

LAPORAN
TUGAS AKHIR
APLIKASI PENYEWAAN KAMERA BERBASIS DESKTOP



Mata Kuliah Pemograman Berorientasi Objek

Dosen Pengampu :

M. Bahrul Subkhi, M. Kom

Disusun Oleh :

- | | |
|----------------------------|--------------|
| 1. Ellok Sintha Maydiana | (2213020005) |
| 2. Febriyanti Nur Fadillah | (2213020037) |
| 3. Aldy Sasongko | (2213020061) |

KELAS 2E
PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK DAN ILMU KOMPUTER
UNIVERSITAS NUSANTARA PGRI KEDIRI

2024

DAFTAR ISI

| | |
|--------------------------------------|----|
| DAFTAR ISI..... | 2 |
| BAB I PENDAHULUAN..... | 3 |
| A. Latar Belakang..... | 3 |
| B. Rumusan Masalah..... | 4 |
| BAB II PEMBAHASAN..... | 6 |
| A. Flowchart Sistem..... | 6 |
| B. Class Diagram..... | 7 |
| C. Database..... | 8 |
| D. Hasil Program dan Penjelasan..... | 8 |
| BAB III PENUTUP..... | 16 |
| Kesimpulan..... | 16 |
| DAFTAR PUSTAKA..... | 17 |

BAB I

PENDAHULUAN

A. Latar Belakang

Penyewaan kamera merupakan salah satu aspek yang cukup berkembang, menyadari hal itu setiap pengusaha yang bergerak di bisnis telah mengubah cara kita mengabadikan dan berbagi momen berharga dalam hidup. Peralatan fotografi yang semakin canggih, seperti kamera digital dengan resolusi tinggi dan fitur-fitur inovatif, membuat masyarakat semakin aktif terlibat dalam fotografi. Seiring dengan berkembangnya minat ini, banyak individu dan organisasi dihadapkan pada tantangan untuk mendapatkan peralatan fotografi yang memenuhi kebutuhan mereka.

Pertama-tama, aplikasi ini memiliki antarmuka pengguna yang ramah, memanfaatkan elemen seperti tombol, tabel, dan dialog untuk memudahkan penggunaan. Teknologi informasi membentuk cara kita berinteraksi dengan pekerjaan kita, khususnya di industri fotografi, dengan meningkatnya kebutuhan akan kemudahan akses dan kontrol layanan terus memainkan peran penting peranan penting dalam modernisasi dengan berbagai layanan. Penerapan PyQt5, perangkat pengembangan perangkat lunak berbasis Python untuk pengembangan aplikasi persewaan kamera berbasis desktop.

Aplikasi ini juga menyediakan fitur untuk memesan sewa kamera yang berisi data penyewa kamera seperti nomor antrian, data penyewa, dan type kamera (DSLR, Mirrorless, Hybrid) lalu memilih type merk dan keperluan apa (Fotografi atau Videografi). Setelah itu menciptakan data transaksi tersebut yang telah dipinjam yang memiliki antrian dan riwayat transaksi. Ini membantu pengelola penyewaan kamera untuk dengan cepat melacak status kamera yang masih dipinjam dan mencatat sejarah peminjaman yang telah selesai. Informasi ini tersaji dalam tabel yang rapi dan mudah dibaca. Fungsi utama aplikasi mencakup operasi peminjaman dan pengembalian kamera.

Penggunaan PyQt5 dalam persewaan kamera membuka kemungkinan baru untuk meningkatkan pengalaman pengguna, memudahkan manajemen persewaan, dan meningkatkan efisiensi operasional. Paradigma persewaan kamera berbasis desktop menggunakan PyQt5 memungkinkan pengguna mencari dan memesan perlengkapan fotografi dengan mudah melalui antarmuka yang intuitif dan ramah pengguna.

Selain itu, fitur tambahan seperti pemrosesan transaksi yang aman, pemantauan inventaris secara real-time, dan integrasi dengan sistem pembayaran digital memberikan nilai tambah yang signifikan bagi penyedia layanan dan konsumen. Artikel ini bertujuan untuk menjelaskan bagaimana transformasi ini membentuk dan meningkatkan layanan penyewaan kamera berbasis desktop dengan menerapkan teknologi PyQt5.

Kami berharap dengan memahami potensi dan manfaat PyQt5 dalam konteks ini, pembaca kami akan mendapatkan wawasan lebih dalam tentang kontribusi

teknologi ini dalam membentuk masa depan persewaan kamera yang lebih terintegrasi dan efisien. Dalam konteks ini, artikel ini menggali lebih jauh fenomena persewaan kamera. Dengan memahami sejarah, evolusi, dan manfaat persewaan kamera, kami berharap pembaca mendapatkan pemahaman lebih dalam tentang bagaimana layanan ini telah menjadi bagian integral dari ekosistem fotografi modern.

B. Rumusan Masalah

Berdasarkan latar belakang diatas dapat dirumuskan permasalahan - permasalahan yaitu ;

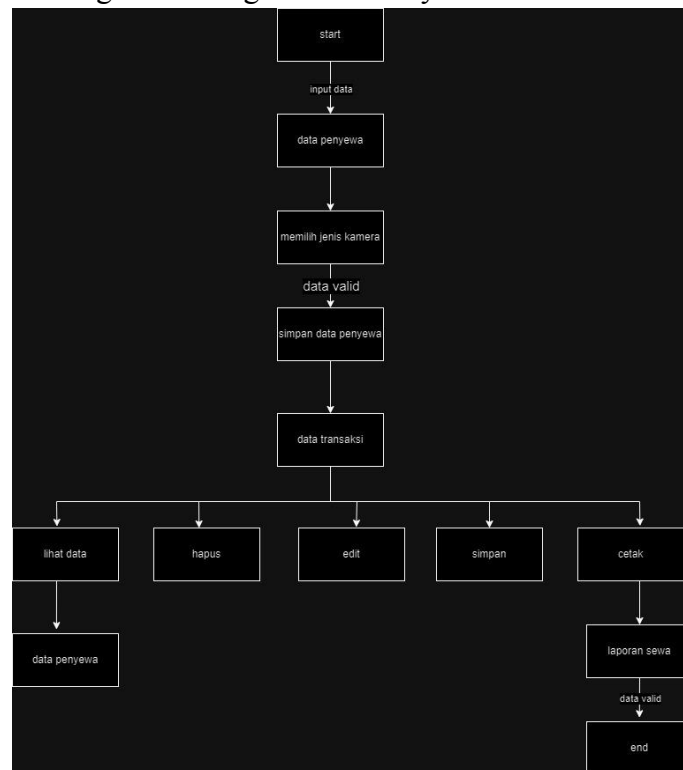
1. Bagaimana cara meningkatkan efisiensi proses pembayaran penyewaan kamera ?
2. Bagaimana data pelanggan semua dapat ditampilkan secara keseluruhan ?
3. Bagaimana mempermudah dan mempercepat proses pengembalian kamera?
4. Apakah sistem dapat memberikan denda otomatis untuk kamera yang sudah melewati batas waktu peminjaman?
5. Bagaimana menyajikan riwayat transaksi peminjaman kamera dengan lebih informatif dan terperinci?

