

# Mata Garuda LPDP Sulsel in Action 2025 - Evaluasi Akhir Kelas Koding

---

## Deskripsi Evaluasi Akhir

Evaluasi akhir ini bertujuan menilai pemahaman dan keterampilan coding dasar menggunakan Python dan Tkinter melalui pembuatan aplikasi berbasis GUI (Graphical User Interface). Peserta bekerja dalam tim untuk menyelesaikan aplikasi sederhana sesuai tema yang dipilih, kemudian mempresentasikan hasilnya sebagai bagian dari evaluasi penutup kegiatan.

## Pembagian Kelompok

<b>Group 01</b>	<b>Group 02</b>	<b>Group 03</b>	<b>Group 04</b>	<b>Group 05</b>
Abdul Aziz	Nazzala Fauziah	Salsabila Dwi Yanti	Meyshita Ivana	Alvin Hidayat
Sepriani Ester Ari	Nabilah Nailatul Izzah	Windy Ayu Alfiasari	Yulianti	Maulana Farhan Wahyudi
Yunisa Nur Anggraini	Salsabila	Sesya Julian Rahman	A. Afifah Syahira Irwan	Zaqa Qyr'an
Muh. Fauzan Saputra	Aldilla Adrian	Alfira S	Ananda Nurul Attaya H	Filza Azhari
Ervalina Tri Ayu Wulandari	Jesika Ayu Lestari	Novri Anggraena Akmar	Olivia	Hikari Alfadiyah
Dina Reski Ramdhani	Nurmita Sari	Fitria Ramadhani	Jenimar Puspita Maharani	Faizahyanti

## Ketentuan Evaluasi

### 1. Pembagian Tim:

- Siswa dibagi ke dalam 5 tim.
- Setiap tim terdiri dari 6 orang.

### 2. Pemilihan Aplikasi:

- Setiap tim menyiapkan **satu aplikasi akhir** yang akan dievaluasi.
- Aplikasi menggunakan bahasa pemrograman Python dengan Tkinter sebagai GUI.

### 3. Waktu Penggeraan:

Hari	Tanggal	Kode Pertemuan	Kegiatan
Jumat	28 November 2025	15	Final Project

Hari	Tanggal	Kode Pertemuan	Kegiatan
	16		Presentation

- Waktu yang diberikan untuk penyelesaian akhir aplikasi adalah **2 jam untuk penyelesaian aplikasi dan 1 jam untuk presentasi masing masing kelompok.**
- Fokuskan waktu pada penyempurnaan fitur dan perapian antarmuka sebelum sesi penilaian.

#### 1. Presentasi:

- Setiap tim mempresentasikan aplikasi yang telah dibuat.
- Presentasi mencakup penjelasan tentang tujuan aplikasi, alur kerja, serta cuplikan kode utama yang mendukung fungsionalitas.

## Panduan Pengembangan Aplikasi

Peserta diharapkan membuat aplikasi dengan menggunakan **Tkinter** yang memiliki antarmuka grafis sederhana. Berikut ini adalah contoh aplikasi sederhana yang dapat digunakan sebagai referensi:

### Contoh Aplikasi: Pesan Selamat Datang dan Printout Tulisan

Aplikasi ini hanya memiliki satu tombol, yang ketika ditekan, akan menampilkan pesan selamat datang di layar.

```
import tkinter as tk

# Fungsi untuk menampilkan pesan
def show_message():
    label.config(text="Selamat datang di aplikasi Tkinter!")

# Membuat window utama
root = tk.Tk()
root.title("Pesanan Selamat Datang")

# Membuat label kosong
label = tk.Label(root, text="", font=("Arial", 14))
label.pack(pady=20)

# Membuat tombol untuk menampilkan pesan
button = tk.Button(root, text="Klik Saya", command=show_message, font=("Arial", 14))
button.pack()

# Menjalankan aplikasi
root.mainloop()
```

