

APLIKASI PEMBUATAN KARTU UCAPAN DIGITAL

T. A. Semester Ganjil 2024/2025



5230411285

OKTABERLIAN DIVA GRACIELA

**PROGRAM STUDI INFORMATIKA
FAKULTAS SAINS & TEKNOLOGI
UNIVERSITAS TEKNOLOGI YOGYAKARTA
2024**

DAFTAR ISI

DAFTAR ISI.....	2
PENDAHULUAN.....	3
1.1 Latar Belakang.....	3
1.2 Tujuan.....	3
CARA KERJA APLIKASI.....	4
2.1 Gambaran Umum.....	4
2.2 Alur Proses Aplikasi.....	4
2.3 Diagram Flowchart.....	5
PENJELASAN FUNGSI PROGRAM.....	6
KODE PROGRAM.....	7
HASIL DAN PEMBAHASAN.....	14
a. Tampilan Awal Aplikasi.....	14
b. Contoh Pratinjau Kartu.....	15
c. Tampilan Dialog Simpan File.....	16
d. Tampilan Kartu Setelah di Simpan dalam File.....	17
KESIMPULAN.....	18

PENDAHULUAN

1.1 Latar Belakang

Pada era digital seperti sekarang, kartu ucapan telah menjadi salah satu cara yang populer untuk menyampaikan pesan emosional, baik untuk merayakan momen spesial maupun sekadar memberi perhatian kepada orang terdekat. Dengan kemajuan teknologi, pembuatan kartu ucapan kini tidak lagi terbatas pada proses manual atau cetak. Sebagai gantinya, aplikasi digital yang dapat disesuaikan dengan preferensi dan kebutuhan pengguna semakin banyak bermunculan. Aplikasi-aplikasi ini memungkinkan siapa saja untuk dengan mudah merancang kartu ucapan pribadi yang lebih kreatif, menarik, dan tentunya lebih cepat diselesaikan. Pengguna dapat memilih berbagai elemen seperti desain, template, teks, warna, dan gambar untuk menciptakan kartu yang sesuai dengan suasana hati atau tema acara. Untuk memenuhi permintaan akan aplikasi yang praktis dan fleksibel, aplikasi ini dikembangkan menggunakan bahasa pemrograman Python, yang terkenal dengan kemudahan penggunaannya, serta Tkinter sebagai antarmuka pengguna grafis (GUI) yang intuitif. Selain itu, aplikasi ini juga memanfaatkan library PIL (Pillow) untuk manipulasi gambar, yang memungkinkan pengguna untuk mengedit dan menyesuaikan elemen visual dengan berbagai efek dan filter secara langsung di dalam aplikasi.

1.2 Tujuan

- Mempermudah pengguna dalam membuat kartu ucapan secara digital.
- Memberikan pilihan tema, template, dan warna latar belakang untuk personalisasi kartu.
- Menyediakan fitur untuk menyimpan kartu ucapan dalam format gambar (.png).

