

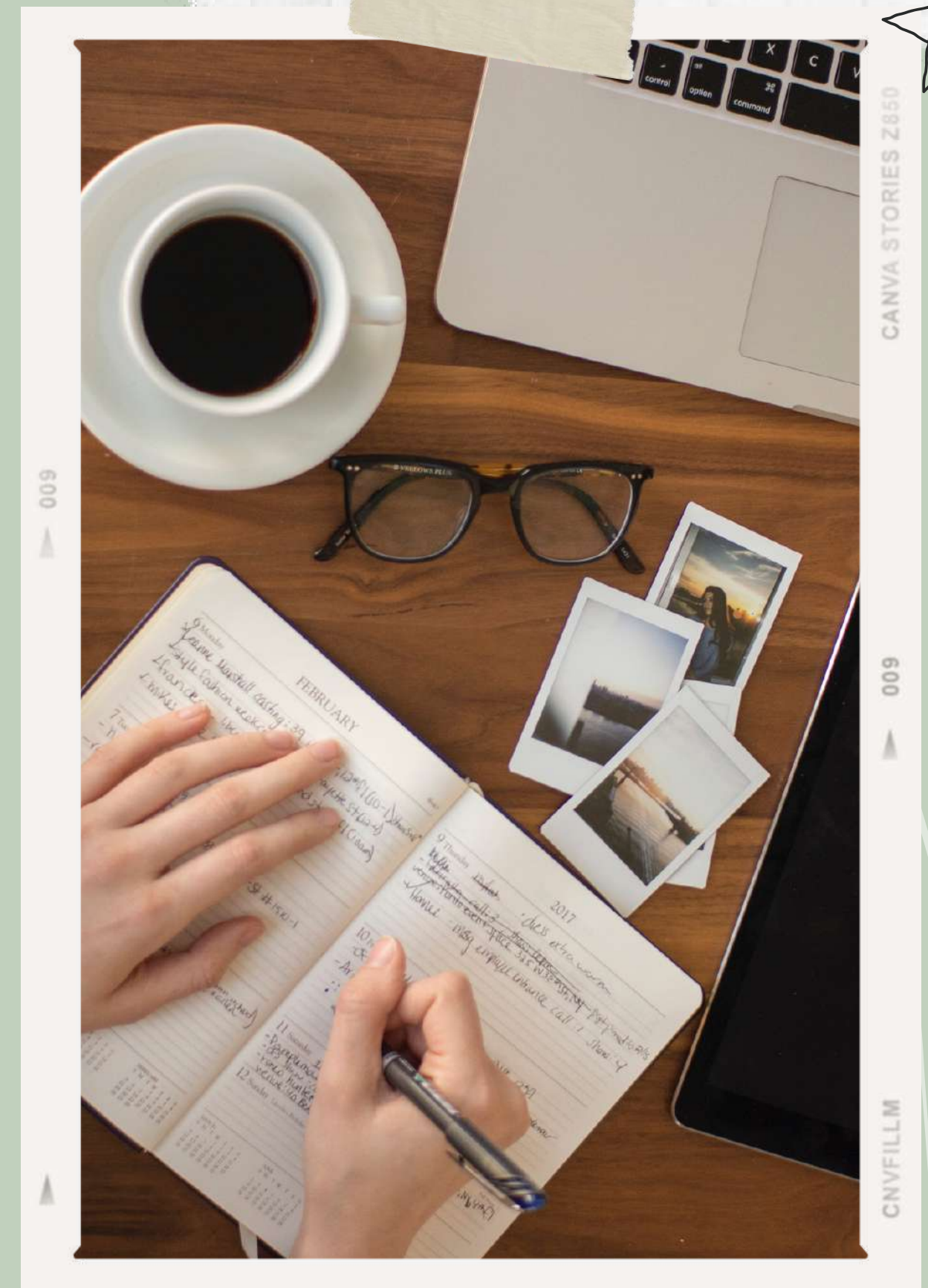
TUGAS BESAR

Aplikasi Pendataan Penduduk

Kelompok 6

* Nama Kelompok

1	Farhan Nanda Prihamudi (1206210006)
2	Ananda Taqhsya Dwiwana (1206210019)
3	Annas Thasya Haafizhah S (1206210022)



Tip: Use links to go to a different page inside your presentation.

How: Highlight text, click on the link symbol on the toolbar, and select the page in your presentation you want to connect.

