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

Latar Belakang

Dibayak daerah air menjadi salah satu isu yang sering dikeluhkan oleh masayarakat. Keluhan ini sering meluputi kesulitan dalam memperoleh air bersih, dan sering matinya aliran air, selain itu tagihan air juga menjadi momok bagi warga sekitar dan banyak yang kaget akan jumlah total yang harus dibayar dalam satu bulan, tidak adanya informasi yang transparan mengenai tarif, serta ketidaksesuaian antara pemakaian dan jumlah tagihan yang dikenakan.

Hal ini menyebabkan ketidakpuasan bagi pengguna warga sekitar, yang merasa kesulitan dalam mengetahui dengan pasti berapa biaya yang harus dibayar dan apakah jumlah tersebut sesuai dengan pemakaian mereka. Sebagai dampaknya, munculnya ketidakpercayaan terhadap pihak penyedia layanan air, dan meningkatnya permintaan untuk adanya sistem yang lebih jelas, transparan, dan akuntabel.

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 dab pandas serta os dengan masing-masing memiliki fungsi yang berbeda

1. Ttk digunakkan untuk widget dengan tampilan
2. Messegebox digunakkan 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 digunakkan untukmemasukkan data ke dalam tabel setelah menvalidasi 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 = debit * 500
                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"ada 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 digunakkan untuk mengecek apakah data yang dimasukkan lebih dari 0 dan jika lebih maka lanjut dan jika tidak maka menampilkan massage 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”

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. 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 digunakkan untuk membuat kolom di exel dengan isian data diatas

Menentukan Lokasi Penyimpanan:

```
       documents_path = os.path.expanduser("C:\Users\Acer Helios Neo
16\Documents\Data Tugas dan Materi Kuliah\Data Semester 3\Pemograman
Berorientasi Objek Praktik\GUI Project")
        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")
```

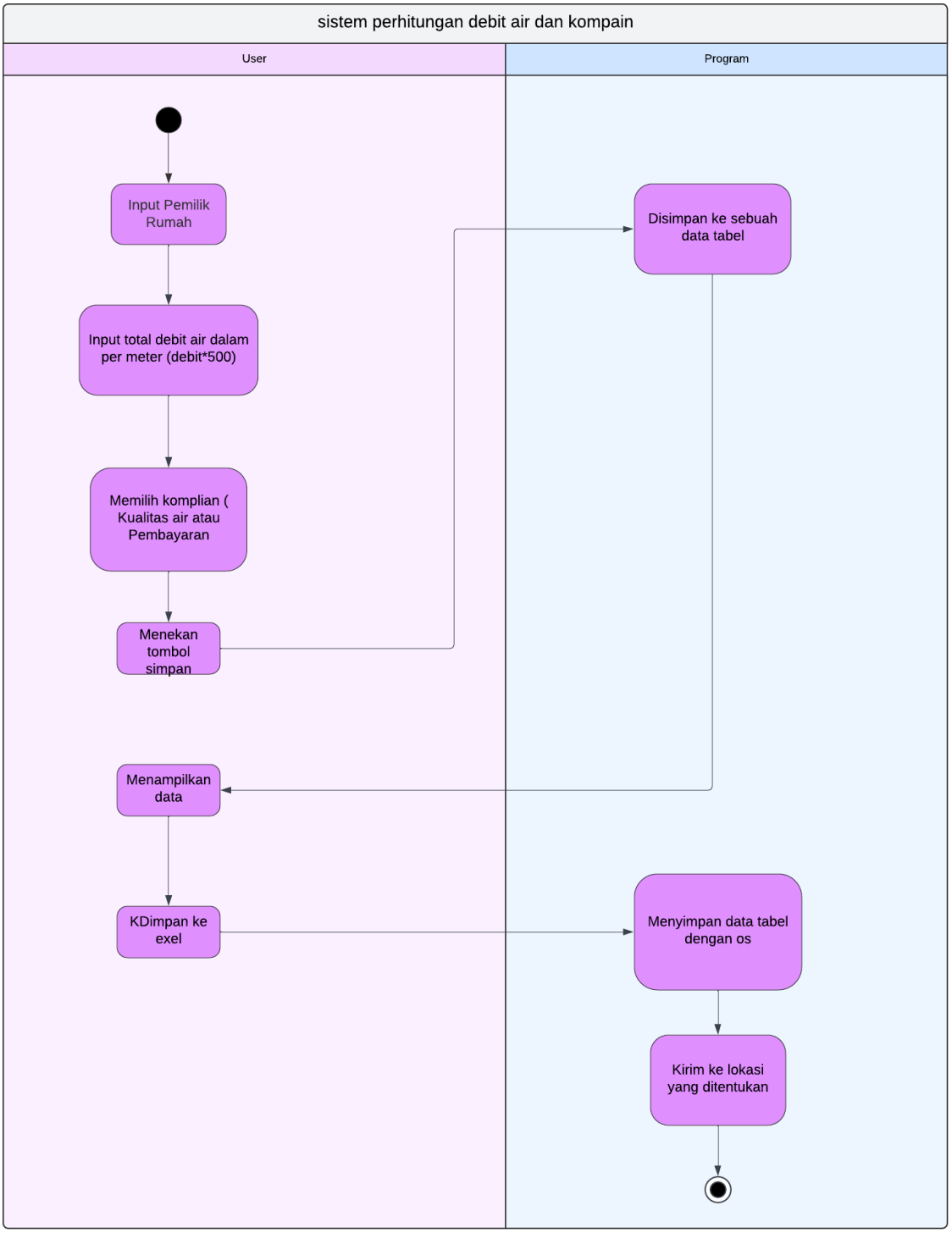
disini fungsi dari import os adalah untuk menyimpan data inputan dari dengan metode os.path.expanduser dan disimpan dengan Lokasi yang ditentukan serta os.path.join berisikan nama file exel yang kita mau.

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_exel dengan ketentuan jika berhasil maka sukses dan jika tidak maka error

ACTIVITY DIAGRAM


sistem perhitungan debit air dan kompain
User
Program
Input Pemilik Rumah
Input total debit air dalam per meter (debit*500)
Memilih komplian (Kualitas air atau Pembayaran
Menekan tombol simpan
Disimpan ke sebuah data tabel
Menampilkan data
KDimpan ke exel
Menyimpan data tabel dengan os
Kirim ke lokasi yang ditentukan