

SURAT PENCATATAN CIPTAAN

Dalam rangka perlindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta, dengan ini menerangkan:

Nomor dan tanggal permohonan : EC002024229131, 19 November 2024

Pencipta

Nama : Ricki Sastra, Rini Martiwi dkk

Alamat : Jl. Kalibata Timur 1 No.44 , Pancoran, Jakarta Selatan, DKI Jakarta, 12740

Kewarganegaraan : Indonesia

Pemegang Hak Cipta

Nama : Universitas Bina Sarana Informatika

Alamat : Jl. Kramat Raya No.98, RW.9, Kwitang, Senen, Jakarta Pusat, Dki Jakarta 10450

Kewarganegaraan : Indonesia

Jenis Ciptaan : Program Komputer

Judul Ciptaan : Aplikasi Certificate Generator(ACG Versi 1.0)

Tanggal dan tempat diumumkan untuk pertama kali : 15 November 2024, di Jakarta Pusat
di wilayah Indonesia atau di luar wilayah Indonesia

Jangka waktu perlindungan : Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.

Nomor pencatatan : 000801563

adalah benar berdasarkan keterangan yang diberikan oleh Pemohon.

Surat Pencatatan Hak Cipta atau produk Hak terkait ini sesuai dengan Pasal 72 Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.



a.n. MENTERI HUKUM DAN HAK ASASI MANUSIA
DIREKTUR JENDERAL KEKAYAAN INTELEKTUAL
u.b

Direktur Hak Cipta dan Desain Industri

IGNATIUS M.T. SILALAHI
NIP. 196812301996031001

LAMPIRAN PENCIPTA

No	Nama	Alamat
1	Ricki Sastra	Jl. Kalibata Timur 1 No.44 , Pancoran, Jakarta Selatan
2	Rini Martiwi	Jl. Kemajuan RT.006/004 Petukangan Selatan , Pesanggrahan, Jakarta Selatan
3	Nyoman Suardhita	Jl Percetakan Negara 1 No 8 Rt 003 Rw 07 , Cempaka Putih, Jakarta Pusat
4	Dwi Indarti	Jl. Cumi-cumi Blok C1 No 6 Pondok Permai Kotabumi , Pasar Kemis, Tangerang
5	Eulin Karlina	Dusun Pahing Rt1 Rw3 Desa Karangsari , Waled, Cirebon



Manual Book Aplikasi Certificate Generator (ACG Versi 1.0)

Pendahuluan

Aplikasi Certificate Generator (ACG) adalah aplikasi desktop untuk membuat sertifikat secara otomatis menggunakan template PDF atau gambar (PNG/JPG). Aplikasi ini dirancang untuk menghasilkan sertifikat dari data peserta yang disimpan dalam file Excel, dengan cepat dan efisien.

Persyaratan Sistem

1. Sistem Operasi: Windows (7/8/10/11) atau macOS.
2. Aplikasi Tidak Membutuhkan Instalasi Python: Aplikasi sudah dalam bentuk standalone executable (.exe).
3. File Pendukung:
 - File template (PDF atau gambar).
 - File Excel yang berisi data nama peserta.

Cara Menggunakan Aplikasi

1. Buka File Aplikasi:

Klik dua kali pada file aplikasi (`ACG\_Versi\_1.0.exe`) untuk menjalankan.



