LAPORAN PEMBUATAN TUGAS ALGORITMA DAN PEMROGRAMAN 1B DENGAN PRODUK GAME TEBAK ANGKA MENGGUNAKAN GUI



Dibuat oleh :

1. Ari Dermawan (50423201) (1IA02)
2. Farras Hudafi Susanto (50423462) (1IA02)

UNIVERSITAS GUNADARMA

FAKULTAS TEKNIK INDUSTRI

PRODI INFORMATIKA

Jl. Margonda Raya 100 - Depok

DAFTAR PUSTAKA

[**1. KASUS** 1](#_Toc152533309)

[**2. PEMBAGIAN TUGAS** 1](#_Toc152533310)

[**3. SOLUSI** 1](#_Toc152533311)

[**4. DESIGN** 1](#_Toc152533312)

[**5. CODING** 4](#_Toc152533313)

[**6. OUTPUT** 7](#_Toc152533314)

[**7. UJI COBA** 10](#_Toc152533315)

[**8. DOKUMENTASI** 11](#_Toc152533316)

[**9. KESIMPULAN DAN SARAN** 15](#_Toc152533317)

# **1. KASUS**

Video game pertama kali muncul pada tahun 1960. Dalam beberapa tahun belakangan video games sangat popular, hingga valuasinya naik terus menerus dari tahun ke tahun. Semakin kesini, video game menjadi sebuah kebutuhan untuk generasi baru. Oleh karena itu kami membuat video game simple mengguankan Bahasa pemrograman python yang berjudul “Guess The Number”

# **2. PEMBAGIAN TUGAS**

Ari Dermawan :

* Project Leader
* Pembuat Laporan
* Pembuat PPT
* Error Tester / Bug Finder
* Presenter

Farras Hudafi Susanto :

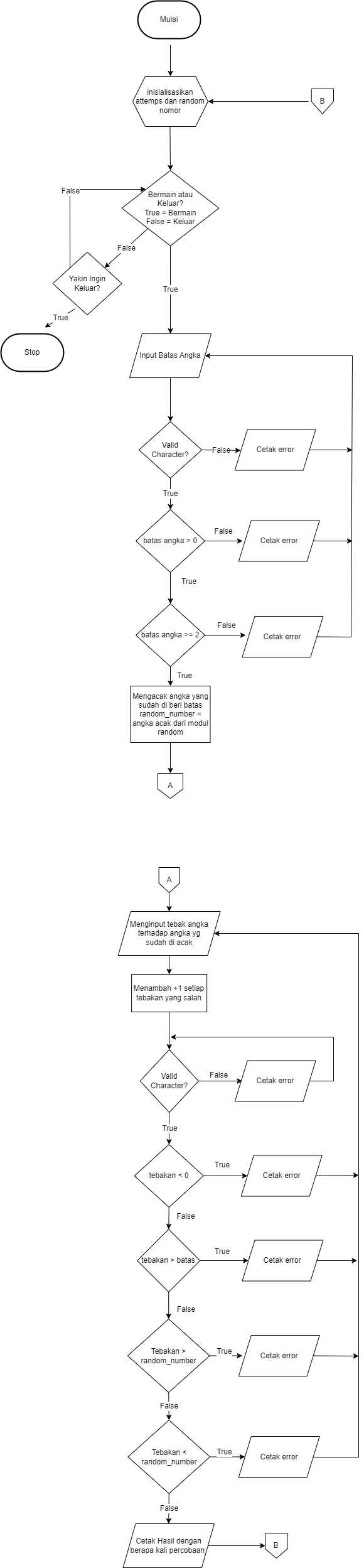
* Coder
* Main Idea
* Flowchart Designer