BAB II

PEMBAHASAN

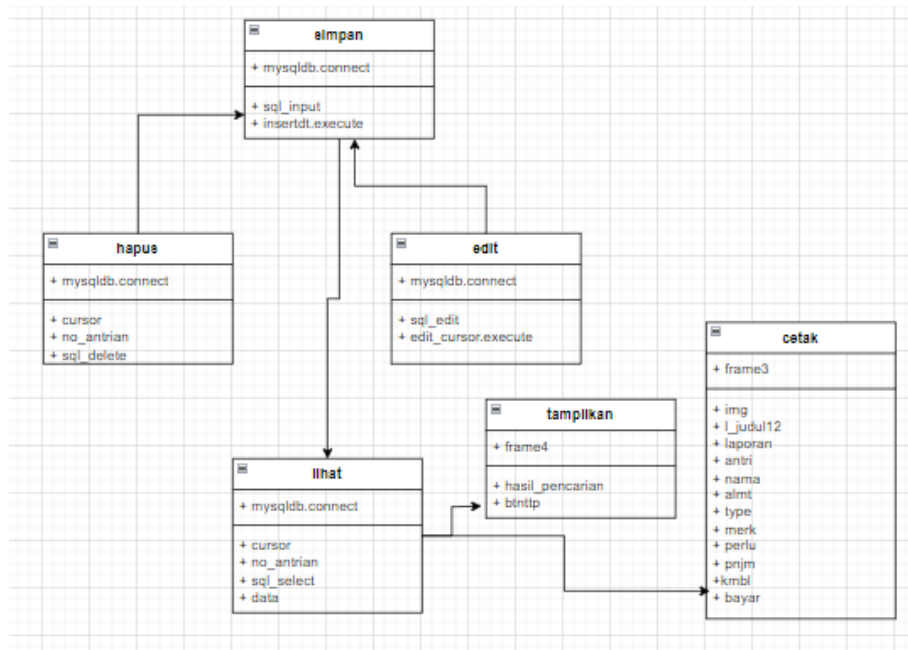
A. Flowchart Sistem

Flowchart adalah representasi grafis dari urutan langkah-langkah dalam sebuah proses atau sistem. Dalam diagram ini, langkah-langkah direpresentasikan dengan simbol-simbol grafis yang terhubung secara berurutan, membentuk aliran atau "flow" dari satu langkah ke langkah berikutnya.



B. Class Diagram

Diagram kelas (class diagram) adalah salah satu jenis diagram yang digunakan dalam pemrograman berorientasi objek untuk menggambarkan struktur statis dari sistem perangkat lunak. Diagram kelas menampilkan kelas-kelas yang ada dalam sistem, hubungan antara kelas-kelas tersebut, serta atribut dan metode yang dimiliki oleh setiap kelas.



C. Database

Berikut data sewa dari pelanggan yang sudah melakukan transaksi atau pembayaran ;

The screenshot shows the phpMyAdmin interface for a database named 'sewakamera'. The 'datasewa' table is selected, and the data is displayed in a table format. The table has 7 columns: No_Antrian, Nama_Lengkap, Alamat, Email, Type_Kamera, Merk_Kamera, and Keperluan. There are 2 rows of data.

| No_Antrian | Nama_Lengkap | Alamat | Email | Type_Kamera | Merk_Kamera | Keperluan |
|------------|--------------|--------|-----------------|-------------|-------------|-----------|
| 1 | ellok | pare | ellok@gmail.com | DSLR | Sony | Fotografi |
| 1 | mawar | pare | mawar@gmail.com | Mirrorless | Nikon | Fotografi |

localhost / 127.0.0.1 / sewakam... Beranda Cariva Diagram Tanpa Judul - drawio Diagram Tanpa Judul - drawio

localhost/phpmyadmin/index.php?route=/database/structure&db=sewakamera

Server: 127.0.0.1 » Database: sewakamera

Struktur SQL Cari Kueri Ekspor Impor Operasi Hak Akses Routine Event Trigger

Filters

Mengandung kata:

| Tabel | Tindakan | Baris | Jenis | Penyortiran | Ukuran | Beban |
|--|---|----------|---------------|---------------------------|----------------|------------|
| <input type="checkbox"/> datasewa | Jelajahi Struktur Cari Tambahkan Kosongkan Hapus | 2 | InnoDB | utf8mb4_general_ci | 16.0 KB | - |
| <input type="checkbox"/> datatransaksi | Jelajahi Struktur Cari Tambahkan Kosongkan Hapus | 0 | InnoDB | utf8mb4_general_ci | 16.0 KB | - |
| 2 tabel | Jumlah | 2 | InnoDB | utf8mb4_general_ci | 32.0 KB | 0 B |

☐ Pilih Semua Dengan pilihan:

Cetak Kamus data

Create new table

Nama tabel Jumlah kolom

Buat

Konsol

31°C Hujan 14:49 03/01/2024

localhost / 127.0.0.1 / sewakam... Beranda Cariva Diagram Tanpa Judul - drawio Diagram Tanpa Judul - drawio

localhost/phpmyadmin/index.php?route=/sql&pos=0&db=sewakamera&table=datatransaksi

Server: 127.0.0.1 » Database: sewakamera » Tabel: datatransaksi

Jelajahi Struktur SQL Cari Tambahkan Ekspor Impor Hak Akses Operasi Trigger

Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Menampilkan baris 0 - 0 (total 1, Pencarian dilakukan dalam 0,0004 detik.)

SELECT * FROM `datatransaksi`

Profil [Edit dikotak] [Ubah] [Jelaskan SQL] [Buat kode PHP] [Segarkan]

☐ Tampilkan semua Jumlah baris: 25 Saring baris: Cari di tabel ini

Extra options

| No_Antrian | Tgl_Pinjam | Tgl_Kembali | Pembayaran |
|------------|------------|-------------|--------------|
| 1 | 0001-04-24 | 0001-04-24 | Tranfer Bank |

☐ Tampilkan semua Jumlah baris: 25 Saring baris: Cari di tabel ini

Operasi hasil kueri

Cetak Salin ke clipboard Ekspor Tampilkan bagan Buat tampilan

Konsol

31°C Hujan 11:10 04/01/2024

D. Program dan Penjelasan

Berikut Program dan penjelasan dari Penyewaan Kamera ;

LIBRARY

```
from cgitb import text
from tkinter import *
from tkinter import ttk, messagebox
from tkcalendar import DateEntry
from PIL import Image, ImageTk
from tkinter import Frame, Label, Text, Button, StringVar, Radiobutton, ttk, Tk, END
import qrcode
import MySQLdb
```

- `cgitb.text`: Modul ini biasanya digunakan untuk menampilkan informasi traceback atau jejak eksekusi saat terjadi kesalahan dalam program Python. Namun, dalam contoh yang Anda berikan, modul ini disertakan tetapi tidak digunakan secara eksplisit.
- `tkinter`: Ini adalah modul standar Python yang digunakan untuk pembuatan antarmuka grafis pengguna (GUI).
- `tkcalendar`: Ini adalah pustaka tambahan yang menyediakan widget kalender untuk digunakan dalam aplikasi Tkinter.
- `PIL (Python Imaging Library)`: Sekarang dikenal sebagai `pillow`, ini adalah pustaka yang umum digunakan untuk memanipulasi gambar dalam Python.
- `qrcode`: Modul ini digunakan untuk membuat kode QR.
- `MySQL`: Ada kemungkinan terjadi kesalahan penulisan. Sepertinya Anda bermaksud menggunakan modul untuk berinteraksi dengan MySQL database. Modul yang tepat untuk ini biasanya adalah `mysql-connector-python` atau `pymysql`.