2. Login:

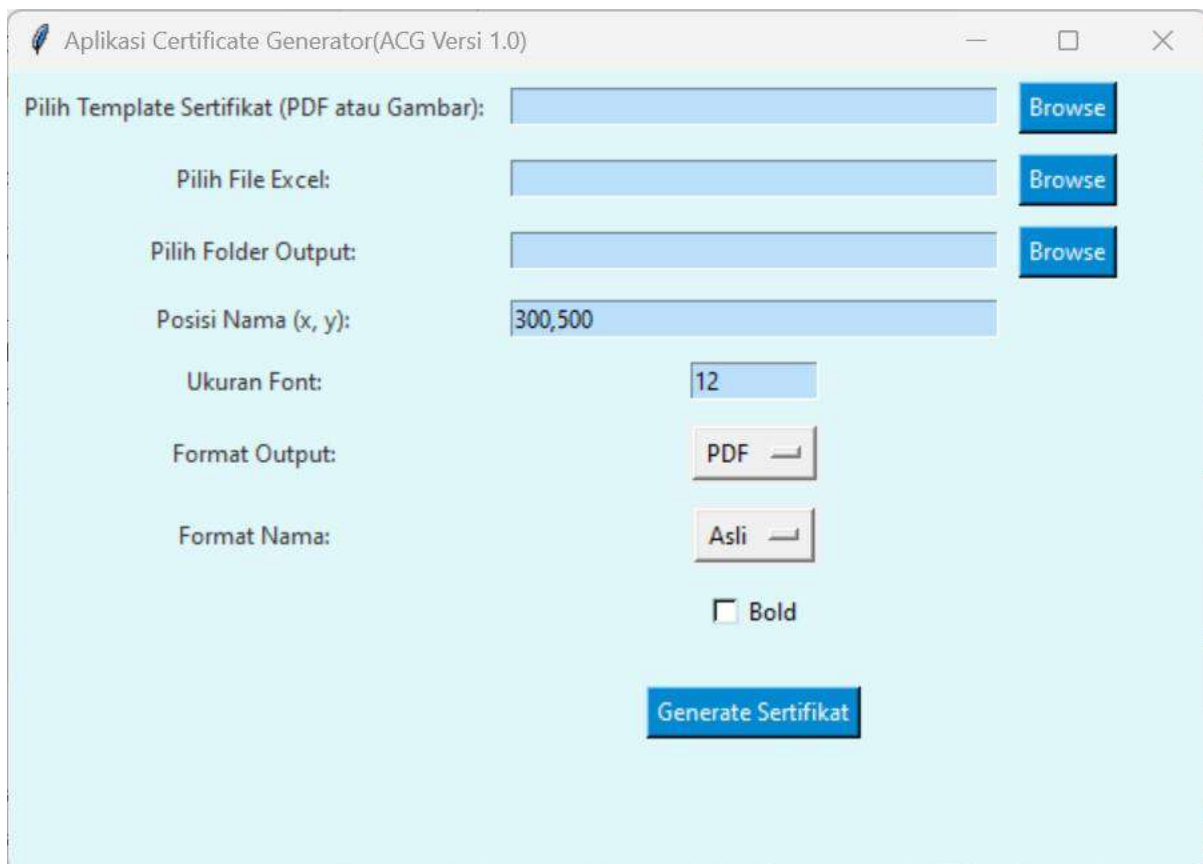
- Username:
- Password:

Klik tombol Login untuk melanjutkan. Jika berhasil, jendela utama aplikasi akan terbuka.



2. Tampilan Utama

Setelah login, Anda akan melihat jendela utama dengan beberapa komponen berikut:



A. Pemilihan Template

1. Pilih Template Sertifikat:

Klik tombol Browse untuk memilih file template:

- Format yang didukung: PDF, PNG, atau JPG.

2. Template akan digunakan sebagai latar belakang untuk menempatkan nama peserta.

B. Pemilihan File Data

1. Pilih File Excel:

Klik tombol Browse untuk memilih file Excel berisi data nama peserta.

2. Format File Excel:

- File harus memiliki kolom bernama `Name` yang memuat nama peserta.

C. Pengaturan Output

1. Pilih Folder Output:

Klik tombol Browse untuk memilih lokasi folder tempat sertifikat yang dihasilkan akan disimpan.

2. Pastikan folder memiliki cukup ruang penyimpanan.

D. Pengaturan Teks

1. Posisi Nama (x, y):

Masukkan posisi nama di template dalam format `x,y`. Contoh: `300,500`.

2. Ukuran Font:

Masukkan ukuran font teks yang akan digunakan (contoh: `12`).

3. Format Output:

Pilih format sertifikat yang dihasilkan: PDF atau PNG.

