### The application will ask the user to input the **member**’**s height**. The **height** must be **numeric** and **between 100 and 240**.

### The application will ask the user to input the **member**’**s income**. The **income** must be **numeric** and **between 0 and 10.000**.

### After that, the application will **update** all data basedon **the selected male member number** to be updated.

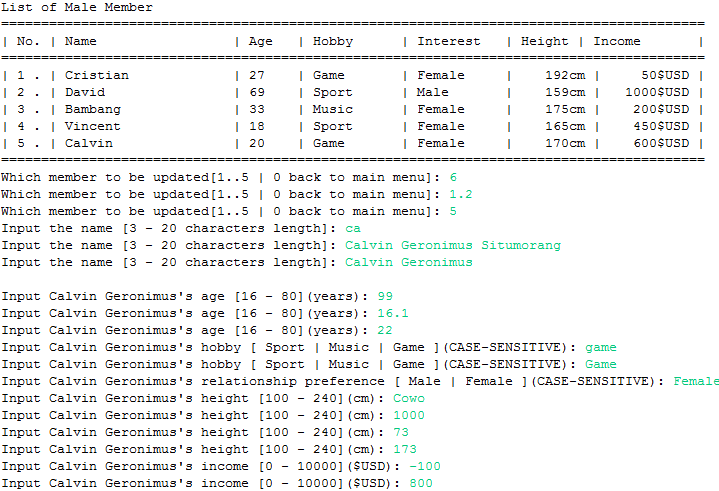


Figure 9. Input New Data for Male Member



Figure 10. Success Message after Update

## If the user chooses the **sub menu 2** (“**Female**”), then:

### The application will **show** **all** **Female** **facts** and ask the user to input the **number** of **female member** **to be updated.** The input must be **numeric** and **between 1 and the latest number of facts**. In here, the user can also input **0** to **back** to **main menu.**

### The application will ask the user to input the **name of the member**. The **length** of the **name** must be **between 3 and 20** characters.

### The application will ask the user to input the **member**’**s age**. The **age** must be **numeric and** **between** **16 and 80**.

### The application will ask the user to input the **member**’**s hobby**. The **hobby** must be **either** “**Sport**”, “**Music**” or “**Game**”. (**CASE-SENSITIVE**)

### The application will ask the user to input the **member**’**s relationship preference**. The **sex interest** must be **either** “**Male**” or “**Female**”. (**CASE-SENSITIVE**)

### The application will ask the user to input the **member**’**s height**. The **height** must be **numeric** and **between 100 and 240**.

### After that, the application will **update** all data basedon **the selected female member** **number** to be updated.

### 

Figure 11. Input New Data for Female Member



Figure 12. Success Message after Update

# If the user chooses **menu 4** “**Delete Member**”, then:

## The application will **show a sub menu** consists of ‘**Male**’ and ‘**Female**’ and ask the user to input the **choice**. The choice must be **numeric** and either **1 and 2**. The user can also input **0** to **back** to **main menu**.



Figure 13. Sub Menu of Delete Member

## If the user chooses the **sub menu 1** (“**Male**”), then:

### The application will **show** **all** **Male** **facts** and ask the user to input the **number** of **the male member to be deleted.** The input must be **numeric** and **between 1 and the latest number of facts**. The user can also input **0** to **back** to **main menu**.

