

LAPORAN STUDI KASUS PYTHON GUI APLIKASI TILANG

Pemrograman Berbasis Objek Praktik



Disusun Oleh :

Hafizh Habiibi Lubis 5230411296

**Fakultas Sains dan Teknologi
Program Studi Informatika
Universitas Teknologi Yogyakarta
2024**

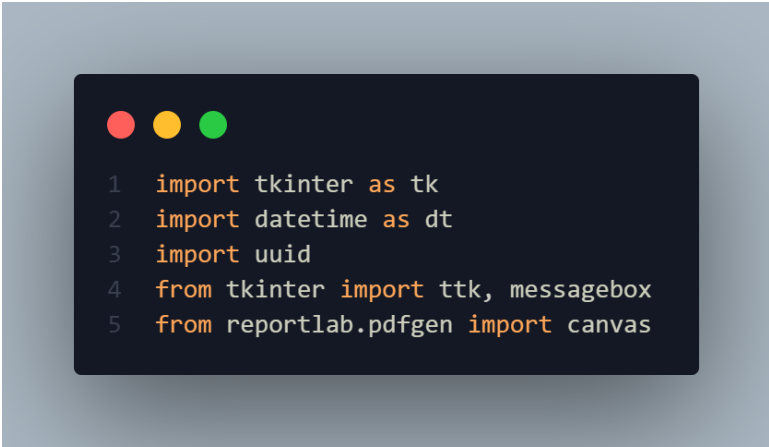
LATAR BELAKANG

Pelanggaran lalu lintas merupakan masalah yang marak terjadi di berbagai negara, termasuk Indonesia. Untuk mengatasi hal ini, pihak kepolisian melakukan operasi penilangan dengan harapan dapat menekan angka pelanggaran. Namun, proses penerbitan tiket tilang masih dilakukan secara manual, yang berisiko menimbulkan kesalahan dan sering kali memakan waktu cukup lama, sehingga terkesan membuang-buang waktu. Oleh karena itu, dibutuhkan sebuah aplikasi untuk mempermudah penulisan dan pencetakan tiket tilang, yang dapat menghemat waktu dan meminimalisir kesalahan.

Aplikasi tilang berbasis Python ini dirancang dengan fitur untuk menginputkan data diri pelanggar, data kendaraan, dan denda sesuai dengan pelanggaran yang dilakukan. Aplikasi ini juga dilengkapi dengan sistem validasi untuk mencegah kesalahan input yang mungkin terjadi. Selain itu, terdapat halaman riwayat yang memungkinkan petugas untuk mengecek tiket tilang yang telah dikeluarkan sebelumnya. Aplikasi ini juga dilengkapi dengan fitur pencetakan tiket tilang dalam format PDF untuk mempermudah proses administrasi.

PEMBAHASAN


1. Import Library



```
1 import tkinter as tk
2 import datetime as dt
3 import uuid
4 from tkinter import ttk, messagebox
5 from reportlab.pdfgen import canvas
```

Kode diatas digunakan untuk mengimport library yang akan digunakan pada aplikasi tilang, library yang saya import tkinter, datetime, uuid, reportlab.pdfgen.

2. Buat Kelas TilangApp



```
1 class TilangApp:
2     def __init__(self, root):
3         self.root = root
4         self.root.title("SITILANG ODNI")
5         self.root.geometry("1366x768")
6         self.root.configure(bg = "white")
7
8         self.cek_nik = []
9         self.cek_plat = []
10        self.default_font = ("Courier New", 10)
11        self.widget_create()
```

Kode diatas digunakan untuk membuat kelas TilangApp dimana didalam kelas ini akan diisi fungsi atau method untuk membangun aplikasi. Didalam class ini juga terdapat method konstruktor yang menginialisasi window untuk aplikasi, list kosong untuk melakukan validasi, mengatur font default pada aplikasi, serta menginialisasi widget.

3. Method widget_create

```
1 # Tampilan
2 def widget_create(self):
3     # Title
4     title_label = tk.Label(text="SATUAN LALU LINTAS KEPOLISIAN NEGERI ODNI", font=("Courier New", 20, "bold"), bg="white")
5     title_app = tk.Label(text="SITILANG ODNI", font=("Courier New", 18, "bold"), bg="white")
6     title_label.pack()
7     title_app.pack()
```

Method widget_create digunakan untuk membuat tampilan pada aplikasi dengan bantuan librari tkinter, disini kita bisa mengatur desain aplikasi sesuai dengan selera dan fungsionalitasnya.

```
1 # Frame
2 frame_left = tk.Frame(root, highlightbackground="black", highlightthickness=3)
3 frame_left.pack(side="left", padx=20, pady=20, anchor="n", expand=True, fill="both")
4
5 frame_right = tk.Frame(root, highlightbackground="black", highlightthickness=3)
6 frame_right.pack(side="right", padx=20, pady=20, anchor="n", expand=True, fill="both")
```

Baris kode diatas digunakan untuk membuat frame yang berfungsi untuk membagi tempat untuk fitur fitur yang akan ditambahkan nantinya, dengan frame ini kita bisa mengelompokkan menu menu yang ada pada aplikasi. Disini saya membagi frame menjadi 2 bagian yaitu kanan dan kiri.

```
1 # Menu Terdakwa
2 td_menu = tk.Label(frame_left, text="DATA DIRI TERDAKWA", font=("Courier New", 12, "bold"))
3 td_menu.pack(anchor="center", pady=20)
4
5 # Nama Terdakwa
6 name_label = tk.Label(frame_left, text="NAMA TERDAKWA", font=self.default_font)
7 name_label.pack(anchor="w", padx=10)
8 self.name_var = tk.StringVar()
9 self.name_entry = tk.Entry(frame_left, textvariable=self.name_var)
10 self.name_entry.pack(fill="x", anchor="w", padx=10, pady=2)
11
12 # NIK Terdakwa
13 nik_label = tk.Label(frame_left, text="NIK TERDAKWA", font=self.default_font)
14 nik_label.pack(anchor="w", padx=10, pady=8)
15 self.nik_var = tk.StringVar()
16 self.nik_entry = tk.Entry(frame_left, textvariable=self.nik_var)
17 self.nik_entry.pack(fill="x", anchor="w", padx=10, pady=2)
18
19 # Alamat Terdakwa
20 alm_label = tk.Label(frame_left, text="ALAMAT TERDAKWA", font=self.default_font)
21 alm_label.pack(anchor="w", padx=10, pady=8)
22 self.alm_var = tk.StringVar()
23 self.alm_entry = tk.Entry(frame_left, textvariable=self.alm_var)
24 self.alm_entry.pack(fill="x", anchor="w", padx=10, pady=2)
```

