

LAPORAN TUGAS PEMROGRAMAN  
BERBASIS OBJEK PRAKTIK  
**“Aplikasi Keluhan Masyarakat”**



AMARULLAH ARSADINATA  
5230411323

PROGRAM STUDI INFORMATIKA FALKUSTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS TEKNOLOGI YOGYAKARTA  
TAHUN AJARAN  
2024/2025

## Pendahuluan

Aplikasi Keluhan Masyarakat adalah sebuah program berbasis GUI (Graphical User Interface) yang dirancang untuk mencatat, mengelola, dan memonitor keluhan yang diajukan oleh masyarakat. Aplikasi ini menggunakan pustaka Tkinter pada Python untuk membangun antarmuka pengguna.

## Penjelasan Program: Aplikasi Keluhan Masyarakat

Dokumen ini berisi penjelasan tentang setiap elemen dan fungsi yang ada di dalam program Aplikasi Keluhan Masyarakat. Tujuan dokumen ini adalah memberikan panduan yang mudah dipahami dan membantu pembaca untuk memahami struktur serta logika dari program.

### 1. Jendela Utama Aplikasi (`root`)

- Jendela utama program yang menjadi wadah seluruh elemen antarmuka pengguna.
- Parameter penting:
  - - `title("Aplikasi Keluhan Masyarakat")`: Menentukan judul aplikasi.
  - - `geometry("700x700")`: Mengatur ukuran jendela (700x700 piksel).

### 2. Label

- Label digunakan untuk menampilkan teks statis pada antarmuka aplikasi.
- Contoh label:
  - - `label\_judul`: Menampilkan "Aplikasi Keluhan Masyarakat".
  - - `label\_nama`: Menampilkan "Nama Pelapor".
  - - `label\_rt\_rw`: Menampilkan "RT/RW Pelapor".
  - - `label\_tanggal`: Menampilkan "Hari/Tanggal/Tahun".
  - - `label\_keluhan`: Menampilkan "Keluhan".
  - - `label\_tabel\_keluhan`: Menampilkan "Daftar Keluhan Masyarakat".

### 3. Entry (Input Teks)

- Input teks digunakan untuk menerima data dari pengguna.
- Field yang tersedia:
  - - `entri\_nama`: Untuk nama pelapor.
  - - `entri\_tanggal`: Untuk tanggal kejadian.
  - - `entri\_keluhan`: Untuk deskripsi keluhan.

### 4. Combobox (Pilihan Dropdown)

- Komponen dropdown untuk memilih RT/RW.
- Contoh: `menu\_rt\_rw` dengan pilihan seperti `01/01`, `01/02`, hingga `10/10`.

## 5. Tabel (Treeview)

- Tabel digunakan untuk menampilkan daftar keluhan masyarakat.
- Kolom dalam tabel:
  - - `Nama`: Nama pelapor.
  - - `RT/RW`: Lokasi pelapor.
  - - `Tanggal`: Hari/Tanggal/Tahun pelaporan.
  - - `Keluhan`: Isi keluhan.
  - - `Status`: Status keluhan (default: "Belum Diterima").
- Fitur tabel:
  - - Menambahkan, menghapus, atau memperbarui data.

## 6. Tombol (Aksi)

- Tombol menyediakan aksi yang dapat dilakukan pengguna.
- - `tombol\_tambah`: Menambahkan keluhan ke tabel.
- - `tombol\_hapus`: Menghapus baris yang dipilih.
- - `tombol\_terima`: Mengubah status keluhan menjadi "Diterima".

## 7. Fungsi-Fungsi

- Logika program dikelola melalui fungsi-fungsi berikut:
- - `tambah\_keluhan`: Menambahkan keluhan ke tabel dengan validasi input.
- - `hapus\_baris\_terpilih`: Menghapus baris yang dipilih di tabel.
- - `tandai\_diterima`: Memperbarui status keluhan menjadi "Diterima".

## 8. Validasi Input

- Validasi memastikan semua input terisi sebelum menambahkan data ke tabel.
- Jika ada input kosong, akan muncul pesan error menggunakan `messagebox.showerror`.

## 9. Struktur Program

- Bagian utama program (`\_\_main\_\_`) melakukan langkah berikut:
- - Membuat objek `Tk()` sebagai jendela utama.
- - Memulai aplikasi dengan `root.mainloop()`.

## Deskripsi Fungsi

### 1. `__init__(self, root)`

Fungsi ini adalah konstruktor kelas yang bertanggung jawab untuk menginisialisasi jendela aplikasi. Fungsi ini mengatur judul, ukuran jendela, dan memanggil fungsi `buat_widget` untuk membuat antarmuka.

*Kode Fungsi: `__init__(self, root)`*

```
def __init__(self, root):  
    self.root = root  
  
    self.root.title("Aplikasi Keluhan Masyarakat")  
  
    self.root.geometry("700x700")  
  
    self.buat_widget()
```

### 2. `buat_widget(self)`

Fungsi ini digunakan untuk membuat elemen-elemen antarmuka pengguna seperti label, input teks, combo box untuk RT/RW, tabel daftar keluhan, dan tombol-tombol untuk interaksi pengguna.