4. Format Nama

Pilih format penulisan nama:

- Asli: Sesuai data di file Excel.
- Kapital: Nama ditulis dalam huruf kapital.
- Kapital Huruf Depan: Huruf pertama setiap kata menggunakan kapital.

5. Bold:

Centang kotak ini untuk menampilkan nama dalam teks tebal (bold).

Aplikasi Certificate Generator(ACG Versi 1.0)

Pilih Template Sertifikat (PDF atau Gambar): C:/Users/ricki sastra/Downloads/image.pdf.p Browse

Pilih File Excel: R:/sertifikat/v2/data nama.xlsx Browse

Pilih Folder Output: R:/sertifikat Browse

Posisi Nama (x, y): 300,500

Ukuran Font: 12

Format Output: PDF

Format Nama: Asli

☒ Bold

Generate Sertifikat

3. Membuat Sertifikat

1. Setelah semua pengaturan selesai, klik tombol Generate Sertifikat.
2. Aplikasi akan memproses data dari file Excel dan menghasilkan sertifikat sesuai template.
3. Sertifikat akan disimpan di folder output yang telah dipilih.

Contoh Penggunaan

1. File Excel

Berikut contoh format data di file Excel:

Name	

John Doe	

| Jane Smith |

| Alexander Lee|

2. Template PDF/Gambar

Gunakan file sertifikat kosong dengan area kosong untuk penempatan nama.

3. Output

- Nama pada sertifikat akan ditempatkan di posisi yang ditentukan (`x, y`) sesuai ukuran dan format font yang diatur.

Troubleshooting

1. Aplikasi Tidak Berjalan

- Pastikan sistem Anda mendukung file aplikasi (.exe).
- Jalankan aplikasi dengan Run as Administrator jika terjadi kendala izin.

2. Error Font

- Jika terjadi error terkait font, pastikan font sistem Anda memiliki `arial.ttf` dan `arialbd.ttf`.

3. Sertifikat Tidak Tersimpan

- Periksa apakah folder output memiliki izin tulis.
- Pastikan template atau folder tidak dalam keadaan terkunci oleh aplikasi lain.

Tips Penggunaan

- Gunakan template beresolusi tinggi untuk hasil sertifikat yang lebih baik.
- Uji posisi teks pada template sebelum memproses data dalam jumlah besar.
- Simpan backup file Excel dan template sebelum memulai.

Kontak Pengembang

Jika ada pertanyaan atau masalah, silakan hubungi:

- Email: ricki.rkt@bsi.ac.id

- Telepon: 08992562232

Selamat menggunakan Aplikasi Certificate Generator (ACG Versi 1.0). Semoga bermanfaat!