Kode diatas digunakan untuk mendesain menu terdakwa yang terdiri dari entry nama, nik, dan alamat terdakwa. Menu ini juga disertai dengan title untuk memudahkan pengguna. Selain itu kita bisa mengatur menu ini akan diletakkan pada frame yang mana

```
1 # Menu Kendaraan
2 td_menu = tk.Label(frame_left, text="DATA KENDARAAN", font=("Courier New", 12, "bold"))
3 td_menu.pack(anchor="center", pady=20)
4
5 # Jenis Kendaraan
6 jnk_list = ["Motor", "Mobil", "Truck"]
7 jnk_label = tk.Label(frame_left, text="JENIS KENDARAAN", font=self.default_font)
8 jnk_label.pack(anchor="w", padx=10, pady=8)
9 self.jnk_menu = ttk.Combobox(frame_left, values=jnk_list, font=self.default_font)
10 self.jnk_menu.pack(fill="x", anchor="w", padx=10, pady=2)
11 self.jnk_menu.bind("<<ComboboxSelected>>", self.update_pelanggaran)
12
13 # Nomor Plat Kendaraan
14 plt_label = tk.Label(frame_left, text="NO PLAT KENDARAAN", font=self.default_font)
15 plt_label.pack(anchor="w", padx=10, pady=8)
16 self.plt_var = tk.StringVar()
17 self.plt_entry = tk.Entry(frame_left, textvariable=self.plt_var)
18 self.plt_entry.pack(fill="x", anchor="w", padx=10, pady=2)
19
20 # Merk Kendaraan
21 mrk_label = tk.Label(frame_left, text="MERK KENDARAAN", font=self.default_font)
22 mrk_label.pack(anchor="w", padx=10, pady=8)
23 self.mrk_var = tk.StringVar()
24 self.mrk_entry = tk.Entry(frame_left, textvariable=self.mrk_var)
25 self.mrk_entry.pack(fill="x", anchor="w", padx=10, pady=2)
26
27 # Seri Kendaraan
28 seri_label = tk.Label(frame_left, text="SERI KENDARAAN", font=self.default_font)
29 seri_label.pack(anchor="w", padx=10, pady=8)
30 self.seri_var = tk.StringVar()
31 self.seri_entry = tk.Entry(frame_left, textvariable=self.seri_var)
32 self.seri_entry.pack(fill="x", anchor="w", padx=10, pady=2)
```

Kode diatas digunakan untuk mendesain menu kendaraan terdakwa yang terdiri dari entry nomor plat, merk kendaraan, seri kendaraan. Menu ini juga memiliki combobox untuk memilih jenis kendaraan, combobox ini ter bind dengan method update_pelanggaran untuk menghubungkan antara combobox jenis kendaraan dengan combobox pelanggaran. Pada kode ini juga kita bisa mengatur lokasi frame yang akan ditempati oleh menu ini.

```

1  # Menu Petugas
2      td_menu = tk.Label(frame_right, text="MENU PETUGAS", font=("Courier New", 12, "bold"))
3      td_menu.pack(anchor="center", pady=20)
4
5      # Jenis Pelanggaran 1
6      plg_label = tk.Label(frame_right, text="JENIS PELANGGARAN 1", font=self.default_font)
7      plg_label.pack(anchor="w", padx=10, pady=8)
8      self.plg_menu = ttk.Combobox(frame_right, font=self.default_font)
9      self.plg_menu.pack(fill="x", anchor="w", padx=10, pady=2)
10
11     # Jenis Pelanggaran 2
12     plg_label2 = tk.Label(frame_right, text="JENIS PELANGGARAN 2", font=self.default_font)
13     plg_label2.pack(anchor="w", padx=10, pady=5)
14     self.plg_menu2 = ttk.Combobox(frame_right, font=self.default_font)
15     self.plg_menu2.pack(fill="x", anchor="w", padx=10, pady=2)
16     self.plg_menu2.bind("<<ComboboxSelected>>", self.cek_pelanggaran1)
17
18     # Nama Petugas
19     ptg_label = tk.Label(frame_right, text="NAMA PETUGAS", font=self.default_font)
20     ptg_label.pack(anchor="w", padx=10, pady=8)
21     self.ptg_var = tk.StringVar()
22     self.ptg_entry = tk.Entry(frame_right, textvariable=self.ptg_var)
23     self.ptg_entry.pack(fill="x", anchor="w", padx=10, pady=2)

```

Kode diatas digunakan untuk mendesain menu petugas yang terdiri dari entry nama petugas dan combobox pelanggaran dan pelanggaran 2. Pada combobox pelanggaran 2 terdapat bind yang digunakan untuk menghubungkan pelanggaran 2 dengan method cek_pelanggaran1 dimana nantinya user tidak bisa menginputkan pelanggaran 2 terlebih dahulu sebelum pelanggaran 1 diisi. Pada kode ini juga terdapat pengaturan frame untuk meletakkan menu petugas ini.

```

1  # riwayat
2      rwy_frame = tk.Label(frame_right, text="RIWAYAT TIKET", font=self.default_font)
3      rwy_frame.pack(fill="both", padx=10, pady=5)
4
5      self.rwy_preview = ttk.Treeview(rwy_frame, columns=("No Tiket", "Nama Terdakwa", "Pelanggaran", "Pelanggaran 2"), show="headings")
6      self.rwy_preview.heading("No Tiket", text="No Tiket")
7      self.rwy_preview.heading("Nama Terdakwa", text="Nama Terdakwa")
8      self.rwy_preview.heading("Pelanggaran", text="Pelanggaran")
9      self.rwy_preview.heading("Pelanggaran 2", text="Pelanggaran 2")
10     self.rwy_preview.pack(fill="x", expand=True, padx=10, pady=10)
11
12     # Setting ke tengah
13     self.rwy_preview.column("No Tiket", anchor="center")
14     self.rwy_preview.column("Nama Terdakwa", anchor="center")
15     self.rwy_preview.column("Pelanggaran", anchor="center")
16     self.rwy_preview.column("Pelanggaran 2", anchor="center")

```

Kode diatas digunakan untuk membuat treeview yang digunakan untuk menampilkan riwayat tiket tilang yang dicetak. Didalam treeview ini terdapat header No Tiket, Nama Terdakwa, Pelanggaran, Pelanggaran 2.