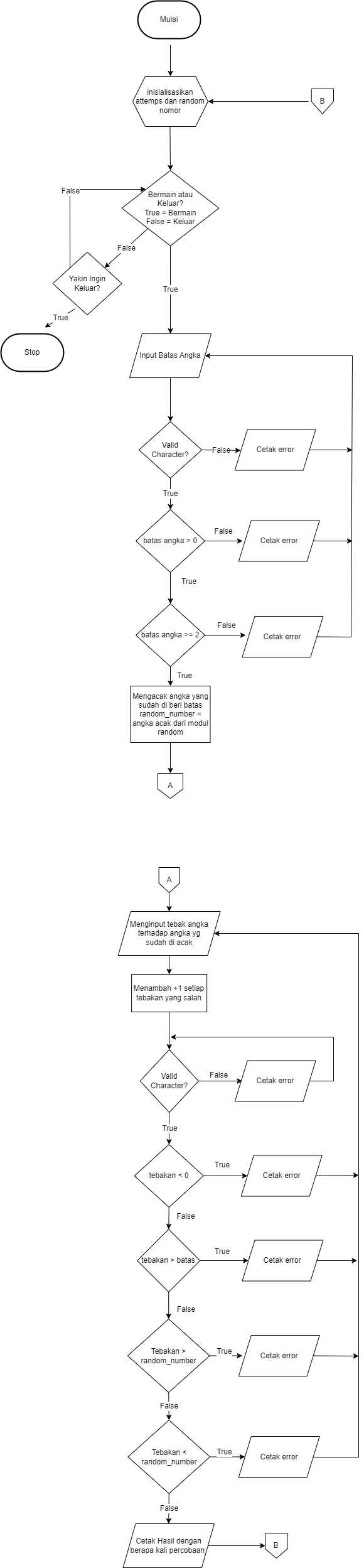
# **3. SOLUSI**

Video game tidak luput dari interaksi antara user dan program, untuk mempercantik program kami menggunakan GUI (Graphical User Interface). Kami menggunakan library Tkinter untuk membuat GUI. Rancangan video game yang kami buat tidak sulit, garis besarnya kami hanya menggunakan conditioning dan OOP pada program ini. Lebih lengkapnya kami jabarkan dalam bentuk list :

* Types and Comparisons
* Conditional Statements
* Looping
* Exception Handling
* Function
* OOP (Object Oriented Programming

# **4. DESIGN**





# **5. CODING**

Berikut merupakan final code yang sudah melewati berbagai trial and error

import tkinter as tk

from tkinter import ttk, messagebox, simpledialog

from PIL import Image, ImageTk

import random

class TebakanGame:

    def \_\_init\_\_(self, master):

        self.master = master

        self.master.title("Guess The Number")

        self.button\_color = "#507590"

        self.hover\_color = "#28495c"

        self.font\_style = ('Open Sans', 12)

        image\_path = r"login\_bg.png"

        self.load\_and\_display\_image(image\_path)

        self.master.tk\_setPalette(background='#b8d3e1')

        style = ttk.Style(self.master)

        style.element\_create("Transparent.TLabel", "from", "default")

        style.layout("Transparent.TLabel", [("Transparent.TLabel", {"sticky":

"news"})])

        style.configure("Transparent.TLabel", background="#b8d3e1")

        tk.Label(self.master, text="Masukkan batas angka", font=('Open Sans',

18),bg='#b8d3e1').pack(pady=20)

        self.entry = tk.Entry(self.master, font=('Open Sans', 15), )

        self.entry.pack(pady=10)

        self.entry.bind("<Return>", lambda event: self.start\_game())

        self.master.protocol("WM\_DELETE\_WINDOW", self.on\_closing)

        self.start\_button = tk.Button(self.master, text="Mulai Game",

command=self.start\_game, font=('Open Sans', 15), bg=self.button\_color)

        self.start\_button.pack(pady=15)

        self.start\_button.bind("<Enter>", lambda event: self.on\_hover(event,

self.hover\_color))

        self.start\_button.bind("<Leave>", lambda event: self.on\_hover(event,

self.button\_color))

        self.random\_number = 0

        self.attempts = 0

    def on\_hover(self, event, color):

        event.widget.config(bg=color)

    def load\_and\_display\_image(self, image\_path):

        image = Image.open(image\_path)

        image = ImageTk.PhotoImage(image)

        label = tk.Label(self.master, image=image)

        label.image = image

        label.place(relwidth=1, relheight=1)

    def start\_game(self):

        try:

            batas\_atas = int(self.entry.get())

            if batas\_atas < 0:

                messagebox.showerror("Error", "Masukkan bilangan positif.")

                return

            if batas\_atas < 2:

                messagebox.showerror("Error", "Masukkan angka batas yang lebih

besar dari 1.")

                return

            self.random\_number = random.randint(1, batas\_atas)

            self.attempts = 0

            game\_over = False

            while not game\_over:

                tebakan = self.get\_guess(batas\_atas)

                self.attempts += 1

                game\_over = self.update\_feedback(tebakan)

            self.display\_results()

        except ValueError:

            messagebox.showerror("Error", "Masukkan karakter yang sesuai.")

    def get\_guess(self, batas\_atas):

        while True:

            try:

                tebakan = int(simpledialog.askstring("Tebakan", f'Tebak angka

antara 1 sampai {batas\_atas}'))

                if tebakan <= 0:

                    messagebox.showerror("Error", "Masukkan angka di atas 0.")