Latar Belakang



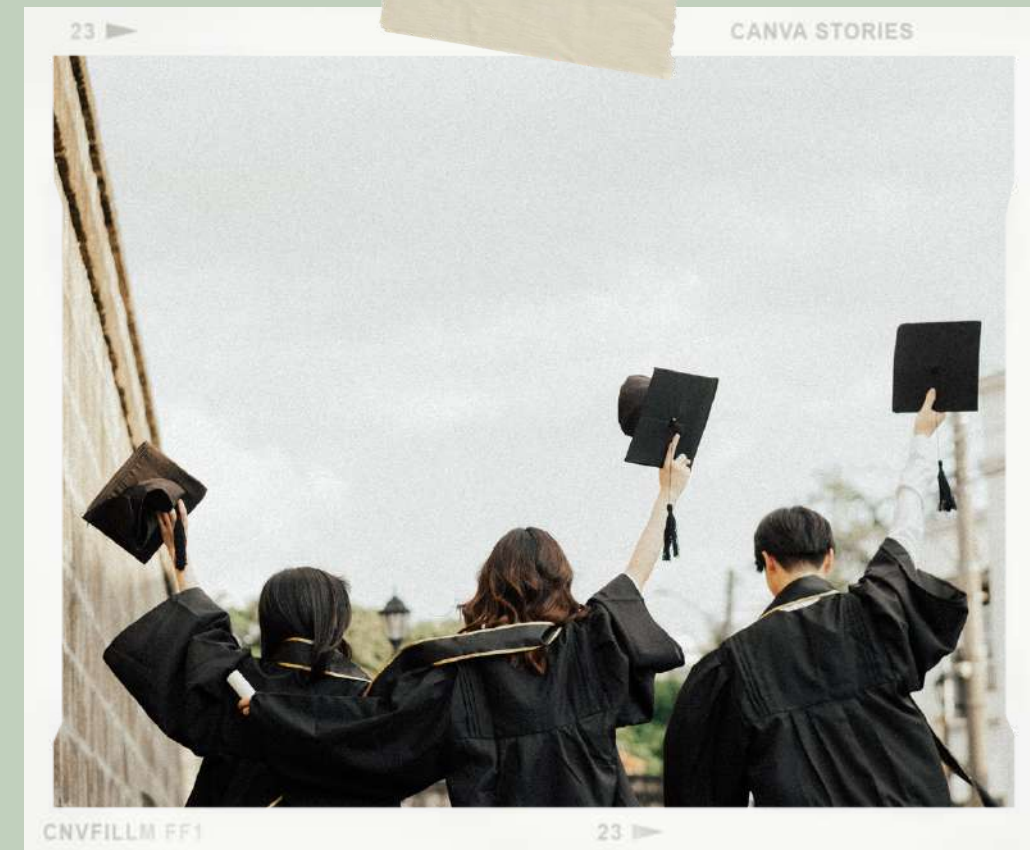
Banyak desa yang menghadapi tantangan dalam mengelola data penduduk secara efisien dan akurat karena minimnya akses serta pemahaman terhadap teknologi informasi. Mengingat tantangan-tantangan ini, penting untuk merancang dan mengembangkan aplikasi pendataan penduduk menggunakan dan MongoDB yang sederhana dan mudah digunakan. Sehingga aplikasi tersebut dapat membantu desa mengelola data penduduk dengan lebih baik, meminimalkan risiko kesalahan, meningkatkan efisiensi, dan memungkinkan akses lebih cepat terhadap informasi yang diperlukan untuk pengambilan keputusan yang lebih baik di tingkat desa.



signs of the island being brought back to life anyway, I had

- **Peningkatan Akurasi Data:** Aplikasi pendataan digital meminimalkan risiko kesalahan manusia yang umumnya terjadi dalam proses manual.
- **Efisiensi dan Produktivitas:** Dengan aplikasi pendataan, proses pengumpulan, pengolahan, dan penyimpanan data menjadi lebih cepat dan efisien.
- **Akses Mudah dan Cepat:** Masyarakat desa dapat dengan mudah mengakses informasi mengenai data penduduk mereka melalui aplikasi, tanpa harus datang ke kantor desa atau menunggu laporan manual.
- **Transparansi dan Akuntabilitas:** Masyarakat dapat melihat dan memverifikasi data mereka sendiri, memastikan bahwa informasi yang tercatat benar dan tidak terjadi manipulasi.