Pada kode ini juga diatur posisi treeviewnya berdasarkan frame yang akan diisi.

```
1 # Tombol Cetak
2 ctk_button = tk.Button(frame_right, text="CETAK TIKET", width=50, command=self.buat_pdf)
3 ctk_button.pack(padx=10, pady=5, anchor="center")
4
5 # Tombol Hapus Riwayat
6 reset_button = tk.Button(frame_right, text="RESET", width=50, command=self.hapus_riwayat)
7 reset_button.pack(padx=10, pady=10, anchor="center")
```

Baris kode diatas digunakan untuk membuat button atau tombol yang nantinya akan digunakan oleh user untuk mengonfirmasi perubahan. Saya membuat 2 buah tombol yaitu tombol cetak untuk mencetak tiket dan tombol reset untuk mereset data yang ada pada treeview. Disini juga kita bisa mengatur frame mana yang akan digunakan untuk menempatkan tombol-tombol ini.

4. Method update_pelanggaran

```
1 def update_pelanggaran(self, event):
2     kendaraan = self.jnk_menu.get()
3
4     pelanggaran = {
5         "Motor": [("Surat Tidak Lengkap", 150000), ("Tidak Pakai Helm", 100000), ("Plat Tidak Standar", 100000)],
6         "Mobil": [("Surat Tidak Lengkap", 250000), ("Tidak Pakai Seatbelt", 120000), ("Kebut Kebutan", 200000)],
7         "Truck": [("Surat Tidak Lengkap", 350000), ("Tidak Pakai Seatbelt", 220000), ("Muatan Berlebih", 300000)]
8     }
9
10    if kendaraan in pelanggaran:
11        pelanggaran_list = [f"{plg[0]} - Denda : RP.{plg[1]:,}" for plg in list(pelanggaran[kendaraan])]
12        self.plg_menu['values'] = pelanggaran_list
13        self.plg_menu2['values'] = pelanggaran_list
14
15        #Reset Pilihan ketika pilihan jenis kendaraan berubah
16        self.plg_menu.set('')
17        self.plg_menu2.set('')
```

Method update_pelanggaran digunakan untuk membuat fungsionalitas pada combobox jnk_kendaraan dimana ketika user memilih jenis kendaraan tertentu maka list pelanggaran yang tampil akan berbeda, hal ini akan diterapkan pada combobox pelanggaran dimana terdapat 2 combobox yaitu plg_menu dan plg_menu2. Pada method ini terdapat sebuah dictionary

bernama pelanggaran yang memiliki key jenis kendaraan dan valuenya itu adalah list yang berisi tuple jenis pelanggaran serta dendanya. Selanjutnya terdapat pengondisian if untuk mencocokkan key pada dictionary dan jika cocok program akan mencetak pelanggaran yang dipilih pada combobox dengan format yang sudah diatur. Kemudian terdapat kode untuk menangani ketika jenis kendaraan yang dipilih berubah. Method ini dihubungkan dengan event listener (event) sehingga akan otomatis dipanggil jika terdapat perubahan pada combobox jnk_kendaraan.

5. Method cek_pelanggaran1

```
1 def cek_pelanggaran1(self, event):
2     if not self.plg_menu.get():
3         messagebox.showerror("ERROR", "HARAP ISI PELANGGARAN 1 TERLEBIH DAHULU!")
4         self.plg_menu2.set('')
```

Method cek_pelanggaran1 digunakan untuk menangani jika user menginputkan pelanggaran 2 pada combobox plg_menu2 dimana hal tersebut tidak diperbolehkan. Jika user menginputkan pelanggaran 2 sebelum pelanggaran 1 maka akan muncul sebuah messagebox error. Method ini dihubungkan dengan event listener (event) sehingga akan secara otomatis dipanggil jika pada combobox plg_menu2 diisi sebelum mengisi combobox plg_menu.

6. Method nomor_tiket

```
1 def nomor_tiket(self):
2     now = dt.date.today().strftime("%d-%m-%Y")
3     return f"TILANG-{now}-{str(uuid.uuid4())[4:]}"
```


Method ini digunakan untuk membuat nomor tiket dengan kombinasi TILANG-tanggal hari ini-id, pembuatan nomor tiket ini menggunakan 2 library yaitu datetime untuk mendapatkan tanggal hari ini dan library uuid untuk mendapatkan kode acak.

7. Method ekstrak_denda

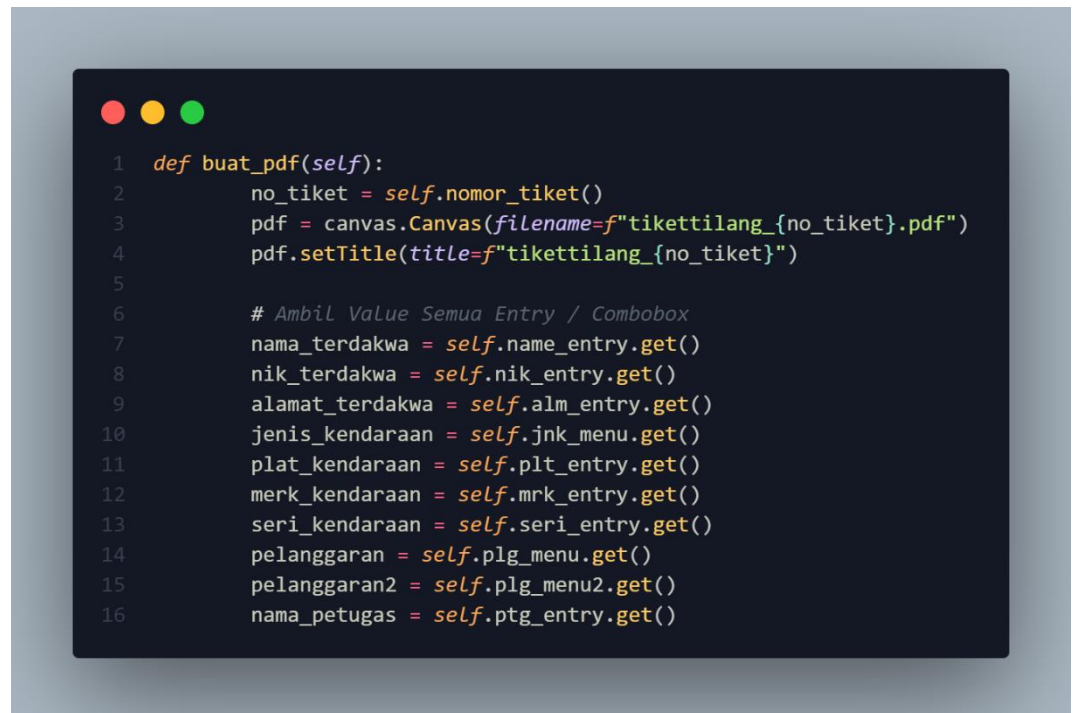


```
1 # Ekstrak Denda dari Pelanggaran
2 def ekstrak_denda(self, pelanggaran):
3     if "Denda : RP." in pelanggaran:
4         try:
5             # Mengambil angka setelah "Denda : RP."
6             return int(pelanggaran.split("Denda : RP.")[-1].replace(",", ""))
7         except ValueError:
8             return 0 # Jika format tidak sesuai, anggap denda 0
9     return 0
```