                elif tebakan > batas\_atas:

                    messagebox.showerror("Error", f'Tebakan anda terlalu

besar. Tebak angka antara 1 sampai {batas\_atas}.')

                else:

                    return tebakan

            except ValueError:

                messagebox.showerror("Error", "Masukkan karakter yang

sesuai.")

    def update\_feedback(self, tebakan):

        if tebakan < self.random\_number:

            messagebox.showinfo("Coba Lagi!", "Tebakan anda terlalu rendah,

Tebak lagi")

        elif tebakan > self.random\_number:

            messagebox.showinfo("Coba Lagi!", "Tebakan anda terlalu tinggi,

Tebak lagi")

        else:

            return True

        return False

    def display\_results(self):

        messagebox.showinfo("Selamat!!", f'Tebakan anda benar!\n Angka yang

benar adalah {self.random\_number}.\n Jumlah percobaan:

{self.attempts}')

        self.entry.delete(0, tk.END)

        self.entry.focus\_set()

    def on\_closing(self):

        if messagebox.askokcancel("Keluar", "Apakah Anda yakin ingin

keluar?"):

            self.master.destroy()

if \_\_name\_\_ == "\_\_main\_\_":

    root = tk.Tk()

    app = TebakanGame(root)

    root.geometry("300x200")

    root.resizable(False, False)

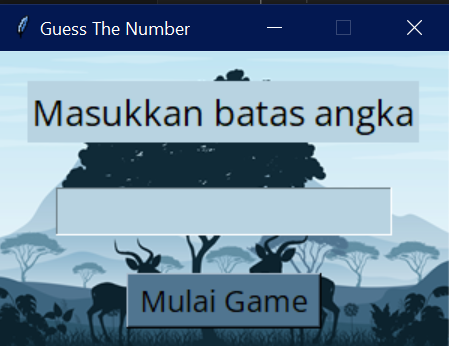
    root.mainloop()

\*Kode diatas tidak akan bisa dijalankan, dikarenakan ada perbedaan dalam spacing dan tab

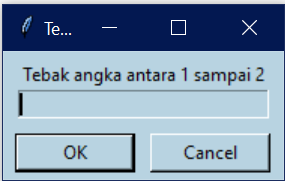
# **6. OUTPUT**

Pada output, kami akan pecah menjadi 4 Non-Error Page, Error Page, “Coba Lagi” Page, dan Quit Page, dengan rincian sebagai berikut :

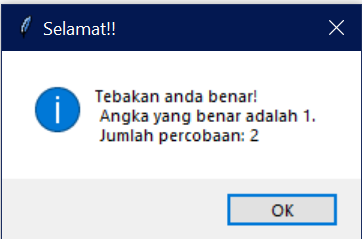
1. **Non-Error Page**
2. Akan muncul windows, kami menyebutnya sebagai Page 1 dimana Ketika pertama kali kodenya dijalankan,



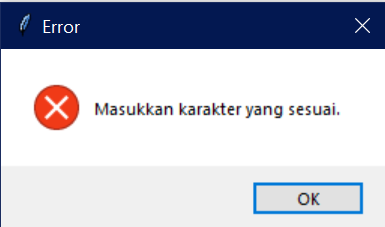
1. Jika kita memasukkan bilangan, maka akan memunculkan windows baru, kami menyebutnya sebagai Page 2 (Kemungkinan 1)



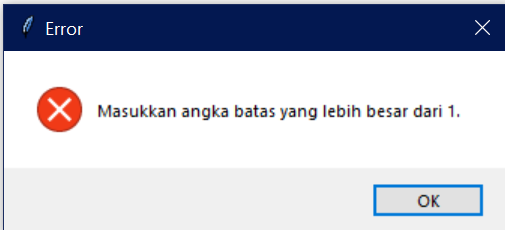
1. Jika kita berhasil menebak dengan benar, maka akan muncul window seperti ini



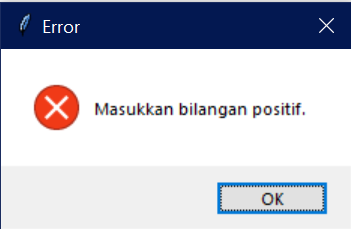
1. **Error Page**
2. Ketika kita menginput tipe data yang tidak sesuai



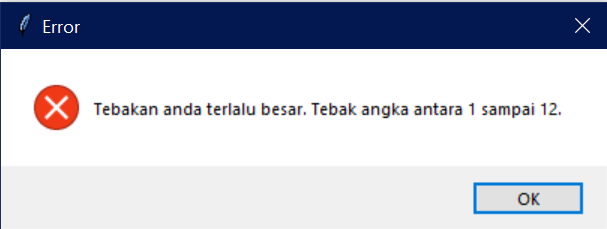
1. Ketika kita menginput angka di bawah 1