DEF SIMPAN

```
def simpan():
    db = MySQLdb.connect("localhost", "root", "", "sewakamera")
    insertdt = db.cursor()
    try:
        sql_input = "insert into datasewa(No_Antrian, Nama_Lengkap, Alamat, Email, Type_Kamera, Merk_Kamera, Keperluan) value('"+e_antri.get()+"', '"+e_nama.get()+"', '"+e_almt.get()+"',
        insertdt.execute(sql_input)
        db.commit()
        messagebox.showinfo("Sukses", "Data berhasil disimpan")
    except Exception as e:
        db.rollback()
        messagebox.showerror("Error", f"Error: {str(e)}")
    finally:
        db.close()
```

- Fungsi `simpan()`: dipanggil ketika suatu aksi (seperti klik tombol simpan) dilakukan dalam antarmuka pengguna.
- Koneksi ke Database: Pertama, kode membuka koneksi ke database MySQL menggunakan modul `MySQLdb` dengan parameter host (`localhost`), username (`root`), password kosong (`""`), dan nama database (`sewakamera`).
- Cursor: Setelah koneksi berhasil dibuat, cursor database (`insertdt`) dibuat untuk mengeksekusi perintah SQL.
- Perintah SQL: Baris `sql_input` adalah string yang berisi perintah SQL untuk

memasukkan data ke dalam tabel datasewa di database sewakamera. Nilai-nilai yang diambil dari field-field dalam antarmuka pengguna seperti e_antri, e_nama, e_almt, dan sebagainya dimasukkan ke dalam perintah SQL ini.

- Eksekusi SQL: Perintah SQL dieksekusi menggunakan insertdt.execute(sql_input).
- Commit dan Rollback: Jika eksekusi berhasil, perubahan dikonfirmasi (db.commit()) ke dalam database. Namun, jika ada kesalahan (Exception), maka transaksi tersebut dibatalkan (db.rollback()).
- Pesan Sukses atau Error: Setelah operasi selesai, akan muncul kotak pesan (messagebox) untuk memberi tahu pengguna apakah operasi penyimpanan berhasil atau ada error dalam proses tersebut.
- Penutup Koneksi: Terakhir, koneksi ke database ditutup dengan db.close() dalam blok finally. Ini dilakukan agar sumber daya terkait dengan koneksi database dapat dibebaskan setelah selesai digunakan.

DEF EDIT

```
def edit():
    db = MySQLdb.connect("localhost", "root", "", "sewakamera")
    edit_cursor = db.cursor()
    try:
        sql_edit = "UPDATE datasewa SET Nama_Lengkap=%s, Alamat=%s, Email=%s, Type_Kamera=%s, Merk_Kamera=%s, Keperluan=%s WHERE No_Antrian=%s"
        edit_cursor.execute(sql_edit, (e_nama.get(), e_almt.get(), e_mail.get(), r.get(), c_merk.get(), r2.get(), e_antri.get()))
        db.commit()
        messagebox.showinfo("Sukses", "Data berhasil diupdate")
    except Exception as e:
        db.rollback()
        messagebox.showerror("Error", f"Error: {str(e)}")
    finally:
        db.close()
```

- Kedua fungsi tersebut berinteraksi dengan elemen-elemen antarmuka pengguna (seperti e_nama, e_almt, e_mail, r, c_merk, r2, e_antri) untuk mendapatkan nilai yang akan dimasukkan, diupdate, atau dihapus dari database.

DEF HAPUS

```
def hapus():
    db = MySQLdb.connect("localhost", "root", "", "sewakamera")
    cursor = db.cursor()
    try:
        no_antrian = e_antri.get()
        sql_delete = f"DELETE FROM datasewa WHERE No_Antrian = '{no_antrian}'"
        cursor.execute(sql_delete)
        db.commit()
        messagebox.showinfo("Sukses", "Data berhasil dihapus")
    except Exception as e:
        db.rollback()
        messagebox.showerror("Error", f"Error: {str(e)}")
    finally:
        db.close()
```

- Fungsi edit() bertugas untuk melakukan operasi pembaruan (UPDATE) data dalam tabel datasewa.
- Pertama-tama, kode membuka koneksi ke database MySQL dan membuat cursor

untuk melakukan operasi pada database.

- Kemudian, terdapat perintah SQL `sql_edit` yang merupakan perintah `UPDATE` untuk memperbarui data sesuai `No_Antrian` tertentu.
- `edit_cursor.execute(sql_edit, (e_nama.get(), e_almt.get(), e_mail.get(), r.get(), c_merk.get(), r2.get(), e_antri.get()))` menjalankan perintah SQL dengan nilai-nilai yang diambil dari elemen-elemen antarmuka pengguna.
- Jika pembaruan berhasil, perubahan dikonfirmasi (`db.commit()`). Jika terjadi kesalahan, transaksi dibatalkan (`db.rollback()`).
- Terakhir, koneksi ke database ditutup dengan `db.close()`.
- Fungsi `hapus()` bertugas untuk menghapus data dari tabel `datasewa` berdasarkan `No_Antrian` tertentu.
- Mirip dengan fungsi `edit()`, fungsi ini juga membuka koneksi ke database MySQL, membuat cursor, dan mengeksekusi perintah SQL (dalam hal ini, `DELETE`) untuk menghapus data sesuai dengan `No_Antrian` yang diperoleh dari antarmuka pengguna.
- Jika operasi `hapus` berhasil, perubahan dikonfirmasi (`db.commit()`). Jika terjadi kesalahan, transaksi dibatalkan (`db.rollback()`).
- Terakhir, koneksi ke database ditutup dengan `db.close()`.

DEF LIHAT

```
def lihat():
    db = MySQLdb.connect("localhost", "root", "", "sewakamera")
    cursor = db.cursor()
    no_antrian = e_antri.get()
    sql_select = f"SELECT * FROM datasewa WHERE No_Antrian = '{no_antrian}'"
    try:
        cursor.execute(sql_select)
        data = cursor.fetchone()
        cursor.close()
        db.close()

        tampilkan(data)

    except Exception as e:
        cursor.close()
        db.close()

        messagebox.showerror("Error", f"Error: {str(e)}")
```

Pastikan telah mendefinisikan fungsi `tampilkan()` agar dapat menampilkan data yang diterima dari query SQL ke antarmuka pengguna.