Method ekstrak_denda digunakan untuk mengolah nilai string yang berasal dari combobox plg_menu dan plg_menu2, yang nantinya akan dikirimkan sebagai parameter pelanggaran dalam fungsi buat_pdf. Setelah string tersebut dimasukkan ke parameter pelanggaran, fungsi ini akan melakukan validasi untuk memastikan formatnya sesuai. Jika string tersebut mengandung teks "Denda : RP.", maka string akan dipecah menggunakan .split("Denda : RP.") untuk mengambil nilai dendanya. Terdapat juga [-1] yang digunakan untuk mendapatkan nilai terakhir dari hasil split. Nilai denda yang diperoleh, misalnya "150,000", akan dihapus tanda komanya dengan .replace(",", ""), sehingga menjadi "150000", dan kemudian dikonversi menjadi integer menggunakan int(). Jika string tidak mengandung "Denda : RP." atau terjadi kesalahan dalam proses, method ini akan mengembalikan nilai 0. Method ekstrak_denda berperan penting dalam mengolah string dari combobox menjadi nilai denda dalam bentuk integer, yang kemudian akan dijumlahkan di dalam method buat_pdf untuk menghitung total denda yang harus dibayarkan. Dengan cara ini, method ini

memastikan bahwa data pelanggaran dapat diproses dengan benar dan dihitung sesuai ketentuan.

8. Method Buat PDF



Method ini digunakan untuk memberikan fungsi pada tombol cetak tiket yang sudah dihubungkan dengan command=, didalam method ini seperti digambar pertama kita memanggil method nomor_tiker dengan memasukkanya kedalam sebuah variabel, kemudian menginisiasi librari reportlaab dengan membuat variabel yang berisi canvas yang merupakan halaman kosong, disini saya sekalian memberikan nama file PDF ketika disimpan dengan kombinasi string dan no_tiket sehingga nama file akan berbeda beda. Selanjutnya kita akan mengambil semua nilai atau value dari entry user dan dimasukan kedalam variabel seperti di gambar.

```

1  # Validasi Form Kosong
2      if not all([nama_terdakwa, nik_terdakwa, alamat_terdakwa, jenis_kendaraan,
3                  plat_kendaraan, merk_kendaraan, seri_kendaraan, nama_petugas]):
4          messagebox.showerror("ERROR", "PASTIKAN SEMUA DATA TERISI!")
5          return
6
7      # Validasi Format NIK
8      if not nik_terdakwa.isdigit() or len(nik_terdakwa) != 5:
9          messagebox.showerror("ERROR", "NIK TERDAKWA HARUS ANGKA DAN BERJUMLAH 5 DIGIT!")
10         return
11
12     # Validasi Pelanggaran
13     if pelanggaran == pelanggaran2:
14         messagebox.showerror("ERROR", "INPUTAN PELANGGARAN HARUS BERBEDA!")
15         self.plg_menu2.set('')
16         return
17
18     # Validasi NIK
19     if nik_terdakwa in self.cek_nik:
20         messagebox.showerror("ERROR", "NIK TERDAKWA SUDAH ADA!")
21         return
22
23     # Validasi Plat
24     if plat_kendaraan in self.cek_plat:
25         messagebox.showerror("ERROR", "NOMOR PLAT SUDAH ADA!")
26         return

```

Pada gambar diatas bias dilihat bahwa pada method buat_pdf terdapat validasi untuk menangani berbagai macam hal seperti jika semua inputan kosong dan user menekan tombol cetak maka muncul peringatan, validasi untuk NIK yang harus sesuai format, validasi ketika user memilih pelanggaran yang sama pad combobox, dan validasi untuk NIK dan nomor plat agar unik dan tidak duplikat.

```

1  # Simpan NIK dan Plat setelah semuanya valid
2      self.cek_plat.append(plat_kendaraan)
3      self.cek_nik.append(nik_terdakwa)

```

Kode diatas berkaitan dengan validasi NIK dan nomor plat yang harus unik, jadi ketika NIK dan nomor plat yang unik dimasukan, nilai atau value dari

entry tadi akan disimpan kedalam list cek_plat dan cek_nik yang ada pada method konstruktor. Sehingga ketika user memasukan NIK atau nomor plat yang sama dua kali program validasi tadi akan membandingkan inputan user tadi dengan isi list cek_plat dan cek_nik.

```
1  # Hapus Isi Form Setelah Pencet Tombol Cetak Tiket
2      self.name_var.set("")
3      self.nik_var.set("")
4      self.alm_var.set("")
5      self.plt_var.set("")
6      self.mrk_var.set("")
7      self.seri_var.set("")
8      self.ptg_var.set("")
9      self.plg_menu.set("")
10     self.plg_menu2.set("")
11     self.jnk_menu.set("")
```

Kode diatas digunakan untuk menghapus isian pada semua entry ketika tombol cetak tiket ditekan sehingga semua entry menjadi bersih kembali dan siap untuk diisi.

```
1  # menghitung total denda
2      denda_1 = self.ekstrak_denda(pelanggaran)
3      denda_2 = self.ekstrak_denda(pelanggaran2) if pelanggaran2 else 0
4      total_denda = denda_1 + denda_2
```