Manfaat Aplikasi



being brought back to life anyway, I had

Tahap-tahap implementasi aplikasi

[Back to Agenda Page](#)

Pada tahap ini akan dilakukan perancangan sebuah aplikasi yang diinginkan dengan menggunakan penulisan Bahasa pemrograman python serta akan dikaitkan dengan MongoDB yang berfungsi untuk menyimpan data dan memperhatikan fitur-fitur yang di butuhkan.

Proses Implementasi

```
import tkinter as tk
from tkinter import messagebox
from tkinter import ttk
from pymongo import MongoClient
```

Library yang dipakai

```
MONGO_HOST = 'localhost'
MONGO_PORT = 27017
MONGO_DB = 'Data_Penduduk'
MONGO_COLLECTION = 'mahasiswa'
```

konfigurasi MongoDB


```
def connect_to_mongo():
    try:
        client = MongoClient(MONGO_HOST, MONGO_PORT)
        db = client[MONGO_DB]
        collection = db[MONGO_COLLECTION]
        return collection
    except Exception as e:
        messagebox.showerror('Error', f'Gagal terhubung ke MongoDB: {str(e)}')
        return None
```

Fungsi untuk
menghubungkan ke MongoDB

Fungsi Log in

```
users = {
    'nanda': 'nanda',
    'farhan': 'farhan',
    'thasya': 'thasya',
    # Add more users here...
}

def login():
    username = entry_username.get()
    password = entry_password.get()

    if username in users and users[username] == password:
        # Hide the login frame and show the main application
        login_frame.pack_forget()
        show_main_app()
    else:
        messagebox.showwarning('Peringatan', 'Login gagal. Coba lagi.')
```


Fungsi untuk menampilkan data pada box

```
def show_data():
    collection = connect_to_mongo()
    if collection:
        data = collection.find({}, {"_id": 0})
        data_list = []
        for item in data:
            data_list.append(item)
        data_text.delete(1.0, tk.END)
        # Membuat judul kolom
        column_names = ["Name", "NIK", "Tempat Lahir", "Tanggal Lahir", "Alamat"]
        data_text.insert(tk.END, "{:<8} {:<8} {:<15} {:<15} {:<15}\n".format(*column_names))
        data_text.insert(tk.END, "="*60 + "\n")

        # Memasukkan data ke dalam tabel
        for data in data_list:
            data_text.insert(tk.END, "{:<8} {:<8} {:<15} {:<15} {:<15}\n".format(data["name"], data["nik"], data["tl"], data["tgl"]
```

```
def add_data():
    Name = name_entry.get()
    NIK = nik_entry.get()
    Tempat_Lahir = tempat_lahir_entry.get()
    Tanggal_Lahir = tanggal_lahir_entry.get()
    Alamat = alamat_entry.get()
    if Name and NIK and Tempat_Lahir and Tanggal_Lahir and Alamat:
        collection = connect_to_mongo()
        if collection:
            data = {"name": Name, "nik": NIK, "tl": Tempat_Lahir, "tgl": Tanggal_Lahir, "alamat": Alamat}
            collection.insert_one(data)
            show_data()
            # Show success messagebox
            messagebox.showinfo("Success", "Data has been added successfully!")

            name_entry.delete(0, tk.END)
            nik_entry.delete(0, tk.END)
            tempat_lahir_entry.delete(0, tk.END)
            tanggal_lahir_entry.delete(0, tk.END)
            alamat_entry.delete(0, tk.END)

        else:
            messagebox.showwarning("Warning", "Name, NIK, Tempat Lahir, Tanggal Lahir and Alamat must be filled!")
```