- Fungsi `lihat()` berperan dalam menampilkan data dari tabel `datasewa` berdasarkan nomor antrian tertentu.
- Pertama-tama, fungsi ini membuka koneksi ke database MySQL menggunakan `MySQLdb`.
- Selanjutnya, nomor antrian diperoleh dari elemen antarmuka pengguna dengan

menggunakan `e_antri.get()`.

- Setelah itu, dibuat sebuah perintah SQL `sql_select` untuk melakukan SELECT data dari tabel `datasewa` berdasarkan nomor antrian yang telah diperoleh.
- Kemudian, perintah SQL dieksekusi menggunakan `cursor.execute(sql_select)`.
- Jika eksekusi berhasil, `cursor.fetchone()` digunakan untuk mengambil satu baris data yang sesuai dengan kriteria pencarian.
- Setelah data diperoleh, koneksi dan cursor ditutup (`cursor.close()` dan `db.close()`).
- Data yang telah ditemukan kemudian diteruskan ke fungsi `tampilkan()` untuk ditampilkan di antarmuka pengguna.
- Jika terjadi kesalahan pada saat eksekusi query, pesan error akan ditampilkan menggunakan `messagebox.showerror()`.

DEF TAMPILAN

```
def tampilkan(data):
    frame4 = Frame(notebook, width=400, height=400)
    frame4.pack_propagate(0)
    notebook.add(frame4, text="Hasil Pencarian")

    if data:
        l_judul3 = Label(frame4, text="HASIL PENCARIAN")
        l_judul3.place(x=120, y=10)

        hasil_pencarian = Text(frame4, height=10, width=40)
        hasil_pencarian.place(x=30, y=60)

        hasil_pencarian.insert(END, f"No. Antrian           : {data[0]}\n")
        hasil_pencarian.insert(END, f>Nama Lengkap       : {data[1]}\n")
        hasil_pencarian.insert(END, f"Alamat            : {data[2]}\n")
        hasil_pencarian.insert(END, f>Email             : {data[3]}\n")
        hasil_pencarian.insert(END, f"Type Kamera       : {data[4]}\n")
        hasil_pencarian.insert(END, f"Merk Kamera       : {data[5]}\n")
        hasil_pencarian.insert(END, f"Keperluan         : {data[6]}\n")

        btnutup = Button(frame4, text="tutup", command=frame4.destroy).place(x=300, y=330)
    else:
        l_tidak_ditemukan = Label(frame4, text="Data tidak ditemukan")
        l_tidak_ditemukan.place(x=120, y=10)
        btnutup = Button(frame4, text="tutup", command=frame4.destroy).place(x=300, y=330)
```

Fungsi `tampilkan()` ini bertanggung jawab untuk membuat tampilan hasil pencarian data pada antarmuka pengguna berdasarkan data yang diterima.

- Fungsi `tampilkan(data)` memiliki satu parameter masukan yaitu `data`, yang seharusnya berisi informasi yang akan ditampilkan di antarmuka.
- Pertama, fungsi ini membuat sebuah Frame baru (`frame4`) di dalam objek `notebook` (diasumsikan `notebook` adalah objek Tkinter seperti `Tk()` atau `Toplevel()` yang memiliki fungsi `.add()` untuk menambahkan frame) dengan ukuran 400x400 piksel.
- Selanjutnya, fungsi tersebut menambahkan frame yang telah dibuat ke dalam objek `notebook` dengan teks "Hasil Pencarian".
- Fungsi kemudian melakukan pemeriksaan kondisional (`if data:`) untuk memastikan bahwa data yang diterima tidak kosong (`None`).

- Jika data tidak kosong, fungsi akan membuat label judul ("HASIL PENCARIAN") di frame4 dan menampilkan informasi dari data ke dalam sebuah Text dengan tinggi 10 baris dan lebar 40 karakter.
- Setiap informasi yang ada dalam data dimasukkan ke dalam hasil_pencarian menggunakan hasil_pencarian.insert().
- Jika data kosong (tidak ada hasil pencarian), fungsi akan menampilkan label "Data tidak ditemukan" di frame4.
- Baik jika ada hasil pencarian maupun tidak, fungsi ini juga membuat tombol "tutup" yang akan menghapus frame4 saat ditekan.

DEF GENERATE

```
def generate_qr_code(data):
    qr = qrcode.QRCode(
        version= 1,
        error_correction=qrcode.constants.ERROR_CORRECT_L,
        box_size= 2,
        border= 2,
    )
    qr.add_data(data)
    qr.make(fit=True)
    img = qr.make_image(fill_color="black", back_color="white")
    return img
```

Fungsi ini merupakan implementasi sederhana untuk membuat kode QR dari sebuah teks/data tertentu menggunakan modul qrcode di Python.

- Fungsi ini menerima satu parameter, yaitu data, yang akan menjadi isi dari kode QR yang akan dibuat.
- Pertama, objek qr dibuat menggunakan qrcode.QRCode(). Pada saat pembuatan objek, beberapa parameter dikonfigurasi:
- version: Menentukan versi dari kode QR yang akan dibuat. Versi 1 adalah ukuran paling kecil, sementara versi yang lebih tinggi menunjukkan ukuran yang lebih besar.
- error_correction: Menentukan level koreksi kesalahan. qrcode.constants.ERROR_CORRECT_L adalah level koreksi kesalahan rendah.
- box_size: Ukuran satu kotak (square) dalam kode QR.
- border: Ketebalan garis pinggiran (border) dari kode QR.
- Kemudian, data yang diberikan dimasukkan ke dalam kode QR menggunakan qr.add_data(data).
- qr.make(fit=True) digunakan untuk menghasilkan kode QR dengan memfittingkan data ke ukuran terbaik.
- Selanjutnya, qr.make_image(fill_color="black", back_color="white") digunakan untuk membuat gambar kode QR dengan warna isi (fill_color) hitam dan latar belakang (back_color) putih.
- Akhirnya, gambar kode QR yang telah dibuat dikembalikan sebagai hasil dari fungsi tersebut.