Kode diatas sangat berhubungan dengan method ekstrak_denda, karena nilai parameter pelanggaran di method ekstrak_denda berasal dari sini. Jadi kode ini pada variabel denda1 akan memanggil method ekstrak_denda dan memasukan parameternya yaitu pelanggaran, pelanggaran ini merupakan

variabel yang berisi nilai dari entry `plg_menu` yang didapatkan dengan `.get()`. Hal ini juga berlaku pada variabel `denda2`. Terdapat `if` disana digunakan untuk menangani kondisi jika `pelanggaran2` memiliki nilai maka akan dirporses dengan `self.ekstrak_denda(pelanggaran2)` jika kosong maka nilai akan di set menjadi 0, terakhir `denda1` akan dijumlah dengan `denda2` dan mendapatkan total denda.

```
1  # Header
2      pdf.setFont('Courier', 20)
3      pdf.drawString(180,770, "KEPOLISIAN NEGERI ODNI")
4
5      pdf.setFont('Courier', 12)
6      pdf.drawString(163,750, "SATUAN LALU LINTAS KEPOLISIAN NEGERI ODNI")
7
8      pdf.setFont('Courier', 12)
9      pdf.drawString(225, 730, f"{no_tiket}")
10     pdf.drawString(100, 710, "=*60)
```

```
1  # isi
2      pdf.setFont('Courier', 12)
3      pdf.drawString(100, 670, f>Nama Terdakwa: {nama_terdakwa}")
4      pdf.drawString(100, 650, f"NIK: {nik_terdakwa}")
5      pdf.drawString(100, 630, f"Alamat: {alamat_terdakwa}")
6      pdf.drawString(100, 610, f"Jenis Kendaraan: {jenis_kendaraan}")
7      pdf.drawString(100, 590, f"Nomor Plat: {plat_kendaraan}")
8      pdf.drawString(100, 570, f"Merk: {merk_kendaraan}")
9      pdf.drawString(100, 550, f"Seri: {seri_kendaraan}")
10     pdf.drawString(100, 530, f"Pelanggaran: {pelanggaran}")
11     if pelanggaran2:
12         pdf.drawString(100, 510, f"Pelanggaran 2: {pelanggaran2}")
13
14     pdf.drawString(225, 470, f"Total Denda : RP.{total_denda:,}")
15
16     pdf.setFont('Courier',15)
17     pdf.drawString(140, 430, "SEGERA BAYARKAN UANG DENDA KE PETUGAS :)")
```

```

1  # Footer
2      pdf.setFont('Courier', 12)
3      pdf.drawString(420, 390, "PETUGAS")
4      pdf.drawString(420, 370, f"{nama_petugas}")
5      pdf.drawString(100, 350, "="*60)
6
7      pdf.save()

```

Kode diatas digunakan untuk mendesain header, isi, dan footer pada tiket tilang, pdf disini merupakan variabel yang berisi canvas dari library reportlab, disini kita juga harus mengatur koordinat teks secara manual untuk mendapatkan posisi yang pas. Pada bagian paling bawah di bagian footer terdapat pdf.save() kode ini berfungsi untuk mengexport desain PDF ini menjadi file PDF.

```

1  # tambah ke treeview
2      self.rwy_preview.insert("", "end", values=(no_tiket, nama_terdakwa, pelanggaran, pelanggaran2))

```

Kode diatas digunakan untuk menambahkan data ke dalam treeview berdasarkan values yang diminta.

9. Method hapus_riwayat

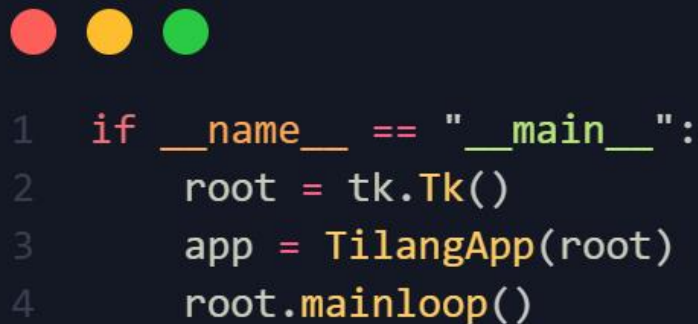
```

1  # Hapus Isi Treeview + Validasi
2      def hapus_riwayat(self):
3          confirm = messagebox.askquestion("", "YAKIN HAPUS SEMUA RIWAYAT?")
4          if confirm == "YES".lower():
5              for item in self.rwy_preview.get_children():
6                  self.rwy_preview.delete(item)

```

Method `hapus_riwayat` digunakan untuk menghapus riwayat cetak tiket tilang yang ada pada treeview. Method ini akan dipanggil ketika tombol reset ditekan karena didalam konfigurasi tombol reset terdapat `command=hapus_riwayat` yang digunakan untuk menghubungkan tombol reset dengan method `hapus_riwayat`. Pada method ini terdapat validasi dimana jika user menekan yes maka semua data akan dihapus, `.get_children()` disini digunakan untuk mendapatkan semua item yang ada di treeview yang nantinya akan dihapus.

10. Membuat Objek

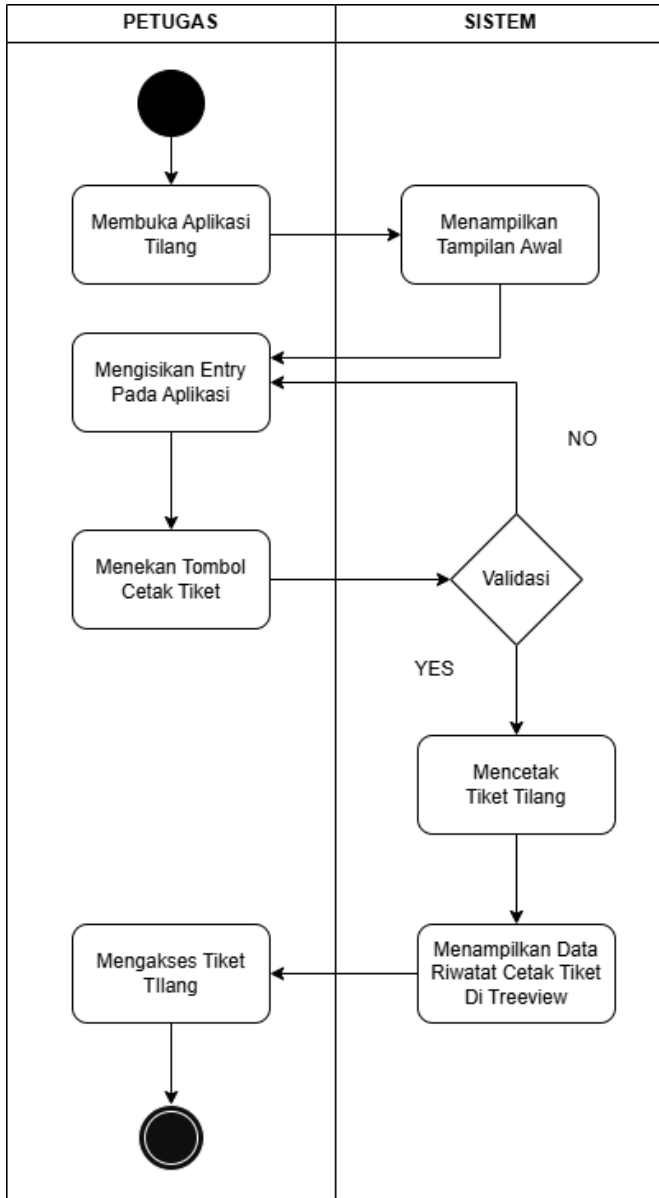


```
1  if __name__ == "__main__":  
2      root = tk.Tk()  
3      app = TilangApp(root)  
4      root.mainloop()
```