CARA KERJA APLIKASI

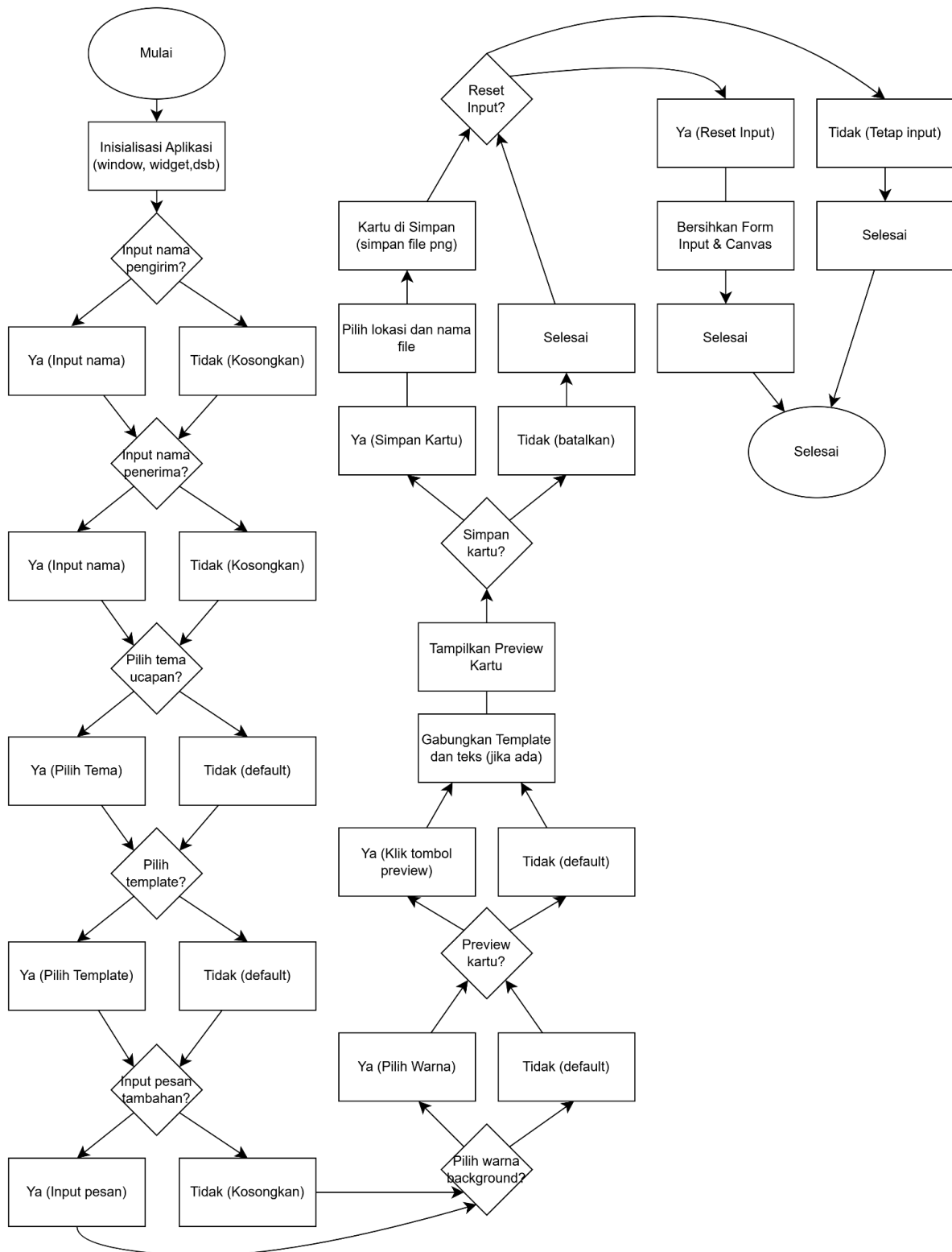
2.1 Gambaran Umum

- Input Data: Pengguna memasukkan nama pengirim, nama penerima, memilih tema ucapan, dan menambahkan pesan.
- Template dan Warna: Pengguna memilih template kartu serta warna latar belakang.
- Preview: Kartu ucapan ditampilkan di canvas sebagai pratinjau.
- Simpan Kartu: Pengguna dapat menyimpan kartu dalam format gambar.

2.2 Alur Proses Aplikasi

- Program memuat tema dan template kartu yang telah disiapkan.
- Input pengguna divalidasi dan digunakan untuk menghasilkan kartu.
- Fungsi `preview_card()` menampilkan hasil kartu sementara pada layar.
- Fungsi `save_card()` menyimpan kartu ke lokasi yang dipilih pengguna.

2.3 Diagram Flowchart



PENJELASAN FUNGSI PROGRAM

Berikut adalah penjelasan fungsi utama pada program:

1. `widget_create()`
 - Membuat elemen GUI seperti label, entry, combobox, tombol, dan canvas.
 - Menentukan tata letak komponen dalam antarmuka aplikasi.
2. `preview_card()`
 - Menghasilkan pratinjau kartu ucapan berdasarkan input pengguna.
 - Menggunakan library PIL untuk menggambar teks dan memuat template kartu.
3. `save_card()`
 - Menyimpan kartu ucapan sebagai file gambar dalam format PNG.
 - Menggunakan dialog file untuk memilih lokasi penyimpanan.
4. `reset_input()`
 - Menghapus semua input pengguna dan mengembalikan pengaturan ke kondisi awal.
5. `choose_bg_color()`
 - Membuka pemilih warna untuk memilih warna latar belakang kartu.

