

LAPORAN TUGAS
MEMBUAT SISTEM PERHITUNGAN AIR DESA DAN KOMPLAIN PENGGUNA



Disusun oleh:

❖ **Fauzi Kurniawan (5230411251)**

PROGRAM STUDI S1 INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS TEKNOLOGI YOGYAKARTA
2024

Penjelasan Kode Program

Program ini menggunakan **Tkinter**, pustaka GUI untuk Python, untuk membuat aplikasi yang memungkinkan pengguna untuk memasukkan data terkait air rumah tangga dan keluhan air. Aplikasi ini menyertakan beberapa fitur utama, seperti menambahkan data, menghapus data, serta menyimpan data ke file Excel. Berikut adalah penjelasan setiap bagian dari kode:

0. Import

```
1. import tkinter as tk
2. from tkinter import ttk, messagebox
3. import pandas as pd
4. import os
5.
```

Bagian ini memberikan Pustaka untuk program ini dan isian tkinter dan pandas serta os dengan masing-masing memiliki fungsi yang berbeda

1. Tk digunakan untuk widget dengan tampilan
2. Messagebox digunakan untuk menampilkan kotak pesan
3. Os digunakan antarmuka untuk berinteraksi dengan sistem operasi

1. Inisialisasi Aplikasi

```
class AppOrder:
    def __init__(self, root):
        self.root = root
        self.root.title("Tools PADKA 1.0")
        self.root.geometry("1000x800")
        self.root.configure(bg="#87CEEB")
        self.widget_create()
```

Bagian ini mendefinisikan kelas AppOrder yang menerima parameter root, yaitu objek utama dari aplikasi. root adalah jendela utama aplikasi yang dibuat menggunakan Tkinter. Di sini juga dilakukan pengaturan judul jendela dan ukuran jendela serta warna untuk background berwarna biru langit

2. Membuat Widget (Elemen GUI)

Metode widget_create digunakan untuk mendefinisikan berbagai elemen antarmuka pengguna (UI) seperti label, input field, tombol, dan tabel.

- **Judul:** Label yang menampilkan judul aplikasi.
- **Input Data:** Tiga elemen utama untuk input:
 - Nama Pemilik Rumah
 - Debit Air dalam 1 Bulan

- Pilihan Keluhan
- **Tombol:** Tombol untuk menambahkan data, menyimpan ke Excel, dan menghapus data.
- **Tabel:** Treeview digunakan untuk menampilkan data yang dimasukkan, yang berisi informasi seperti Nama Pemilik, Debit Air, Total Pembayaran, dan Keluhan.

3. Menambahkan Data

Metode tambah_data digunakan untuk memasukkan data ke dalam tabel setelah memvalidasi input.

```
def tambah_data(self):
    name = self.name_entry.get()
    debit = self.debit_entry.get()
    keluhan = self.Keluhan_Menu.get()

    try:
        debit = float(debit)
        if debit <= 0:
            raise ValueError("Debit harus lebih dari 0.")
        if name and keluhan != "Pilih Keluhan":
            total_pembayaran = f"Rp {debit * 500:,.0f}".replace(',', '.')
            self.data_Tabel.insert("", "end", values=(name, debit,
total_pembayaran, keluhan))
            self.name_entry.delete(0, tk.END)
            self.debit_entry.delete(0, tk.END)
            self.Keluhan_Menu.set("Pilih Keluhan")
        else:
            messagebox.showerror("Error", "Semua kolom harus diisi dengan
benar.")
    except ValueError as e:
        messagebox.showerror("Error", f"Kesalahan input: {e}")
```

Mengambil Input:

- Nama pemilik rumah diambil dari `self.name_entry.get()`.
- Debit air diambil dari `self.debit_entry.get()`.
- Keluhan diambil dari `self.Keluhan_Menu.get()`, yang berisi pilihan keluhan yang dipilih pengguna.

try:

debit = float(debit)

if debit <= 0:

raise ValueError("Debit harus lebih dari 0.")

