

**LAPORAN**  
**TUGAS AKHIR**  
**APLIKASI PERPUSTAKAAN BERBASIS DESKTOP**



**Mata Kuliah Pemograman Berorientasi Objek**

**Dosen Pengampu :**

M. Bahrul Subkhi, M. Kom

**Disusun Oleh :**

- |                                    |              |
|------------------------------------|--------------|
| 1. Muhammad Andrian Bhakti Maulana | (2213020200) |
| 2. Muchammad Rafli Alfiansyah      | (2213020161) |
| 3. Yoga Firnanda Wicaksono         | (2213020128) |

**KELAS 2D**  
**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
Fungsi – Fungsi Dalam Class .....	18

# **BAB I**

## **PENDAHULUAN**

### **A. Latar Belakang**

Perpustakaan adalah kumpulan sumber daya informasi yang disusun dan dikelola untuk keperluan studi, penelitian, dan pembelajaran. Latar belakang perpustakaan mencakup sejarah dan evolusi peran perpustakaan dalam masyarakat. Tradisional, perpustakaan seringkali terkait dengan koleksi buku fisik. Namun, dengan kemajuan teknologi, perpustakaan modern juga mencakup sumber daya digital seperti e-book, jurnal online, dan basis data elektronik. Perpustakaan menjadi pusat pengetahuan yang mendukung akses terbuka terhadap informasi untuk berbagai keperluan.

Peran perpustakaan juga berkembang dari sekadar menyediakan bahan bacaan menjadi penyedia layanan informasi, pelatihan, dan bimbingan dalam menggunakan sumber daya informasi. Dengan demikian, perpustakaan bertransformasi menjadi ruang pembelajaran dan kolaborasi di mana masyarakat dapat mengakses, berbagi, dan memanfaatkan pengetahuan. Penggunaan teknologi dalam perpustakaan, seperti sistem manajemen perpustakaan (Library Management System) dan aplikasi berbasis komputer, telah memodernisasi cara perpustakaan menyediakan layanan dan berinteraksi dengan pengguna. Ini memungkinkan pencarian informasi yang lebih efisien dan pengelolaan koleksi buku yang lebih baik..

Aplikasi manajemen peminjaman perpustakaan merupakan solusi modern untuk membantu mengelola dan memantau aktivitas peminjaman buku serta memberikan layanan yang lebih efisien kepada pengguna perpustakaan. Kode program yang dikembangkan menggunakan PyQt5, sebuah framework GUI untuk Python, dan MySQL sebagai sistem manajemen basis data. Aplikasi ini memiliki tujuan utama untuk menyederhanakan proses peminjaman dan pengembalian buku, meningkatkan keakuratan catatan transaksi, dan memberikan akses cepat terhadap informasi peminjaman.

Pertama-tama, aplikasi ini memiliki antarmuka pengguna yang ramah,

memanfaatkan elemen seperti tombol, tabel, dan dialog untuk memudahkan penggunaan. Pengguna dapat dengan mudah mencari dan memilih anggota atau buku menggunakan fungsi pencarian yang terintegrasi. Dengan menekan tombol, pengguna dapat memanggil dialog khusus untuk mencari anggota atau buku berdasarkan ID, nama, atau ISBN.

Aplikasi ini juga menyediakan fitur untuk melihat buku yang belum dikembalikan dan riwayat transaksi. Ini membantu pengelola perpustakaan untuk dengan cepat melacak status buku 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 buku. Dalam proses peminjaman, stok buku secara otomatis dikurangi setelah transaksi berhasil. Sebaliknya, dalam pengembalian, stok buku akan ditambahkan kembali ke dalam perpustakaan. Selain itu, aplikasi ini menyediakan opsi pembatalan transaksi, yang dapat digunakan untuk membatalkan peminjaman yang belum selesai.

Aplikasi ini juga memastikan keamanan dan validasi data. Pengecekan stok buku sebelum peminjaman dan mekanisme konfirmasi sebelum pembatalan transaksi memberikan perlindungan terhadap kesalahan dan keadaan tak terduga. Pesan dan kotak dialog memberikan umpan balik kepada pengguna, membantu mereka memahami hasil operasi dengan lebih baik. Selain itu, pengembangan aplikasi ini melibatkan integrasi MySQL untuk menyimpan dan mengelola data transaksi, anggota, dan buku. Modul ``mysql.connector`` digunakan untuk menjalankan kueri SQL dan mengelola koneksi ke basis data. Dengan kata lain, aplikasi manajemen peminjaman perpustakaan ini memberikan solusi terintegrasi untuk meningkatkan efisiensi pengelolaan perpustakaan. Pengguna dapat dengan mudah melacak status buku, melakukan peminjaman, dan mengelola transaksi dengan lebih baik, menjadikan aplikasi ini sebagai alat yang berharga dalam meningkatkan layanan perpustakaan.