### Then, the application will **delete** a data basedon **which male member number** to be deleted.

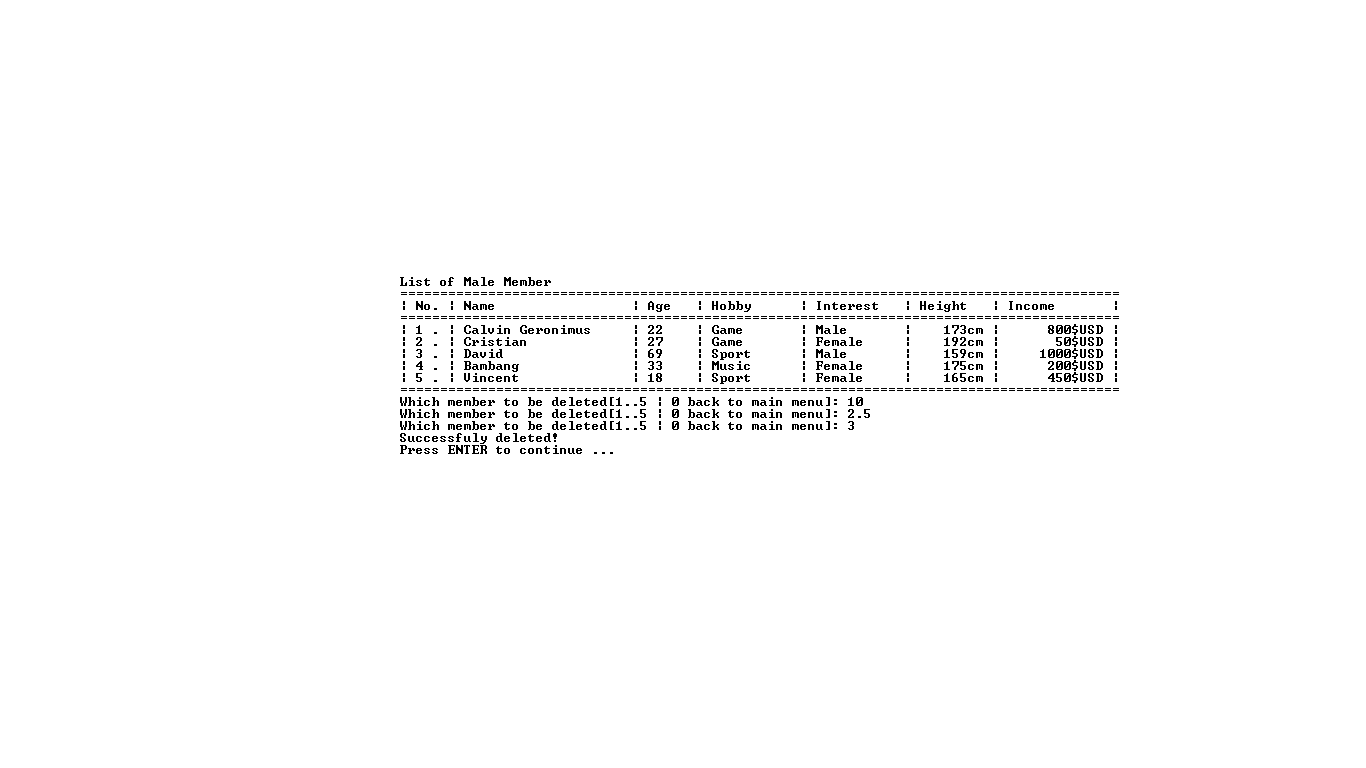


Figure 14. Input Male fact to be deleted

## If the user chooses the **sub menu 2** (“**Female**”), then:

### The application will **show** **all** **Female** **facts** and ask the user to input the **number** of **female member** **to be deleted.** The input must be **numeric** and **between 1 and the latest number of facts**. The user can also input **0** to **back** to **main menu**.

