

KELOMPOK 11



GO GOALS

PROJECT TEAM PROGRAMA KOMPUTER

DEVI MAULIDAH ROMADHONA (I0322037)

DIMAS ARYA SANTOSA (I0322038)

GHAISAN NABIEL ALFIAN ARSYAD (I0322050)

JOURDY LUKIANO MATTHEW SIRAIT (I0322060)

KHALIFAHTUR ALFI (I0322065)

2023



BAB I

DESKRIPSI MASALAH

Bab ini menjelaskan tentang latar belakang, tujuan program, dan manfaat dari program “Go Goals”

1.1 Latar Belakang

Dalam lingkungan perguruan tinggi, para mahasiswa sering menghadapi tantangan mengenai penyelesaian tugas yang tepat waktu. Meskipun mereka telah membuat daftar dengan tanggal jatuh tempo (*due date*) tugas-tugas mereka, seringkali mereka mengalami kesulitan dalam memenuhi tenggat waktu yang ditetapkan. Permasalahan ini dapat berdampak negatif terhadap kualitas pekerjaan mahasiswa, tingkat stres yang tinggi, dan pencapaian tujuan akademik yang tidak tercapai. Para mahasiswa juga seringkali menghadapi masalah dalam memanajemenkan sebuah *goals* atau tujuan yang ingin dicapai oleh mahasiswa tersebut.

1.2 Rumusan Masalah

Masalah yang kerap kali muncul adalah sebagai berikut:

1. Mahasiswa sering merasa kebingungan dalam perihal menyusun daftar-daftar *deadline* tugas perkuliahan secara urut dan *goals* yang ingin mereka capai.
2. Mahasiswa mengalami kurangnya visibilitas dan pemantauan kemajuan terhadap *deadline* tersebut sehingga mahasiswa tidak mengetahui seberapa jauh tingkat penyelesaian dari *deadline* tugas tersebut dan tidak mengetahui sejauh mana mereka telah mencapai target yang ditetapkan.
3. Mahasiswa kerap kali merasa kesulitan dalam menetapkan prioritas yang mana terdapat banyak pengumpulan tugas dan *goals* yang harus dicapai sehingga hal ini dapat mengarah pada rasa bingung, pekerjaan yang tertunda, dan ketidakmampuan untuk mencapai hasil yang diharapkan.
4. Terakhir, mahasiswa biasanya akan mengalami rasa kurangnya sebuah *reminder* atau pengingat untuk *deadline* tugas dan pencapaian *goals* mereka. Terkadang mahasiswa



dapat melewati tenggat waktu untuk pengumpulan tugas atau tidak terlaksananya *goals* yang ingin mereka raih karena kurangnya sebuah pengingat yang efektif.

1.3 Tujuan Program

Untuk menyelesaikan beberapa masalah yang dihadapi para mahasiswa sekarang terkait *deadline* tugas dan pencapaian *goals*, maka diperlukan sebuah program yang dapat menyelesaikan permasalahan tersebut. Program "*Go Goals*" adalah sebuah program yang dapat membantu para mahasiswa agar tugas dan goals yang ingin mereka capai dapat terselesaikan dengan tepat waktu, efektif, dan efisien. Program "*Go Goals*" ini juga akan membantu para mahasiswa untuk bisa:

- Menyusun daftar-daftar *deadline* tugas dan *goals* yang ingin dicapai mahasiswa.
- Memantau perkembangan mahasiswa dalam mengerjakan tugas dan *goals*.
- Menetapkan prioritas dalam memilih tugas mana yang harus dikerjakan terlebih dahulu dan *goals* apa yang ingin dicapai sehingga mendapatkan hasil yang diharapkan.
- Membuat sebuah "*Reminder*" atau pengingat *deadline* tugas-tugas dan *goals* mahasiswa agar tidak melewati tenggat waktu yang telah ditetapkan.