Fungsi untuk menambahkan data

Fungsi untuk menghapus data

```
def delete_data():
    Name = name_entry.get()
    if Name:
        collection = connect_to_mongo()
        if collection:
            collection.delete_one({"name": Name})
            show_data()
            # Show success messagebox
            messagebox.showinfo("Success", "Data has been deleted successfully!")

        name_entry.delete(0, tk.END)
    else:
        messagebox.showwarning("Warning", "Name must be filled!")
```

```
def update_data():
    Name = name_entry.get()
    NIK = nik_entry.get()
    Tempat_Lahir = tempat_lahir_entry.get()
    Tanggal_Lahir = tanggal_lahir_entry.get()
    Alamat = alamat_entry.get()
    if Name and NIK and Tempat_Lahir and Tanggal_Lahir and Alamat:
        collection = connect_to_mongo()
        if collection:
            collection.update_one({"name": Name}, {"$set": {"nik": NIK,
                                                            "tl": Tempat_Lahir,
                                                            "tgl": Tanggal_Lahir,
                                                            "alamat": Alamat }})

            show_data()
            # Show success messagebox
            messagebox.showinfo("Success", "Data has been uptaded successfully!")

        name_entry.delete(0, tk.END)
        nik_entry.delete(0, tk.END)
        tempat_lahir_entry.delete(0, tk.END)
        tanggal_lahir_entry.delete(0, tk.END)
        alamat_entry.delete(0, tk.END)
    else:
        messagebox.showwarning("Warning", "Name, NIK, Tempat Lahir, Tanggal Lahir and Alamat must be filled!")
```

Fungsi untuk mengupdate data

Fungsi untuk mencari data berdasarkan nama

```
def search_data():
    Name = name_entry.get()
    data_text.delete(1.0, tk.END)
    # Membuat judul kolom
    column_names = ["Name", "NIK", "Tempat Lahir", "Tanggal Lahir", "Alamat"]
    data_text.insert(tk.END, "{:<8} {:<8} {:<15} {:<15} {:<15}\n".format(*column_names))
    data_text.insert(tk.END, "="*60 + "\n")
    if Name:
        collection = connect_to_mongo()
        if collection:
            data = collection.find_one({"name": Name})
            if data:
                data_text.insert(tk.END, "{:<8} {:<8} {:<15} {:<15} {:<15}\n".format(data["name"], data["nik"], data["tl"], data["tg"], data["alamat"]))
            else:
                messagebox.showinfo("Info", "Data not found!")
        name_entry.delete(0, tk.END)
    else:
        messagebox.showwarning("Warning", "Name must be filled!")
```

```
def show_main_app():
    login_frame.grid_forget()
    main_app_frame.grid()

    # Create a scrolled frame
    global data_frame
    data_frame = ttk.Frame(main_app_frame)
    data_frame.grid(row=1, column=0, columnspan=2, padx=5, pady=5)

    # Create a vertical scrollbar for the scrolled frame
    scrollbar = ttk.Scrollbar(data_frame, orient=tk.VERTICAL)

    # Create a text widget to display data inside the scrolled frame
    global data_text
    data_text = tk.Text(data_frame, width=60, height=10, yscrollcommand=scrollbar.set)
    data_text.pack(side=tk.LEFT)

    # Configure the scrollbar to scroll the text widget
    scrollbar.config(command=data_text.yview)
    scrollbar.pack(side=tk.RIGHT, fill=tk.Y)

def set_background_color(widget, color):
    widget.config(bg=color)
```

