

LAPORAN STUDI KASUS
APLIKASI PENYEWAAN KENDARAAN RODA DUA DENGAN GUI
MODERN



Maulana Rizal Alvani

5230411264

Sarjana Informatika
Universitas Teknologi Yogyakarta

KATA PENGANTAR

Puji syukur kami panjatkan ke hadirat Tuhan Yang Maha Esa atas limpahan rahmat-Nya sehingga laporan dengan judul **“Aplikasi Penyewaan Kendaraan Roda Dua dengan GUI Modern”** dapat diselesaikan dengan baik.

Laporan ini disusun untuk menjelaskan pengembangan aplikasi berbasis GUI menggunakan Python dan Tkinter, yang dirancang untuk mempermudah proses penyewaan kendaraan. Kami berharap laporan ini dapat memberikan manfaat bagi pembaca, terutama dalam memahami implementasi teknologi GUI sederhana.

Kami menyadari bahwa laporan ini masih memiliki kekurangan, sehingga kritik dan saran yang membangun sangat kami harapkan. Terima kasih kepada semua pihak yang telah memberikan dukungan dalam penyusunan laporan ini.

DAFTAR ISI

KATA PENGANTAR.....	2
DAFTAR ISI	3
BAB I.....	4
1.1 Latar Belakang Masalah	4
1.2 Rumusan Masalah.....	4
1.3 Tujuan Penelitian	4
BAB II.....	5
2.1 Latar Belakang Masalah	5
2.2 Sistem Rental Motor	5
BAB III	6
3.1 Tools yang Digunakan	6
3.2 Alur Sistem.....	6
BAB IV	7
4.1 Kode Program	7
4.2 Tampilan Antarmuka.....	10
4.3 Penjelasan Kode Program	10
BAB V	12
5.1 Hasil	12
5.2 Pembahasan.....	12
BAB VI	13
6.1 Kesimpulan	13

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Rental kendaraan bermotor merupakan layanan yang terus berkembang di era modern. Dengan meningkatnya mobilitas masyarakat, khususnya di kawasan perkotaan atau tempat wisata, kebutuhan akan motor sebagai alat transportasi yang praktis semakin tinggi. Perusahaan rental sering menghadapi tantangan dalam mengelola data transaksi, seperti perhitungan biaya dan pencatatan pesanan pelanggan, yang apabila dilakukan secara manual dapat memakan waktu dan rawan kesalahan.

Oleh karena itu, diperlukan sebuah aplikasi yang mempermudah proses perhitungan biaya rental sekaligus mencatat transaksi. Dalam studi kasus ini, dikembangkan sebuah aplikasi Rental Motor menggunakan bahasa Python dan pustaka Tkinter. Aplikasi ini memungkinkan pengguna untuk menghitung biaya sewa motor berdasarkan jenis motor dan durasi peminjaman, serta mencatat semua transaksi dalam tabel riwayat yang dapat dipantau langsung melalui antarmuka aplikasi.

1.2 Rumusan Masalah

- Bagaimana merancang aplikasi GUI berbasis Python yang dapat mempermudah perhitungan biaya rental motor?
- Bagaimana menciptakan antarmuka pengguna yang intuitif dan mudah dipahami, bahkan oleh pengguna yang tidak memiliki latar belakang teknis?
- Bagaimana mencatat setiap transaksi rental motor dalam bentuk tabel riwayat yang dapat diakses kapan saja?

1.3 Tujuan Penelitian

- Membuat aplikasi GUI sederhana yang mampu menghitung biaya rental motor secara otomatis.
- Menyediakan antarmuka yang user-friendly untuk mendukung pengalaman pengguna.
- Menambahkan fitur pencatatan riwayat transaksi yang terintegrasi dalam aplikasi.

BAB II

LANDASAN TEORI

2.1 Latar Belakang Masalah

GUI adalah antarmuka yang memungkinkan pengguna berinteraksi dengan perangkat lunak melalui elemen grafis seperti tombol, teks, dan tabel. Dalam pengembangan perangkat lunak berbasis GUI, estetika dan kenyamanan penggunaan menjadi aspek penting. Python menyediakan pustaka **Tkinter** sebagai salah satu solusi pengembangan aplikasi GUI. Tkinter mendukung berbagai elemen interaktif seperti label, tombol, kotak teks, dan tabel, serta memungkinkan pengaturan warna, font, dan tata letak.

2.2 Sistem Rental Motor

Rental motor adalah layanan yang menyediakan penyewaan kendaraan bermotor dalam jangka waktu tertentu. Sistem ini mencakup:

- Pilihan Jenis Motor: Penyediaan motor sesuai kebutuhan pelanggan (misalnya, motor matic, bebek, atau sport).
- Durasi Sewa: Biaya sewa dihitung berdasarkan jumlah hari penyewaan.
- Pencatatan Riwayat: Setiap transaksi perlu dicatat untuk keperluan dokumentasi dan audit.

BAB III METODOLOGI

3.1 Tools yang Digunakan

- Python: Digunakan sebagai bahasa pemrograman utama.
- Tkinter: Pustaka bawaan Python untuk membangun GUI.
- Treeview (Tkinter): Elemen tabel yang digunakan untuk mencatat riwayat pesanan.