DEF CETAK

```
def cetak():
    frame3 = Frame(notebook, width=400, height=400)
    frame3.pack_propagate(0)
    notebook.add(frame3, text="Laporan Sewa")
    frame3.place_forget()

    l_judul2 = Label(frame3, text="LAPORAN PERSEWAAN")
    l_judul2.place(x=120, y=10)
    laporan = Text(frame3, height=10, width=40)
    laporan.place(x=30, y=60)
    antri = e_antri.get()
    nama = e_nama.get()
    almt = e_almt.get()
    mail = e_mail.get()
    typek = r.get()
    merk = e_merk.get()
    perlu = r2.get()
    pnjm = d_pinjam.get()
    kabl = d_kembali.get()
    bayar = c_metode.get()

    laporan.insert(END, f"No. Antrian      : {antri}\n")
    laporan.insert(END, f>Nama Lengkap      : {nama}\n")
    laporan.insert(END, f"Alamat      : {almt}\n")
    laporan.insert(END, f"Email      : {mail}\n")
    laporan.insert(END, f"Type Kamera      : {typek}\n")
    laporan.insert(END, f"Merk Kamera      : {merk}\n")
    laporan.insert(END, f"Keperluan      : {perlu}\n")
    laporan.insert(END, f"Tanggal Peminjaman      : {pnjm}\n")
    laporan.insert(END, f"Tanggal Pengembalian      : {kabl}\n")
    laporan.insert(END, f"Metode Pembayaran      : {bayar}\n")

    data_to_encode = f"No. Antrian: {antri}, Nama: {nama}, Alamat: {almt}, Email: {mail}, Type Kamera: {typek}, Merk Kamera: {merk}, Keperluan: {perlu}, Tanggal Peminjaman: {pnjm}, Tangg"

    qr_code_image = generate_qr_code(data_to_encode)
    qr_code_image = ImageTk.PhotoImage(qr_code_image)

    qr_label = Label(frame3, image=qr_code_image)
    qr_label.image = qr_code_image
    qr_label.place(x= 230, y= 225)

    btnhttp = Button(frame3, text="tutup", command=frame3.destroy).place(x=30, y=330)
```

```
root = Tk()
root.geometry("380x500")
root.title("Project UAS")

notebook = ttk.Notebook(root)

# Frame for 1
frame1 = Frame(notebook, width=400, height=400)
frame1.pack_propagate(0)
notebook.add(frame1, text="Data Sewa")

l_judul = Label(frame1, text= "PERSEWAAN KAMERA")
l_antri = Label(frame1, text= "No. Antrian")
l_nama = Label(frame1, text= "Nama Lengkap")
l_almt = Label(frame1, text= "Alamat")
l_mail = Label(frame1, text= "Email")
l_type = Label(frame1, text= "Type Kamera")
l_merk = Label(frame1, text= "Merk Kamera")
l_perlu = Label(frame1, text= "Keperluan")

e_antri = Entry(frame1, width= 10)
e_nama = Entry(frame1, width= 35)
e_almt = Entry(frame1, width= 35)
e_mail = Entry(frame1, width= 35)

l_judul.place(x= 120, y= 10)
l_antri.place(x= 30, y= 60)
l_nama.place(x= 30, y= 90)
l_almt.place(x= 30, y= 120)
l_mail.place(x= 30, y= 150)
l_type.place(x= 30, y= 180)
l_merk.place(x= 30, y= 210)
l_perlu.place(x= 30, y= 240)

e_antri.place(x= 125, y= 60)
e_nama.place(x= 125, y= 90)
e_almt.place(x= 125, y= 120)
e_mail.place(x= 125, y= 150)

r = StringVar()
r.set(None)

r2 = StringVar()
```

```

rdtype1 = Radiobutton(frame1, text= "DSLR", variable= r, value= "DSLR").place(x= 125, y= 180)
rdtype2 = Radiobutton(frame1, text= "Mirrorless", variable= r, value= "Mirrorless").place(x= 190, y= 180)
rdtype3 = Radiobutton(frame1, text= "Hybrid", variable= r, value= "Hybrid").place(x= 270, y= 180)

rdperlu1 = Radiobutton(frame1, text= "Fotografi", variable= r2, value= "Fotografi").place(x= 125, y= 240)
rdperlu2 = Radiobutton(frame1, text= "Videografi", variable= r2, value= "Videografi").place(x= 220, y= 240)

btnsimpan = Button(frame1, text= "simpan", command= simpan).place(x= 203, y= 330)
btndedit = Button(frame1, text= "Edit", command= edit).place(x=170, y=330)
btnhapus = Button(frame1, text= "hapus", command= hapus).place(x= 125, y= 330)
btnlihat = Button(frame1, text= "lihat data", command= lihat).place(x= 300, y= 330)
btnklr = Button(frame1, text= "keluar", command= root.destroy).place(x= 30, y= 330)

merk = ["Canon", "Nikon", "FujiFilm", "Sony", "Lumix", "Olimpus"]
perlu = ["Fotografer", "Videografer"]

c_merk = ttk.Combobox(frame1, values= merk)
c_merk.place(x= 125, y= 210)

# Frame for 2
frame2 = Frame(notebook, width=400, height=400)
frame2.pack_propagate(0)
notebook.add(frame2, text="Data Transaksi")

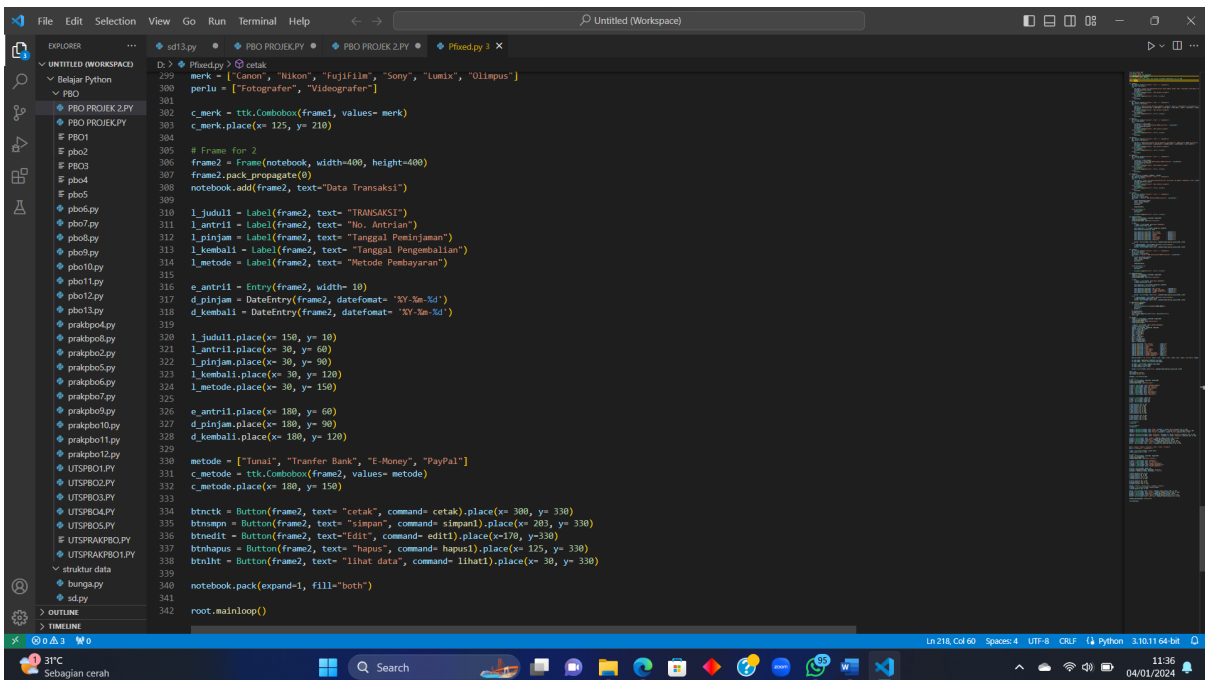
l_judul1 = Label(frame2, text= "TRANSAKSI")
l_antri1 = Label(frame2, text= "No. Antrian")
l_pinjam = Label(frame2, text= "Tanggal Pinjaman")
l_kembali = Label(frame2, text= "Tanggal Pengembalian")
l_metode = Label(frame2, text= "Metode Pembayaran")

e_antri1 = Entry(frame2, width= 10)
d_pinjam = DateEntry(frame2, dateformat= '%Y-%m-%d')
d_kembali = DateEntry(frame2, dateformat= '%Y-%m-%d')

l_judul1.place(x= 150, y= 10)
l_antri1.place(x= 30, y= 60)
l_pinjam.place(x= 30, y= 90)
l_kembali.place(x= 30, y= 120)
l_metode.place(x= 30, y= 150)

e_antri1.place(x= 180, y= 60)
d_pinjam.place(x= 180, y= 90)
d_kembali.place(x= 180, y= 120)