1.4 Manfaat Program

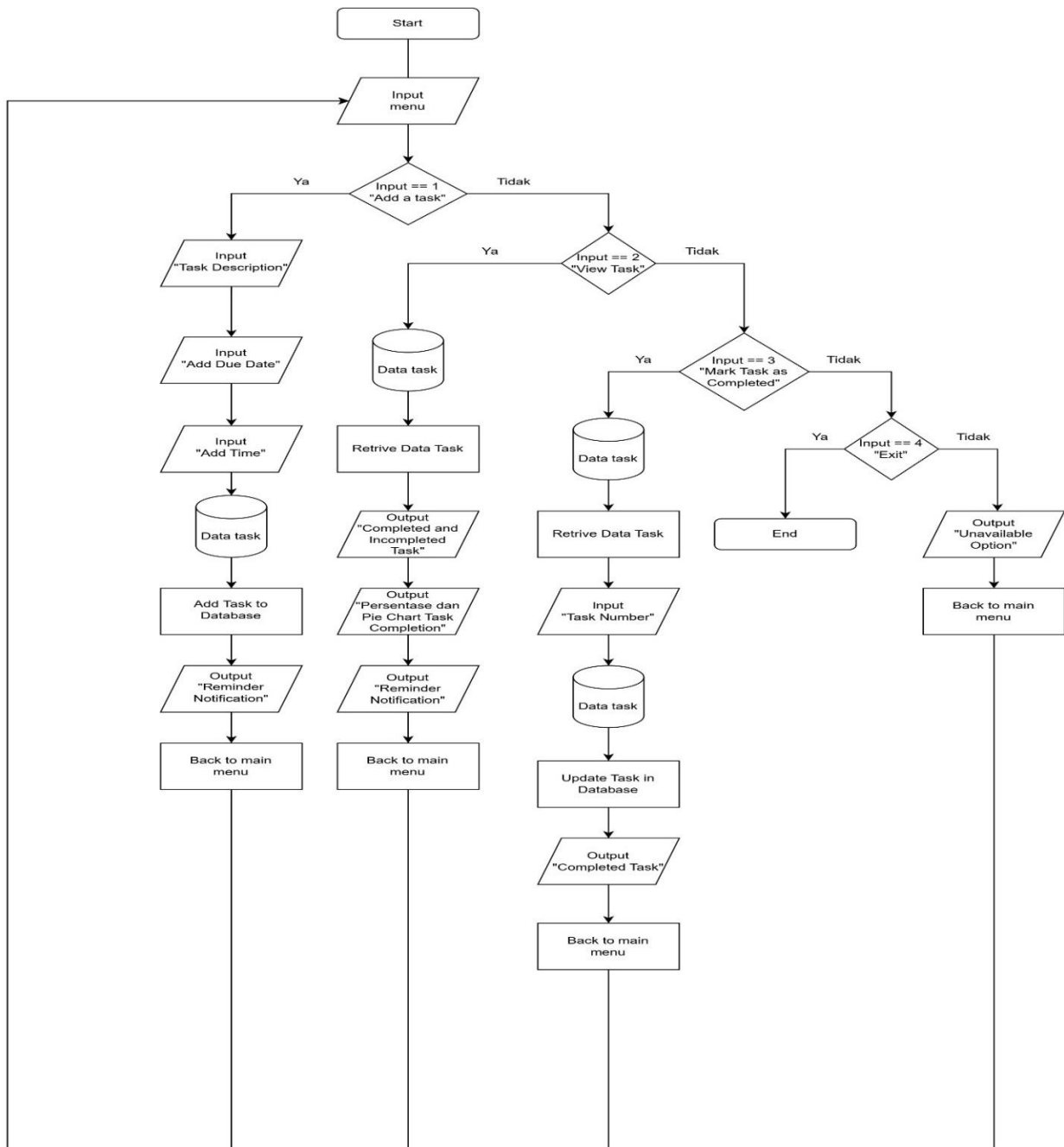
Penting bagi mahasiswa untuk mengatasi permasalahan ini agar dapat mencapai tujuan akademik mereka dengan lebih efektif. Hal ini melibatkan pengembangan keterampilan manajemen waktu, seperti membuat jadwal yang teratur, memprioritaskan tugas berdasarkan urgensi dan pentingnya, serta menghindari penundaan. Dengan dukungan program "*Go Goals*" dan upaya dari mahasiswa itu sendiri, diharapkan agar permasalahan terkait *deadline* tugas yang tertunda dan *goals-goals* yang tidak tercapai dapat dikurangi. Mahasiswa akan dapat meningkatkan efisiensi mereka dalam menyelesaikan tugas, mengurangi tingkat stres, dan mencapai tujuan akademik yang telah ditetapkan.



BAB II

FLOWCHART

Bab ini menjelaskan tentang penjelasan alur flowchart pada program python “Go Goals”.





Gambar 2.1 Flowchart program “Go Goals”

2.1 Flowchart “Go Goals”

Flowchart di atas ini menggambarkan langkah-langkah dalam program “Go Goals” untuk membantu mahasiswa dalam menyusun tugas, Menyusun goals yang ingin dicapai dan memberikan reminder pengerjaan. Berikut merupakan penjelasan *flowchart*:

1. Memulai program
2. Memulai *Loop* pada *Main Menu*:
 - Apakah pengguna memilih Menu 1 “Add a Task?”. Jika tidak maka lanjut ke Langkah 4.
 - Jika iya, minta pengguna untuk memasukkan deskripsi tugas.
 - Minta pengguna untuk memasukkan tanggal dan waktu *deadline*.
 - Data tugas akan masuk ke dalam *database* dalam bentuk csv.
 - Melalui data csv akan menghasilkan *output* berupa *reminder notification* 1 hari dan 1 jam sebelum *deadline* berakhir.
 - Keluar dari *Loop*.
3. Kembali ke *Main Menu*.
4. Memulai *Loop* pada *Main Menu*:
 - Apakah pengguna memilih Menu 2 “View Task and reminder?”. Jika tidak, maka lanjut ke Langkah 6.
 - Jika iya, program akan menampilkan data dari *database* csv berupa tabel tugas yang telah selesai (*completed*) dan tugas yang belum (*incompleted*), lengkap dengan keterangan deskripsi tugas, batas waktu, serta statusnya.
 - Setelah menampilkan tabel, selanjutnya menampilkan output berupa persentase dan pie chart.
 - Keluar dari *Loop*.
5. Kembali ke *Main Menu*.
6. Memulai *Loop* pada *Main Menu*:



- Apakah pengguna memilih Menu 3 “*Mark Task as Completed?*”. Jika tidak, maka lanjut ke Langkah 8.
 - Jika iya, program akan menampilkan data dari *database* csv berupa tabel tugas yang telah selesai (*completed*) dan tugas yang belum (*incompleted*), lengkap dengan keterangan deskripsi tugas, batas waktu, serta statusnya.
 - Pengguna diminta untuk memasukkan nomor tugas yang ingin ditandai selesai dan tanggal selesai.
 - Input dari pengguna akan masuk ke dalam *database* csv dan memperbarui data.
 - Program menampilkan *output* data yang sudah diperbarui.
 - Keluar dari *Loop*.
7. Kembali ke *Main Menu*.
8. Memulai *Loop* pada *Main Menu*:
- Apakah pengguna memilih Menu 4 “*Exit?*”. Jika tidak, maka lanjut ke Langkah 9.
 - Jika iya, program akan berhenti.
9. Menampilkan *output* “*Unavailable Option*”.
10. Keluar dari *Loop*.
11. Menampilkan *Main Menu*.
12. Selesai.