Source Code Aplikasi :

```
import tkinter as tk
from tkinter import filedialog, messagebox
from PyPDF2 import PdfReader, PdfWriter
from reportlab.lib.pagesizes import letter
from reportlab.pdfgen import canvas
from PIL import Image, ImageDraw, ImageFont
import pandas as pd
import os
import io

# Username dan password yang valid
VALID_USERNAME = ""
VALID_PASSWORD = ""

# Fungsi untuk memeriksa login
def check_login(username_entry, password_entry):
    username = username_entry.get()
    password = password_entry.get()

    if username == VALID_USERNAME and password == VALID_PASSWORD:
        messagebox.showinfo("Login Sukses", "Selamat datang!")
        login_window.destroy()
        open_main_app() # Membuka aplikasi utama
    else:
        messagebox.showerror("Login Gagal", "Username atau password salah.")

# Fungsi untuk membuka aplikasi utama setelah login berhasil
def open_main_app():
    # Fungsi lainnya untuk aplikasi utama
    def select_template():
        file_path = filedialog.askopenfilename(filetypes=[("PDF files",
        "*.pdf"), ("Image files", "*.png;*.jpg;*.jpeg")])
        if file_path:
            template_entry.delete(0, tk.END)
            template_entry.insert(0, file_path)

    def select_excel():
        file_path = filedialog.askopenfilename(filetypes=[("Excel files",
        "*.xlsx")])
        if file_path:
            excel_entry.delete(0, tk.END)
            excel_entry.insert(0, file_path)

    def select_output_folder():
```

```

folder_path = filedialog.askdirectory()
if folder_path:
    output_entry.delete(0, tk.END)
    output_entry.insert(0, folder_path)

def get_position_from_input(position_text):
    try:
        x, y = map(int, position_text.split(","))
        return (x, y)
    except ValueError:
        messagebox.showerror("Error", "Posisi tidak valid. Gunakan format 'x,y'.")
    return None

def add_text_to_pdf(template_path, output_path, name, position,
font_size=12, bold=False):
    reader = PdfReader(template_path)
    writer = PdfWriter()
    packet = io.BytesIO()
    c = canvas.Canvas(packet, pagesize=letter)

    font = "Helvetica-Bold" if bold else "Helvetica"
    c.setFont(font, font_size)
    text_width = c.stringWidth(name, font, font_size)
    x_position = position[0] - text_width / 2
    y_position = position[1]
    c.drawString(x_position, y_position, name)
    c.save()

    packet.seek(0)
    new_pdf = PdfReader(packet)
    page = reader.pages[0]
    page.merge_page(new_pdf.pages[0])

    writer.add_page(page)
    with open(output_path, "wb") as output_file:
        writer.write(output_file)

def add_text_to_png(template_path, output_path, name, position,
font_size=40, bold=False):
    template = Image.open(template_path)
    draw = ImageDraw.Draw(template)
    font_path = "arialbd.ttf" if bold else "arial.ttf"
    font = ImageFont.truetype(font_path, font_size)
    text_width, text_height = draw.textsize(name, font)
    x_position = position[0] - text_width / 2
    y_position = position[1]
    draw.text((x_position, y_position), name, font=font, fill=(0, 0, 0))

```

```

template.save(output_path)

def format_name(name, format_type):
    """Fungsi untuk memformat nama berdasarkan pilihan."""
    if format_type == "Asli":
        return name
    elif format_type == "Kapital":
        return name.upper()
    elif format_type == "Kapital Huruf Depan":
        return name.title()
    return name

def generate_certificates():
    try:
        template_path = template_entry.get()
        excel_path = excel_entry.get()
        output_folder = output_entry.get()
        format_output = format_combobox.get()
        bold = bold_var.get()
        name_format = name_format_combobox.get()

        if not template_path or not excel_path or not output_folder:
            messagebox.showerror("Error", "Semua field harus diisi!")
            return

        data = pd.read_excel(excel_path)
        font_size = int(font_size_entry.get())
        position_text = position_entry.get()
        name_position = get_position_from_input(position_text)
        if name_position is None:
            return

        for index, row in data.iterrows():
            name = row['Name']
            formatted_name = format_name(name, name_format) # Format nama
sesuai pilihan

            if format_output == "PDF" and
template_path.lower().endswith('.pdf'):
                output_path = os.path.join(output_folder,
f"{formatted_name}_certificate.pdf")
                add_text_to_pdf(template_path, output_path,
formatted_name, name_position, font_size, bold)
            elif format_output == "PNG" and
template_path.lower().endswith(('png', '.jpg')):
                output_path = os.path.join(output_folder,
f"{formatted_name}_certificate.png")