```



1. Tampilan Awal:

- **root** adalah jendela utama aplikasi.
- **notebook** adalah komponen dari Tkinter untuk menampung beberapa tab.
- Ada tiga tab di **notebook**: "Data Sewa", "Data Transaksi", dan "Laporan Sewa".

2. Frame 1: Data Sewa:

- Terdapat label dan entry untuk memasukkan data seperti No. Antrian, Nama, Alamat, Email, Type Kamera, Merk Kamera, dan Keperluan.
- Terdapat juga beberapa radiobuttons untuk memilih jenis kamera dan

keperluannya.

- Terdapat tombol "Simpan", "Edit", "Hapus", "Lihat Data", dan "Keluar".

3. **Frame 2: Data Transaksi:**

- Ini adalah frame untuk memasukkan data transaksi seperti No. Antrian, Tanggal Peminjaman, Tanggal Pengembalian, dan Metode Pembayaran.
- Terdapat tombol "Simpan", "Edit", "Hapus", "Lihat Data", dan "Cetak".

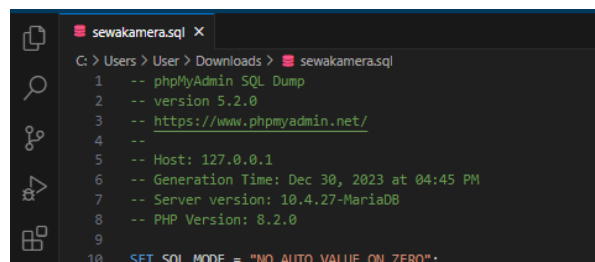
4. **Fungsi-fungsi:**

- **cetak()** adalah fungsi yang dipanggil saat tombol "cetak" ditekan pada Frame Data Transaksi. Ini mengumpulkan informasi yang dimasukkan, membuat laporan teks, membuat QR code dari informasi tersebut, dan menempatkannya di dalam "Laporan Sewa" (Frame 3).
- Fungsi-fungsi lain seperti **simpan**, **edit**, **hapus**, dan **lihat** sepertinya terkait dengan manipulasi data dan tindakan yang berhubungan dengan tombol-tombol pada antarmuka.

5. **Penggunaan Komponen GUI Tkinter:**

- **Label**, **Entry**, **Button**, **Radiobutton**, **Combobox**, **Frame**, **Notebook**, dan **Place()** digunakan untuk menampilkan elemen-elemen ini di jendela GUI.
- Pengaturan **geometry**, **title**, dan **mainloop** untuk konfigurasi jendela

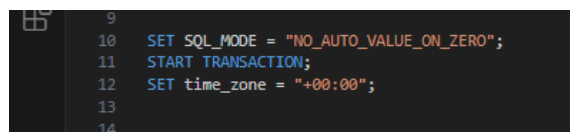
Berikut Program dan penjelasan yang akan disambungkan ke MYSQL ;



```
sewakamera.sql X
C:\Users\User>Downloads>sewakamera.sql
1  -- phpMyAdmin SQL Dump
2  -- version 5.2.0
3  -- https://www.phpmyadmin.net/
4  --
5  -- Host: 127.0.0.1
6  -- Generation Time: Dec 30, 2023 at 04:45 PM
7  -- Server version: 10.4.27-MariaDB
8  -- PHP Version: 8.2.0
9
10 SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
```

1. Informasi Header

Bagian ini berisi informasi tentang file dump SQL, termasuk versi phpMyAdmin, link ke situs resmi phpMyAdmin, informasi host, waktu pembuatan, versi server, dan versi PHP.



```
9
10 SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
11 START TRANSACTION;
12 SET time_zone = "+00:00";
13
14
```

2. Pengaturan Konfigurasi SQL: Pernyataan ini menetapkan mode SQL tertentu, memulai transaksi baru, dan menetapkan zona waktu.

- **SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";**: Menyetel mode SQL yang mencegah penambahan nilai nol secara otomatis.
- **MULAI TRANSAKSI;**: Memulai transaksi SQL.
- **SET time_zone = "+00:00";**: Tetapkan zona waktu ke UTC.

```

14
15 /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
16 /*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
17 /*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;

```

3. Simpan konfigurasi sebelumnya: Baris ini adalah instruksi yang diberi komentar untuk menyimpan konfigurasi kumpulan karakter dan susunan karakter sebelumnya.

```

18 /*!40101 SET NAMES utf8mb4 */;
19

```

4. Pengaturan dump default, pernyataan ini menyetel set karakter ke utf8mb4. `/*!40101 SET NAMES utf8mb4 */;` Setel karakter ke utf8mb4.

```

19
20 --
21 -- Database: `sewakamera`
22 --
23
24 -- -----
25
26 --

```

5. Pembuatan basis data, ini menunjukkan bahwa pernyataan SQL berikut mereferensikan basis data penyewaan kamera.
-- Basis Data: sewakamera --: Memberikan pengenalan yang menunjukkan bahwa pernyataan berikut merujuk tabel untuk pelanggan yaitu sewakamera

```

25
26 --
27 -- Table structure for table `datasewa`
28 --
29

```

6. Buat Tabel (datasewa), ini membuat tabel bernama **datasewa** dengan kolom dan tipe data yang ditentukan.

```

28 --
29
30 CREATE TABLE `datasewa` (
31   `No_Antrian` int(5) NOT NULL,
32   `Nama_Lengkap` varchar(25) NOT NULL,
33   `Alamat` varchar(25) NOT NULL,
34   `Email` varchar(30) NOT NULL,
35   `Type_Kamera` varchar(15) NOT NULL,
36   `Merk_Kamera` varchar(15) NOT NULL,
37   `Keperluan` varchar(15) NOT NULL
38 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
39

```

7. Buat tabel (datasewa), ini akan membuat tabel bernama **datasewa** dengan kolom dan tipe data yang mirip dengan contoh gambar di atas.