## **B. Rumusan Masalah**

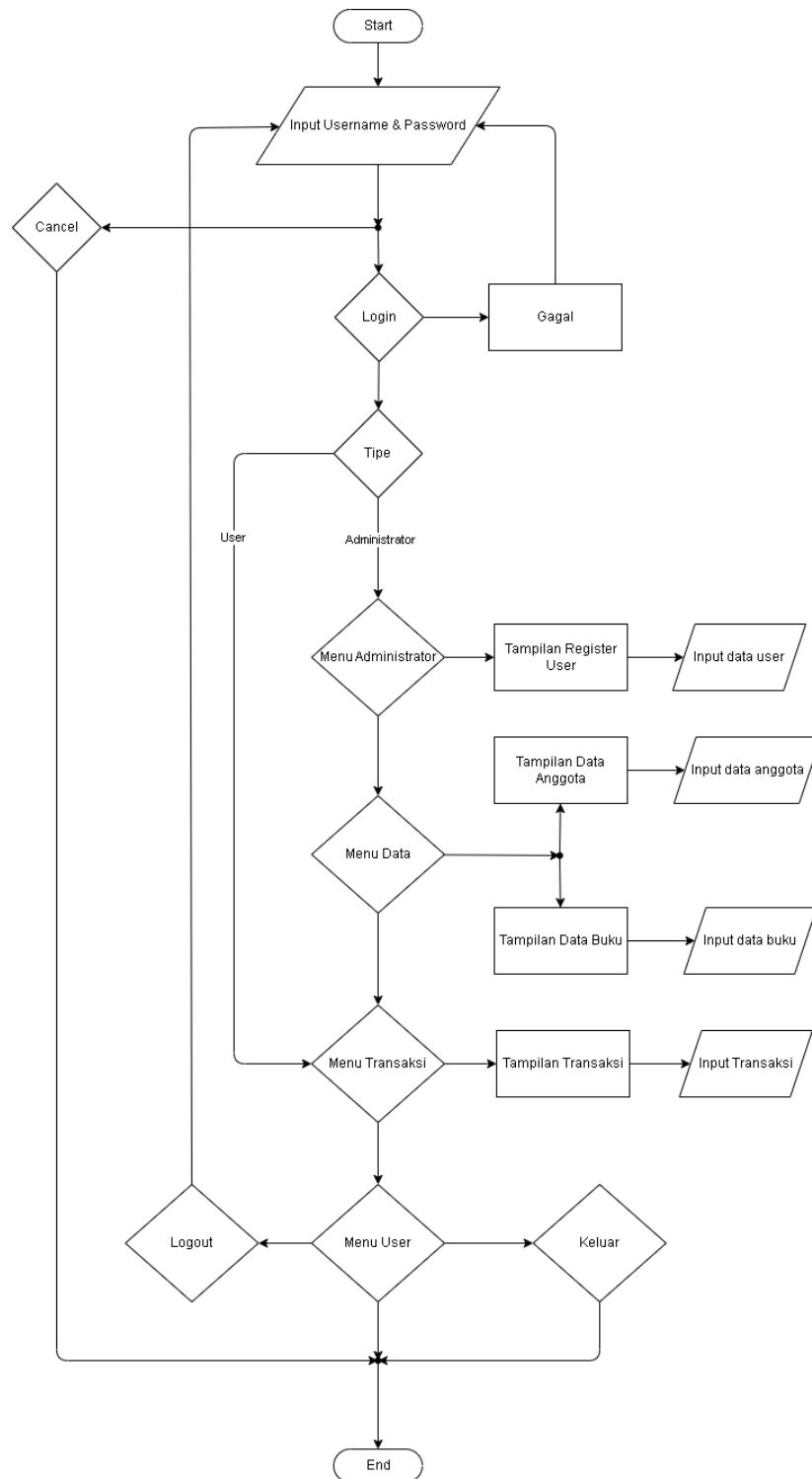
1. Proses Pencarian Anggota Perpustakaan:
  - Bagaimana meningkatkan efisiensi proses pencarian anggota perpustakaan?

2. Manajemen Peminjaman Buku:
  - Apakah ada cara untuk mengoptimalkan integrasi antara pengelolaan peminjaman dan pembaruan stok buku secara otomatis?
3. Pengembalian Buku:
  - Bagaimana mempermudah dan mempercepat proses pengembalian buku?
  - Apakah sistem dapat memberikan denda otomatis untuk buku-buku yang sudah melewati batas waktu peminjaman?
4. Pencarian Buku Belum Kembali:
  - Apakah dapat memberikan tampilan yang lebih terstruktur dan mudah dimengerti terkait buku-buku yang belum dikembalikan?
  - Bagaimana meningkatkan keakuratan dalam identifikasi buku yang harus diingatkan agar segera dikembalikan?
5. Riwayat Transaksi:
  - Bagaimana menyajikan riwayat transaksi peminjaman dengan lebih informatif dan terperinci?
6. Manajemen Stok Buku:
  - Bagaimana memastikan bahwa pembaruan stok buku terjadi secara konsisten dan akurat?
  - Apakah ada cara untuk memberikan notifikasi atau peringatan jika stok buku mencapai batas tertentu?
7. Manajemen Kesalahan:
  - Bagaimana meningkatkan manajemen kesalahan agar pengguna dapat dengan mudah mengatasi kendala teknis atau kesalahan dalam proses pengelolaan perpustakaan?