contoh lain Aplikasi ini hanya memiliki satu tombol, yang ketika ditekan, akan menampilkan pesan sesuai input di layar.

```
import tkinter as tk

def ambil_input():
    # Mengambil teks yang dimasukkan ke dalam entry
    nilai = entry.get()
    print(f"Nilai yang dimasukkan: {nilai}")

root = tk.Tk()

# Membuat Entry widget
entry = tk.Entry(root)
entry.pack(pady=20)

# Tombol untuk mengambil nilai dari entry
button = tk.Button(root, text="Ambil Input", command=ambil_input)
button.pack()

root.mainloop()
```

## Contoh Aplikasi Advance dengan Style

Berikut contoh aplikasi dengan menggunakan style dari ttk.

```
import tkinter as tk
from tkinter import ttk

root = tk.Tk()
root.title("Biodata Form")
root.geometry("650x400")

# Mengatur style menggunakan ttk.Style
style = ttk.Style()

# Menetapkan tema default
style.theme_use("classic")

# Mengatur warna dan font pada widget
style.configure("TButton",
                background="#1E93AB", # Warna biru
                foreground="#000000", # Teks putih
                font=("Arial", 12, "bold"))

style.configure("TLabel",
                background="#F3F2EC", # Warna krem
                foreground="#E62727", # Warna merah terang
                font=("Arial", 12))

style.configure("TEntry",
                fieldbackground="#DCDCDC", # Warna abu-abu muda
                foreground="#000000", # Warna teks hitam
```

```
        font=("Arial", 12))

style.configure("TSpinbox",
    fieldbackground="#DCDCDC", # Warna abu-abu muda
    foreground="#000000", # Warna teks hitam
    font=("Arial", 12))

style.configure("TCombobox",
    fieldbackground="#DCDCDC", # Warna abu-abu muda
    foreground="#000000", # Warna teks hitam
    font=("Arial", 12))

# ---- main container ----
main_frame = ttk.Frame(root, padding=10)
main_frame.pack(fill="both", expand=True)

# 2-column layout frames
left_frame = ttk.Frame(main_frame)
right_frame = ttk.Frame(main_frame)

left_frame.grid(row=0, column=0, sticky="nwe", padx=(0, 10))
right_frame.grid(row=0, column=1, sticky="nwe")

main_frame.columnconfigure(0, weight=1)
main_frame.columnconfigure(1, weight=1)

# ===== STATE VARIABLES =====
name_var = tk.StringVar()
age_var = tk.IntVar(value=18)
gender_var = tk.StringVar(value="Not specified")
phone_var = tk.StringVar()
email_var = tk.StringVar()
city_var = tk.StringVar()

hobby_music = tk.BooleanVar()
hobby_sport = tk.BooleanVar()
hobby_read = tk.BooleanVar()

# ----- LEFT SIDE -----
ttk.Label(left_frame, text="Name:").grid(row=0, column=0, sticky="w")
ttk.Entry(left_frame, textvariable=name_var, width=25).grid(row=0,
column=1, sticky="we", pady=2)

ttk.Label(left_frame, text="Age:").grid(row=1, column=0, sticky="w")
ttk.Spinbox(left_frame, from_=1, to=120, textvariable=age_var,
width=5).grid(row=1, column=1, sticky="w", pady=2)

ttk.Label(left_frame, text="Phone:").grid(row=2, column=0, sticky="w")
ttk.Entry(left_frame, textvariable=phone_var, width=25).grid(row=2,
column=1, sticky="we", pady=2)

ttk.Label(left_frame, text="Email:").grid(row=3, column=0, sticky="w")
ttk.Entry(left_frame, textvariable=email_var, width=25).grid(row=3,
column=1, sticky="we", pady=2)
```

```
left_frame.columnconfigure(1, weight=1)

# ----- RIGHT SIDE -----
ttk.Label(right_frame, text="Gender:").grid(row=0, column=0, sticky="w")
gender_frame = ttk.Frame(right_frame)
gender_frame.grid(row=0, column=1, sticky="w")