3.2 Alur Sistem

- Input: Pengguna memilih jenis motor dan memasukkan lama sewa (dalam hari).
- Proses Perhitungan: Aplikasi menghitung total biaya berdasarkan tarif harian jenis motor yang dipilih.
- Output: Menampilkan total biaya pada layar aplikasi.
- Pencatatan Riwayat: Transaksi dicatat dalam tabel riwayat yang dapat dipantau langsung oleh pengguna.

BAB IV

KODE DAN IMPLEMENTASI

4.1 Kode Program

```
# Import pustaka
import tkinter as tk
from tkinter import ttk
from tkinter import messagebox

class RentalMotorApp:
    def __init__(self, root):
        self.root = root
        self.root.title("RENTAL MOTOR BERKELAZZ")
        self.root.geometry("1500x1000")
        self.root.resizable(False, False)

        # Warna tema
        self.bg_color = "#ffffff"
        self.primary_color = "#4a4a4a"
        self.accent_color = "#1a73e8"
        self.secondary_color = "#f1f1f1"

        # Atur latar belakang
        self.root.configure(bg=self.bg_color)

        # Harga motor per hari
        self.motor_prices = {
            "Motor Matic": 50000,
            "Motor bebek": 40000,
            "Motor Sport": 100000
        }

        # Daftar Riwayat Pesanan
        self.history = []

        # Label Judul
        header_label = tk.Label(
            root, text="RENTAL MOTOR BERKELAZZ", font=("Helvetica", 25,
"bold"),
            bg=self.bg_color, fg=self.primary_color, pady=10
        )
        header_label.pack()

        # Pilihan Motor
        motor_frame = tk.Frame(root, bg=self.bg_color)
        motor_frame.pack(pady=10)
```

```

        tk.Label(motor_frame, text="Pilih Jenis Motor:", font=("Helvetica",
14), bg=self.bg_color, fg=self.primary_color).pack(anchor="w")

        self.motor_var = tk.StringVar(value="Motor Matic")
        for motor in self.motor_prices.keys():
            tk.Radiobutton(
                motor_frame, text=motor, variable=self.motor_var, value=motor,
font=("Helvetica", 12),
                bg=self.bg_color, fg=self.primary_color, anchor="w",
selectcolor=self.secondary_color
            ).pack(anchor="w")

        # Input Lama Sewa
        days_frame = tk.Frame(root, bg=self.bg_color)
        days_frame.pack(pady=10)
        tk.Label(days_frame, text="Lama Sewa (Hari):", font=("Helvetica", 14),
bg=self.bg_color, fg=self.primary_color).pack(anchor="w")

        self.days_entry = tk.Entry(days_frame, font=("Helvetica", 12),
width=10)
        self.days_entry.pack(anchor="w", pady=5)

        # Tombol Hitung dan Reset
        self.button_frame = tk.Frame(root, bg=self.bg_color)
        self.button_frame.pack(pady=10)

        self.calculate_button = tk.Button(
            self.button_frame, text="Hitung Biaya", font=("Helvetica", 12,
"bold"),
            bg=self.accent_color, fg="white", command=self.calculate_cost,
relief="flat", padx=10, pady=5
        )
        self.calculate_button.grid(row=0, column=0, padx=10)

        self.reset_button = tk.Button(
            self.button_frame, text="Reset", font=("Helvetica", 12, "bold"),
            bg=self.secondary_color, fg=self.primary_color,
command=self.reset, relief="flat", padx=10, pady=5
        )
        self.reset_button.grid(row=0, column=1, padx=10)

        # Label Hasil
        self.result_label = tk.Label(root, text="", font=("Helvetica", 14,
"bold"), fg=self.accent_color, bg=self.bg_color)
        self.result_label.pack(pady=10)

        # Tabel Riwayat Pesanan

```



```

        tk.Label(root, text="Riwayat Pesanan:", font=("Helvetica", 14),
bg=self.bg_color, fg=self.primary_color).pack(pady=5)

        self.history_frame = tk.Frame(root, bg=self.bg_color)
        self.history_frame.pack()

        self.tree = ttk.Treeview(self.history_frame, columns=("Motor", "Hari",
"Total"), show="headings", height=7)
        self.tree.pack()

        # Konfigurasi kolom
        self.tree.heading("Motor", text="Jenis Motor")
        self.tree.heading("Hari", text="Lama Sewa (Hari)")
        self.tree.heading("Total", text="Total Biaya (Rp)")
        self.tree.column("Motor", anchor="center", width=150)
        self.tree.column("Hari", anchor="center", width=100)
        self.tree.column("Total", anchor="center", width=150)

        # Style Treeview
        style = ttk.Style()
        style.theme_use("clam")
        style.configure(
            "Treeview", background=self.secondary_color,
foreground=self.primary_color, rowheight=25,
            fieldbackground=self.secondary_color, font=("Helvetica", 11)
        )
        style.map("Treeview", background=[("selected", self.accent_color)],
foreground=[("selected", "white")])

    def calculate_cost(self):
        try:
            days = int(self.days_entry.get())
            if days <= 0:
                raise ValueError
            motor = self.motor_var.get()
            cost_per_day = self.motor_prices[motor]
            total_cost = cost_per_day * days
            self.result_label.config(text=f"Total Biaya: Rp {total_cost:,}")

            # Tambahkan ke riwayat
            self.history.append((motor, days, total_cost))
            self.tree.insert("", "end", values=(motor, days, f"Rp
{total_cost:,}"))

        except ValueError:
            messagebox.showerror("Input Tidak Valid", "Masukkan jumlah hari
yang valid (angka positif).")