BAB III

KODE PROGRAM PYTHON

Bab ini menjelaskan tentang penjelasan modul yang digunakan, fungsi kode program, dan fitur yang digunakan pada program “Go Goals”.

3.1 Modul

Program “Go Goals” menggunakan beberapa modul yang di-import dalam menjalankan beberapa proses kode yang dipakai. Berikut modul yang digunakan:

```
import csv
import datetime
import time
import threading
from plyer import notification
from tabulate import tabulate
import matplotlib.pyplot as plt
```

Gambar 3.1 Modul

3.2 Fungsi Kode Program

Dalam menjalankan program, program “Go Goals” menggunakan beberapa fungsi kode yang dimasukkan. Berikut fungsi yang digunakan:

1) Fungsi `display_menu()`

Fungsi ini digunakan untuk menampilkan menu yang dapat dipilih oleh pengguna.

```
def display_menu():
    print("==== To-Do List =====")
    print("1. Add a task")
    print("2. View tasks and reminders")
    print("3. Mark task as completed")
    print("4. Exit")
```

Gambar 3.2 Fungsi `display_menu()`



2) Fungsi `save_task_to_csv()`

Fungsi ini digunakan untuk menyimpan “task” yang diinput oleh pengguna ke dalam file csv. Fungsi ini dilakukan dengan membuka file “tasks.csv” dalam mode “w”.

```
def save_tasks_to_csv():  
    with open('tasks.csv', mode='w', newline='') as file:  
        writer = csv.writer(file)  
        writer.writerow(['Description', 'Reminder Date', 'Reminder Time', 'Completed'])  
        for task in tasks:  
            writer.writerow([task['description'], task['reminder_date'], task['reminder_time'], task['completed']])
```

Gambar 3.3 Fungsi `save_task_to_csv()`

3) Fungsi `load_task_from_csv()`

Fungsi ini dilakukan untuk memuat “task” yang sebelumnya telah diinput oleh pengguna. Fungsi ini dilakukan dengan membuka file “tasks.csv” dalam mode “r”.

```
def load_tasks_from_csv():  
    tasks.clear()  
    try:  
        with open('tasks.csv', mode='r') as file:  
            reader = csv.DictReader(file)  
            for row in reader:  
                tasks.append({  
                    'description': row['Description'],  
                    'reminder_date': row['Reminder Date'],  
                    'reminder_time': row['Reminder Time'],  
                    'completed': row['Completed'] == 'True'  
                })  
    except FileNotFoundError:  
        pass
```

Gambar 3.4 Fungsi `load_task_from_csv()`

4) Fungsi `add_task()`

Fungsi ini memungkinkan pengguna untuk menginput deskripsi tugas, tanggal dan waktu pengumpulan. Pada fungsi ini, digunakan juga penanganan eksepsi yang



akan memberikan keterangan mengenai input yang dimasukkan pengguna *invalid* apabila tidak sesuai dengan format yang ditentukan.

```
def add_task():
    task = input("Enter task description: ")
    while True:
        try:
            date_str = input("Enter the due date (YYYY-MM-DD): ")
            datetime.datetime.strptime(date_str, "%Y-%m-%d")
            break
        except ValueError:
            print("Invalid date format. Please enter the date in YYYY-MM-DD format.")
    while True:
        try:
            time_str = input("Enter the time (HH:MM): ")
            datetime.datetime.strptime(time_str, "%H:%M")
            break
        except ValueError:
            print("Invalid time format. Please enter the time in HH:MM format.")
    reminder_datetime = datetime.datetime.strptime(date_str + " " + time_str, "%Y-%m-%d %H:%M")
    tasks.append({
        "description": task,
        "reminder_date": date_str,
        "reminder_time": time_str,
        "completed": False
    })
    print("Task added successfully!")
    threading.Thread(target=schedule_notification, args=(tasks[-1], reminder_datetime)).start()
    threading.Thread(target=schedule_notification2, args=(tasks[-1], reminder_datetime)).start()
```

Gambar 3.5 Fungsi add_task()

5) Fungsi schedule_notification(task, reminder_datetime)

Fungsi ini digunakan untuk menjadwalkan *reminder* yang akan ditampilkan kepada pengguna ketika tugas yang diinputkan oleh pengguna waktunya tersisa 1 jam.



```
def schedule_notification(task, reminder_datetime):
    current_datetime = datetime.datetime.now()
    time_difference = reminder_datetime - current_datetime
    remaining_time = time_difference.total_seconds()
    status = "Incomplete"

    if remaining_time > 0 and not task['completed']:
        notification_time = remaining_time - 3600 # Subtract 1 hour from remaining time

        if notification_time > 0:
            remaining_hours = int(remaining_time / 3600)
            remaining_minutes = int((remaining_time / 60) - (remaining_hours * 60))

            time.sleep(notification_time)
            notification.notify(
                title="Task Reminder",
                message=f"{task['description']} - {status} (Reminder in {remaining_hours} hours, {remaining_minutes} minutes)",
                timeout=30
            )
        )
```