### Then, the application will **delete** a data basedon **which female member number** to be deleted.

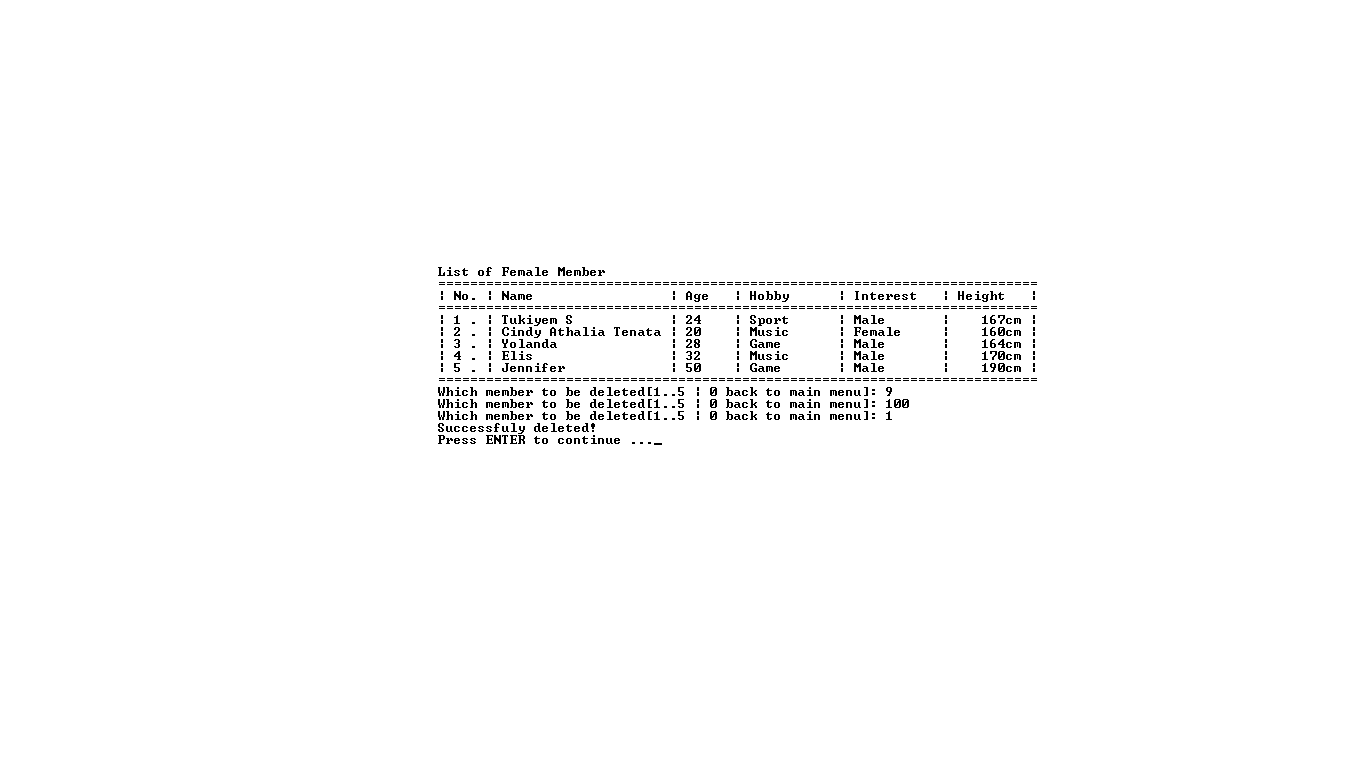


Figure 15. Input Female fact to be deleted

# If the user chooses **menu 5** “**Search Match**”, then:

## The application will ask the user to input **his**/**her** **name**. The **length** of the **name** must be **between 3 and 20**.

## The application will ask the user to input **his**/**her** **gender**. The **gender** must be either “**Male**” or “**Female**”. (**CASE-SENSITIVE**)

## The application will ask the user to input **his**/**her** **relationship preference**. The **preference** must be either “**Male**” or “**Female**”. (**CASE-SENSITIVE**)

## The application will ask the user to input **his**/**her age**. The **age** must be **numeric and** **between** **16 and 80**.

## The application will ask the user to input **his**/**her height**. The **height** must be **numeric and** **between** **100 and 240**.

## The application will ask the user to input **his**/**her** **hobby**. The **hobby** must be either “**Sport**”, “**Music**” or “**Female**”. (**CASE-SENSITIVE**)

## If the **user**’**s** **relationship preference** is **Male**, then the application will ask the user to input **his**/**her** **preferred income**. The **preferred income** must be **numeric and** **between** **0 and 10.000**.

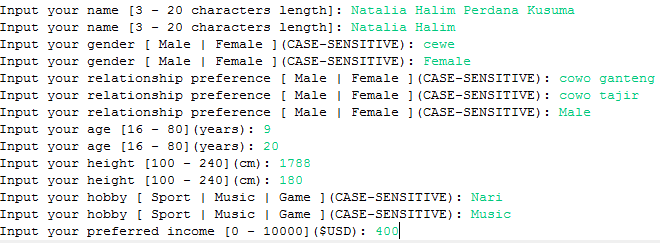


Figure 16. Input desired Couple's data

## Then, the application will search for the corresponding member from **Male Facts** if **the user**’**s relationship preference** is **Male. I**f the **user**’**s** **relationship preference** is **Female the application will search for Female Facts**. The fact must fulfill these following criteria:

### The **age gap between user** and **member must be less than 40**.

### The **member** has the **same preference** as **user**’**s gender**.

### The **height gap between user** and **member must be less than 40**.

### Then **compute** **user** and **member** **match** **rate** by following these rules:

* **Initial** **match rate value is 100.**
* If **user**’**s** **hobby** and **member**’**s** **hobby** **is not equals**, **decrease** **match** **rate** **by** **20.**
* **Decrease match rate with user and member’s age gap.**

**Example**: If current **match rate is 90**, **the** **user age is 40** and **the member age is 60**, then the match rate is **90** – (**|60 – 40|)** (**the age gap is absolute number**)

* **Decrease match rate with user and member’s height gap.**

**Example**: If current **match rate is 90**, **the** **user height is 160** and **the member height is 180**, then the match rate is **90** – (**|60 – 40**|) (**the age gap is absolute number**)

* If **user**’**s relationship preference interest** is **Male**,then **decrease match rate** with **user**’**s preferred income** and **member**’**s income gap if member**’**s income** is **less than user**’**s preferred income**.