KODE PROGRAM

Berikut adalah kode program dari aplikasi pembuat kartu ucapan dari saya :

```
import os
import textwrap
import tkinter as tk
from tkinter import ttk, messagebox, colorchooser, filedialog
from PIL import Image, ImageDraw, ImageFont, ImageTk

# Warna latar belakang
bg_color = "pink"

class AppOrder:
    def __init__(self, root):
        self.root = root
        self.root.title("Aplikasi Pembuat Kartu Ucapan")
        self.root.geometry("900x1000")
        self.root.config(bg=bg_color)

    # Tema
    self.card_themes = {
        "Birthday": "Selamat Ulang Tahun! Semoga hari-harimu menyenangkan.",
        "Wedding": "Selamat atas pernikahanmu! Semoga cinta selalu menyertai.",
        "Condolence": "Turut berduka cita. Semoga keluarga diberi kekuatan.",
        "Thank You": "Terima kasih atas segalanya. Kami sangat menghargainya.",
        "Get Well Soon": "Semoga lekas sembuh dan kembali sehat.",
        "Anniversary": "Selamat Hari Jadi! Semoga cinta terus bertumbuh.",
        "Graduation": "Selamat atas kelulusanmu! Semoga sukses di masa depan.",
        "New Baby": "Selamat atas kelahiran bayi mungilmu! Semoga bahagia selalu.",
        "Christmas": "Selamat Natal! Semoga damai dan sukacita Natal menyertai kita
semua.",
        "Eid al-Fitr": "Selamat Idul Fitri! Mohon maaf lahir dan batin.",
        "Eid al-Adha": "Selamat Idul Adha! Semoga keberkahan selalu menyertai.",
```

```

        "Chinese New Year": "Selamat Tahun Baru Imlek! Gong Xi Fa Cai, semoga
keberuntungan melimpah.",
        "Diwali": "Selamat Hari Diwali! Semoga hidupmu terang dan penuh kebahagiaan.",
        "Ramadhan": "Selamat menunaikan ibadah Ramadan. Semoga diberi keberkahan
dan kekuatan.",
        "Hanukkah": "Selamat Hanukkah! Semoga kebahagiaan dan keberkahan
bersamamu.",
    }

    # Template
    self.template_folder = "Kartu_Ucapan/images"
    self.templates = {f"Template {i}": os.path.join(self.template_folder, f"template{i}.png")
for i in range(1, 9)}

    # Warna background default
    self.bg_color = "#FFFFFF"

    self.widget_create()

def widget_create(self):
    # Frame utama
    main_frame = tk.Frame(self.root, bg=self.bg_color)
    main_frame.place(relx=0.5, rely=0.5, anchor="center")

    # Input dan pilihan
    self.create_label_entry(main_frame, "Nama Pengirim:", 0, "sender_entry")
    self.create_label_entry(main_frame, "Nama Penerima:", 1, "receiver_entry")
    self.create_label_combobox(main_frame, "Pilih Tema Ucapan:", 2,
self.card_themes.keys(), "theme_var", "theme_menu")
    self.create_button(main_frame, "Pilih Warna Background", self.choose_bg_color, 3)

    # Template
    self.create_label_combobox(main_frame, "Pilih Template:", 4, self.templates.keys(),
"template_var", "template_menu")

    # Pesan tambahan

```



```

        self.create_label_entry(main_frame, "Pesan Tambahan:", 5, "message_entry",
width=40)

# Tombol dan Canvas
self.create_button(main_frame, "Preview Kartu", self.preview_card, 6)
self.canvas = tk.Canvas(main_frame, width=500, height=300, bg="white",
highlightbackground="black")
self.canvas.grid(row=7, column=0, columnspan=2, pady=10, sticky="w")

# Tombol Reset Input
self.create_button(main_frame, "Reset Input", self.reset_input, 8)

# Tombol Simpan Kartu
self.create_button(main_frame, "Simpan Kartu", self.save_card, 9)

def create_label_entry(self, parent, text, row, attr_name, width=30):
    tk.Label(parent, text=text, bg=bg_color, anchor="w").grid(row=row, column=0,
sticky="w", padx=10, pady=5)
    entry = tk.Entry(parent, width=width)
    entry.grid(row=row, column=1, padx=10, pady=5, sticky="w")
    setattr(self, attr_name, entry)

def create_label_combobox(self, parent, text, row, values, var_name, menu_name):
    tk.Label(parent, text=text, bg=bg_color, anchor="w").grid(row=row, column=0,
sticky="w", padx=10, pady=5)
    var = tk.StringVar()
    menu = ttk.Combobox(parent, textvariable=var, values=list(values), state="readonly")
    menu.grid(row=row, column=1, padx=10, pady=5, sticky="w")
    setattr(self, var_name, var)
    setattr(self, menu_name, menu)

def create_button(self, parent, text, command, row):
    tk.Button(parent, text=text, command=command).grid(row=row, column=0,
columnspan=2, pady=10, sticky="w")

def choose_bg_color(self):