```

```

def reset(self):
    self.motor_var.set("Motor Matic")
    self.days_entry.delete(0, tk.END)
    self.result_label.config(text="")

if __name__ == "__main__":
    root = tk.Tk()
    app = RentalMotorApp(root)
    root.mainloop()

```

4.2 Tampilan Antarmuka

Aplikasi ini dirancang dengan tata letak yang sederhana tetapi efektif, terdiri dari:

1. Judul Aplikasi: Memberikan identitas aplikasi kepada pengguna.
2. Pilihan Motor: Menggunakan elemen Radiobutton untuk memungkinkan pengguna memilih jenis motor.
3. Input Lama Sewa: Kotak input untuk memasukkan jumlah hari penyewaan.
4. Tombol:
 - Hitung Biaya: Untuk memproses perhitungan biaya rental.
 - Reset: Untuk menghapus semua input dan mengatur ulang aplikasi ke keadaan awal.
5. Tabel Riwayat Pesanan: Menyimpan semua transaksi yang telah dilakukan.

4.3 Penjelasan Kode Program

- Import Pustaka

Pustaka **Tkinter** digunakan untuk membangun aplikasi GUI. **MessageBox** ditambahkan untuk menampilkan pesan kesalahan.

```

• import tkinter as tk
• from tkinter import ttk
• from tkinter import messagebox

```

- Fungsi Utama

Konstruktor `__init__` digunakan untuk menginisialisasi seluruh elemen aplikasi. Parameter `root` mewakili jendela utama.

```

• class RentalMotorApp:
•     def __init__(self, root):

```

- Jenis Motor dan Harga

Tiga jenis motor disediakan dalam aplikasi, masing-masing dengan tarif per hari yang berbeda

```

• self.motor_prices = {
•     "Motor Matic": 50000,
•     "Motor Bebek": 40000,
•     "Motor Sport": 100000
• }

```

- Pilihan Motor

Elemen **Radiobutton** memungkinkan pengguna memilih satu jenis motor. Variabel `self.motor_var` digunakan untuk menyimpan pilihan pengguna.

```

• for motor in self.motor_prices.keys():
•     tk.Radiobutton(root, text=motor, variable=self.motor_var,
value=motor).pack(anchor="w")

```

- Fungsi Perhitungan

1. Input Validasi: Memastikan bahwa jumlah hari adalah angka positif.
2. Perhitungan: Mengalikan harga motor per hari dengan lama sewa untuk mendapatkan total biaya.
3. Output: Hasil ditampilkan di layar dan dicatat dalam tabel riwayat.

```

• def calculate_cost(self):
•     try:
•         days = int(self.days_entry.get())
•         if days <= 0:
•             raise ValueError
•         motor = self.motor_var.get()
•         cost_per_day = self.motor_prices[motor]
•         total_cost = cost_per_day * days
•         self.result_label.config(text=f"Total Biaya: Rp
{total_cost:,}")
•
•         # Tambahkan ke riwayat
•         self.tree.insert("", "end", values=(motor, days, f"Rp
{total_cost:,}")
•     except ValueError:
•         messagebox.showerror("Input Tidak Valid", "Masukkan jumlah hari
yang valid (angka positif).")

```

- Fungsi Reset

Fungsi ini menghapus semua input dan mengatur ulang aplikasi ke keadaan semula.

```

• def reset(self):
•     self.motor_var.set("Motor Matic")
•     self.days_entry.delete(0, tk.END)
•     self.result_label.config(text="")

```

BAB V

HASIL DAN PEMBAHASAN

5.1 Hasil

Aplikasi ini berhasil dibuat dengan fitur-fitur berikut

1. Menghitung biaya rental motor berdasarkan jenis motor dan lama penyewaan.
2. Menampilkan hasil perhitungan secara langsung di layar.
3. Mencatat setiap transaksi ke tabel riwayat yang dapat diakses pengguna kapan saja.

5.2 Pembahasan

1. Kelebihan:
 - Aplikasi ini user-friendly dan mudah digunakan.
 - Tabel riwayat memberikan kemudahan dalam memantau transaksi sebelumnya.
2. Kekurangan
 - Data riwayat belum disimpan secara permanen (hilang ketika aplikasi ditutup).

BAB VI

PENUTUP

6.1 Kesimpulan

Aplikasi rental motor berbasis GUI ini berhasil membantu perhitungan biaya rental motor dengan cepat dan efisien. Antarmuka yang sederhana mendukung kemudahan penggunaan, sementara fitur pencatatan riwayat menambah nilai fungsional aplikasi.