Fungsi untuk menampilkan aplikasi utama setelah login berhasil

Fungsi untuk desain aplikasi

```
# Inisialisasi Tkinter
root = tk.Tk()
root.title("Aplikasi Pendataan Penduduk")

# Mengubah warna latar belakang jendela utama (root)
set_background_color(root, "lightblue")

# Frame untuk login
login_frame = tk.Frame(root)
login_frame.grid(row=4, column=0, padx=30, pady=30)

# Label dan input untuk login
lbl_username = tk.Label(login_frame, text="Username:")
lbl_username.grid()
entry_username = tk.Entry(login_frame, width=30)
entry_username.grid()

lbl_password = tk.Label(login_frame, text="Password:")
lbl_password.grid()
entry_password = tk.Entry(login_frame, width=30, show="*")
entry_password.grid()

# Tombol login
btn_login = tk.Button(login_frame, text="Login", command=login)
btn_login.grid()

# Buat frame untuk aplikasi utama
main_app_frame = tk.Frame(root)

# Mengubah warna latar belakang jendela utama (root)
set_background_color(main_app_frame, "lightblue")
```



```
label_texts = [
    "Name :",
    "NIK :",
    "Tempat Lahir :",
    "Tanggal Lahir :",
    "Alamat :"]

# Calculate the maximum length of label texts
max_label_width = max(len(text) for text in label_texts)

# Create and align the labels
# for row, label_text in enumerate(label_texts):
#     label = tk.Label(main_app_frame, text=label_text)
#     label.grid(row=1, column=0, padx=5, pady=5, sticky="w")
#     label.config(width=max_label_width) # Set a common width for all labels

# Label
for row, label_text in enumerate(label_texts):
    name_label = tk.Label(main_app_frame, text="Name :")
    name_label.grid(row=3, column=0, padx=5, pady=5, sticky="w")
    name_label.config(width=max_label_width) # Set a common width for all labels

    nik_label = tk.Label(main_app_frame, text="NIK :")
    nik_label.grid(row=4, column=0, padx=5, pady=5, sticky="w")
    nik_label.config(width=max_label_width) # Set a common width for all labels

    tempat_lahir_label = tk.Label(main_app_frame, text="Tempat Lahir :")
    tempat_lahir_label.grid(row=5, column=0, padx=5, pady=5, sticky="w")
    tempat_lahir_label.config(width=max_label_width) # Set a common width for all labels

    tanggal_lahir_label = tk.Label(main_app_frame, text="Tanggal Lahir :")
    tanggal_lahir_label.grid(row=6, column=0, padx=5, pady=5, sticky="w")
    tanggal_lahir_label.config(width=max_label_width) # Set a common width for all labels

    alamat_label = tk.Label(main_app_frame, text="Alamat :")
    alamat_label.grid(row=7, column=0, padx=5, pady=5, sticky="w")
    alamat_label.config(width=max_label_width) # Set a common width for all labels

    data_text_label = tk.Label(main_app_frame, text="Data")
    data_text_label.grid(row=0, column=0, columnspan=2, padx=5, pady=5)
    data_text_label.config(width=max_label_width) # Set a common width for all labels
```

Label dalam penginputan data penduduk

Kotak teks input data penduduk

```
name_entry = tk.Entry(main_app_frame)
name_entry.grid(row=3, column=0, colspan=2, padx=5, pady=5)

nik_entry = tk.Entry(main_app_frame)
nik_entry.grid(row=4, column=0, colspan=2, padx=5, pady=5)

tempat_lahir_entry = tk.Entry(main_app_frame)
tempat_lahir_entry.grid(row=5, column=0, colspan=2, padx=5, pady=5)

tanggal_lahir_entry = tk.Entry(main_app_frame)
tanggal_lahir_entry.grid(row=6, column=0, colspan=2, padx=5, pady=5)

alamat_entry = tk.Entry(main_app_frame)
alamat_entry.grid(row=7, column=0, colspan=2, padx=5, pady=5)
```

```
add_button = tk.Button(main_app_frame, text="Add", command=add_data)
add_button.grid(row=3, column=1, colspan=2, padx=5, pady=5)
add_button.config(width=max_label_width) # Set a common width for all labels

update_button = tk.Button(main_app_frame, text="Update", command=update_data)
update_button.grid(row=4, column=1, colspan=2, padx=5, pady=5)
update_button.config(width=max_label_width) # Set a common width for all labels

delete_button = tk.Button(main_app_frame, text="Delete", command=delete_data)
delete_button.grid(row=5, column=1, colspan=2, padx=5, pady=5)
delete_button.config(width=max_label_width) # Set a common width for all labels

search_button = tk.Button(main_app_frame, text="Search", command=search_data)
search_button.grid(row=6, column=1, colspan=2, padx=5, pady=5)
search_button.config(width=max_label_width) # Set a common width for all labels
```

