


Ujian Akhir Praktikum - Semester Ganjil 2018/2019 <i>Practicum Final Exam - Odd Semester Year 2018/2019</i>		
Matakuliah <i>Subject</i> COMP6153 - Operating System		
Kelas <i>Class</i> : BT01	Tanggal <i>Date</i> : 1/12/2019	
Dosen <i>Lecturer</i> : D1279 - Tatang Gunar Setiadji, M.Eng.	Waktu <i>Time</i> : 07.20 - 09.00	

PERATURAN UJIAN:

Exam Regulations:

- Mahasiswa tidak diperbolehkan membuka **BUKU** atau catatan apapun
Student is not allowed to open or use any books or notes
- Mahasiswa tidak diperbolehkan menggunakan disket/flash disk/tempat penyimpanan data apapun di dalam ruangan ujian
Student is not allowed to use disc/flash disc/or any other storage devices in exam room
- Mahasiswa tidak diperbolehkan berdiskusi dengan peserta ujian lainnya
Student is not allowed to discuss with other exam participants
- Asisten **BERHAK** memberi nilai 0 (**NOL**) bagi peserta ujian yang melakukan segala bentuk kecurangan
Assistant is able to give 0 (ZERO) score for exam participant who does any cheating actions
- Kumpulkan jawaban dan soal tepat pada waktunya
Submit the case and answer on time
- 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

PERHATIAN!

Attention!

- Kerjakan sesuai dengan soal
Work in accordance with the exam case
- Bagi yang mengerjakan tidak sesuai dengan soal akan diberikan sanksi sebagai berikut:
For those who do not work in accordance with the exam case will be punished as follows:
 - **Jika ada** penilaian logika, maka penilaian logika adalah **NOL (0)**
If there is logic marking, it will be marked as ZERO (0)
 - Untuk penilaian yang lain (selain logika) akan **dikalikan** setengah dari nilai yang diperoleh
Another marking (beside logic) will be multiplied by half the value obtained

Important Notes:

1. Use your **own Console and Scheduler (FIFO)**.
2. You must use **Semaphore** to **control** operation.
3. You must use **Timer** to generate tick of time.
4. Use **KThread**.

Soal*Case***ToDo List**

Mr. TD wants to develop an app named **ToDo List** in BlueJack land, Mr. TD wanted to make an application with the intention of **managing Mr. TD's tasks**. The features provided in this application are insert, view and complete task. As a programmer, you are asked to create the application using **nachOS** in **Java programming language**.

In the start of the application, the program will look for a file named **"task.txt"** which will be **converted into an acceptable task data** and load it into the system with the help of file system if the file is available.

After the said process has been completed, the application will then show **4 menus** which represent features in the application which are:

- ❖ **Insert Task**
- ❖ **View Task**
- ❖ **Complete Task**
- ❖ **Exit**

```

ToDo list
=====
1. Insert Task
2. View Task
3. Complete Task
4. Exit
>>

```

Figure 1. Main menu

1. Insert Task

The first menu option let the user **inserts a new task into the system**. This option requires the user to input information regarding the task that follows the following detail which are:

- Ask the user to **input Task's Title**. The task's title must be **between 5 and 15 characters**.

```
Input task title [5 - 15]: Bobo
Input task title [5 - 15]: Mancing
```

Figure 2. Title input

- Ask the user to **input Task's Description**. The task's description **must contains at least 2 words**.

```
Input task description [more than 1 word]: Mancing
Input task description [more than 1 word]: Mancing ikan
```

Figure 3. Description input

- Ask the user to **input Task's type**. The task's type **must be "Important", or "Unimportant" (Case Insensitive)**.

```
Input task type [Important | Unimportant]: Import
Input task type [Important | Unimportant]: important
```

Figure 4. Type input

- After all the required input has been filled, the task's status will set to **"Not Done"**, and **show success message** to the user.

```
Task has been added successfully!
```

Figure 5. Success message

2. View Task

The second option of the menu (**View Task**) allows the user to **view all data in the system**. The process follows the following steps which are:

- **Check** if book list **has any data**. If the book list **didn't have any data**, the application will **show an error message** to the user.
- Otherwise, if the task list **already filled or have a task data**, the application will **show all task**.

```
Task List
=====

No. 1
=====
Title       : Mancing
Description  : Mancing ikan
Type        : Important
Status      : Not Done

No. 2
=====
Title       : Minum
Description  : Minum bersama angkatan
Type        : Unimportant
Status      : Not Done

No. 3
=====
Title       : Masak
Description  : masak nasi
Type        : Unimportant
Status      : Done

Press enter to continue...
```

Figure 6. View all data

3. Complete Task

The third option in the menu allows the user to **complete an uncompleted task from the list**. The completing process follows the following steps which are:

- **Check** if there at least one uncompleted task in task list. If there is no any uncompleted task in the list, the application will **show an error message** to the user.
- Otherwise, the application **will print all uncompleted task only on task list** and ask the user what **index of a task that want to be completed**. The **index starts from 1 to number of uncompleted task(s)**.

```

Task List
=====

No. 1
=====
Title       : Mancing
Description  : Mancing ikan
Type        : Important
Status      : Not Done

No. 2
=====
Title       : Minum
Description  : Minum bersama angkatan
Type        : Unimportant
Status      : Not Done

Input task number [1..2]:
  
```

Figure 7. View uncompleted task

- After the process has been **completed successfully**, **Update the task's status to "Done"** using **KThread** and direct the user back **to the main menu**.

```
Input task number [1..2]: 6  
Input task number [1..2]: 1  
  
Task with title: Mancing has been mark as 'Done'
```

Figure 8. Complete task

4. Exit

If the user chooses the forth **Menu (Exit)**, the application will **print tick of time in second using a timer**.

```
Your application has been running for 110 second(s).
```

Figure 9. Exit