* Code ini digunakan untuk mengecek apakah data yang dimasukkan lebih dari 0 dan jika lebih maka lanjut dan jika tidak maka menampilkan message error

Perhitungan

- Memasukkan debit air per meter dikali dengan 500 rupiah

Menambahkan Data ke dalam Tabel

- `self.data_Tabel.insert("", "end", values=(name, debit, total_pembayaran, keluhan))`

Setelah melakukan inputan dan perhitungan maka data akan dimasukkan ke dalam table menggunakan `self.data_Table.insert` dengan meliputi (nama ,debit,total_pembayaran, Keluhan

Mengrisert Form

- `self.name_entry.delete(0, tk.END)`
- `self.debit_entry.delete(0, tk.END)`
- `self.Keluhan_Menu.set("Pilih Keluhan")`

code ini digunakan untuk mengosongkan data setelah melakukan inputan demham `.delete` dan pilihan diset ulang ke default “Pilih Keluhan”

Menambahkan Tulisan Rp untuk total pembayaran

```
total_pembayaran = f"Rp {debit * 500:,.0f}".replace(',', '.')
```

untuk bagian , adalah memisahkan angka ribuan misalkan 1000 maka 1.000 dan of digunakan untuk membulatkan angka dan tidak desimal

4. Menghapus Data

Metode `hapus_data` memungkinkan pengguna untuk menghapus data yang telah dimasukkan di tabel.

```
def hapus_data(self):
    selected_item = self.data_Tabel.selection()
    if selected_item:
        for item in selected_item:
            self.data_Tabel.delete(item)
    else:
        messagebox.showerror("Error", "Pilih data yang ingin dihapus.")
```

Code ini digunakan untuk menghapus code dengan mengampil id dari item yang dipilih dan jika ada maka program akan menghapus baris menggunakan metode `self.data_Tabel.delete(item)`

5. Mencari data berdasarkan nama

```
def cari_data(self):
    # Membuka jendela input untuk meminta nama pemilik rumah
    cari_window = tk.Toplevel(self.root)
    cari_window.title("Cari Data")

    cari_window.geometry("400x150")
    cari_window.configure(bg="#87CEEB")

    # Label dan entry digunakan untuk mencari pemilik rumah
    label = tk.Label(cari_window, text="Masukkan Nama Pemilik Rumah:",
bg="#87CEEB", font=("Arial", 12))
    label.pack(pady=10)

    nama_input = tk.Entry(cari_window, width=30, font=("Arial", 12))
    nama_input.pack(pady=5)

    # code ini digunakan untuk mencari
    def search():
        nama_cari = nama_input.get().strip()
        for item in self.data_Tabel.get_children():
            values = self.data_Tabel.item(item)['values']
            if nama_cari.lower() in str(values[0]).lower():
                self.data_Tabel.selection_set(item)
                self.data_Tabel.see(item)
                messagebox.showinfo("Hasil Pencarian", f>Data ditemukan:
{values}")

                cari_window.destroy()
                return
        messagebox.showerror("Tidak Ditemukan", f"Tidak ada data dengan nama:
{nama_cari}")
        cari_window.destroy()
```

- **Toplevel**

Ini adalah widget dari tkinter untuk membuat jendela baru

- **nama_cari = nama_input.get().strip()**

code ini digunakan untuk mengambil text yg dimasukkan user

- **Value[0]**

Adalah kolom pertama ditabel yang isinya nama pemilik rumah

- **Windows destroy**

Digunakan untuk menutup setelah pencarian selesai dan return untuk menghentikan fungsi search()

6. Menyimpan Data ke Excel

Mengambil Data dari Tabel:

```
data = []
for item in self.data_Tabel.get_children():
    data.append(self.data_Tabel.item(item)['values'])
```

- Program pertama-tama mengumpulkan semua data yang ada dalam tabel
- (self.data_Tabel). Data diambil menggunakan metode self.data_Tabel.get_children(), yang akan mengembalikan semua item (baris) dalam tabel.
- Kemudian, data untuk setiap item diambil dengan self.data_Tabel.item(item)['values'] yang mengembalikan nilai-nilai kolom dari setiap baris.