Tombol untuk
melakukan operasi
CRUD

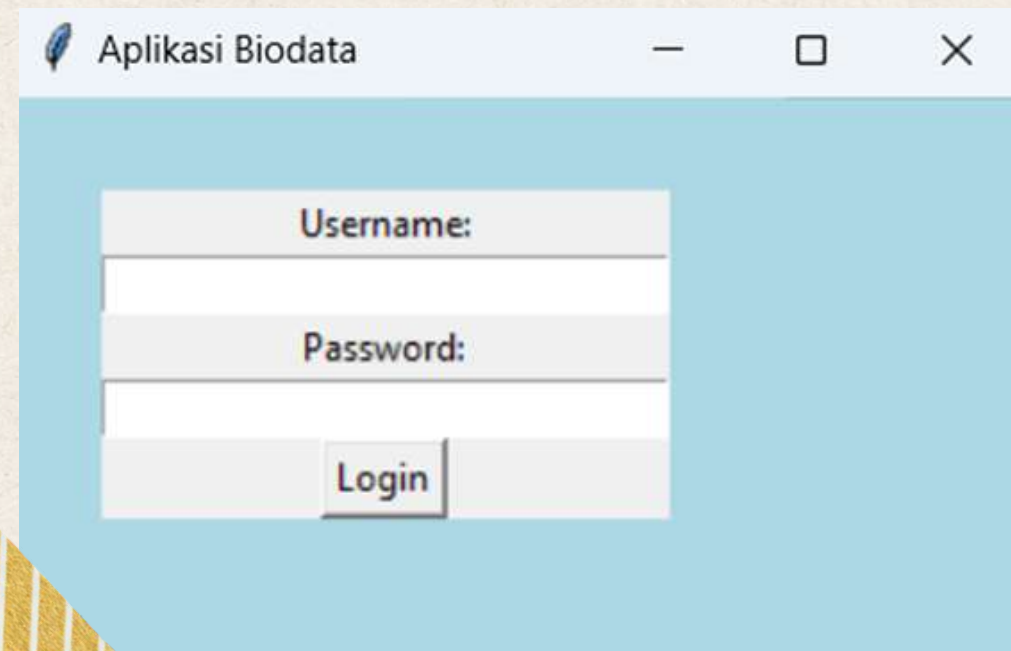
Fungsi untuk
menampilkan data
dan menjalankan
aplikasi

```
# Tombol untuk menampilkan data
show_data_button = tk.Button(main_app_frame, text="Show Data", command=show_data)
show_data_button.grid(row=4, column=3, columnspan=4, padx=5, pady=5)

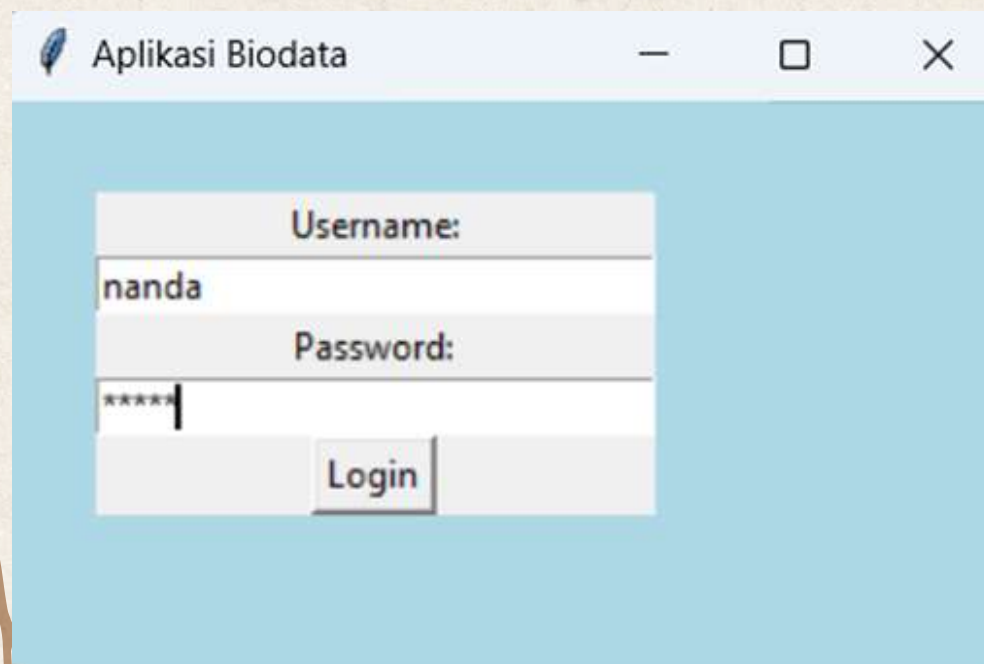
# Jalankan aplikasi
root.mainloop()
```

Setelah semua proses implementasi telah dikerjakan, maka aplikasi sederhana pendataan penduduk dengan menggunakan python dan MongoDB telah berhasil dibuat, diharapkan dengan berhasilnya dibuat aplikasi sederhana pendataan penduduk dapat digunakan dengan baik.