Gambar 3.6 Fungsi `schedule_notification(task, reminder_datetime)`

6) Fungsi `schedule_notification2(task, reminder_datetime)`

Fungsi ini digunakan untuk menjadwalkan *reminder* yang akan ditampilkan kepada pengguna ketika tugas yang diinputkan oleh pengguna waktunya tersisa 1 hari.

```
def schedule_notification2(task, reminder_datetime):
    current_datetime = datetime.datetime.now()
    time_difference = reminder_datetime - current_datetime
    remaining_time = time_difference.total_seconds()
    status = "Incomplete"

    if remaining_time > 0 and not task['completed']:
        notification_time2 = remaining_time - 86400 # Subtract 1 day from remaining time

        if notification_time2 > 0:
            remaining_hours2 = int(remaining_time / 3600)
            remaining_minutes2 = int((remaining_time / 60) - (remaining_hours2 * 60))

            time.sleep(notification_time2)
            notification.notify(
                title="Task Reminder",
                message=f"{task['description']} - {status} (Reminder in {remaining_hours2} hours, {remaining_minutes2} minutes)",
                timeout=30
            )
        )
```

Gambar 3.7 Fungsi `schedule_notification2(task, reminder_datetime)`



7) Fungsi **display_tasks_table()**

Fungsi ini digunakan ketika pengguna ingin menampilkan tugas-tugas yang telah diinput sebelumnya. Fungsi ini menggunakan modul **tabulate** sehingga memungkinkan data berupa tugas ditampilkan dalam bentuk tabel.

```
def display_tasks_table():
    headers = ['Task Description', 'Due Date', 'Time', 'Status']
    rows = []
    sorted_tasks = sorted(tasks, key=lambda x: (x['reminder_date'], x['reminder_time']))

    for index, task in enumerate(sorted_tasks, start=1):
        status = "Completed" if task["completed"] else "Incomplete"
        rows.append([index, task['description'], task['reminder_date'], task['reminder_time'], status])

    table = tabulate(rows, headers=headers, tablefmt='grid')
    print(table)
```

Gambar 3.8 Fungsi **display_tasks_table()**

8) Fungsi **view_tasks_and_reminders()**

Fungsi ini digunakan untuk menampilkan tugas dan *reminder*. Fungsi ini memanggil fungsi **display_task_table()** yang telah didefinisikan sebelumnya.

```
def view_tasks_and_reminders():
    display_tasks_table()
    input("Press Enter to continue...")
```

Gambar 3.9 Fungsi **view_tasks_and_reminders()**

9) Fungsi **mark_task_as_completed()**

Fungsi ini akan memanggil fungsi **display_task_table()** yang kemudian pengguna akan menginput nomor dari “task” yang akan ditandai sebagai “completed”. Pada fungsi ini, digunakan juga penanganan eksepsi yang apabila



pengguna menginput dalam bentuk bukan angka, maka pada terminal akan tertulis *“Invalid input. Please enter a valid task number.”*

```
def mark_task_as_completed():
    display_tasks_table()
    while True:
        try:
            task_number = int(input("Enter the task number to mark as completed: "))
            sorted_tasks = sorted(tasks, key=lambda x: (x['reminder_date'], x['reminder_time']))
            if 1 <= task_number <= len(sorted_tasks):
                sorted_tasks[task_number - 1]["completed"] = True
                tasks[tasks.index(sorted_tasks[task_number - 1])]["completed"] = True
                print("Task marked as completed!")
                break
            else:
                print("Invalid task number.")
        except ValueError:
            print("Invalid input. Please enter a valid task number.")
```

Gambar 3.10 Fungsi mark_task_as_completed()

10) Fungsi calculate_task_completion_percentage()

Fungsi kode ini digunakan untuk menghitung persentase penyelesaian tugas dan persentase tugas yang belum selesai. Fungsi ini mengasumsikan bahwa ada variabel *"tasks"* yang berisi daftar tugas yang akan dianalisis serta memberi informasi tentang sejauh mana penyelesaian tugas yang telah dilakukan, dan persentase tugas yang masih harus diselesaikan.

```
def calculate_task_completion_percentage():
    total_tasks = len(tasks)
    completed_tasks = sum(1 for task in tasks if task['completed'])
    incomplete_tasks = total_tasks - completed_tasks

    if total_tasks == 0:
        complete_percentage = 0
        incomplete_percentage = 0
    else:
        complete_percentage = (completed_tasks / total_tasks) * 100
        incomplete_percentage = (incomplete_tasks / total_tasks) * 100

    return complete_percentage, incomplete_percentage
```



Gambar 3.11 Fungsi `calculate_task_completion_percentage()`

11) Fungsi `display_task_completion_graph (complete_percentage, incomplete_percentage)`

Fungsi ini digunakan untuk menampilkan grafik lingkaran dari tugas yang sudah ditandai sebagai “*completed*” dan tugas yang masih “*incomplete*”. Untuk menampilkan grafik lingkaran tersebut, digunakan modul **matplotlib.pyplot**.

```
def display_task_completion_graph(complete_percentage, incomplete_percentage):
    labels = ['Complete', 'Incomplete']
    sizes = [complete_percentage, incomplete_percentage]
    colors = ['#1f77b4', '#ff7f0e']

    plt.pie(sizes, labels=labels, colors=colors, autopct='%1.1f%%', startangle=90)
    plt.axis('equal')
    plt.title('Task Completion')
    plt.show()
```

Gambar 3.12 Fungsi `display_task_completion_graph (complete_percentage, incomplete_percentage)`

12) Fungsi `overdue_tasks()`

Fungsi ini digunakan untuk mencari dan memberikan notifikasi terhadap tugas-tugas yang sudah lewat batas waktu pengerjaannya (*overdue tasks*).

```
def overdue_tasks():
    current_datetime = datetime.datetime.now()
    status = 'The deadline has been passed'
    overdue_tasks = [task for task in tasks if not task['completed'] and datetime.datetime.strptime(task['reminder_date'] + ' ' + task['reminder_time'], "%Y-%m-%d %H:%M") < current_datetime]
    if overdue_tasks:
        for task in overdue_tasks:
            notification.notify(
                title="Overdue Task",
                message=f"{task['description']} - {status}",
                timeout=30
            )
```