**Example**:

**Current user fact**:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Name | Gender | Age | Hobby | Interest | Height | Income |
| Jeffry | Male | 21 | Sport | Male | 168 | 1000 |

**Member fact** that being matched:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Name | Gender | Age | Hobby | Interest | Height | Income |
| Vincent | Male | 18 | Sport | Male | 165 | 450 |

First, the application will check the match validity:

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | User | Member | Result |
| Age Gap | 21 | 18 | True ((|21 – 18|) < 40) |
| Height Gap | 168 | 165 | True ((|168-165|) < 40) |
| Interest | Male (Gender) | Male (Interest) | True |
| Final Result | | | True |

* The application will check above Final Result:
* If the result is ‘**FALSE**’, then the application will **terminate current matching process** and **continue to next matching process**.
* If the result is ‘**TRUE**’, then the application will calculate the match rate:

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | User | Member | Result |
| Hobby | Sport | Sport | 0 |
| Age | 21 | 18 | 3 |
| Height | 168 | 165 | 3 |
| Match Rate | | | **94** (100 – (0 + 3 + 3)) |

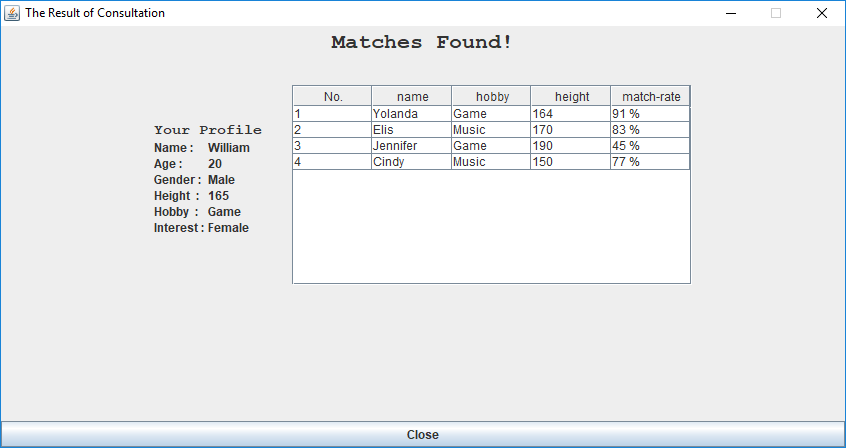
* Because the current user interest is ‘**Male**’, the user’s gender is ‘**Male**’ and member’s gender is ‘**Male**’, then the application will count their income match rate. If the **current user income is 350** and the **member income is 450**. Then **the match rate will be decreased** with following syntax:

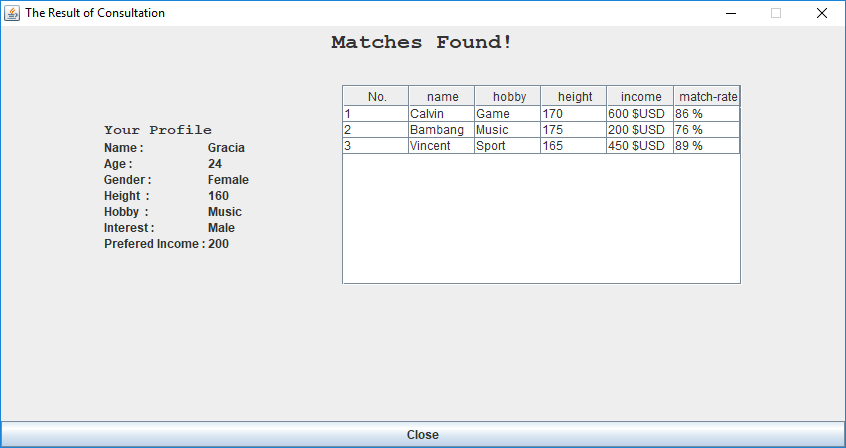
**Current Match Rate – (|User’s Income – Member’s Income|)**

**Final Match Rate** = 94 – (|1000 – 450|) = -456 (Not Match because below 0 or equals 0)

* If **match** **rate** **value** managed to **reach** **zero** **or** **below**, **exclude** **member** from the result.

## After that, the application will **show** the **result of match searching** in the **GUI** (**Graphical User Interface**) view and go back to the main menu.





## Otherwise, if there’s **no member** that fulfill the criteria, then the application will **show** the image that describes **not available** instead of **table** that list **the members**.

## 

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, such as: building facts (facts, assert, modify, retract, deffact, deftemplate), building rule (defrule), knowledge representation (forward chaining, backward chaining), and building expert system.
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

**If you have any question, please ask your assistant!**