

**FITUR LAPORAN HARIAN APLIKASI PEMESANAN MENU
KEDAI FMR**



Dosen Pengampu:

Hasanuddin Al-Habib, S.Si., M.Si.

Dr. Heri Purnawan, S.Si., M.Si.

Nama :

Jauzah Rifda (25031554007)

KELAS 2025G

MATA KULIAH PEMROGRAMAN DASAR

PROGRAM STUDI S1 SAINS DATA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS NEGERI SURABAYA

DESEMBER 2025

1. Pendahuluan

Aplikasi pemesanan menu Kedai FMR merupakan aplikasi kasir berbasis Python dengan antarmuka grafis menggunakan Tkinter. Aplikasi ini dirancang untuk membantu proses pemesanan makanan, pengelolaan menu, serta pencatatan transaksi. Untuk mendukung kebutuhan administrasi, aplikasi dilengkapi dengan fitur laporan harian yang hanya dapat diakses oleh admin.

2. Tujuan Laporan Harian

Fitur laporan harian yang saya tambahkan bertujuan untuk:

1. Mencatat seluruh transaksi pemesanan yang terjadi dalam satu hari.
2. Menyimpan riwayat pesanan secara otomatis.
3. Menghasilkan laporan penjualan harian dalam bentuk PDF.
4. Membantu admin dalam melakukan evaluasi pendapatan harian.

3. Mekanisme Penyimpanan Data Pesanan

Setiap transaksi yang dilakukan akan terlebih dahulu disimpan ke dalam file riwayat harian dengan format JSON pada folder *history*. Format JSON ini digunakan sebagai media penyimpanan data transaksi karena mudah dikelola dan dapat menampung banyak pesanan dalam satu hari. Selanjutnya, data riwayat tersebut diolah dan dikonversi menjadi laporan harian dalam bentuk PDF ketika admin memilih fitur laporan harian.

Setiap data pesanan yang disimpan berisi:

1. Order ID.
2. Waktu pemesanan.
3. Daftar item yang dipesan.
4. Total pembayaran.

File riwayat dibuat berdasarkan tanggal transaksi, sehingga setiap hari memiliki file history tersendiri.

4. Proses Pembuatan Laporan Harian

Admin dapat menghasilkan laporan harian dengan menekan tombol **LAPORAN HARIAN** pada aplikasi. Ketika tombol ditekan, maka sistem akan:

1. Membaca file riwayat pesanan sesuai tanggal hari ini.
2. Mengambil seluruh data transaksi pada hari tersebut.

3. Menghitung total pendapatan harian.
4. Menghasilkan laporan dalam bentuk file PDF.

Laporan pdf berisi informasi sebagai berikut:

1. Judul laporan.
2. Tanggal laporan.
3. Daftar seluruh pesanan (Order ID, waktu, dan item).
4. Total pendapatan harian.

5. Hak Akses Admin

Fitur laporan harian hanya dapat diakses oleh pengguna dengan peran admin. Akses ini diatur melalui sistem login, dimana admin harus memasukkan password terlebih dahulu. Setelah berhasil login, admin akan mendapatkan tambahan tombol laporan harian dan edit menu.