Pengecekan Data:

- Jika tidak ada data yang diambil (misalnya, jika tabel kosong), maka akan muncul pesan kesalahan menggunakan messagebox.showerror().

Menyusun Data ke dalam Format DataFrame:

```
df = pd.DataFrame(data, columns=["Nama Pemilik Rumah", "Debit Air (m³)", "Total Pembayaran", "Keluhan"])
```

- Intinya program ini digunakan untuk membuat kolom di excel dengan isian data diatas

Menentukan Lokasi Penyimpanan:

```
documents_path = os.path.expanduser("C:Project GUI")
data_folder = os.path.join(documents_path, "Data")
os.makedirs(data_folder, exist_ok=True)
file_path = os.path.join(data_folder, "Data Air Desa 2024.xlsx")
```

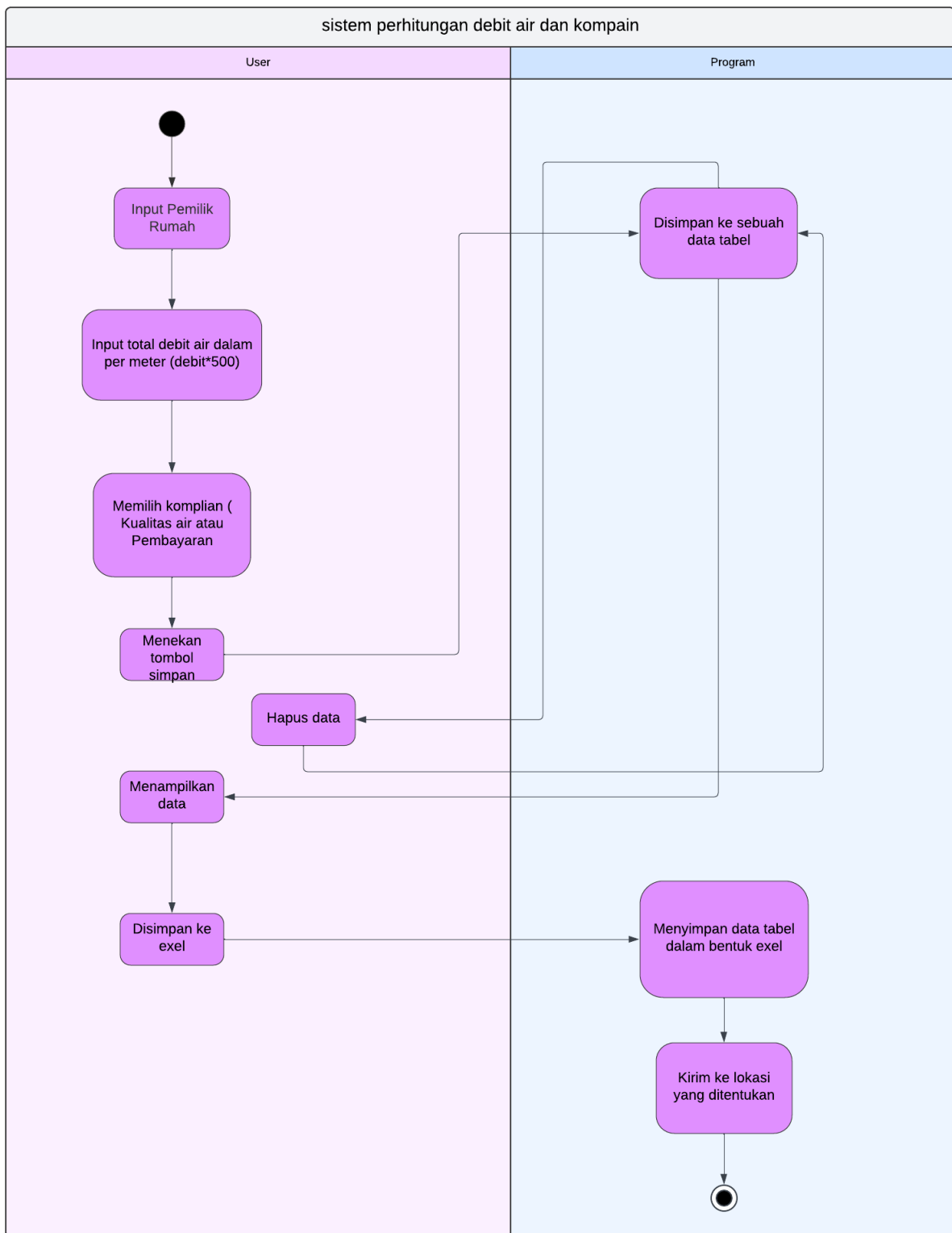
disini fungsi dari import os adalah untuk menyimpan data inputan dari dengan metode os.path.expanduser dan disimpan dengan Lokasi yang ditentukan serta os.path.join berisikan nama file excel yang kita mau.