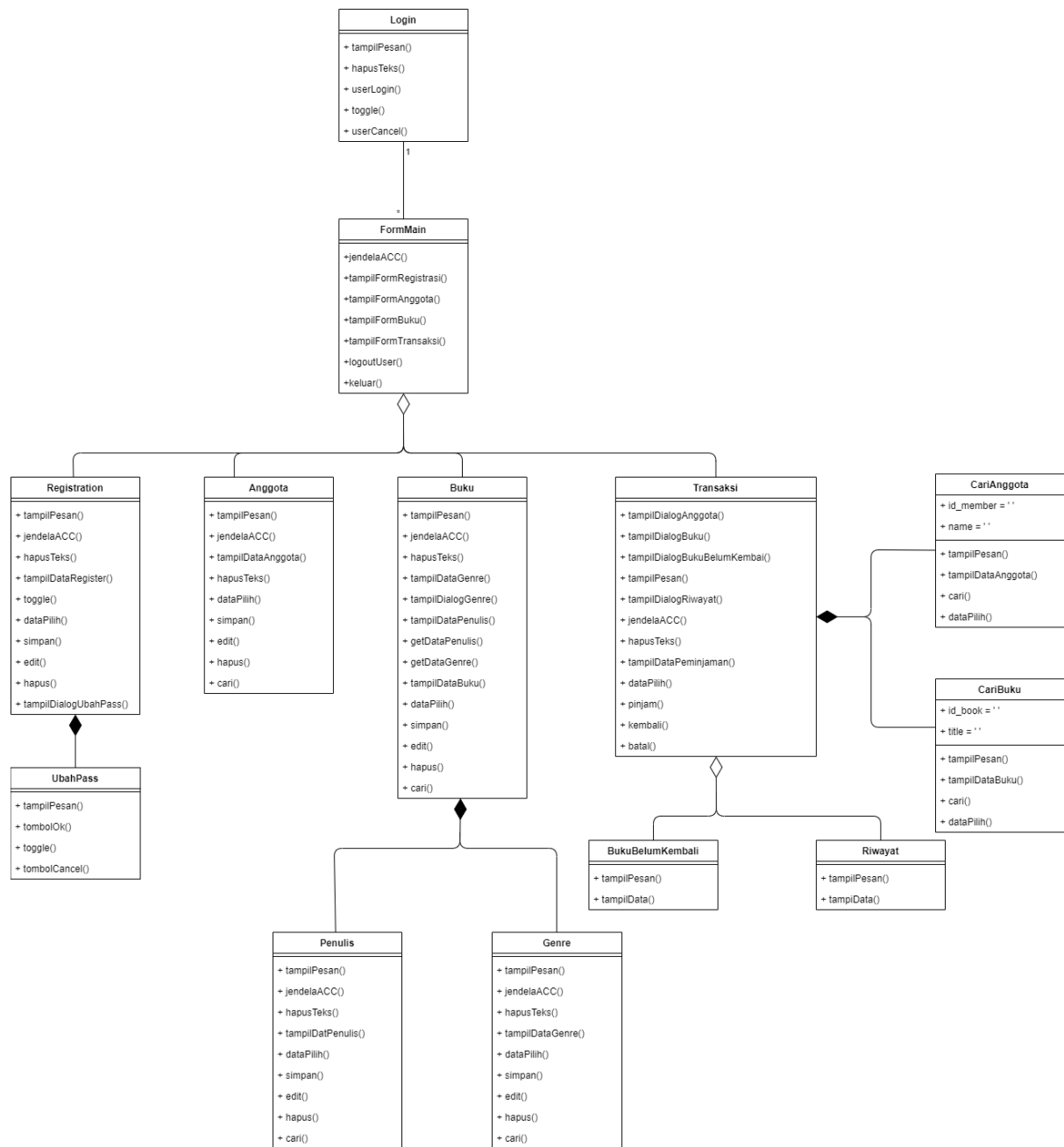
## BAB II

### PEMBAHASAN

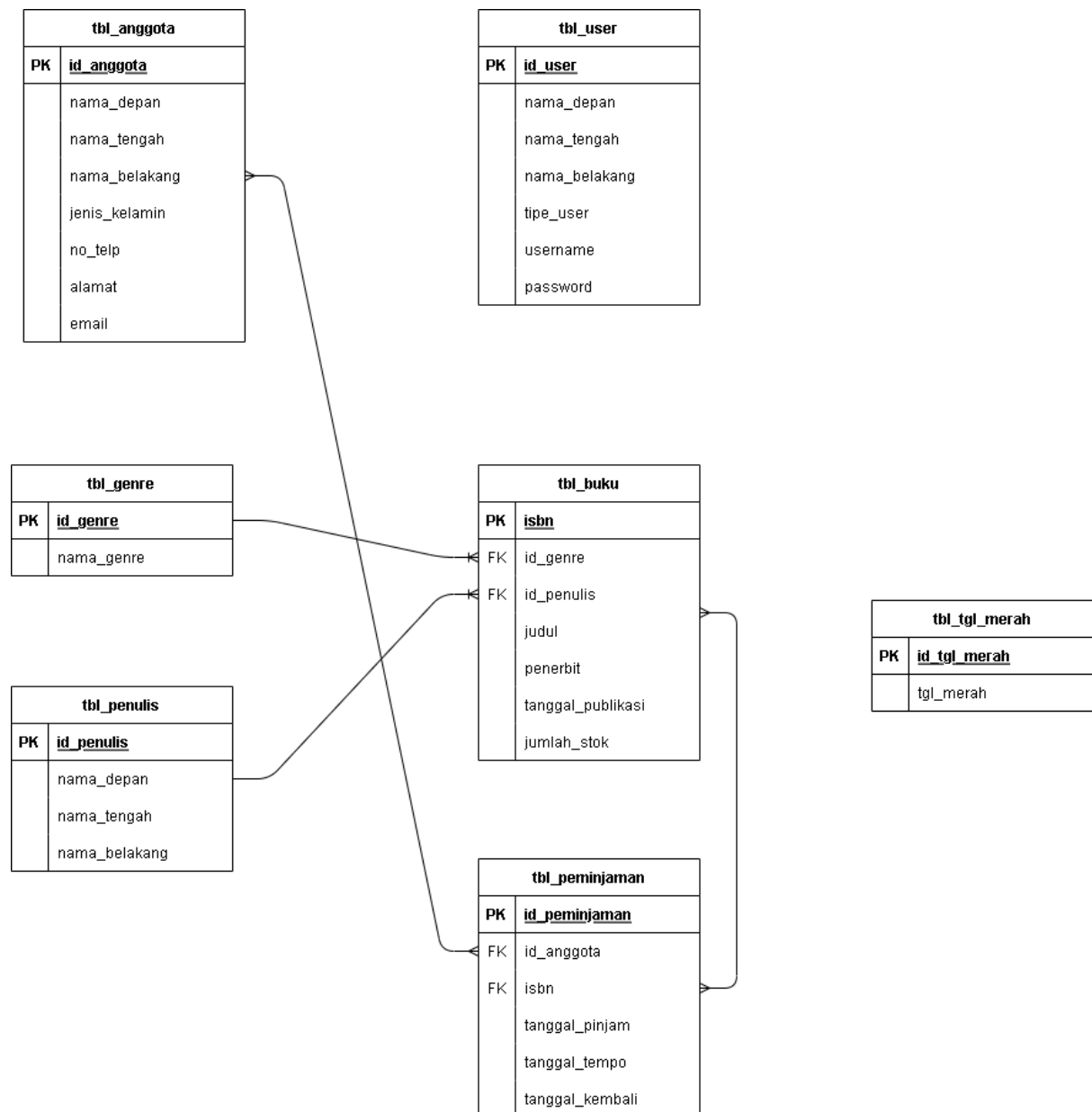
#### A. Flowchart Sistem



## B. Class Diagram

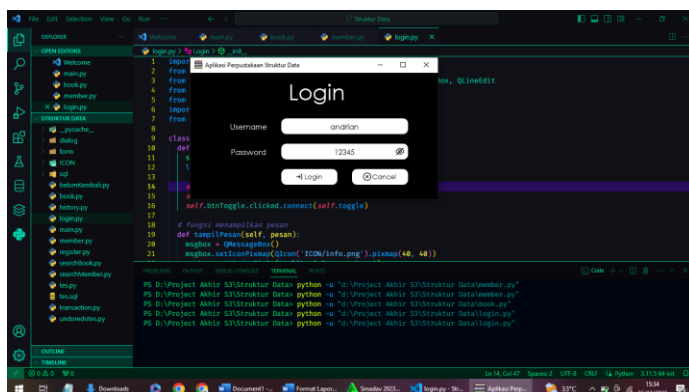


## C. Database



## D. Hasil Program dan Penjelasan

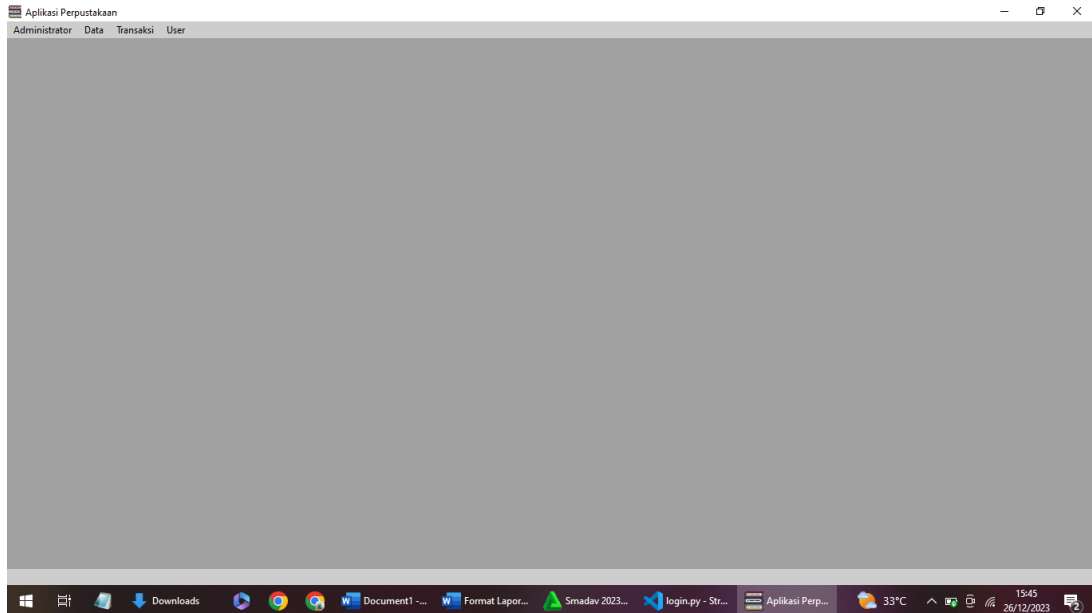
### 1. Form Login





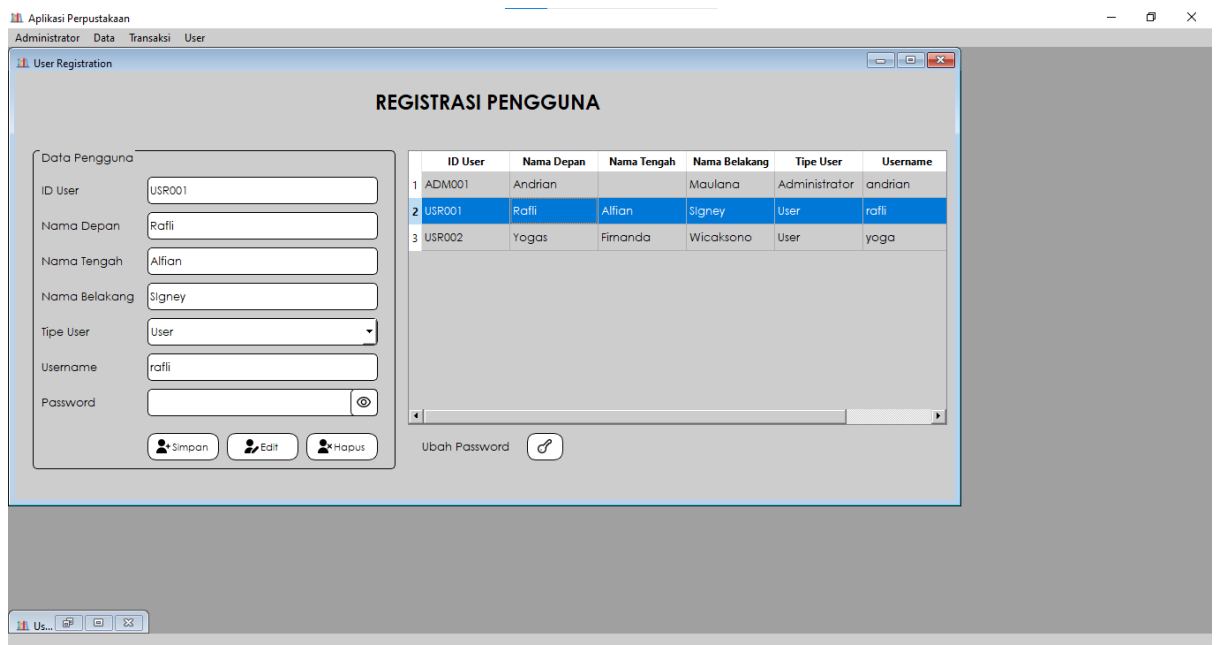
Login menggunakan username dan password yang sudah di masukkan dalam database. Jika username dan password sudah di isi klik login.

## 2. Form main menu



Jika login benar tampilan akan seperti di atas

## 3. Menu Registrasi



**REGISTRASI PENGGUNA**

Data Pengguna

ID User:

Nama Depan:

Nama Tengah:

Nama Belakang:

Tipe User:

Username:

Password:

ID User	Nama Depan	Nama Tengah	Nama Belakang	Tipe User	Username
1 ADM001	Andrian		Maulana	Administrator	andrian
2 USR001	Rafli	Alfian	Signey	User	rafli
3 USR002	Yogas	Firmanda	Wicaksono	User	yoga

Ubah Password

Tampilan di atas digunakan untuk resigtrasi pengguna yang terdapat fitur simpan, edit,ubah Password,dan hapus data pengguna.

#### 4. Menu Tampilan ubah Password

The screenshot shows a window titled "User Registration" with a tab "REGISTRASI PENGGUNA". On the left is a form for user registration with fields: ID User (USR001), Nama Depan (Rafli), Nama Tengah (Alfian), Nama Belakang (Signey), Tipe User (User), Username (rafli), and Password. Below the form are buttons: Simpan, Edit, and Hapus. On the right is a table of users:

	ID User	Nama Depan	Nama Tengah	Nama Belakang	Tipe User	Username
1	ADM001	Andrian		Maulana	Administrator	andrian
2	USR001	Rafli	Alfian	Signey	User	rafli
3	USR002	Yogas	Firmanda	Wicaksono	User	yoga

A "Changed Password" dialog box is open over the table, showing a password field with the value "12072003" and buttons "OK" and "Cancel". Below the table is a "Ubah Password" button with a key icon.

Tampilan di atas digunakan untuk mengubah password user dan admin.