Gambar 3.13 Fungsi `overdue_tasks()`

13) Fungsi `on_going_task()`

Fungsi ini digunakan untuk memilah “*task*” yang belum ditandai sebagai “*completed*” dan “*task*” tersebut belum melewati batas waktu yang telah ditentukan



untuk kemudian dikirimkan notifikasi *reminder* ketika pengguna memilih opsi ke - 2 pada fungsi **main()**. Fungsi ini memanggil fungsi **notify_task_reminder()** untuk memungkinkan notifikasi dikirimkan ke pengguna.

```
def on_going_task():
    current_datetime = datetime.datetime.now()
    for index, task in enumerate(tasks, start=1):
        if not task['completed']:
            status = "Incomplete"
            reminder_datetime = datetime.datetime.strptime(task['reminder_date'] + " " + task['reminder_time'], "%Y-%m-%d %H:%M")
            time_difference = reminder_datetime - current_datetime
            remaining_time = time_difference.total_seconds()
            if remaining_time <= 0:
                pass
            else:
                remaining_hours = int(remaining_time / 3600)
                remaining_minutes = int(remaining_time / 60 - remaining_hours * 60)
                notify_task_reminder(task['description'], status, remaining_hours, remaining_minutes)
```

Gambar 3.14 Fungsi **on_going_task()**

14) Fungsi **notify_task_reminder()**

Fungsi ini digunakan untuk mengirimkan notifikasi kepada pengguna. Fungsi ini menggunakan **plyer** untuk memungkinkan notifikasi masuk ke pengguna.

```
def notify_task_reminder(description, status, hours, minutes):
    notification.notify(
        title="Task Reminder",
        message=f"{description} - {status} (Reminder in {hours} hours, {minutes} minutes)",
        timeout=30 # Durasi notifikasi dalam detik
    )
```

Gambar 3.15 Fungsi **notify_task_reminder()**

15) Fungsi **main()**

Fungsi **main()** digunakan untuk titik masuk dari program. Di dalam fungsi ini terdapat perulangan berupa “*while True*” yang akan menjalankan program sampai program diminta untuk berhenti oleh pengguna dengan memilih opsi 4 pada menu. Fungsi ini memanggil **load_tasks_from_csv()** yang bertujuan untuk memuat “*tasks*” dari file csv. Di setiap iterasinya, program akan menampilkan menu



menggunakan fungsi **display_menu()**, dan pengguna akan diminta untuk menginput opsi yang dipilih.

Jika pengguna memilih opsi 1, maka program akan menjalankan fungsi **add_task()** yang memungkinkan pengguna untuk menambahkan tugas baru, yang kemudian akan disimpan dalam file csv dengan memanggil fungsi **save_task_to_csv()**.

Jika pengguna memilih opsi 2, maka program akan menjalankan fungsi **view_tasks_and_reminders()** untuk menampilkan tugas dan reminder yang telah diinput. Setelah itu, program akan menjalankan fungsi **calculate_task_completion_percentage()** yang akan menampilkan persentase tugas yang telah selesai dan belum selesai dan hasilnya akan di-*print*. Setelah itu, program akan menjalankan fungsi **display_task_completion_graph()** yang menampilkan grafik persentase penyelesaian tugas dalam bentuk grafik lingkaran. Kemudian, program akan menjalankan fungsi **overdue_task** (yang akan mengirimkan notifikasi kepada pengguna mengenai tugas yang belum selesai dan sudah melewati batas waktu) dan **on_going_task** (yang akan mengirimkan notifikasi kepada pengguna mengenai tugas yang belum selesai dan belum melewati batas waktu yang telah ditentukan).

Jika pengguna memilih opsi 3, program akan menjalankan fungsi **mark_task_as_completed()** yang akan menandai tugas yang sudah selesai dan kemudian program akan menjalankan fungsi **save_task_to_csv()** untuk menyimpan tugas yang telah diubah statusnya menjadi “completed” ke dalam file csv.

Jika pengguna memilih opsi 4, program akan menjalankan fungsi **save_task_to_csv** untuk menyimpan daftar tugas terbaru dan kemudian *Loop* “*while True*” akan berhenti karena dilakukan **break**, sehingga program berakhir.



Jika pengguna memilih opsi di luar 1-4, maka program akan melakukan print *“Invalid choice, Please try again”* dan kembali ke awal *Loop*.

```
def main():
    load_tasks_from_csv()

    while True:
        display_menu()
        choice = input("Enter your choice (1-4): ")
        if choice == "1":
            add_task()
            save_tasks_to_csv()
        elif choice == "2":
            load_tasks_from_csv()
            view_tasks_and_reminders()
            complete_percentage, incomplete_percentage = calculate_task_completion_percentage()
            print("-----Task Completion-----")
            print(f"Complete: {complete_percentage:.2f}%")
            print(f"Incomplete: {incomplete_percentage:.2f}%")
            display_task_completion_graph(complete_percentage, incomplete_percentage)
            overdue_tasks()
            on_going_task()
        elif choice == "3":
            mark_task_as_completed()
            save_tasks_to_csv()
        elif choice == "4":
            save_tasks_to_csv()
            break
        else:
            print("Invalid choice. Please try again.")

    print()

if __name__ == "__main__":
    main()
```

Gambar 3.16 Fungsi main ()

3.3 Fitur Program

```
===== To-Do List =====
1. Add a task
2. View tasks and reminders
3. Mark task as completed
4. Exit
Enter your choice (1-4):
```