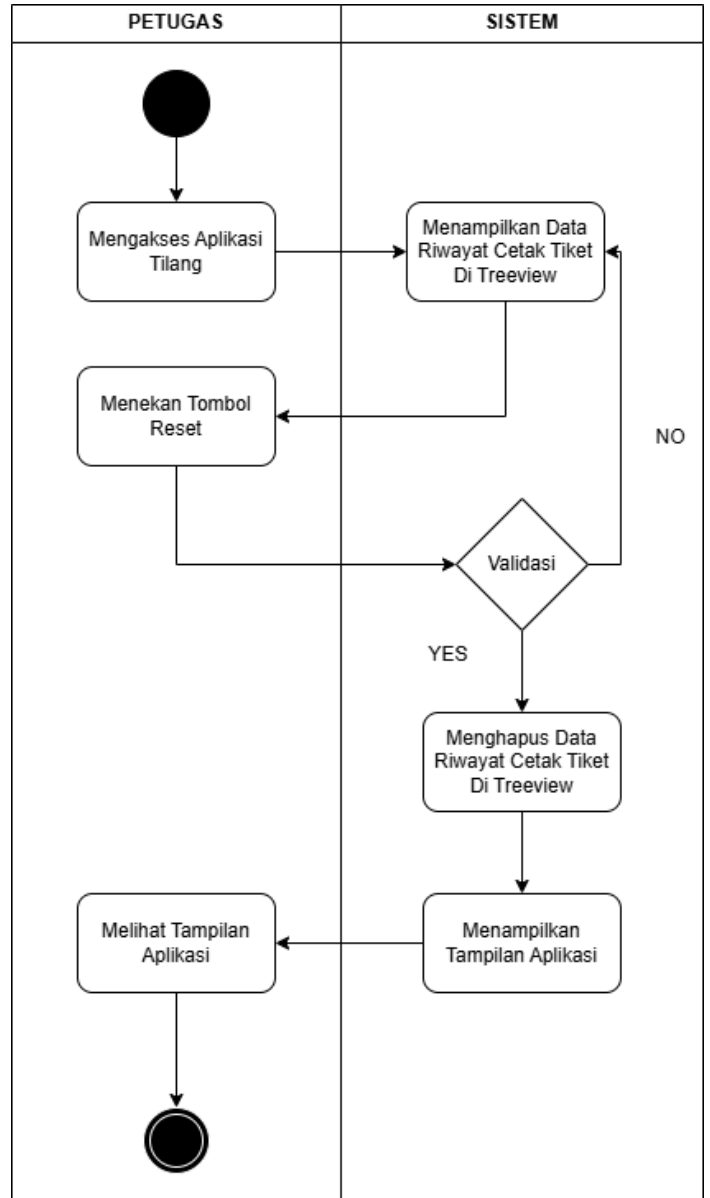
Kode ini digunakan untuk memulai dan menjalankan aplikasi Tkinter. Ketika file ini dijalankan, ia akan membuat jendela utama (`root`), menginisialisasi aplikasi dengan kelas `TilangApp`, dan menjalankan loop utama Tkinter untuk menunggu interaksi pengguna. Tanpa `mainloop()`, aplikasi Tkinter tidak akan tampil dan tidak dapat berfungsi. `root = tk.Tk()` digunakan untuk membuat objek `Tk()` yang mewakili jendela utama aplikasi Tkinter.

ACTIVITY DIAGRAM

Mencetak Tiket Tilang



Hapus Riwayat Cetak Tiket



IMPLEMENTASI

1. Membuat Tiket Tilang 2 Pelanggaran

The screenshot shows a web application window titled "SITILANG ODNI". The main header reads "SATUAN LALU LINTAS KEPOLISIAN NEGERI ODNI" and "SITILANG ODNI". The interface is divided into two main panels. The left panel, titled "DATA DIRI TERDAKWA", contains two sections: "DATA DIRI TERDAKWA" with fields for "NAMA TERDAKWA", "NIK TERDAKWA", and "ALAMAT TERDAKWA"; and "DATA KENDARAAN" with fields for "JENIS KENDARAAN" (a dropdown menu), "NO PLAT KENDARAAN", "MERK KENDARAAN", and "SERI KENDARAAN". The right panel, titled "MENU PETUGAS", contains two dropdown menus for "JENIS PELANGGARAN 1" and "JENIS PELANGGARAN 2", a text field for "NAMA PETUGAS", and a table with four columns: "No Tiket", "Nama Terdakwa", "Pelanggaran", and "Pelanggaran 2". Below the table are two buttons: "CETAK TIKET" and "RESET".

Setelah menjalankan aplikasi tilang maka akan muncul tampilan awal seperti gambar diatas.

The screenshot shows the same application window as before, but with data entered into the form. In the "DATA DIRI TERDAKWA" section, the fields are filled with: "RRI" for "NAMA TERDAKWA", "12345" for "NIK TERDAKWA", "KULON PROGO" for "ALAMAT TERDAKWA", "Mobil" for "JENIS KENDARAAN", "AB 12345 CB" for "NO PLAT KENDARAAN", "LEXUS" for "MERK KENDARAAN", and "LFA" for "SERI KENDARAAN". In the "MENU PETUGAS" section, the "JENIS PELANGGARAN 1" dropdown is set to "Surat Tidak Lengkap - Denda : RP.250,000" and the "JENIS PELANGGARAN 2" dropdown is set to "Rebut Kebutan - Denda : RP.200,000". The "NAMA PETUGAS" field is filled with "CikQ". The table remains empty. The "CETAK TIKET" and "RESET" buttons are still present at the bottom.

Isikan entry dengan lengkap lalu klik tombol Cetak Tiket.

