

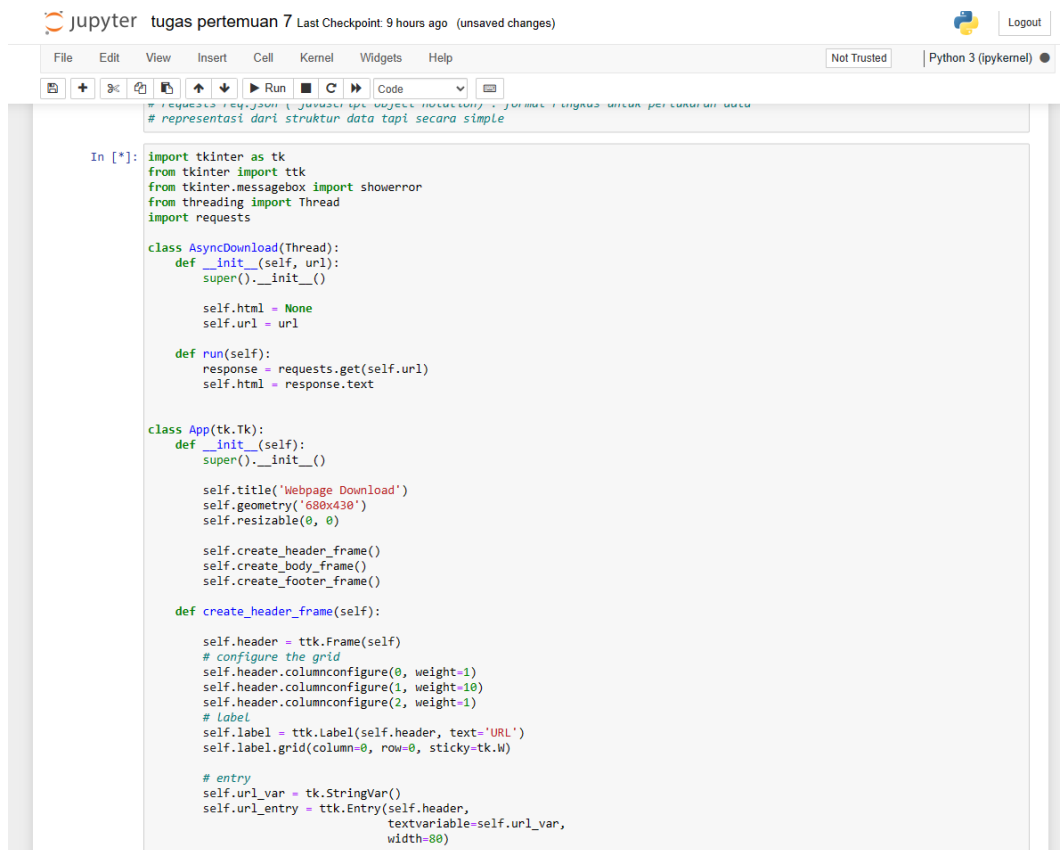
Nama : Muhammad Tarmidzi Bariq

NPM : 51422161

Kelas : 1IA13

Tanggal : 11/5/2023

## Tugas pertemuan 7



The screenshot shows a Jupyter Notebook window titled "tugas pertemuan 7" with a last checkpoint of 9 hours ago. The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help), a toolbar with icons for file operations and execution, and a status bar indicating "Not Trusted" and "Python 3 (ipykernel)". The code cell contains the following Python code:

```
In [*]: import tkinter as tk
from tkinter import ttk
from tkinter.messagebox import showerror
from threading import Thread
import requests

class AsyncDownload(Thread):
    def __init__(self, url):
        super().__init__()

        self.html = None
        self.url = url

    def run(self):
        response = requests.get(self.url)
        self.html = response.text

class App(tk.Tk):
    def __init__(self):
        super().__init__()

        self.title('Webpage Download')
        self.geometry('680x430')
        self.resizable(0, 0)

        self.create_header_frame()
        self.create_body_frame()
        self.create_footer_frame()

    def create_header_frame(self):

        self.header = ttk.Frame(self)
        # configure the grid
        self.header.columnconfigure(0, weight=1)
        self.header.columnconfigure(1, weight=10)
        self.header.columnconfigure(2, weight=1)
        # Label
        self.label = ttk.Label(self.header, text='URL')
        self.label.grid(column=0, row=0, sticky=tk.W)

        # entry
        self.url_var = tk.StringVar()
        self.url_entry = ttk.Entry(self.header,
                                   textvariable=self.url_var,
                                   width=80)
```

```
File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

self.url_entry.grid(column=1, row=0, sticky=tk.EW)

# download button
self.download_button = ttk.Button(self.header,
                                   text='Download')
self.download_button['command'] = self.handle_download
self.download_button.grid(column=2, row=0, sticky=tk.E)

# attach the header frame
self.header.grid(column=0, row=0, sticky=tk.NSEW, padx=10, pady=10)

def handle_download(self):
    url = self.url_var.get()
    if url:
        self.download_button['state'] = tk.DISABLED
        self.html.delete(1.0, "end")

        download_thread = AsyncDownload(url)
        download_thread.start()

        self.monitor(download_thread)
    else:
        showerror(title='Error',
                  message='Masukkan URL Webpage')

def monitor(self, thread):
    if thread.is_alive():
        self.after(100, lambda: self.monitor(thread))
    else:
        self.html.insert(1.0, thread.html)
        self.download_button['state'] = tk.NORMAL