#### 5. Menu Data Anggota

The screenshot shows a window titled "Aplikasi Perpustakaan" with a menu bar: Administrator, Data, Transaksi, User. The "Member Data" tab is active, showing a form for member registration and a table of members.

**Form Anggota:**

- ID Anggota
- Nama Depan
- Nama Tengah
- Nama Belakang
- Jenis Kelamin
- No. Telp / HP
- Alamat
- Email
- Buttons: Simpan, Edit, Hapus

**Table of Members:**

ID / Nama Anggota	ID Anggota	Nama Depan	Nama Tengah	Nama Belakang	Jenis Kelamin	No.Telp/HP	Alamat	Email
1	ANG001	Budi	Santoso	Wibowo	Laki-laki	081234567890	Jl. Merdeka ...	budi@gn
2	ANG002	Citra	Rahayu	Utami	Perempuan	085678901234	Jl. Raya ...	citra@ya
3	ANG003	Dharma	Adi	Prabowo	Laki-laki	081345678901	Jl. Mawar ...	dharma@
4	ANG004	Eka	Sari	Wijaya	Perempuan	087890123456	Jl. Puri ...	eka@gm
5	ANG005	Fajar	Wisnu	Saputra	Laki-laki	081234567890	Jl. Kenanga...	fajar@gn
6	ANG006	Gita	Ayunda	Kusuma	Perempuan	085678901234	Jl. Anggrek ...	gita@yah
7	ANG007	Hendra	Wijaya	Gunawan	Laki-laki	081345678901	Jl. Dahlia N...	hendra@
8	ANG008	Infan	Rahayu	Sari	Perempuan	087890123456	Jl. Jambu ...	intan@gr
9	ANG009	Joko	Susilo	Saputra	Laki-laki	081234567890	Jl. Raya ...	joko@gn
10	ANG010	Kartika	Dewi	Putri	Perempuan	085678901234	Jl. Puspa ...	kartika@
11	ANG011	Lukman	Firmansyah	Rahario	Laki-laki	081345678901	Jl. Melati N...	lukman@

Tampilan di atas digunakan untuk menampilkan data anggota yang terdapat fitur simpan, edit, dan hapus data anggota

## 6. Menu Data Buku

The screenshot shows the 'DATA BUKU' application window. On the left is a 'Form Buku' with fields for ISBN, Judul, Genre, Penulis, Penerbit, Tgl Terbit, and Jumlah Stok. Below the form are buttons for Simpan, Edit, and Hapus. On the right is a table of book data with columns: ISBN, Judul, Genre, Penulis, Penerbit, Tgl Terbit, and Jumlah Stok. The table contains 10 rows of data.

	ISBN	Judul	Genre	Penulis	Penerbit	Tgl Terbit	Jumlah Stok
1	9780061120077	One Hundre...	Agama	Gabriel ...	Harper ...	30-05-1967	22
2	9780061122415	Murder on t...	Fiksi	Agatha ...	HarperCollins	01-01-1934	10
3	9780061577079	And Then ...	Self-Help	Agnes ...	HarperCollins	01-01-1939	15
4	9780062502179	A Return to ...	Ekonomi	Marianne ...	HarperOne	01-03-1992	14
5	9780099582070	Norwegian ...	Biografi	Haruki ...	Vintage	11-08-1987	18
6	9780141439556	Pride and ...	Sastra	Jane Austen	Penguin ...	28-01-1813	25
7	9780143105954	Matilda	Kesehatan	Roald Dahl	Penguin Books	01-10-1988	18
8	9780156881807	To Kill a ...	Filosofi	Virginia ...	Harper ...	11-07-1960	30
9	9780375706772	I Know Why...	Hukum	Maya ...	Random ...	01-04-1969	16
10	9780451524935	1984	Sejarah	George ...	Signet Classic	08-06-1949	12

Tampilan di atas digunakan untuk menampilkan data buku yang terdapat fitur simpan, edit, dan hapus data buku

## 7. Menu Data Genre Buku

The screenshot shows the 'DATA BUKU' application window with the 'DATA GENRE' sub-window open. The 'Form Genre Buku' in the sub-window has fields for ID Genre and Nama Genre, with Simpan, Edit, and Hapus buttons. A table of genre data is also visible in the sub-window. The main 'Form Buku' and the main table of book data are partially visible in the background.

	ID Genre	Nama Genre
1	GNR001	Fiksi
2	GNR002	Non-Fiksi
3	GNR003	Sains
4	GNR004	Sejarah

Menu Data Genre bisa dibuka dengan memencet tombol (+) di Form Buku Genre  
Tampilan di atas digunakan untuk menampilkan data Genre yang terdapat fitur simpan, edit, dan hapus data buku

## 8. Menu Data Penulis

The screenshot shows the 'DATA BUKU' application interface. A modal window titled 'Author Data' is open, displaying the 'DATA PENULIS' form. The form includes fields for ID Penulis, Nama Depan, Nama Tengah, and Nama Belakang, along with buttons for Simpan, Edit, and Hapus. A table of existing authors is also visible, listing ID Penulis, Nama Depan, Nama Tengah, and Nama Belakang.