SITILANG ODNI
— □ ×

SATUAN LALU LINTAS KEPOLISIAN NEGERI ODNI
SITILANG ODNI

DATA DIRI TERDAKWA

NAMA TERDAKWA

NIK TERDAKWA

ALAMAT TERDAKWA

DATA KENDARAAN

JENIS KENDARAAN

NO PLAT KENDARAAN

MERK KENDARAAN

SERI KENDARAAN

MENU PETUGAS

JENIS PELANGGARAN 1

JENIS PELANGGARAN 2

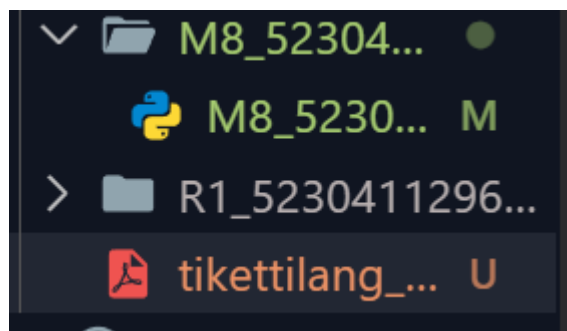
NAMA PETUGAS

No Tiket	Nama Terdakwa	Pelanggaran	Pelanggaran 2
TILANG-29-11-2024-a332	RIRI	Surat Tidak Lengkap - Denda : RP.250,000	Kebut Kebutan - Denda : RP.200,000

CETAK TIKET

RESET

Jika sudah maka akan muncul tampilan seperti berikut, beberapa data yang ada pada entry akan ditampilkan ke dalam treeview riwayat. Dan file tiket tilang akan dicetak kedalam format pdf.



Akan muncul file baru dengan nama tikettilang_id yang berisi data tiket tilang yang sudah di masukan tadi.

KEPOLISIAN NEGERI ODNI
SATUAN LALU LINTAS KEPOLISIAN NEGERI ODNI
TILANG-29-11-2024-a332

Nama Terdakwa: RIRI
NIK: 12345
Alamat: KULON PROGO
Jenis Kendaraan: Mobil
Nomor Plat: AB 12345 CB
Merk: LEXUS
Seri: LFA
Pelanggaran: Surat Tidak Lengkap - Denda : RP.250,000
Pelanggaran 2: Kebut Kebutan - Denda : RP.200,000

Total Denda : RP.450,000

SEGERA BAYARKAN UANG DENDA KE PETUGAS :)

PETUGAS
CIKO

Ketika file PDF tadi dibuka maka akan muncul tampilan lengkap seperti gambar diatas ini.

2. Membuat Tiket Tilang 1 Pelanggaran

SITILANG ODNI

SATUAN LALU LINTAS KEPOLISIAN NEGERI ODNI
SITILANG ODNI

DATA DIRI TERDAKWA

NAMA TERDAKWA

MIMI

NIK TERDAKWA

12347

ALAMAT TERDAKWA

BANTUL

DATA KENDARAAN

JENIS KENDARAAN

Truck

NO PLAT KENDARAAN

AA 6754 CK

MERK KENDARAAN

HINO

SERI KENDARAAN

DUTRO

MENU PETUGAS

JENIS PELANGGARAN 1

Surat Tidak Lengkap - Denda : RP.350,000

JENIS PELANGGARAN 2

NAMA PETUGAS

CIKO

No Tiket	Nama Terdakwa	Pelanggaran	Pelanggaran 2
TILANG-29-11-2024-a332	RIRI	Surat Tidak Lengkap - Denda : RP.250,000	Kebut Kebutan - Denda : RP.200,000

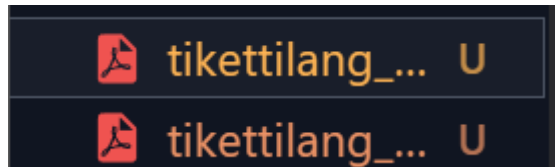
CETAK TIKET

RESET

Masukan semua entry seperti langkah sebelumnya namun kosongi pada bagian pelanggaran 2 seperti gambar diatas.

No Tiket	Nama Terdakwa	Pelanggaran	Pelanggaran 2
TILANG-29-11-2024-a332	RIRI	Surat Tidak Lengkap - Denda : RP.250,000	Kebut Kebut - Denda : RP.200,000
TILANG-29-11-2024-3900	MIMI	Surat Tidak Lengkap - Denda : RP.350,000	

Selanjutnya klik tombol Cetak Tiket maka data tadi akan dicetak sebagai tiket tilang dengan format PDF dan beberapa data akan disimpan kedalam treeview sebagai riwayat.



Maka akan muncul file PDF baru dengan nama `tiketttilang_id` yang menandakan proses pembuatan tiket berhasil

KEPOLISIAN NEGERI ODNI
SATUAN LALU LINTAS KEPOLISIAN NEGERI ODNI
TILANG-29-11-2024-3900
=====

Nama Terdakwa: MIMI
NIK: 12347
Alamat: BANTUL
Jenis Kendaraan: Truck
Nomor Plat: AA 6754 CK
Merk: HINO
Seri: DUTRO
Pelanggaran: Surat Tidak Lengkap - Denda : RP.350,000

Total Denda : RP.350,000

SEGERA BAYARKAN UANG DENDA KE PETUGAS :)

PETUGAS
CIKI
=====

Seperti ini tampilan dari tiket tilang dengan 1 pelanggaran.

3. Validasi Entry Kosong

SITILANG ODNI

SATUAN LALU LINTAS KEPOLISIAN NEGERI ODNI
SITILANG ODNI

DATA DIRI TERDAKWA

NAMA TERDAKWA
NIK TERDAKWA
ALAMAT TERDAKWA

DATA KENDARAAN
JENIS KENDARAAN
NO PLAT KENDARAAN
MERK KENDARAAN
SERI KENDARAAN

MENU PETUGAS

JENIS PELANGGARAN 1
JENIS PELANGGARAN 2
NAMA PETUGAS

No Tiket TILANG-29-11-2024- TILANG-29-11-2024-	Pelanggaran Surat Tidak Lengkap - Denda : RP.250,000 Surat Tidak Lengkap - Denda : RP.350,000	Pelanggaran 2 Kebut Kebutan - Denda : RP.200,000
--	---	---

CETAK TIKET
RESET

ERROR
PASTIKAN SEMUA DATA TERISI!

Seperti ini tampilan dari aplikasi tilang ketika semua entry kosong dan user menklik tombol Cetak Tiket.

4. Validasi Combobox Pelanggaran 2 Diisi Terlebih Dahulu

The screenshot shows the 'SATUAN LALU LINTAS KEPOLISIAN NEGERI ODNI SITILANG ODNI' application. On the left is the 'DATA DIRI TERDAKWA' form with fields for Name, NIK, Address, and Vehicle Data. On the right is the 'MENU PETUGAS' form. In the 'MENU PETUGAS' form, 'JENIS PELANGGARAN 1' is set to 'Tidak Pakai Seatbelt - Denda : RP.220,000'. 'JENIS PELANGGARAN 2' is currently empty. An error dialog box is displayed in the center with the message: 'HARAP ISI PELANGGARAN 1 TERLEBIH DAHULU!' (Please fill in violation 1 first). Below the dropdowns are buttons for 'CETAK TIKET' and 'RESET'.