- CREATE TABLE datasewa (
.
.
.) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci; Membuat tabel data sewa dengan kolom yang ditentukan menggunakan mesin penyimpanan InnoDB dan rangkaian karakter utf8mb4.


```

39
40 -----
41
42 --
43 -- Table structure for table `datatransaksi`
44 --
45
46 CREATE TABLE `datatransaksi` (
47   `No_Antrian` int(5) NOT NULL,
48   `Tgl_Pinjam` date NOT NULL,
49   `Tgl_Kembali` date NOT NULL,
50   `Pembayaran` varchar(20) NOT NULL
51 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
52

```

8. Buat Tabel (Data Transaksi): Ini akan membuat tabel lain bernama Data Transaksi dengan kolom berbeda.

- CREATE TABLE Data Transaksi (.

.

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4

COLLATE=utf8mb4_general_ci; Membuat tabel data transaksi

dengan berbagai kolom menggunakan mesin penyimpanan InnoDB dan kumpulan karakter utf8mb4.

```

57
58 -- Indexes for table `datasewa`
59 --
60 ALTER TABLE `datasewa`
61   ADD PRIMARY KEY (`No_Antrian`);
62
63 --
64 -- Indexes for table `datatransaksi`
65 --
66 ALTER TABLE `datatransaksi`
67   ADD PRIMARY KEY (`No_Antrian`);
68

```

9. Indeks pada tabel, pernyataan ini menambahkan indeks kunci utama ke tabel terkait.

- ALTER TABLE Data Sewa ADD PRIMARY KEY (No_Antrian);: Menambahkan indeks kunci utama pada kolom No_Antrian pada tabel Lease Data.
- ALTER TABLE Data Transaksi ADD PRIMARY KEY (No_Antrian);: Menambahkan indeks kunci utama pada kolom No_Antrian tabel data transaksi.

```

67   ADD PRIMARY KEY (`No_Antrian`);
68 COMMIT;
69

```

10. Komit transaksi:

COMMIT; Untuk menyelesaikan transaksi dan menyimpan perubahan yang dilakukan.

```

69
70 /*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
71 /*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
72 /*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
73

```

11. Pulihkan konfigurasi sebelumnya, baris ini berisi instruksi untuk memulihkan rangkaian karakter dan konfigurasi pemeriksaan sebelumnya.

E. Implementasi Hasil

1. Menu Mengisi Data Sewa Kamera

Project UAS

Data Sewa Data Transaksi

PERSEWAAN KAMERA

No. Antrian 1

Nama Lengkap Aldy

Alamat XXXXXXX

Email XXXXXXX

Type Kamera ☒ DSLR ☐ Mirorles ☐ Hybrid

Merk Kamera Nikon

Keperluan Nikon fotografi

Canon

FujiFilm

Sony

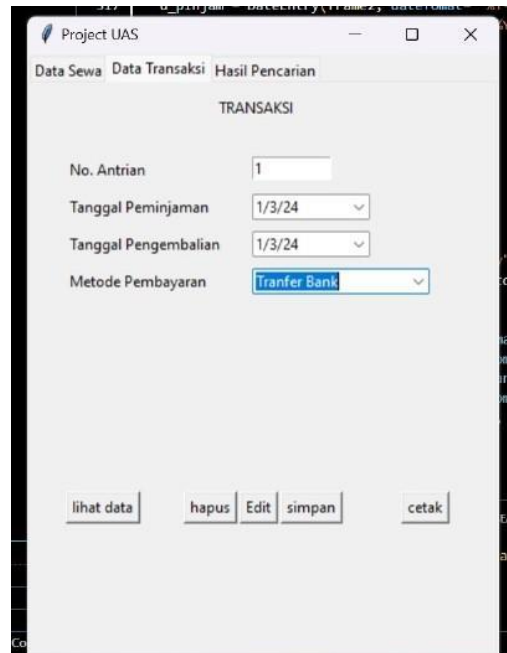
Lumix

Olimpus

keluar hapus Edit simpan lihat data

Tampilan di atas digunakan untuk mengisi data saat akan melakukan transaksi berfungsi penyimpanan data sewa pemesanan sewa kamera dan step awal dalam akan melakukan penyewaan. Sebelum itu kita sudah memberitahu secara langsung kepada pelanggan tentang perdendaan jika melakukan pengembalian telat dan juga sebelum dipinjam akan ada pengecekan barang aman kepada pelanggan, jika melakukan kerusakan pada kamera juga akan menanggung resikonya. Dan saat meminjam kita pada pihak pemilik, melakukan syarat - syarat saat meminjam kamera dengan jaminan barang berharga, seperti KTP, dll. Selanjutnya menu ini dapat mencetak struk kamera nantinya dengan nomor antrian yang diberikan oleh pemilik, dan jenis merk kamera yang kami punya yaitu Nikon, FujiFilm, Sony, Lumix, dan Olympus. Setelah itu memilih keperluan peminjaman kamera seperti apa. Selain itu juga terdapat fitur simpan, edit, lihat data, dan hapus data pelanggan.