Gambar 3.17 Fitur program *“Go Goals”*



Go Goals memiliki 4 fitur utama, yaitu “*Add a Task*”, “*View Tasks and Reminder*”, “*Mark Task as Completed*”, dan “*Exit*”. Berikut penjelasan dan keunikannya:

1) *Add a Task*

Untuk memunculkan fitur ini, pengguna dapat memasukkan opsi angka 1 pada perintah “*Enter your choice (1-4)*”. Kemudian pengguna akan diminta untuk memasukkan deskripsi tugas, batas tanggal dan waktu yang ingin di-*input*.

```
Enter your choice (1-4): 1
Enter task description: Tubes Prokom
Enter the due date (YYYY-MM-DD): 2023-06-19
Enter the time (HH:MM): 00:00
Task added successfully!
```

Gambar 3.18 Tampilan fitur *add a task*

Keunikannya pada fitur ini adalah telah menggunakan eksepsi, di mana saat pengguna salah dalam menginput format tanggal dan waktu maka akan muncul keterangan “*Invalid date/time format*”, kemudian pengguna akan diminta untuk meng-*input* informasi yang benar dan sesuai format. Apabila berhasil akan muncul keterangan “*Task added successfully*”.

```
Enter your choice (1-4): 1
Enter task description: Tubes Prokom
Enter the due date (YYYY-MM-DD): 233333
Invalid date format. Please enter the date in YYYY-MM-DD format.
Enter the due date (YYYY-MM-DD): 2023-06-17
Enter the time (HH:MM): 213
Invalid time format. Please enter the time in HH:MM format.
Enter the time (HH:MM):
```

Gambar 3.19 Tampilan ketika pengguna salah memasukkan *input*

Selain itu, setelah pengguna menginput data tugas, data akan langsung disimpan pada *file* csv. Hal ini bertujuan untuk memudahkan pemanggilan pada notifikasi, tabel, dan persentase serta mencegah hilangnya data saat pengguna meninggalkan program.



```
tasks.csv M x
tasks.csv
1 Description,Reminder Date,Reminder Time,Completed
2 Tubes Prokom,2023-06-18,01:03,False
3 Tugas Kalkulus,2023-06-17,02:05,True
4 Tugas PRD,2023-06-16,00:00,False
5 Tubes Prokom,2023-06-19,00:00,False
6
```

3.20 Tampilan tasks.csv

2) View Tasks and Reminder

Untuk memunculkan fitur ini, pengguna dapat memasukkan opsi angka 2 pada perintah “Enter your choice (1-4)”. Kemudian akan muncul tabel yang menampilkan lengkap deskripsi tugas, batas tanggal dan waktu yang telah pengguna input serta status *completed* atau *incomplete*. Keunikannya adalah tugas dengan *due date* terdekat akan secara otomatis berada pada tabel paling atas sedangkan tugas dengan *due date* terjauh berada pada tabel paling bawah. Hal ini akan membantu pengguna untuk menentukan prioritas tugas yang harus dikerjakan.

	Task Description	Due Date	Time	Status
1	Tugas PRD	2023-06-16	00:00	Completed
2	Tugas Kalkulus	2023-06-17	02:05	Completed
3	Tugas Mekanika Teknik	2023-06-17	12:23	Incomplete
4	Tubes Prokom	2023-06-18	01:03	Incomplete
5	Tugas Prokom	2023-06-18	12:16	Incomplete
6	Tugas Laporan Prokom	2023-06-18	12:21	Incomplete

Gambar 3.21 Tampilan fitur *view tasks and reminder*

Selain itu, keunikan pada fitur ini ada setelah pengguna mengklik tombol *enter*. Akan muncul persentase tugas *task completion* berupa tulisan dan grafik *pie chart*. Hal ini akan membantu pengguna untuk mengetahui seberapa jauh *progress* tugas yang sudah dan harus dikerjakan.



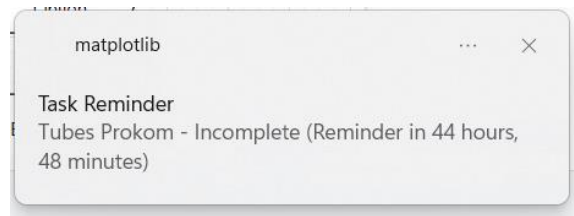
```
-----Task Completion-----
Complete: 33.33%
Incomplete: 66.67%
```

Gambar 3.22 Tampilan persentase berupa teks

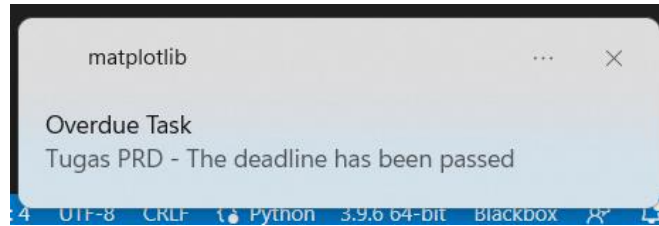


Gambar 3.23 Tampilan persentase berupa pie chart

Untuk mengingatkan pengguna, setelah menutup grafik *pie chart* akan muncul *Pop Up* notifikasi berupa tugas dengan status *incomplete* atau belum selesai. Untuk tugas yang belum melewati batas waktu akan ditampilkan dengan keterangan *message* deskripsi tugas, status, dan sisa waktu dalam jam dan menit. Sedangkan untuk tugas yang telah melewati batas waktu akan ditampilkan keterangan *message* deskripsi tugas dan status “*This deadline has been passed*”.



