Practicum Case	
COMP6178 | COMP6178003 Introduction to Programming	
Computer Science	**C1-COMP6178-SH01-08**
Valid on *Even Semester Year 2019/2020*	**Revision 00**

Learning Outcomes
- LO1 – Explain the kind of the algorithm in problem solving
- LO2 – Explain the usefulness of java syntax and OOPs
- LO3 – Explain the algorithm using java syntax
- LO4 – Explain the best algorithm in problem solving

Topic
- Session 08 – Static and Dynamic Array

Sub Topics
- ArrayList
- ArrayList Method
- Vector
- Vector Method

Soal
Case

Food Order

Food Order is a program to input food order from the customer. As a programmer, you are asked to create the program that consists of **5 menus**:

1. **Insert New Food**
2. **List of food**
3. **Search food**
4. **Delete food**
5. **Exit**

```
               Food Order
               ==========
1. Insert new food
2. List of food
3. Search food
4. Delete food
5. Exit
Choose :
```

- If user chooses **menu 1** (**insert new food**), then:
  - The program will ask user to input:
    - **Food's type** must be "**Appetizer**", "**Main Couse**", or "**Dessert**" (**case insensitive**)
    - **Food's name** must be **between 3 and 20 characters**
    - **Food's calories** must be **between 1.0 and 4.0**
    - **Food's price** must be **between 10000 and 50000**
  - Next, the program will show message "**Data has been successfully inserted!**".

```
Input food's type[Appetizer | Main Course | Dessert]: side dish
Input food's type[Appetizer | Main Course | Dessert]: appetizer
Input food's name[3-20]: to
Input food's name[3-20]: tofu tart
Input food's calories[1.0..4.0]: 0.8
Input food's calories[1.0..4.0]: 2.5
Input food's price[10000..50000]: 900000
Input food's price[10000..50000]: 45000
Data has been successfully inserted!
```

- If user chooses **menu 2** (**List of food**):
  - The program will show **no**, **type**, **name**, **calories**, and **price** of the food.

```
No    : 1
==========
1. Type     : appetizer
2. Name     : tofu tart
3. Calories : 2.5
4. Price    : 45000

No    : 2
==========
1. Type     : dessert
2. Name     : fruit punch
3. Calories : 3.1
4. Price    : 20000
```

  - If there is no data, the program will show message "**No Data!**".

```
No data!
```

- If user chooses **menu 3** (**search food**), the program will ask user to input **food's name** that want to be search. Validate the input **must be between 3 and 20 characters**.
  - If the inputted name is **not found**, the program will show "**No Data Found!**".

```
Input food's name [3-20] : fried chicken
No data found!
```

  - **Otherwise**, the program will show the **no**, **type**, **name**, **calories**, and **price** of the food.

```
Input food's name [3-20] : tofu tart
No    : 1
==========
1. Type     : appetizer
2. Name     : tofu tart
3. Calories : 2.5
4. Price    : 45000
```

- If user chooses **menu 4** (**delete food**), then:
  - The program will show **no**, **type**, **name**, **calories**, and **price** of the food and ask user to input number that want to be deleted. Validate the input **must be between 1 and the total of food** and user can **choose 0 for cancel**.

```
No    : 1
==========
1. Type     : appetizer
2. Name     : tofu tart
3. Calories : 2.5
4. Price    : 45000


No    : 2
==========
1. Type     : dessert
2. Name     : fruit punch
3. Calories : 3.1
4. Price    : 20000


Input number(No) to be deleted [1..2][0 for cancel] :
```

  - Delete the chosen data and show "**Data has been successfully deleted!**"

```
Input number(No) to be deleted [1..2][0 for cancel] : 1
Data has been successfully deleted!
```

  - If there is no data, the program will show message "**No Data!**"

```
No data!
```

- If user chooses **menu 5** (**exit**), the program will be closed.

Please run the EXE file to see the sample program.