**Important Notes**:

1. Use your **own Console** and **Scheduler**.
2. Youmustuse **Semaphore** to **control** operation.
3. You must use **Timer** to print tick of time.
4. Use **KThread** to view existing files.

**Soal**

*Case*

**File Manager**

**File Manager** is an application to manage files such as sending and receiving files between user and view files which the user receives. As a programmer, you are being asked to create an application to help them manage their order using **nachOS**.

First, this application will show the **computer number** (based on the network link address) and **3 main** **menus** consist of:

* **Send File**
* **View Existing File(s)**
* **Exit**

A close up of a logo

Description generated with very high confidence

**Figure 1. File Manager Main Menu**

1. **Send File**

If the user chooses **Menu 1** (**Send File**), then the application will:

* Ask the user to **input file name**. The **file name** must contain **dot** (‘**.**’). Dot **must not be** **in** **front** **of** or **in the end** of the **file name**.

A close up of a piece of paper

Description generated with high confidence

**Figure 2. Input file name**

* Ask the user to **input file size**. The file size must be between **1 and 500 Megabytes**.

A screenshot of a cell phone

Description generated with high confidence**Figure 3. Input file size**

* Ask the user to **input file type**. The file type must be between “**Word Document**”, “**Excel Worksheet**”, or “**Unspecified File**” (**case sensitive**).

**A screenshot of a cell phone

Description generated with high confidenceFigure 4. Input file type**

* Ask the user to input the receiver’s **computer name**.

**A screenshot of a cell phone

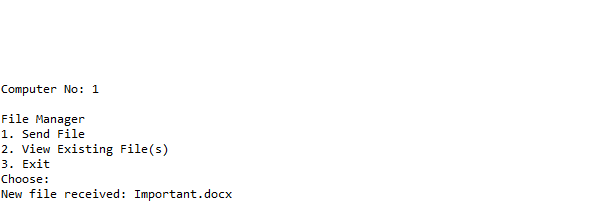
Description generated with very high confidenceFigure 5. Input computer name**

* After fulfilling all validations, the application will **send** the file and **show a message**.

**A screenshot of a cell phone

Description generated with very high confidenceFigure 6. Success Message after Insert**

* Then, a notification will appear in the receiver’s main menu.

**Figure 7. Receiver’s view after receiving the file**

1. **View Existing File(s)**

If the user chooses **Menu 2** (**View Existing File(s)**), then the application will:

* **Check** if the current user **has received a file**.If the user **has not received** any file, the application will **show error message**.

A close up of a logo

Description generated with very high confidence

**Figure 8. No Available File Error Message**

* Otherwise, if the current user **has available file(s)**, the application will **show all files received** using **KThread** and **Scheduler** (**FIFO concept**). The files **will be shown with delay 1000 milliseconds** (use **Thread sleep**).

A screenshot of a cell phone

Description generated with very high confidence

**Figure 9. View Existing Files Received**

1. **Exit**

If the user chooses **Menu 3** (**Exit**), the application will **print tick of time using Timer**. After that, the application will **exit**.

A close up of a logo

Description generated with very high confidence

**Figure 10. Ticks of Time Application has been running**