def create_body_frame(self):
    self.body = ttk.Frame(self)
    self.html = tk.Text(self.body, height=20)
    self.html.grid(column=0, row=1)

    scrollbar = ttk.Scrollbar(self.body,
                              orient='vertical',
                              command=self.html.yview)

    scrollbar.grid(column=1, row=1, sticky=tk.NS)
    self.html['yscrollcommand'] = scrollbar.set
    self.body.grid(column=0, row=1, sticky=tk.NSEW,
                    padx=10, pady=10)

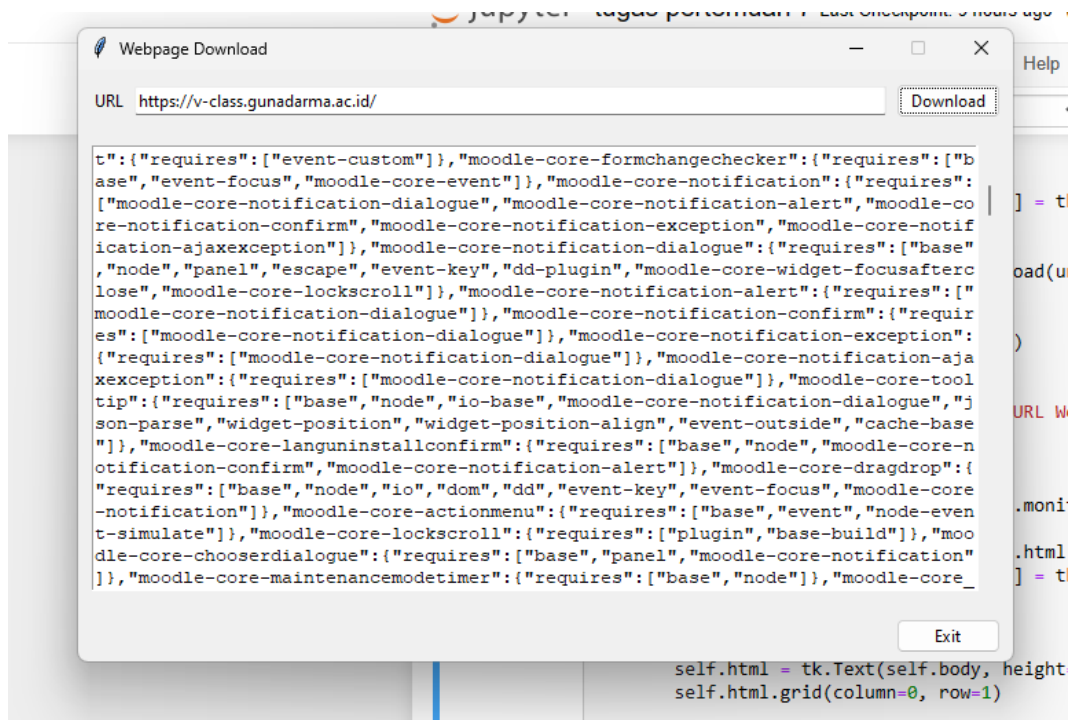
def create_footer_frame(self):
    self.footer = ttk.Frame(self)
    self.footer.columnconfigure(0, weight=1)
    self.exit_button = ttk.Button(self.footer
```

```
        self.exit_button.grid(column=0, row=0, sticky=tk.E)
    self.footer.grid(column=0, row=2, sticky=tk.NSEW, padx=10, pady=10)

if __name__ == "__main__":
    app = App()
    app.mainloop()
```

In [ ]:

## Output



## Logika pemograman

Pertama, program mengimpor modul-modul yang dibutuhkan yaitu “tkinter” , “requests” ,

“Thread” , dan “tkk. Tkinter”

Requests digunakan untuk mengambil data dari web

Thread digunakan untuk membuat proses sekuensial dan asynchronous

Program ini terdiri dari tiga bagian utama yaitu header frame, body frame, dan footer frame.

Pada header frame, terdapat label yang menunjukkan teks "URL", sebuah entry untuk memasukkan URL, dan sebuah tombol "Download". Ketika tombol "Download" ditekan, program akan memanggil fungsi handle\_download() untuk men-download halaman web dan menampilkan konten halaman tersebut.

Pada fungsi handle\_download(), URL yang dimasukkan oleh pengguna akan diambil dari entry menggunakan metode get() dan dimasukkan ke dalam sebuah

objek thread baru dari kelas AsyncDownload. Objek thread tersebut kemudian dimulai menggunakan metode start(). Ketika thread masih hidup (dalam keadaan belum selesai), program akan memanggil fungsi monitor() secara terus-menerus menggunakan metode after() untuk memantau kemajuan proses download. Ketika thread sudah selesai, konten halaman web akan ditampilkan di dalam sebuah textbox di body frame.

Pada body frame, terdapat sebuah textbox untuk menampilkan konten halaman web. Terdapat juga sebuah scrollbar untuk memudahkan pengguna dalam mengakses teks yang panjang.

Pada footer frame, terdapat sebuah tombol "Exit" yang akan menutup aplikasi jika ditekan.

Kelas AsyncDownload merupakan kelas turunan dari kelas Thread yang digunakan untuk men-download halaman web secara asynchronous. Ketika objek thread tersebut dimulai dengan metode start(), program akan menggunakan library requests untuk mengambil konten halaman web dari URL yang diberikan. Konten halaman web tersebut kemudian disimpan dalam sebuah atribut html pada objek thread.