2. Menu Data Transaksi



Project UAS

Data Sewa Data Transaksi Hasil Pencarian

TRANSAKSI

No. Antrian 1

Tanggal Peminjaman 1/3/24

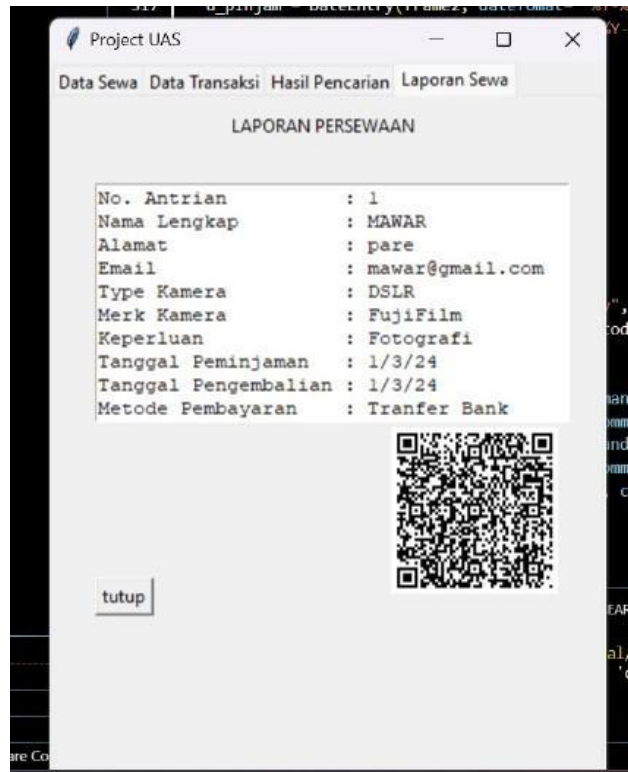
Tanggal Pengembalian 1/3/24

Metode Pembayaran Transfer Bank

lihat data hapus Edit simpan cetak

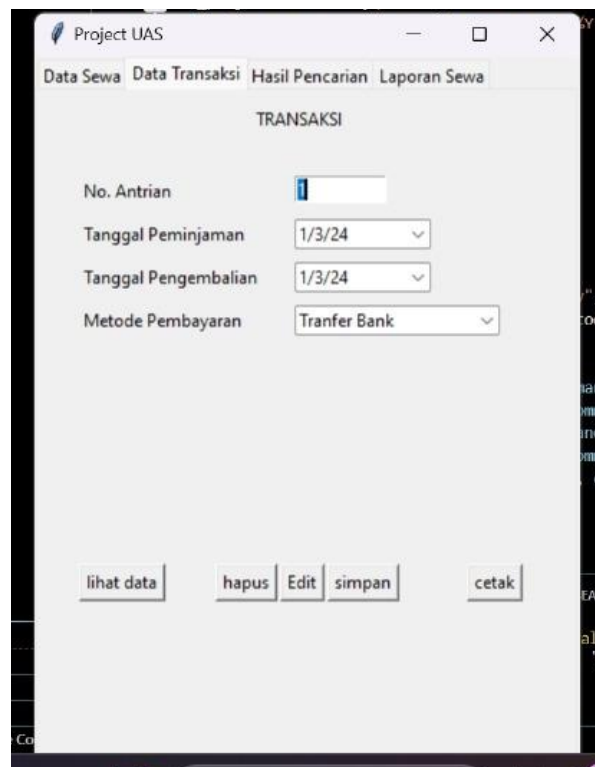
Tampilan di atas digunakan untuk mengisi data pelanggan saat peminjaman dan pengembalian kamera dihari kapan dan memilih metode pembayaran melalui apa bisa pilih pembayaran yang akan dilakukan oleh pelanggan. Setelah diisi semuanya seperti gambar diatas, lalu klik simpan maka otomatis data akan tersimpan di database yang sudah kita sediakan dan sambungkan ke php localhost tersebut. Selain itu terdapat fitur simpan, edit, lihat data dan hapus data pelanggan (jika sudah dikembalikan, klik hapus secara manual).

3. Menu Data Laporan Sewa (Pembayaran / Struk Penyewaan)



Selanjutnya Menu Data Laporan Sewa ini muncul ketidak data pelanggan yang valid disimpan lalu pencetakan struk akan muncul memencet tombol pilihan diatas yaitu Laporan Sewa dan sudah beserta barcode untuk melakukan pembayarannya, dalam transaksi transfer atau uang tunai (bayar ditempat). Tampilan di atas digunakan untuk menampilkan hasil data pelanggan saat akan melakukan pembayaran lalu kamera dapat dipinjam dan memahami prosedur peminjaman dari kami seperti adanya struk penyewaan secara online atau lewat aplikasi tersebut yang sudah dibuat oleh pemilik ini terdapat fitur tutup.

4. Menu masuk Data Transaksi untuk memilih mode pembayaran dan Tanggal Peminjaman dan Pengembalian Kamera



The screenshot shows a web application window titled "Project UAS". It has four tabs: "Data Sewa", "Data Transaksi", "Hasil Pencarian", and "Laporan Sewa". The "Data Transaksi" tab is active. The form is titled "TRANSAKSI" and contains the following fields:

- No. Antrian: A text input field with the value "1".
- Tanggal Peminjaman: A date picker showing "1/3/24".
- Tanggal Pengembalian: A date picker showing "1/3/24".
- Metode Pembayaran: A dropdown menu showing "Tranfer Bank".

At the bottom of the form, there are five buttons: "lihat data", "hapus", "Edit", "simpan", and "cetak".

Tampilan pada Data Transaksi tersebut dapat dibuka kembali dengan satu-satu jika ingin mencari kembali datanya, dengan mengetikkan nomor antrian terlebih dahulu di Form Data Transaksi lalu akan dapat mengetahuinya satu persatu data transaksi atau struk yang sudah dilakukan. Untuk cara lainnya bisa melihat langsung pada database yang sudah disambungkan ke php. Tampilan di atas digunakan untuk menampilkan satu data pelanggan yang ingin dicari.

5. Menu masuk Data Kamera belum kembali lewat Data transaksi

| | |
|----------------------|--------|
| Tanggal Peminjaman | 1/3/24 |
| Tanggal Pengembalian | 1/3/24 |

| No_Antrian | Tgl_Pinjam | Tgl_Kembali | Pembayaran |
|------------|------------|-------------|--------------|
| 1 | 0001-04-24 | 0001-04-24 | Tranfer Bank |

Menu Data penyewaan kamera yang belum dikembalikan bisa dibuka dengan memencet tombol data transaksi , Form Data Kamera Tampilan di atas digunakan untuk menampilkan data kamera yang dipinjam dan dikembalikan.

6. Menu Laporan Penyewaan

- tutup

Jika sudah selesai bisa klik tombol keluar ditekan akan langsung keluar dari program.

BAB III

PENUTUP

Kesimpulan

Sistem Database MySQL, data pelanggan disimpan dalam sebuah tabel dengan primary key berupa nomor antrian. Informasi transaksi seperti tanggal peminjaman dan pengembalian juga disimpan dalam tabel terpisah dengan kunci yang terkait dengan nomor antrian.

Antarmuka Pengguna (UI), pengguna dapat melihat data kamera yang tersedia dan melakukan transaksi sewa melalui antarmuka pengguna. Setelah transaksi selesai, nomor antrian pelanggan adalah kunci untuk mencari dan menyimpan data transaksi ke tabel terkait di MySQL.

Proses Transaksi , proses transaksi diawali dengan pelanggan memilih kamera yang ingin disewa dan memasukkan informasi pribadinya. Pelanggan diberi nomor antrian unik untuk digunakan sebagai kunci dalam sistem. Informasi transaksi seperti tanggal peminjaman dan pengembalian dicatat dan disimpan dalam tabel transaksi dengan menggunakan nomor antrian sebagai acuan.

Laporan dan Penerimaan Sewa, pengguna dapat mengakses laporan persewaan dari antarmuka pengguna dengan memilih opsi laporan ini dapat mencakup informasi seperti daftar transaksi, tanggal sewa, tanggal pengembalian, detail sewa kamera, dan lainnya. Setelah persewaan selesai, Anda dapat membuat dan melihat tanda terima pembayaran atau konfirmasi transaksi.

Kesimpulan Umum Sistem penyewaan kamera ini menggunakan database MySQL untuk menyimpan dan mengelola data pelanggan dan transaksi. Pengguna dapat dengan mudah melacak riwayat sewa dan menerima tanda terima pembayaran melalui antarmuka yang mudah digunakan.

DAFTAR PUSTAKA

Muhammad Bisrul Hafi. “RANCANGAN BANGUN SISTEM INFORMASI APLIKASI PENYEWAAN KAMERA GRESIK BERBASIS MOBILE” bisa diakses :

<https://repository.dinamika.ac.id/id/eprint/6344/1/18390100061-2022-UNIVERSITASDINA MIKA.pdf>

Rony Setiawan. 4 Agustus 2021. “Flowchart Adalah: Fungsi, Jenis, Simbol, dan Contohnya”. Bisa diakses melalui link: <https://www.dicoding.com/blog/flowchart-adalah/>

Lutfi Hanif. 22 September 2021. “Apa itu Localhost? Pengertian dan Cara Menggunakannya”. Bisa diakses melalui link : <https://blog.rumahweb.com/localhost-adalah/>