```

```

color_code = colorchooser.askcolor(title="Pilih Warna Background")[1]
if color_code:
    self.bg_color = color_code

def preview_card(self):
    # Hapus semua elemen di canvas
    self.canvas.delete("all")

    # Membuat gambar sementara (ukuran sesuai canvas)
    img = Image.new("RGB", (500, 300), color=self.bg_color)
    draw = ImageDraw.Draw(img)

    # Memuat template jika dipilih
    selected_template = self.template_var.get()
    if selected_template and selected_template in self.templates:
        template_path = self.templates[selected_template]
        if os.path.exists(template_path):
            template_img = Image.open(template_path).resize((500, 300))
            img.paste(template_img, (0, 0), template_img.convert("RGBA"))

    # Mengambil data input
    sender = self.sender_entry.get()
    receiver = self.receiver_entry.get()
    message = self.message_entry.get()
    theme = self.theme_var.get()

    # Mengatur font (sama seperti save_card)
    try:
        font = ImageFont.truetype("arial.ttf", 18) # Gunakan font Arial jika tersedia
    except IOError:
        font = ImageFont.load_default() # Font default jika Arial tidak tersedia

    # Membuat teks kartu
    text = self.card_themes.get(theme, "")
    combined_message = f"Dear {receiver},\n{text}\n\n{message}\n\nFrom, {sender}"

```

```

# Membungkus teks agar sesuai dengan lebar kartu
max_width = 480 # Sesuai lebar canvas
wrapped_text = "\n".join(
    textwrap.fill(line, width=max_width // 12) # Perkiraan karakter per baris
    for line in combined_message.split("\n")
)

# Menggambar teks pada gambar
draw.multiline_text((70, 80), wrapped_text, font=font, fill="black", spacing=5)

# Konversi gambar menjadi format yang bisa ditampilkan di Canvas
self.preview_image = ImageTk.PhotoImage(img)
self.canvas.create_image(252, 152, image=self.preview_image)

def save_card(self):
    # Meminta user memilih lokasi dan nama file
    file_path = filedialog.asksaveasfilename(
        initialdir=os.getcwd(), # Lokasi default
        defaultextension=".png",
        filetypes=[("PNG files", "*.png")],
        title="Simpan Kartu Ucapan"
    )

    # Jika user tidak memilih file, hentikan proses
    if not file_path:
        return

    # Pastikan nama file memiliki ekstensi .png
    if not file_path.endswith(".png"):
        file_path += ".png"

    # Debug: cek path yang dipilih
    print(f"File akan disimpan di: {file_path}")

```

```

# Membuat gambar kartu
img = Image.new('RGB', (500, 300), color=self.bg_color)
draw = ImageDraw.Draw(img)

# Memuat template yang dipilih
selected_template = self.template_var.get()
if selected_template and selected_template in self.templates:
    template_path = self.templates[selected_template]
    if os.path.exists(template_path):
        template_img = Image.open(template_path).resize((500, 300))
        img.paste(template_img, (0, 0), template_img.convert("RGBA"))

# Mengambil data dari input
sender = self.sender_entry.get()
receiver = self.receiver_entry.get()
message = self.message_entry.get()
theme = self.theme_var.get()

# Mengatur font
try:
    font = ImageFont.truetype("arial.ttf", 18) # Pastikan Arial tersedia di sistem
except IOError:
    font = ImageFont.load_default() # Gunakan font default jika Arial tidak tersedia

# Membuat teks kartu
text = self.card_themes.get(theme, "")
combined_message = f"Dear {receiver},\n{text}\n\n{message}\n\nFrom, {sender}"

# Membungkus teks agar sesuai dengan lebar kartu
max_width = 480 # Lebar maksimum teks dalam piksel
wrapped_text = "\n".join(
    textwrap.fill(line, width=max_width // 12) # Estimasi karakter per baris
    for line in combined_message.split("\n")
)

# Menggambar teks pada kartu