```

```

        add_text_to_png(template_path, output_path,
formatted_name, name_position, font_size, bold)

        messagebox.showinfo("Sukses", "Sertifikat berhasil dibuat!")

    except Exception as e:
        messagebox.showerror("Error", f"Terjadi kesalahan: {e}")

# Membuat window utama
root = tk.Tk()
root.title("Aplikasi Certificate Generator(ACG Versi 1.0)")
root.geometry("600x400")
root.configure(bg="#e0f7fa")

theme_bg = "#0277bd"
theme_fg = "#ffffff"
entry_bg = "#bbdefb"
button_bg = "#0288d1"

template_label = tk.Label(root, text="Pilih Template Sertifikat (PDF atau
Gambar):", bg="#e0f7fa", fg="#333")
template_label.grid(row=0, column=0, pady=5, padx=5)
template_entry = tk.Entry(root, width=40, bg=entry_bg)
template_entry.grid(row=0, column=1, pady=5, padx=5)
template_button = tk.Button(root, text="Browse", command=select_template,
bg=button_bg, fg=theme_fg)
template_button.grid(row=0, column=2, pady=5, padx=5)

excel_label = tk.Label(root, text="Pilih File Excel:", bg="#e0f7fa",
fg="#333")
excel_label.grid(row=1, column=0, pady=5, padx=5)
excel_entry = tk.Entry(root, width=40, bg=entry_bg)
excel_entry.grid(row=1, column=1, pady=5, padx=5)
excel_button = tk.Button(root, text="Browse", command=select_excel,
bg=button_bg, fg=theme_fg)
excel_button.grid(row=1, column=2, pady=5, padx=5)

output_label = tk.Label(root, text="Pilih Folder Output:", bg="#e0f7fa",
fg="#333")
output_label.grid(row=2, column=0, pady=5, padx=5)
output_entry = tk.Entry(root, width=40, bg=entry_bg)
output_entry.grid(row=2, column=1, pady=5, padx=5)
output_button = tk.Button(root, text="Browse",
command=select_output_folder, bg=button_bg, fg=theme_fg)
output_button.grid(row=2, column=2, pady=5, padx=5)

position_label = tk.Label(root, text="Posisi Nama (x, y):", bg="#e0f7fa",
fg="#333")

```

```

position_label.grid(row=3, column=0, pady=5, padx=5)
position_entry = tk.Entry(root, width=40, bg=entry_bg)
position_entry.grid(row=3, column=1, pady=5, padx=5)
position_entry.insert(0, "300,500")

font_size_label = tk.Label(root, text="Ukuran Font:", bg="#e0f7fa",
fg="#333")
font_size_label.grid(row=4, column=0, pady=5, padx=5)
font_size_entry = tk.Entry(root, width=10, bg=entry_bg)
font_size_entry.grid(row=4, column=1, pady=5, padx=5)
font_size_entry.insert(0, "12")

format_label = tk.Label(root, text="Format Output:", bg="#e0f7fa",
fg="#333")
format_label.grid(row=5, column=0, pady=5, padx=5)
format_combobox = tk.StringVar()
format_combobox.set("PDF")
format_dropdown = tk.OptionMenu(root, format_combobox, "PDF", "PNG")
format_dropdown.grid(row=5, column=1, pady=5, padx=5)

name_format_label = tk.Label(root, text="Format Nama:", bg="#e0f7fa",
fg="#333")
name_format_label.grid(row=6, column=0, pady=5, padx=5)
name_format_combobox = tk.StringVar()
name_format_combobox.set("Asli")
name_format_dropdown = tk.OptionMenu(root, name_format_combobox, "Asli",
"Kapital", "Kapital Huruf Depan")
name_format_dropdown.grid(row=6, column=1, pady=5, padx=5)

bold_var = tk.BooleanVar()
bold_checkbox = tk.Checkbutton(root, text="Bold", variable=bold_var,
bg="#e0f7fa")
bold_checkbox.grid(row=7, column=1, pady=5, padx=5)

generate_button = tk.Button(root, text="Generate Sertifikat",
command=generate_certificates, bg=button_bg, fg=theme_fg)
generate_button.grid(row=8, column=1, pady=20)

root.mainloop()

# Fungsi untuk membuka jendela login
def open_login_window():
    global login_window, username_entry, password_entry
    login_window = tk.Tk()
    login_window.title("Aplikasi Certificate Generator(ACG Versi 1.0)")
    login_window.geometry("400x200")
    login_window.configure(bg="#e0f7fa")

```

```
username_label = tk.Label(login_window, text="Username:")
username_label.pack(pady=10)
username_entry = tk.Entry(login_window)
username_entry.pack()

password_label = tk.Label(login_window, text="Password:")
password_label.pack(pady=10)
password_entry = tk.Entry(login_window, show="*")
password_entry.pack()

login_button = tk.Button(login_window, text="Login", command=lambda:
check_login(username_entry, password_entry))
login_button.pack(pady=20)

login_window.mainloop()

open_login_window()
```