Seperti ini tampilan aplikasi tilang ketika user menginputkan pelanggaran 2 terlebih dahulu sebelum menginputkan pelanggaran 1.

5. Validasi Format NIK

The screenshot shows the same application interface. In the 'MENU PETUGAS' form, both 'JENIS PELANGGARAN 1' and 'JENIS PELANGGARAN 2' are set to 'Surat Tidak Lengkap - Denda : RP.350,000'. The 'NIK TERDAKWA' field in the 'DATA DIRI TERDAKWA' form is highlighted with a red border, indicating a validation error. An error dialog box is displayed with the message: 'NIK TERDAKWA HARUS ANGKA DAN BERJUMLAH 5 DIGIT!' (The defendant's NIK must be a number and consist of 5 digits). The 'CETAK TIKET' and 'RESET' buttons are visible at the bottom.

Seperti ini tampilan aplikasi tilang ketika user memasukan entry NIK dengan format yang salah, dimana format yang benar yaitu harus 5 digit angka.

6. Validasi Value Combobox Pelanggaran dan Pelanggaran 2 Sama

The screenshot shows the 'SATUAN LALU LINTAS KEPOLISIAN NEGERI ODNI' application. On the left is the 'DATA DIRI TERDAKWA' form with fields for Name, NIK, Address, and Vehicle Data. On the right is the 'MENU PETUGAS' form. It has two dropdowns for 'JENIS PELANGGARAN 1' and 'JENIS PELANGGARAN 2', both currently set to 'Surat Tidak Lengkap - Denda : RP.350,000'. Below these is a table for violations. An 'ERROR' dialog box is displayed in the center, stating 'INPUTAN PELANGGARAN HARUS BERBEDA!' (Violation input must be different!). At the bottom of the 'MENU PETUGAS' form are buttons for 'CETAK TIKET' and 'RESET'.

Seperti ini tampilan aplikasi tilang ketika user menginputkan pelanggaran yang sama pada pelanggaran 1 dan pelanggaran 2.

7. Validasi NIK Unik

This screenshot shows the same application interface. In the 'MENU PETUGAS' form, the 'JENIS PELANGGARAN 1' dropdown is set to 'Surat Tidak Lengkap - Denda : RP.350,000' and the 'JENIS PELANGGARAN 2' dropdown is empty. The 'No Tiket' field shows 'TILANG-29-11-2024-'. An 'ERROR' dialog box is displayed, stating 'NIK TERDAKWA SUDAH ADA!' (Defendant's NIK already exists!). The 'CETAK TIKET' and 'RESET' buttons are visible at the bottom.

Seperti ini tampilan aplikasi tilang ketika user memasukan NIK yang sama dengan NIK data yang sudah diinput sebelumnya.

8. Validasi Nomor Plat Unik

The screenshot shows the 'SATUAN LALU LINTAS KEPOLISIAN NEGERI ODNI SITILANG ODNI' application. On the left is the 'DATA DIRI TERDAKWA' form with fields for Name, NIK, Address, and Vehicle Data. On the right is the 'MENU PETUGAS' form with dropdowns for 'JENIS PELANGGARAN 1' and 'JENIS PELANGGARAN 2', and a 'NAMA PETUGAS' field. An 'ERROR' dialog box is displayed in the center, stating 'NOMOR PLAT SUDAH ADA!' (License Plate Number Already Exists!). Below the dialog, the 'No Tiket' section shows 'TILANG-29-11-2024-a1'. The 'Pelanggaran' section shows 'Surat Tidak Lengkap - Denda : RP.250,000' and 'Pelanggaran 2' shows 'Kebut Kebutan - Denda : RP.200,000'. At the bottom are 'CETAK TIKET' and 'RESET' buttons.

Seperti ini tampilan aplikasi tilang ketika user memasukan No Plat yang sama seperti No Plat yang sudah diinputkan sebelumnya.

9. Reset Riwayat

The screenshot shows the same application interface. The 'DATA DIRI TERDAKWA' form is on the left. The 'MENU PETUGAS' form on the right now displays a table of violation history. The table has four columns: 'No Tiket', 'Nama Terdakwa', 'Pelanggaran', and 'Pelanggaran 2'. The data rows are as follows:

No Tiket	Nama Terdakwa	Pelanggaran	Pelanggaran 2
TILANG-29-11-2024-a332	RIRI	Surat Tidak Lengkap - Denda : RP.250,000	Kebut Kebutan - Denda : RP.200,000
TILANG-29-11-2024-3900	MIMI	Surat Tidak Lengkap - Denda : RP.350,000	

Below the table are 'CETAK TIKET' and 'RESET' buttons.

Gambar diatas menampilkan aplikasi tilang ketika data riwayat pada treeview belum dihapus. Untuk menghapus semua data klik tombol Reset.

SITILANG ODNI

SATUAN LALU LINTAS KEPOLISIAN NEGERI ODNI
SITILANG ODNI

DATA DIRI TERDAKWA

NAMA TERDAKWA

NIK TERDAKWA

ALAMAT TERDAKWA

DATA KENDARAAN

JENIS KENDARAAN

NO PLAT KENDARAAN

MERK KENDARAAN

SERI KENDARAAN

MENU PETUGAS

JENIS PELANGGARAN 1

JENIS PELANGGARAN 2

NAMA PETUGAS

No Tiket	Pelanggaran	Pelanggaran 2
TILANG-29-11-2024	Surat Tidak Lengkap - Denda : RP.250,000	Kebut Kebutan - Denda : RP.200,000
TILANG-29-11-2024	Surat Tidak Lengkap - Denda : RP.350,000	

CETAK TIKET

RESET

YAKIN HAPUS SEMUA RIWAYAT?

Yes No

Setelah mengklik tombol Reset maka akan muncul pemberitahuan, jika kita memilih yes maka semua data akan terhapus.

SITILANG ODNI

SATUAN LALU LINTAS KEPOLISIAN NEGERI ODNI
SITILANG ODNI

DATA DIRI TERDAKWA

NAMA TERDAKWA

NIK TERDAKWA

ALAMAT TERDAKWA

DATA KENDARAAN

JENIS KENDARAAN

NO PLAT KENDARAAN

MERK KENDARAAN

SERI KENDARAAN

MENU PETUGAS

JENIS PELANGGARAN 1

JENIS PELANGGARAN 2

NAMA PETUGAS

No Tiket	Nama Terdakwa	Pelanggaran	Pelanggaran 2
----------	---------------	-------------	---------------

CETAK TIKET

RESET

Seperti ini aplikasi tilang ketika semua data riwayat berhasil dihapus.