Menyimpan ke Excel:

```
try:
    df.to_excel(file_path, index=False, engine='openpyxl')
    messagebox.showinfo("Sukses", f"Data berhasil disimpan ke {file_path}!")
except Exception as e:
    messagebox.showerror("Error", f"Terjadi kesalahan: {e}")
```

cara untuk menyimpannya adalah dengan df.to_excel dengan ketentuan jika berhasil maka sukses dan jika tidak maka error

ACTIVITY DIAGRAM



PERCOBAAN

Tampilan utama

Tools PADKA 1.0

Tools Perhitungan Air Desa dan Keluhan Air

Nama Pemilik Rumah:

Debit Air dalam 1 Bulan (m³):

Pilih Keluhan:

Daftar Data

Nama Pemilik	Debit Air (m³)	Total Pembayaran	Keluhan
koi	100.0	Rp 50.000	Masalah Tagihan
mifa	355.0	Rp 177.500	Masalah Tagihan
3242	100.0	Rp 50.000	Kualitas Air
35	5415.0	Rp 2.707.500	Masalah Tagihan
eefwe	55.0	Rp 27.500	Masalah Tagihan
erfer	353.0	Rp 176.500	Kualitas Air

Pencarian data

Daftar Data

Nama Pemilik	Debit Air (m³)	Total Pembayaran	Keluhan
koi	100.0	Rp 50.000	Masalah Tagihan
mifa	355.0	Rp 177.500	Masalah Tagihan
3242	100.0	Rp 50.000	Kualitas Air
35	5415.0	Rp 2.707.500	Masalah Tagihan
eefwe	55.0	Rp 27.500	Masalah Tagihan
erfer	353.0	Rp 176.500	Kualitas Air

Hasil Pencarian

Data ditemukan: [35, '5415.0', 'Rp 2.707.500', 'Masalah Tagihan']

Hapus data

Tools Perhitungan Air Desa dan Keluhan Air

Nama Pemilik Rumah:

Debit Air dalam 1 Bulan (m³):




Pilih Keluhan:

Daftar Data

Nama Pemilik	Debit Air (m³)	Total Pembayaran	Keluhan
koi	100.0	Rp 50.000	Masalah Tagihan
mifa	355.0	Rp 177.500	Masalah Tagihan
3242	100.0	Rp 50.000	Kualitas Air
eefwe	55.0	Rp 27.500	Masalah Tagihan
erfer	353.0	Rp 176.500	Kualitas Air

Simpan di folder c:Project GUI

*Berhasil namun python eror saat menyimpan

Name	Date modified	Type	Size
 .env	29/11/2024 21:37	File folder	
 Data Air Desa 2024.xlsx	29/11/2024 21:35	Microsoft Excel Work...	6 KB
 M8_5230411251_Fauzi_Kurniawan.py	29/11/2024 21:36	Python Source File	9 KB