Hasil



Tampilan login untuk
petugas desa



Tampilan login pada
aplikasi sederhana
pendataan penduduk



Tampilan log in gagal jika
username dan password salah



Tampilan menambahkan
data penduduk

Aplikasi Pendataan Penduduk

Data

Name	NIK	Tempat Lahir	Tanggal Lahir	Alamat
nanda	357	Surabaya	08-06-2003	Surabaya
farhan	358	Trenggalek	16-03-2001	Trenggalek
thasya	359	Jakarta	24-07-2002	Gresik
fikrie	360	Medan	12-02-2003	Medan

Show Data

Name: Add

NIK: Update

Tempat Lahir: Delete

Tanggal Lahir: Search

Alamat:

Daftar nama penduduk yang berhasil diinput pada aplikasi

Penduduk.mahasiswa

Documents Data_Penduduk....

Data_Penduduk.mahasiswa

3 DOCUMENTS 1 INDEXES

Documents Aggregations Schema Explain Plan Indexes Validation

Filter Type a query: { field: 'value' } Reset Find Options

ADD DATA EXPORT DATA 1 - 3 of 3

```

_id: ObjectId('64d102164175bbd989af42e1')
name: "nanda"
nik: "357"
tl: "Surabaya"
tgl: "08-06-2003"
alamat: "Surabaya"

_id: ObjectId('64d102434175bbd989af42e4')
name: "farhan"
nik: "358"
tl: "Trenggalek"
tgl: "16-03-2001"
alamat: "Trenggalek"

```

Daftar nama penduduk yang berhasil diinput pada MongoDB

Aplikasi Pendataan Penduduk

Data

Name	NIK	Tempat Lahir	Tanggal Lahir	Alamat
nanda	357	Surabaya	08-06-2003	Surabaya
farhan	358	Trenggalek	16-03-2001	Trenggalek
thasya	359	Jakarta	24-07-2002	Gresik
fikrie	360	Medan	12-02-2003	Medan

Show Data

Name: Add

NIM: Update

Tempat Lahir: Delete

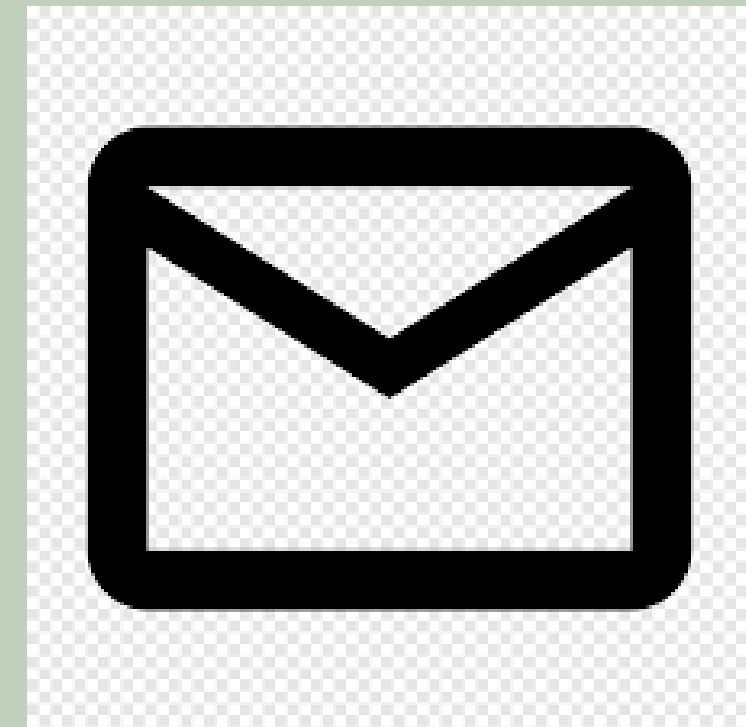
Tanggal Lahir: Search

Alamat:

KESIMPULAN

Pembuatan aplikasi sederhana pendataan penduduk berbasis python dan mongoDB memiliki potensi besar untuk mengatasi tantangan dalam pengelolaan data penduduk. Dengan mengadopsi solusi teknologi informasi, desa dapat meningkatkan akurasi data, efisiensi, dan transparansi, serta merencanakan pembangunan yang lebih tepat sasaran. Manfaat ini tidak hanya dirasakan oleh pemerintah desa, tetapi juga oleh masyarakat dalam bentuk akses yang lebih mudah terhadap informasi dan pemberdayaan teknologi.

Aplikasi yang digunakan



LINK GITHUB

https://github.com/nandatqhsy/APLIKASI_PENDATAAN_PENDUDUK



**Thank
You**