```

```

draw.multiline_text((70, 80), wrapped_text, font=font, fill="black", spacing=5)

# Menyimpan file gambar
try:
    img.save(file_path, "PNG")
    messagebox.showinfo("Simpan", f"Kartu berhasil disimpan di lokasi: {file_path}")
except Exception as e:
    messagebox.showerror("Error", f"Gagal menyimpan file: {str(e)}")

def reset_input(self):
    #untuk mereset semua input pada form
    self.sender_entry.delete(0, tk.END)
    self.receiver_entry.delete(0, tk.END)
    self.theme_var.set("")
    self.template_var.set("")
    self.message_entry.delete(0, tk.END)
    self.bg_color = "#FFFFFF" # Reset warna background ke default
    self.canvas.delete("all") # Bersihkan canvas preview

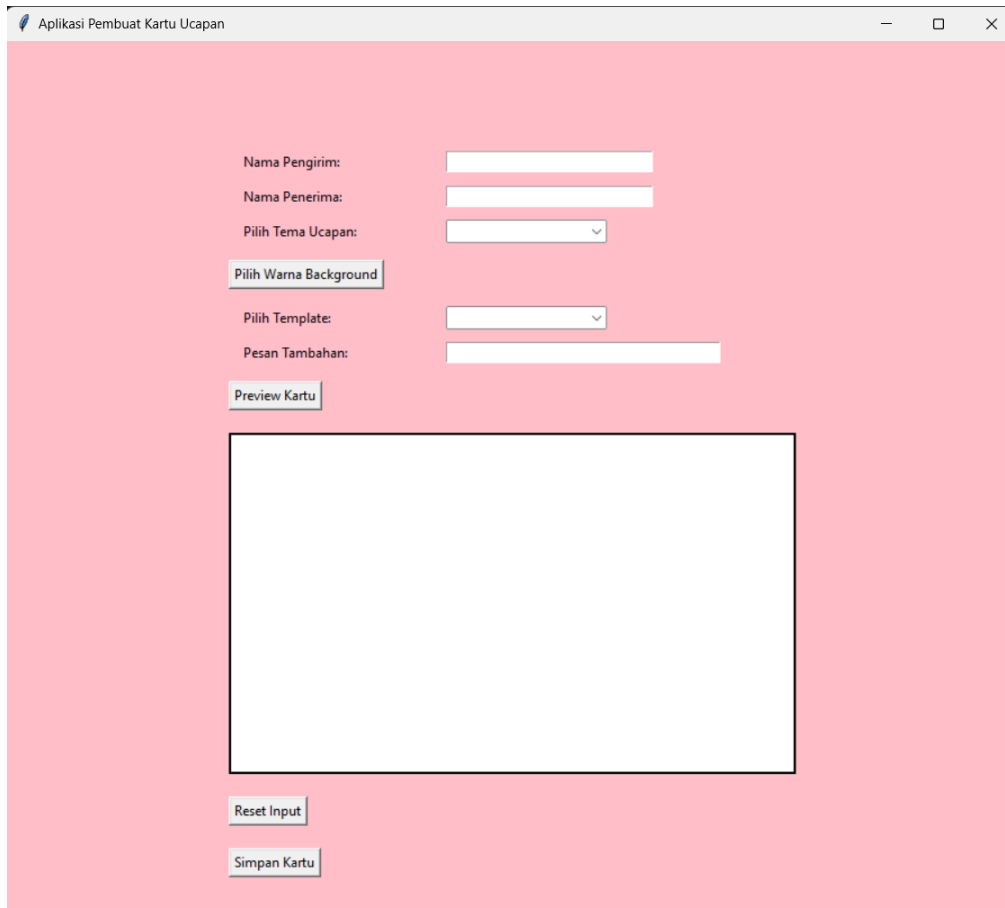
if __name__ == "__main__":
    root = tk.Tk()
    app = AppOrder(root)
    root.mainloop()

```

HASIL DAN PEMBAHASAN

a. Tampilan Awal Aplikasi

Berikut adalah tampilan awal aplikasi ketika pertama kali dijalankan:



The screenshot shows a web application window titled "Aplikasi Pembuat Kartu Ucapan". The interface is set against a solid pink background. It includes the following elements:

- Input Fields:**
 - "Nama Pengirim:" followed by a text input field.
 - "Nama Penerima:" followed by a text input field.
 - "Pilih Tema Ucapan:" followed by a dropdown menu.
 - "Pilih Warna Background:" followed by a dropdown menu.
 - "Pilih Template:" followed by a dropdown menu.
 - "Pesan Tambahan:" followed by a text input field.
- Buttons:**
 - "Preview Kartu": A button located below the message input field.
 - "Reset Input": A button located at the bottom left of the form area.
 - "Simpan Kartu": A button located at the bottom left of the form area, below the "Reset Input" button.
- Preview Area:** A large, empty rectangular box with a black border, intended for displaying the generated greeting card.

b. Contoh Pratinjau Kartu

Setelah pengguna mengisi data dan menekan tombol *Preview Kartu*, hasil pratinjau ditampilkan seperti gambar berikut:

- Input:
 - a. Nama Pengirim: "Diva"
 - b. Nama Penerima: "Graciela"
 - c. Tema: "Birthday"
 - d. Pesan Tambahan: "Semoga sukses selalu!"
 - e. Warna Latar Belakang: Pink
 - f. Template: Template 8

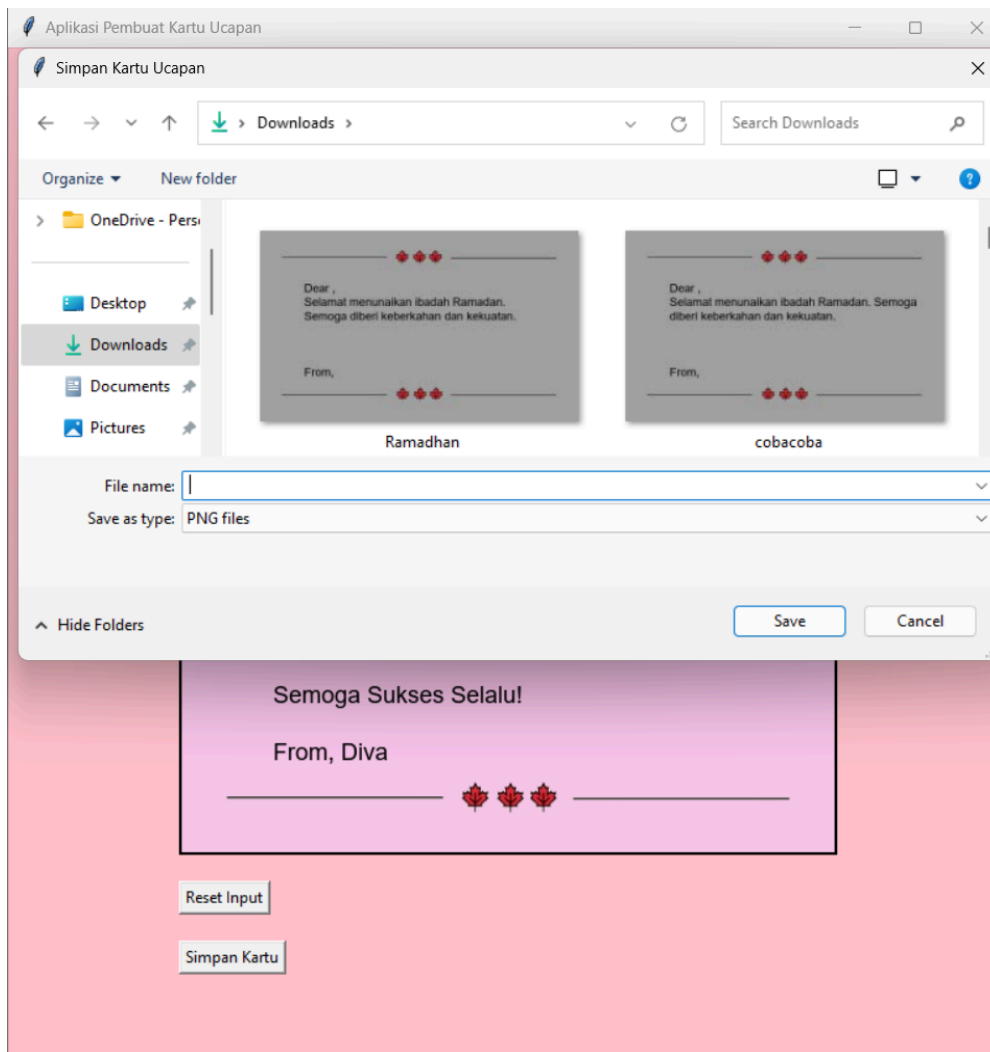
- Hasil Kartu:

Kartu ucapan akan menampilkan pesan berikut:

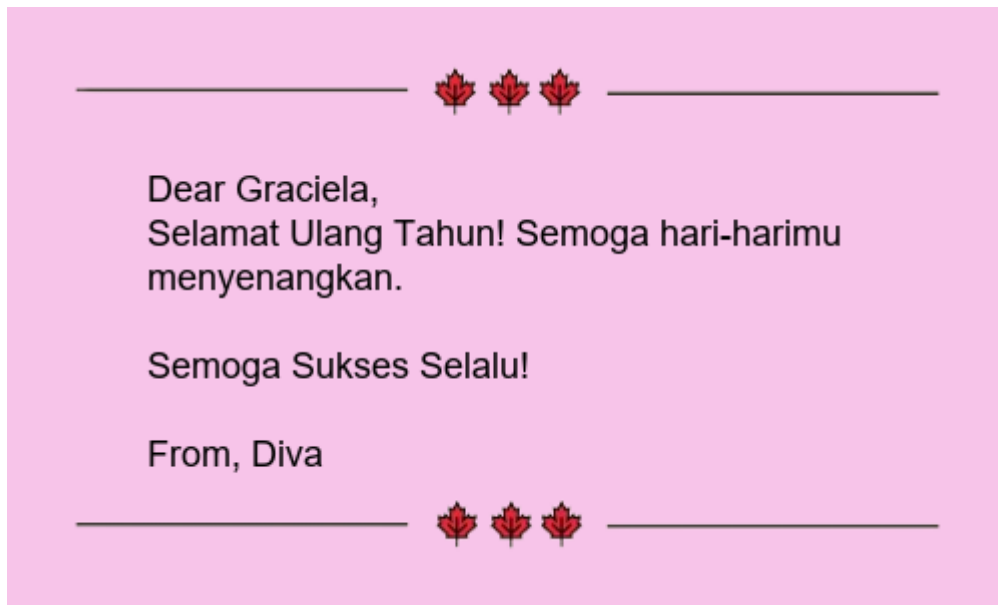
The screenshot shows a web application titled "Aplikasi Pembuat Kartu Ucapan". The interface has a light pink background. On the left, there are input fields for "Nama Pengirim:" (Diva), "Nama Penerima:" (Graciela), "Pilih Tema Ucapan:" (Birthday), "Pilih Warna Background:" (a button), "Pilih Template:" (Template 8), and "Pesan Tambahan:" (Semoga Sukses Selalu). Below these is a "Preview Kartu" button. To the right of the button is a preview box with a light purple background. Inside the preview box, the text reads: "Dear Graciela, Selamat Ulang Tahun! Semoga hari-harimu menyenangkan. Semoga Sukses Selalu! From, Diva". The text is centered and flanked by decorative horizontal lines with three red maple leaves in the middle. Below the preview box are two more buttons: "Reset Input" and "Simpan Kartu".

c. Tampilan Dialog Simpan File

Ketika tombol *Simpan Kartu* ditekan, dialog berikut muncul untuk memilih lokasi penyimpanan:



d. Tampilan Kartu Setelah di Simpan dalam File



KESIMPULAN

Aplikasi ini berhasil mengembangkan sebuah platform yang memungkinkan pengguna untuk membuat kartu ucapan digital secara praktis dengan berbagai opsi personalisasi, termasuk pilihan tema, warna, dan template. Fitur-fitur yang disediakan memudahkan pengguna untuk membuat kartu ucapan dengan sentuhan pribadi, menjadikannya lebih spesial dan sesuai dengan kebutuhan mereka. Pengguna tidak hanya dapat menciptakan kartu ucapan dengan cepat, tetapi juga dapat menambahkan elemen-elemen kreatif untuk membuat kartu tersebut unik. Penggunaan Tkinter sebagai framework untuk antarmuka pengguna memberikan pengalaman yang interaktif dan responsif, sementara Python Imaging Library (PIL) memungkinkan manipulasi gambar yang lebih fleksibel dan canggih. Kombinasi kedua teknologi ini memberikan aplikasi ini kekuatan untuk berkembang lebih lanjut dan menawarkan berbagai kemungkinan dalam pengembangan aplikasi kartu ucapan yang lebih kompleks dan menarik.