DHIVA KESUMA PERTIWI
0806022410025
DAILY TASK MANAGER

The process of creating this program begins with designing a basic concept, namely a list of tasks that can be updated, marked as completed, canceled, and features for adding and deleting tasks. To implement this feature, the program uses data structures such as arrays, Stacks, and LinkedLists to store and manage tasks.
In its implementation, this program displays an interactive menu that allows users to select various options such as viewing a list of tasks, updating tasks, marking tasks as completed, canceling tasks that have been marked as completed, adding and deleting tasks, and undoing the last changes. The input collection process is carried out using a Scanner, and the program is also equipped with input validation so that users can only enter numbers in the available menu.
One of the interesting features of this program is the cat animation that is displayed when the program is first run. This animation uses ASCII characters to provide a more interactive experience for users. In addition, the program also uses ANSI color codes to clarify task status, such as green for completed tasks and red for unfinished tasks.

Development Process
The development process begins with designing the data structures to be used, including selecting an array for the task list, a Stack for the undo and task completion features, and a LinkedList for the flexibility to add and delete tasks. Once the basic structure is determined, coding begins by building the main menu and basic features such as displaying the task list and marking tasks as complete.

Difficulties Encountered
During the development process, several difficulties were encountered, including:
Input Validation – Initially, the program did not have good input validation, so users could enter values other than numbers which caused errors. This was fixed by adding validation using hasNextInt() on the Scanner object.
Undo Implementation – One of the biggest challenges was implementing the undo feature. To overcome this problem, a Stack was used which stores a copy of the task list before changes were made, so users can revert to the previous state.
ANSI Display and Colors – The use of ANSI colors in the terminal is not always compatible with all systems, so testing is needed to ensure that colors are displayed correctly in various environments.
Optimal Data Structure – Choosing the right data structure is a challenge because each has its own advantages and disadvantages. Arrays are used for fixed task lists, Stacks for undo and tracking completed tasks, and LinkedLists allow flexible addition and removal of tasks.