Gambar 3.24 Tampilan *Pop Up* notifikasi tugas *incomplete*



Gambar 3.25 Tampilan *Pop Up* notifikasi ketika tugas telah lewat

3) *Mark Task as Completed*

Untuk memunculkan fitur ini, pengguna dapat memasukkan opsi angka 3 pada perintah “*Enter your choice (1-4)*”. Kemudian akan muncul tabel yang menampilkan lengkap deskripsi tugas, batas tanggal dan waktu yang telah pengguna *input* serta status *completed* atau *incomplete*. Keunikan pada fitur ini adalah dapat mengubah status tugas dari *incomplete* menjadi *completed* hanya dengan memasukkan nomor tugas yang ingin diubah pada perintah “*Enter the task number to mark as completed:*”.

```
Enter your choice (1-4): 3
```

	Task Description	Due Date	Time	Status
1	Tugas PRD	2023-06-16	00:00	Completed
2	Tugas Kalkulus	2023-06-17	02:05	Completed
3	Tugas Mekanika Teknik	2023-06-17	12:23	Incomplete
4	Tubes Prokom	2023-06-18	01:03	Incomplete
5	Tugas Prokom	2023-06-18	12:16	Incomplete
6	Tugas Laporan Prokom	2023-06-18	12:21	Incomplete

```
Enter the task number to mark as completed: 
```

Gambar 3.26 Tampilan fitur *mark task as completed*

Setelah memasukkan nomor tugas yang ingin diubah, data secara otomatis akan diubah pada database csv. Sehingga akan merubah status pada tabel dan menghambat notifikasi untuk muncul.

```
Enter the task number to mark as completed: 1
```

3	Tubes Prokom	2023-06-18	01:03	Incomplete
4	Tubes Prokom	2023-06-19	00:00	Incomplete

```
Task marked as completed!
```



Gambar 3.27 Tampilan tugas yang telah ditandai selesai

Apabila pengguna salah memasukkan nomor tugas, maka akan muncul keterangan “*Invalid task number*”. Program kemudian akan meminta pengguna untuk memasukkan nomor tugas yang benar.

3	Tubes Prokom	2023-06-18	01:03	Incomplete
4	Tubes Prokom	2023-06-19	00:00	Incomplete

Enter the task number to mark as completed: 12
Invalid task number.
Enter the task number to mark as completed: █

Gambar 3.28 Tampilan pengguna salah menginput nomor tugas

4) *Exit*

Untuk memunculkan fitur ini, pengguna dapat memasukkan opsi angka 4 pada perintah “*Enter your choice (1-4)*”. Maka program secara otomatis akan berhenti.

```
===== To-Do List =====
1. Add a task
2. View tasks and reminders
3. Mark task as completed
4. Exit
Enter your choice (1-4): 4
PS C:\prokom\I0322037_Devi-Maulidah-Romadhona_Praktikum-Prokom-2023---Kelompok-11>
```

Gambar 3.29 Tampilan fitur *exit*



BAB IV

HASIL RUNNING PYTHON

Bab ini menjelaskan tentang hasil *running* program python “Go Goals” berupa output yang akan dijabarkan di bawah ini.

4.1 Hasil *Running* Program

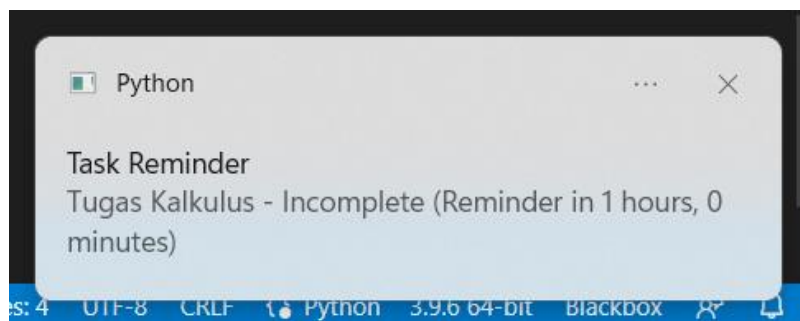
A. *Pop Up* Notifikasi

1) Notifikasi 1 jam sebelum *due date*

Output berupa notifikasi yang muncul 1 jam sebelum *due date* yang telah di *input*. Notifikasi hanya akan keluar apabila *task incomplete* atau belum selesai. Notifikasi muncul dengan keterangan *message* deskripsi tugas, status, serta sisa waktu dalam jam dan menit.

```
===== To-Do List =====  
1. Add a task  
2. View tasks and reminders  
3. Mark task as completed  
4. Exit  
Enter your choice (1-4): 1  
Enter task description: Tugas Kalkulus  
Enter the due date (YYYY-MM-DD): 2023-06-17  
Enter the time (HH:MM): 02:05  
Task added successfully!
```

Gambar 4.1 Tampilan ketika *task* telah ditambahkan



Gambar 4.2 Tampilan *Pop Up* notifikasi 1 jam sebelum *due date*

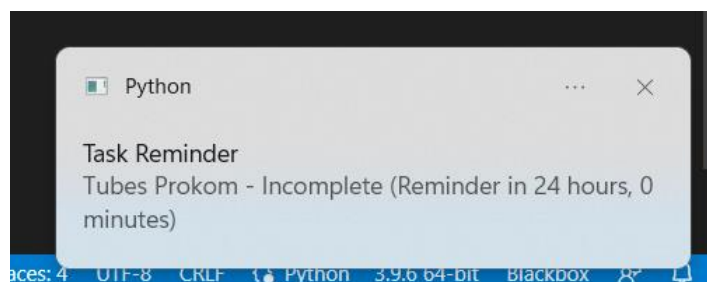


2) Notifikasi 1 hari sebelum *due date*

Output berupa notifikasi yang muncul 1 hari sebelum *due date* yang telah di *input*. Notifikasi hanya akan keluar apabila *task incomplete* atau belum selesai. Notifikasi muncul dengan keterangan *message* deskripsi tugas, status, serta sisa waktu dalam jam dan menit.

```
===== To-Do List =====
1. Add a task
2. View tasks and reminders
3. Mark task as completed
4. Exit
Enter your choice (1-4): 1
Enter task description: Tubes Prokom
Enter the due date (YYYY-MM-DD): 2023-06-18
Enter the time (HH:MM): 01:03
Task added successfully!
```