*Kode Fungsi: `buat_widget(self)`*

```
def buat_widget(self):  
    # Label Judul  
  
    label_judul = tk.Label(self.root, text="Aplikasi  
Keluhan Masyarakat", font=("Arial", 16, "bold"))  
  
    label_judul.pack(pady=10)  
  
    # Nama Pelapor
```

```
        label_nama = tk.Label(self.root, text="Nama  
Pelapor:")  
  
        label_nama.pack()  
  
        self.entri_nama = tk.Entry(self.root, width=40)  
        self.entri_nama.pack(pady=5)  
  
        # RT/RW Pelapor  
  
        label_rt_rw = tk.Label(self.root, text="RT/RW  
Pelapor:")  
  
        label_rt_rw.pack()  
  
        opsi_rt_rw = [f"{i:02}/{j:02}" for i in range(1, 11)  
for j in range(1, 11)]  
  
        self.menu_rt_rw = ttk.Combobox(self.root,  
values=opsi_rt_rw, state="readonly", width=15)  
        self.menu_rt_rw.pack(pady=5)  
  
        # Hari/Tanggal/Tahun  
  
        label_tanggal = tk.Label(self.root,  
text="Hari/Tanggal/Tahun:")  
  
        label_tanggal.pack()  
  
        self.entri_tanggal = tk.Entry(self.root, width=40)
```

```
self.entri_tanggal.pack(pady=5)

# Keluhan

label_keluhan = tk.Label(self.root, text="Keluhan:")
label_keluhan.pack()

self.entri_keluhan = tk.Entry(self.root, width=40)
self.entri_keluhan.pack(pady=5)

# Tabel Daftar Keluhan

label_tabel_keluhan = tk.Label(self.root,
text="Daftar Keluhan Masyarakat:")

label_tabel_keluhan.pack(pady=10)

self.tabel_keluhan = ttk.Treeview(
    self.root,
    columns=("Nama", "RT/RW", "Tanggal", "Keluhan",
"Status"),
    show="headings"
)

self.tabel_keluhan.heading("Nama", text="Nama
Pelapor")
```

```
        self.tabel_keluhan.heading("RT/RW", text="RT/RW  
Pelapor")

        self.tabel_keluhan.heading("Tanggal",  
text="Hari/Tanggal/Tahun")

        self.tabel_keluhan.heading("Keluhan",  
text="Keluhan")

        self.tabel_keluhan.heading("Status", text="Status")

        self.tabel_keluhan.pack(pady=10, fill="both",  
expand=True)

        # Tombol Tambah Keluhan

        tombol_tambah = tk.Button(self.root, text="Tambah  
Keluhan", command=self.tambah_keluhan)

        tombol_tambah.pack(pady=5)

        # Tombol Hapus Baris Terpilih

        tombol_hapus = tk.Button(self.root, text="Hapus  
Baris Terpilih", command=self.hapus_baris_terpilih)

        tombol_hapus.pack(pady=5)
```

```
# Tombol Tandai Diterima

tombol_terima = tk.Button(self.root, text="Tandai
Diterima", command=self.tandai_diterima)

tombol_terima.pack(pady=5)
```

### 3. tambah\_keluhan(self)

Fungsi ini menambahkan data keluhan baru ke tabel. Data yang ditambahkan berasal dari input pengguna. Jika input tidak lengkap, akan muncul pesan error. Data yang dimasukkan memiliki status awal 'Belum Diterima'.

*Kode Fungsi: tambah\_keluhan(self)*

```
def tambah_keluhan(self):

    # Ambil data dari input pengguna

    nama = self.entri_nama.get()

    rt_rw = self.menu_rt_rw.get()

    tanggal = self.entri_tanggal.get()

    keluhan = self.entri_keluhan.get()


    # Validasi input

    if not nama or not rt_rw or not tanggal or not
keluhan:

        messagebox.showerror("Error", "Semua kolom harus
diisi!")

    return
```



```

        # Tambahkan data ke tabel dengan status awal "Belum
Diterima"

        self.tabel_keluhan.insert("", "end", values=(nama,
rt_rw, tanggal, keluhan, "Belum Diterima"))

    # Bersihkan input

    self.entri_nama.delete(0, tk.END)

    self.menu_rt_rw.set("")

    self.entri_tanggal.delete(0, tk.END)

    self.entri_keluhan.delete(0, tk.END)

```

#### 4. hapus\_baris\_terpilih(self)

Fungsi ini menghapus baris yang dipilih pengguna dari tabel keluhan. Jika tidak ada baris yang dipilih, akan muncul pesan error.

*Kode Fungsi: hapus\_baris\_terpilih(self)*

```

def hapus_baris_terpilih(self):

    # Ambil baris yang dipilih oleh pengguna

    item_terpilih = self.tabel_keluhan.selection()

    # Validasi apakah ada baris yang dipilih

    if not item_terpilih:

```

```

        messagebox.showerror("Error", "Pilih baris yang
ingin dihapus!")

    return

    # Hapus baris yang dipilih

    for item in item_terpilih:

        self.tabel_keluhan.delete(item)

```

### 5. tandai\_diterima(self)

Fungsi ini memperbarui status pada baris yang dipilih menjadi 'Diterima'. Jika tidak ada baris yang dipilih, akan muncul pesan error.

*Kode Fungsi: tandai\_diterima(self)*

```

def tandai_diterima(self):

    # Ambil baris yang dipilih oleh pengguna

    item_terpilih = self.tabel_keluhan.selection()

    # Validasi apakah ada baris yang dipilih

    if not item_terpilih:

        messagebox.showerror("Error", "Pilih baris yang
ingin ditandai sebagai diterima!")

    return

```

```
# Perbarui status menjadi "Diterima"

for item in item_terpilih:

    nilai = self.tabel_keluhan.item(item, "values")

    nilai_diperbarui = (*nilai[:-1], "Diterima")

    self.tabel_keluhan.item(item,
values=nilai_diperbarui)
```

## Penutup

Aplikasi ini dirancang untuk membantu mempermudah masyarakat dalam menyampaikan keluhan secara sistematis dan memudahkan pengelola dalam mencatat dan memonitor status keluhan tersebut.