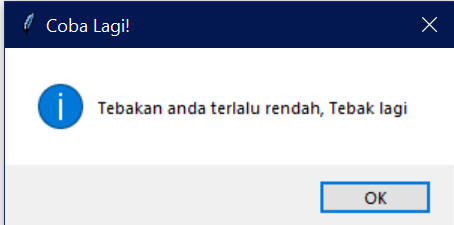
1. Ketika kita menginput angka di bawah 0



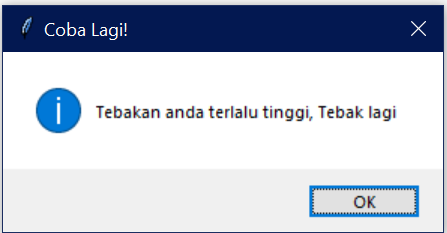
1. Ketika menebak angka di atas batas yang kita tentukan



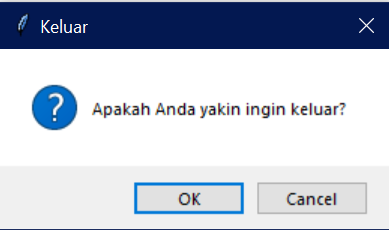
1. **“Coba Lagi” Page**
2. Ketika tebakan terlalu rendah



1. Ketika Tebakan terlalu tinggi



**D. Quit Page**



# **7. UJI COBA**

Uji coba dari program kami dapat dilihat dari table yang kami cantumkan di bawah ini



# **8. DOKUMENTASI**

# Import library yang diperlukan

import tkinter as tk

from tkinter import ttk, messagebox, simpledialog

from PIL import Image, ImageTk

import random

# Definisikan kelas TebakanGame

class TebakanGame:

    def \_\_init\_\_(self, master):

        # Inisialisasi objek TebakanGame.

        # Parameters:

        #     master (tk.Tk): Objek utama Tkinter sebagai master window.

        self.master = master

        self.master.title("Guess The Number") # Judul jendela

        # Gaya tombol dan latar belakang

        self.button\_color = "#507590" #Warna latar belakang tombol.

        self.hover\_color = "#28495c" #Warna latar belakang tombol saat di

hover.

        self.font\_style = ('Open Sans', 12) #Gaya font yang digunakan pada

elemen GUI.

        # Load dan tampilkan gambar latar belakang

        image\_path = r"login\_bg.png"

        self.load\_and\_display\_image(image\_path)

        # Set the background color and style for transparency

        self.master.tk\_setPalette(background='#b8d3e1')

        style = ttk.Style(self.master)

        # Create a transparent label style

        style.element\_create("Transparent.TLabel", "from", "default")

        style.layout("Transparent.TLabel", [("Transparent.TLabel", {"sticky":

"news"})])

        style.configure("Transparent.TLabel", background="#b8d3e1")

        # Menampilkan label dan entry untuk memasukkan batas angka

        tk.Label(self.master, text="Masukkan batas angka", font=('Open Sans',

18),bg='#b8d3e1').pack(pady=20)

        self.entry = tk.Entry(self.master, font=('Open Sans', 15), )

        self.entry.pack(pady=10)

        # Menambahkan event binding untuk tombol Enter

        self.entry.bind("<Return>", lambda event: self.start\_game())

        # Menangani penutupan jendela

        self.master.protocol("WM\_DELETE\_WINDOW", self.on\_closing)

        # Menampilkan tombol untuk memulai game

        self.start\_button = tk.Button(self.master, text="Mulai Game",

command=self.start\_game, font=('Open Sans', 15), bg=self.button\_color)

        self.start\_button.pack(pady=15)

        # Membuat Event binding untuk efek hover pada tombol (akan menjadi

beda warna)

        self.start\_button.bind("<Enter>", lambda event: self.on\_hover(event,

self.hover\_color))  # Set hover background color to a slightly

different shade of green

        self.start\_button.bind("<Leave>", lambda event: self.on\_hover(event,

self.button\_color))  # Set back to the original background color

        # Inisialisasi variabel untuk permainan

        self.random\_number = 0

        self.attempts = 0

    # Metode ini digunakan untuk mengubah warna tombol saat di-hover

    def on\_hover(self, event, color):

        # Objek event Tkinter yang dipicu.

        # Warna yang akan diatur pada tombol.

        event.widget.config(bg=color)