ID Penulis	Nama Depan	Nama Tengah	Nama Belakang
PNL001	Agatha		Christie
PNL002	Ernest	Hemingway	
PNL003	J.K.		Rowling
PNL004	George	Orwell	
PNL005	Haruki		Murakami
PNL006	Jane		Austen

Menu Data Penulis bisa dibuka dengan memencet tombol (+) di Form Buku Penulis. Tampilan di atas digunakan untuk menampilkan data Penulis yang terdapat fitur simpan, edit, dan hapus data buku.

## 9. Menu Data Peminjaman dan Pengembalian Buku

The screenshot shows the 'DATA PEMINJAMAN & PENGEMBALIAN BUKU' application interface. The form includes sections for 'Data Anggota' (ID Anggota, Nama Anggota), 'Data Buku' (ISBN, Judul Buku), and 'Tgl Peminjaman', 'Tgl Pengembalian', 'Tgl Dikembalikan'. There are buttons for Pinjam, Kembali, and Batal. A table of transactions is also visible, listing ID Peminjaman, ISBN, Judul Buku, Tgl Peminjaman, and Tgl Pengembalian.

ID Peminjaman	ISBN	Judul Buku	Tgl Peminjaman	Tgl Pengembalian
---------------	------	------------	----------------	------------------

Tampilan di atas digunakan untuk menampilkan data Peminjaman dan Pengembalian Buku yang terdapat fitur pinjam, kembali, dan batal.

## 10. Menu masuk Data Anggota lewat Data peminjaman dan pengembalian buku

Aplikasi Perpustakaan  
Administrator Data Transaksi User

Transaction Data

### DATA PEMINJAMAN & PENGEMBALIAN BUKU

Data Anggota

ID Anggota

Nama Anggota

Data Buku

ISBN

Judul Buku

Tgl Peminjaman

Tgl Pengembalian

Tgl Dikembalikan

Member Data

ID / Nama Anggota

	ID Anggota	Nama Depan	Nama Tengah	Nama Belakang	Jenis Kelamin	Tlp	Alamat	Email
1	ANG001	Budi	Santoso	Wibowo	Laki-laki	081234567890	Jl. Merdeka ...	budi@gmc
2	ANG002	Citra	Rahayu	Utami	Perempuan	085678901234	Jl. Raya ...	citra@yaho
3	ANG003	Dharma	Adi	Prabowo	Laki-laki	081345678901	Jl. Mawar ...	dharma@r
4	ANG004	Eka	Sari	Wijaya	Perempuan	087890123456	Jl. Puri ...	eka@gma
5	ANG005	Fajar	Wisnu	Saputra	Laki-laki	081234567890	Jl. Kenanga...	fajar@gmc
6	ANG006	Gita	Ayunda	Kusuma	Perempuan	085678901234	Jl. Anggrek ...	gita@yaho
7	ANG007	Hendra	Wijaya	Gunawan	Laki-laki	081345678901	Jl. Dahlia N...	hendra@h
8	ANG008	Intan	Rahayu	Sari	Perempuan	087890123456	Jl. Jambu ...	intan@am

Menu Data Anggota bisa dibuka dengan memencet tombol (+) di Form Data Anggota  
Tampilan di atas digunakan untuk menampilkan data anggota yang ingin dicari

## 11. Menu masuk Data Buku lewat Data peminjaman dan pengembalian buku

Aplikasi Perpustakaan  
Administrator Data Transaksi User

Transaction Data

### DATA PEMINJAMAN & PENGEMBALIAN BUKU

Data Anggota

ID Anggota

Nama Anggota

Data Buku

ISBN

Judul Buku

Tgl Peminjaman

Tgl Pengembalian

Tgl Dikembalikan

Book Data

ISBN / Judul Buku

	ISBN	Judul	Genre	Penulis	Penerbit	Tgl Terbit	Jumlah Stc
1	9780061122415	Murder on t...	Fiksi	Agatha ...	HarperCollins	01-01-1934	10
2	9786029144979	Laskar Pelangi	Fiksi	Andrea ...	Bentang ...	30-08-2005	14
3	9780684801469	The Old Ma...	Non-Fiksi	Ernest ...	Scribner	01-09-1952	14
4	9786020301285	Hafalan ...	Non-Fiksi	Tere Liye	Republika	01-01-2005	20
5	9781408855669	Harry Potter...	Sains	J.K. Rowling	Bloomsbury	26-06-1997	20
6	9786024252527	Supernova: ...	Sains	Dee Lestari	Bentang ...	01-01-2001	18
7	9780451524935	1984	Sejarah	George ...	Signet Classic	08-06-1949	12
8	9789792263228	Bumi Manusia	Sejarah	Pramoedya...	Hasta Mitra	01-01-1980	25

Menu Data Buku bisa dibuka dengan memencet tombol (+) di Form Data Buku.  
Tampilan di atas digunakan untuk menampilkan data buku yang ingin dicari