ttk.Radiobutton(gender_frame, text="Male", value="Male",
variable=gender_var).pack(side="left")
ttk.Radiobutton(gender_frame, text="Female", value="Female",
variable=gender_var).pack(side="left")
ttk.Radiobutton(gender_frame, text="Other", value="Other",
variable=gender_var).pack(side="left")

ttk.Label(right_frame, text="City:").grid(row=1, column=0, sticky="w")
city_combo = ttk.Combobox(right_frame, textvariable=city_var,
state="readonly", width=22)
city_combo["values"] = ("Makassar", "Palu", "Jakarta", "Surabaya",
"Other")
city_combo.grid(row=1, column=1, sticky="w")
city_combo.current(0)

ttk.Label(right_frame, text="Hobbies:").grid(row=2, column=0, sticky="nw")
hobby_frame = ttk.Frame(right_frame)
hobby_frame.grid(row=2, column=1, sticky="w")

ttk.Checkbutton(hobby_frame, text="Music",
variable=hobby_music).pack(anchor="w")
ttk.Checkbutton(hobby_frame, text="Sport",
variable=hobby_sport).pack(anchor="w")
ttk.Checkbutton(hobby_frame, text="Reading",
variable=hobby_read).pack(anchor="w")

ttk.Label(right_frame, text="About you:").grid(row=3, column=0,
sticky="nw", pady=5)
about_text = tk.Text(right_frame, width=30, height=5)
about_text.grid(row=3, column=1, sticky="we")

right_frame.columnconfigure(1, weight=1)

# ===== LIST DISPLAY AREA (RESULT WINDOW) =====
ttk.Label(main_frame, text="Result:").grid(row=2, column=0, columnspan=2,
pady=(20, 5))

result_list = tk.Text(main_frame, width=80, height=8)
result_list.grid(row=3, column=0, columnspan=2, sticky="we")

# ===== SUBMIT BUTTON =====
def on_submit():
    hobbies = []
    if hobby_music.get():
        hobbies.append("Music")
    if hobby_sport.get():
        hobbies.append("Sport")
    if hobby_read.get():
        hobbies.append("Reading")
    result_list.insert(tk.END, f"\n{' '.join(hobbies)}")
```

```

        hobbies.append("Sport")
if hobby_read.get():
    hobbies.append("Reading")

about = about_text.get("1.0", "end-1c")

result_list.delete("1.0", "end") # Clear previous result

result_list.insert("end", f"Name: {name_var.get()}\n")
result_list.insert("end", f"Age: {age_var.get()}\n")
result_list.insert("end", f"Gender: {gender_var.get()}\n")
result_list.insert("end", f"Phone: {phone_var.get()}\n")
result_list.insert("end", f"Email: {email_var.get()}\n")
result_list.insert("end", f"City: {city_var.get()}\n")
result_list.insert("end", f"Hobbies: {', '.join(hobbies)} if hobbies
else '-'}\n")
result_list.insert("end", f"About:\n{about}\n")

submit_btn = ttk.Button(main_frame, text="Show Result", command=on_submit)
submit_btn.grid(row=1, column=0, columnspan=2, pady=10)

root.mainloop()

```

Berikut contoh aplikasi dengan menggunakan style dari tk.

```

import tkinter as tk

root = tk.Tk()
root.title("Biodata Form")
root.geometry("650x400")

# ---- main container ----
main_frame = tk.Frame(root)
main_frame.pack(fill="both", expand=True)

# 2-column layout frames
left_frame = tk.Frame(main_frame)
right_frame = tk.Frame(main_frame)

left_frame.grid(row=0, column=0, sticky="nwe", padx=(0, 10))
right_frame.grid(row=0, column=1, sticky="nwe")

main_frame.columnconfigure(0, weight=1)
main_frame.columnconfigure(1, weight=1)

# ===== STATE VARIABLES =====
name_var = tk.StringVar()
age_var = tk.IntVar(value=18)
gender_var = tk.StringVar(value="Not specified")
phone_var = tk.StringVar()
email_var = tk.StringVar()
city_var = tk.StringVar()