    # Metode untuk memuat dan menampilkan gambar latar belakang.

    def load\_and\_display\_image(self, image\_path):

        # memuat gambar dari file lokal menggunakan PIL

        image = Image.open(image\_path) #Path gambar latar belakang.

        image = ImageTk.PhotoImage(image)

        # Tampilkan gambar menggunakan Label dan meletakan di latar belakang

        label = tk.Label(self.master, image=image)

        label.image = image  # Keep a reference to avoid garbage collection

        label.place(relwidth=1, relheight=1)

    def start\_game(self):

        # Memulai permainan dengan mengambil input batas angka dari pengguna

        # dan melakukan iterasi hingga tebakan benar.

        try:

            # Mengambil batas angka dari input pengguna

            batas\_atas = int(self.entry.get())

            if batas\_atas < 0:

                messagebox.showerror("Error", "Masukkan bilangan positif.")

                return

            if batas\_atas < 2:

                messagebox.showerror("Error", "Masukkan angka batas yang lebih

besar dari 1.")

                return

            # Perintah untuk mengacak angka target

            self.random\_number = random.randint(1, batas\_atas)

            self.attempts = 0

            # Melakukan iterasi hingga tebakan benar

            game\_over = False

            while not game\_over:

                tebakan = self.get\_guess(batas\_atas)

                self.attempts += 1

                game\_over = self.update\_feedback(tebakan)

            self.display\_results()

        # Menangani kesalahan jika input tidak valid

        except ValueError:

            messagebox.showerror("Error", "Masukkan karakter yang sesuai.")

    def get\_guess(self, batas\_atas):

        # Meminta tebakan dari pengguna dengan penanganan kesalahan.

        # Parameters:

        #     batas\_atas (int): Batas atas angka yang dapat ditebak.

        # Returns:

        #     int: Tebakan yang valid dari pengguna.

        while True:

            try:

                tebakan = int(simpledialog.askstring("Tebakan", f'Tebak angka

antara 1 sampai {batas\_atas}'))

                if tebakan <= 0:

                    messagebox.showerror("Error", "Masukkan angka di atas 0.")

                elif tebakan > batas\_atas:

                    messagebox.showerror("Error", f'Tebakan anda terlalu

besar. Tebak angka antara 1 sampai {batas\_atas}.')

                else:

                    return tebakan

            except ValueError:

                # Menangani kesalahan jika input bukan angka

                messagebox.showerror("Error", "Masukkan karakter yang

sesuai.")

    def update\_feedback(self, tebakan):

        # Menampilkan umpan balik berdasarkan tebakan pengguna.

        # Parameters:

        #     tebakan (int): Tebakan pengguna.

        if tebakan < self.random\_number:

            messagebox.showinfo("Coba Lagi!", "Tebakan anda terlalu rendah,

Tebak lagi")

        elif tebakan > self.random\_number:

            messagebox.showinfo("Coba Lagi!", "Tebakan anda terlalu tinggi,

Tebak lagi")

        else:

            return True  # Game is over because the guess is correct

        return False  # Game is still ongoing

    def display\_results(self):

        # Menampilkan hasil tebakan setelah permainan selesai

        messagebox.showinfo("Selamat!!", f'Tebakan anda benar!\n Angka yang

benar adalah {self.random\_number}.\n Jumlah percobaan:

{self.attempts}')

        # Mengatur ulang elemen-elemen untuk permainan baru

        self.entry.delete(0, tk.END)

        self.entry.focus\_set()

    def on\_closing(self):

        if messagebox.askokcancel("Keluar", "Apakah Anda yakin ingin

keluar?"):

            self.master.destroy()

if \_\_name\_\_ == "\_\_main\_\_":

    # Membuat instance Tkinter

    root = tk.Tk()

    app = TebakanGame(root)

    # Menentukan ukuran jendela

    root.geometry("300x200")

    # # untuk tidak bisa di perbesar atau di perkecil

    root.resizable(False, False)

    # Memulai loop utama Tkinter

    root.mainloop()

\*Kode diatas tidak akan bisa dijalankan, dikarenakan ada perbedaan dalam spacing dan tab

# **9. KESIMPULAN DAN SARAN**

Program yang kami buat sangat terbatas dikarenakan ilmu dan informasi. Masih banyak sektor yang dapat dikembangkan. Saran kami jika pembaca ingin mendapatkan hasil grafis yang lebih memuaskan, maka dapat menggunakan Bahasa pemrograman lain yang memang ditujukkan untuk pembuatan game.