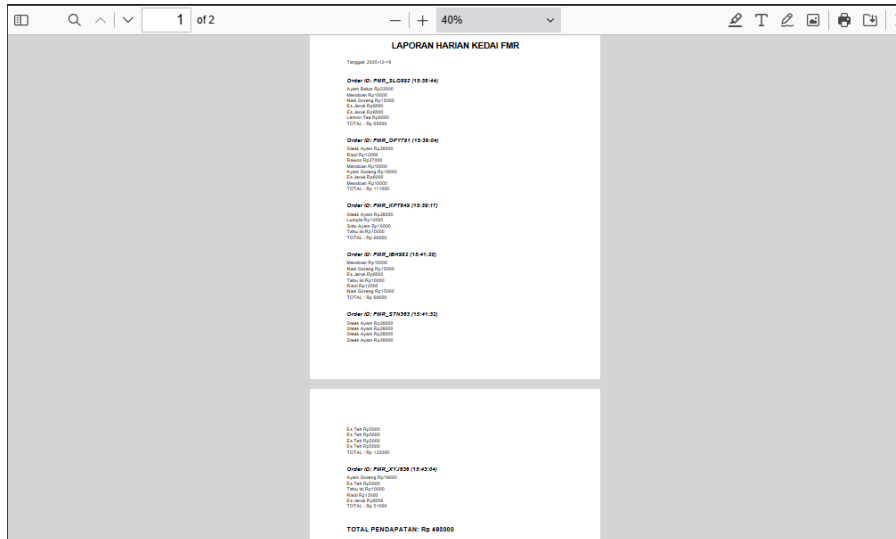
6. Kesimpulan

Fitur laporan harian pada aplikasi pemesanan menu Kedai FMR merupakan komponen penting dalam sistem administrasi. Dengan menyimpan riwayat transaksi secara otomatis dan menghasilkan laporan PDF harian, aplikasi ini mampu meningkatkan efisiensi

7. Tampilan Aplikasi



Gambar.1 Tampilan jendela admin dengan fitur tambahan



Gambar.2 tampilan laporan harian (PDF)

8. Lampiran Kode yang Berubah

```
# folder history harian
HISTORY_FOLDER = "history"
os.makedirs(HISTORY_FOLDER, exist_ok=True)
```

```
def save_order_history(order_id, items, total, order_day, order_time):
    filename = os.path.join(HISTORY_FOLDER, f"history_{order_day}.json")

    order_data = {
        "order_id": order_id,
        "time": order_time.strftime("%H:%M:%S"),
        "items": items,
        "total": total
    }

    if os.path.exists(filename):
        with open(filename, "r") as f:
            history = json.load(f)
    else:
        history = []

    history.append(order_data)

    with open(filename, "w") as f:
        json.dump(history, f, indent=4)
```

```

def generate_daily_report(order_day):
    filename = os.path.join(HISTORY_FOLDER, f"history_{order_day}.json")

    if not os.path.exists(filename):
        messagebox.showerror("Error", "No orders for today")
        return

    with open(filename, "r") as f:
        history = json.load(f)

    pdf_name = f"Laporan_{order_day}.pdf"
    doc = SimpleDocTemplate(pdf_name, pagesize=letter)
    styles = getSampleStyleSheet()
    story = []

    story.append(Paragraph("LAPORAN HARIAN KEDAI FMR", styles["Title"]))
    story.append(Spacer(1, 12))
    story.append(Paragraph(f"Tanggal: {order_day}", styles["Normal"]))
    story.append(Spacer(1, 12))

    grand_total = 0

    for order in history:
        story.append(Paragraph(f"Order ID: {order['order_id']} ({order['time']})", styles["Heading3"]))
        for item in order["items"]:
            story.append(Paragraph(item, styles["Normal"]))
        story.append(Paragraph(order["total"], styles["Normal"]))
        story.append(Spacer(1, 10))

        grand_total += int(order["total"].replace("TOTAL : Rp ", ""))

    story.append(Spacer(1, 12))
    story.append(Paragraph(f"TOTAL PENDAPATAN: Rp {grand_total}", styles["Heading2"]))

    doc.build(story)
    messagebox.showinfo("Success", f"Laporan harian disimpan sebagai {pdf_name}")

```

```

def admin_edit():
    edit_win = Toplevel(root)
    edit_win.title("Admin Edit Menu")
    edit_win.geometry("400x600")

    ttk.Label(edit_win, text="Edit Menu Items").pack(pady=10)

    entries = {}
    for item, price in prices.items():
        frame = ttk.Frame(edit_win)
        frame.pack(fill='x', padx=10, pady=2)
        ttk.Label(frame, text=item, width=15).pack(side='left')
        entry = ttk.Entry(frame)
        entry.insert(0, str(price))
        entry.pack(side='right')
        entries[item] = entry

def order():
    new_receipt = orderIDLabel.cget("text").replace("ORDER ID : ", "")
    transaction_list = orderTransaction.cget("text").split("\n") if
orderTransaction.cget("text") else []

    order_day = date.today()
    order_time = datetime.now()
    total = orderTotalLabel.cget("text")

    generate_pdf(new_receipt, transaction_list, total, order_day,
order_time)
    save_order_history(new_receipt, transaction_list, total, order_day,
order_time)

    # Reset
    orderTotalLabel.configure(text="TOTAL : Rp 0")
    orderIDLabel.configure(text="ORDER ID : " + ORDER_ID())
    orderTransaction.configure(text="")

```