```

```
hobby_music = tk.BooleanVar()
hobby_sport = tk.BooleanVar()
hobby_read = tk.BooleanVar()

# ----- Styling for Widgets -----
bg_color = "#FFFFFF"
btn_color = "#1E93AB"
text_color = "#000000"
label_color = "#E62727"
entry_bg = "#DCDCDC"
font_style = ("Arial", 12)

# ----- LEFT SIDE -----
tk.Label(left_frame, text="Name:", bg=bg_color, fg=label_color,
font=font_style).grid(row=0, column=0, sticky="w")
tk.Entry(left_frame, textvariable=name_var, width=25, font=font_style,
bg=entry_bg, fg=text_color).grid(row=0, column=1, sticky="we", pady=2)

tk.Label(left_frame, text="Age:", bg=bg_color, fg=label_color,
font=font_style).grid(row=1, column=0, sticky="w")
tk.Spinbox(left_frame, from_=1, to=120, textvariable=age_var, width=5,
font=font_style, bg=entry_bg, fg=text_color).grid(row=1, column=1,
sticky="w", pady=2)

tk.Label(left_frame, text="Phone:", bg=bg_color, fg=label_color,
font=font_style).grid(row=2, column=0, sticky="w")
tk.Entry(left_frame, textvariable=phone_var, width=25, font=font_style,
bg=entry_bg, fg=text_color).grid(row=2, column=1, sticky="we", pady=2)

tk.Label(left_frame, text="Email:", bg=bg_color, fg=label_color,
font=font_style).grid(row=3, column=0, sticky="w")
tk.Entry(left_frame, textvariable=email_var, width=25, font=font_style,
bg=entry_bg, fg=text_color).grid(row=3, column=1, sticky="we", pady=2)

left_frame.columnconfigure(1, weight=1)

# ----- RIGHT SIDE -----
tk.Label(right_frame, text="Gender:", bg=bg_color, fg=label_color,
font=font_style).grid(row=0, column=0, sticky="w")
gender_frame = tk.Frame(right_frame)
gender_frame.grid(row=0, column=1, sticky="w")

tk.Radiobutton(gender_frame, text="Male", value="Male",
variable=gender_var, font=font_style, bg=bg_color).pack(side="left")
tk.Radiobutton(gender_frame, text="Female", value="Female",
variable=gender_var, font=font_style, bg=bg_color).pack(side="left")
tk.Radiobutton(gender_frame, text="Other", value="Other",
variable=gender_var, font=font_style, bg=bg_color).pack(side="left")

tk.Label(right_frame, text="City:", bg=bg_color, fg=label_color,
font=font_style).grid(row=1, column=0, sticky="w")
city_combo = tk.OptionMenu(right_frame, city_var, "Makassar", "Palu",
"Jakarta", "Surabaya", "Other")
```

```
city_combo.config(width=22, font=font_style, bg=entry_bg, fg=text_color)
city_combo.grid(row=1, column=1, sticky="w")
city_var.set("Makassar")

tk.Label(right_frame, text="Hobbies:", bg=bg_color, fg=label_color,
font=font_style).grid(row=2, column=0, sticky="nw")
hobby_frame = tk.Frame(right_frame)
hobby_frame.grid(row=2, column=1, sticky="w")

tk.Checkbutton(hobby_frame, text="Music", variable=hobby_music,
font=font_style, bg=bg_color).pack(anchor="w")
tk.Checkbutton(hobby_frame, text="Sport", variable=hobby_sport,
font=font_style, bg=bg_color).pack(anchor="w")
tk.Checkbutton(hobby_frame, text="Reading", variable=hobby_read,
font=font_style, bg=bg_color).pack(anchor="w")

tk.Label(right_frame, text="About you:", bg=bg_color, fg=label_color,
font=font_style).grid(row=3, column=0, sticky="nw", pady=5)
about_text = tk.Text(right_frame, width=30, height=5, font=font_style,
bg=entry_bg, fg=text_color)
about_text.grid(row=3, column=1, sticky="we")

right_frame.columnconfigure(1, weight=1)

# ===== LIST DISPLAY AREA (RESULT WINDOW) =====
tk.Label(main_frame, text="Result:", bg=bg_color, fg=label_color,
font=font_style).grid(row=2, column=0, columnspan=2, pady=(20, 5))

result_list = tk.Text(main_frame, width=80, height=8, font=font_style,
bg=entry_bg, fg=text_color)
result_list.grid(row=3, column=0, columnspan=2, sticky="we")

# ===== SUBMIT BUTTON =====
def on_submit():
    hobbies = []
    if hobby_music.get():
        hobbies.append("Music")
    if hobby_sport.get():
        hobbies.append("Sport")
    if hobby_read.get():
        hobbies.append("Reading")

    about = about_text.get("1.0", "end-1c")

    result_list.delete("1.0", "end") # Clear previous result

    result_list.insert("end", f"Name: {name_var.get()}\n")
    result_list.insert("end", f"Age: {age_var.get()}\n")
    result_list.insert("end", f"Gender: {gender_var.get()}\n")
    result_list.insert("end", f"Phone: {phone_var.get()}\n")
    result_list.insert("end", f"Email: {email_var.get()}\n")
    result_list.insert("end", f"City: {city_var.get()}\n")
    result_list.insert("end", f"Hobbies: {', '.join(hobbies)}\n"
else '-}\n")
```

```
result_list.insert("end", f"About:\n{about}\n")

submit_btn = tk.Button(main_frame, text="Show Result", command=on_submit,
font=font_style, bg=btn_color, fg=text_color)
submit_btn.grid(row=1, column=0, columnspan=2, pady=10)

root.mainloop()
```