Gambar 4.3 Tampilan ketika *task* telah ditambahkan



Gambar 4.4 Tampilan *Pop Up* notifikasi 1 hari sebelum *due date*

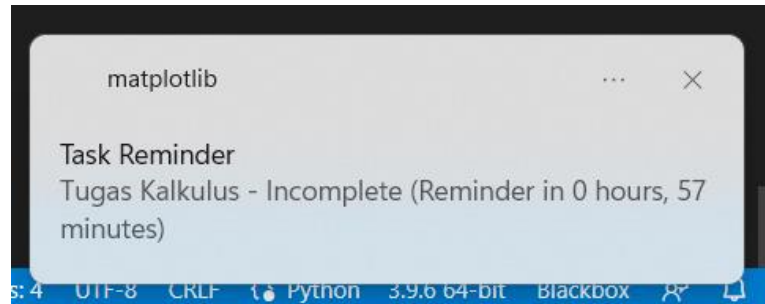
3) Notifikasi saat membuka opsi “*View Tasks and Reminder*”

Output berupa *pop up* notifikasi semua tugas yang *incomplete* atau belum selesai. Notifikasi hanya akan muncul apabila *user* memilih opsi 2 “*View Tasks and Reminder*”. Terdapat dua jenis notifikasi yang akan muncul, yaitu *on going task* dan *overdue task*.

On going task merupakan notifikasi untuk tugas yang belum melewati batas waktu atau *due date* yang diinput oleh *user*. Notifikasi juga disertai dengan

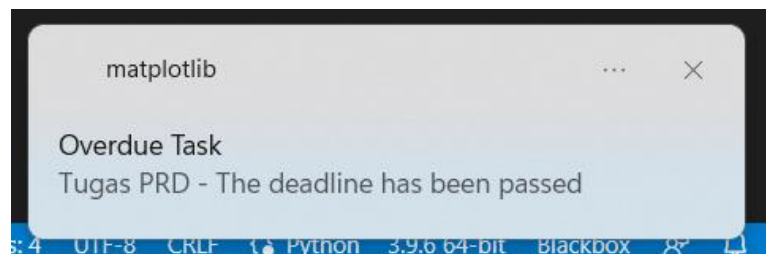


keterangan *message* berupa deskripsi tugas, status, dan sisa waktu dalam jam dan menit.



Gambar 4.5 Tampilan notifikasi muncul ketika memilih menu *View Task and Reminder*

Overdue task merupakan notifikasi untuk tugas yang telah melewati batas waktu atau *due date* yang diinput oleh *user*. Notifikasi juga disertai dengan keterangan *message* berupa deskripsi tugas dan status “*The deadline has been passed*”.



Gambar 4.6 Tampilan notifikasi ketika *deadline* telah melebihi batas waktu

B. Tabel

Output berupa tabel yang akan muncul saat *user* memilih opsi 2 “*View Tasks and Reminder*”. Tabel dilengkapi dengan deskripsi tugas, batas tanggal dan waktu, serta status *completed* atau *incomplete*.



	Task Description	Due Date	Time	Status
1	Tugas PRD	2023-06-16	00:00	Completed
2	Tugas Kalkulus	2023-06-17	02:05	Completed
3	Tugas Mekanika Teknik	2023-06-17	12:23	Incomplete
4	Tubes Prokom	2023-06-18	01:03	Incomplete
5	Tugas Prokom	2023-06-18	12:16	Incomplete
6	Tugas Laporan Prokom	2023-06-18	12:21	Incomplete

Gambar 4.7 Tampilan daftar tugas di dalam tabel

C. Persentase dan *Pie Chart*

Output berupa persentase jumlah tugas dengan status *completed* dan *incomplete*. *Output* akan muncul ketika *user* mengklik tombol *enter* setelah muncul tabel pada opsi 2 “*View Tasks and Reminder*”. Persentase akan ditampilkan dalam bentuk tulisan dan *pie chart*.

```
-----Task Completion-----
Complete: 33.33%
Incomplete: 66.67%
```

Gambar 4.8 Tampilan persentase *complete* dan *incomplete*



Gambar 4.9 Tampilan *pie chart*

4.2 Kelebihan dan Kekurangan Program

A. Kelebihan

Keunggulan atau kelebihan dari program “*Go Goals*” ini adalah sebagai berikut:

- Program dilengkapi dengan notifikasi yang otomatis muncul ketika batas waktu kurang dari 1 jam dan 1 hari untuk tugas dengan status *incomplete* (belum selesai).
- Program telah dilengkapi dengan eksepsi.
- Data pada program ini tersimpan dalam file csv, sehingga tidak akan hilang ketika pengguna keluar dari program.

B. Kekurangan

Kekurangan dari program “*Go Goals*” ini adalah sebagai berikut:

- Tampilan masih berbentuk program, sehingga belum *user friendly*.
- Notifikasi akan hilang dalam 10 detik dan tidak tersimpan pada bar notifikasi walaupun pengguna belum mengkliknya.
- Notifikasi masih belum memiliki suara yang dimodifikasi.