## Penjelasan Kode:

1. **Tkinter** digunakan untuk membuat antarmuka grafis.
2. **Label** menampilkan pesan yang dapat diubah dengan menekan tombol.
3. **Button** memicu fungsi `show_message()` yang mengubah teks pada label.
4. Aplikasi ini sangat sederhana dan memberikan pemahaman dasar tentang widget di Tkinter.

## Tahapan Pengerjaan:

1. **Pilih Aplikasi:** Tentukan aplikasi yang akan dibangun oleh tim Anda. Pilihan aplikasi bisa berupa aplikasi kalkulator, daftar tugas (to-do list), atau aplikasi interaktif lainnya.
2. **Desain dan Kode:** Gunakan Tkinter untuk membuat GUI aplikasi. Fokus pada antarmuka yang sederhana, fungsional, dan rapi.
3. **Presentasi:** Siapkan presentasi singkat yang menjelaskan:
  - Tujuan aplikasi.
  - Bagaimana cara aplikasi bekerja.
  - Potongan kode penting yang digunakan dalam aplikasi.

## Penilaian

Penilaian akan dilakukan berdasarkan kriteria berikut:

1. **Fungsi Aplikasi:** Seberapa baik aplikasi bekerja sesuai dengan tujuan yang ditentukan.
2. **Desain Antarmuka:** Seberapa baik aplikasi didesain dengan menggunakan Tkinter.
3. **Presentasi:** Seberapa jelas dan terstruktur presentasi tentang aplikasi yang dibuat, termasuk kesiapan menjawab pertanyaan.
4. **Kreativitas:** Inovasi dalam fitur atau desain aplikasi yang dibuat.

---

Selamat mengikuti evaluasi akhir dan semoga sukses!