## 12. Menu masuk Data buku belum kembali lewat Data peminjaman dan pengembalian

Aplikasi Perpustakaan  
Administrator Data Transaksi User

Transaction Data

### DATA PEMINJAMAN & PENGEMBALIAN BUKU

Data Anggota

ID Anggota

Nama Anggota

Data Buku

ISBN

Judul Buku

Tgl Peminjaman

Tgl Pengembalian

Tgl Dikembalikan

Book Data That Has Not Been Returned

ID Peminjaman	Judul Buku	Nama Peminjam	Tgl Peminjaman	Tgl Pengembalian
1 2	Laskar Pelangi	Citra Rahay...	20-12-2023	03-01-2024
2 3	The Old Ma...	Citra Rahay...	20-12-2023	03-01-2024

Menu Data Buku yang belum dikembalikan bisa dibuka dengan memencet tombol (Buku belum Kembali) di Form Data Buku Tampilan di atas digunakan untuk menampilkan data buku yang belum kembali

## 13. Menu masuk Riwayat denda Data buku belum kembali lewat Data peminjaman dan pengembalian

Aplikasi Perpustakaan  
Administrator Data Transaksi User

Transaction Data

### DATA PEMINJAMAN & PENGEMBALIAN BUKU

Data Anggota

ID Anggota

Nama Anggota

Data Buku

ISBN

Judul Buku

Tgl Peminjaman

Tgl Pengembalian

Tgl Dikembalikan

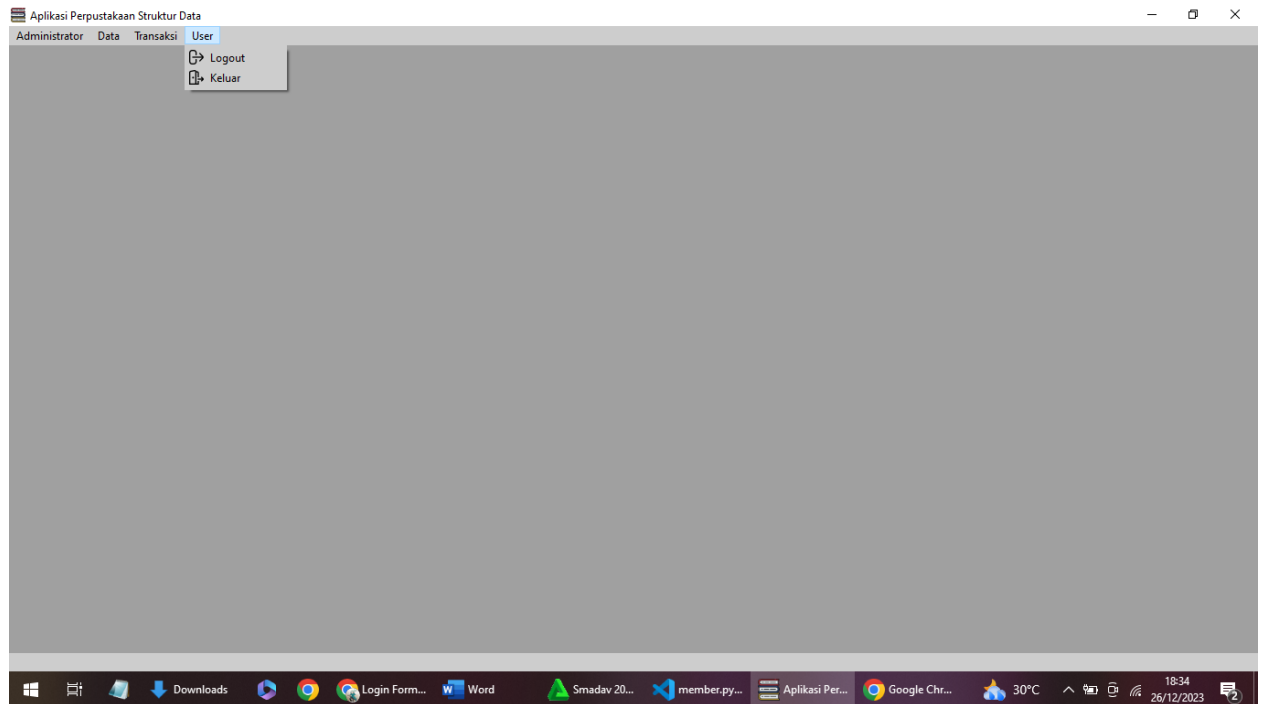
History Data

NB : Jika terlambat pengembalian Denda Rp 2000,00 Perhari

ID Peminjaman	ISBN	Judul Buku	Nama Peminjam	Tgl Peminjaman	Tgl Pengembalian	Tgl Dikembalikan	Denda
1 4	9780143105954	Matilda	Putri Cahay...	20-12-2023	01-01-2024	03-01-2024	Rp 4,000
2 1	9780061122415	Murder on t...	Budi Santos...	20-12-2023	03-01-2024	05-01-2024	Rp 4,000
3 8	9780156881807	To Kill a ...	Nugroho ...	29-12-2023	05-01-2024	06-01-2024	Rp 2,000
4 9	9780679732761	The Stranger	Nugroho ...	29-12-2023	05-01-2024	07-01-2024	Rp 2,000
5 5	9780061122415	Murder on t...	Wulans Mur...	20-12-2023	03-01-2024	08-01-2024	Rp 6,000
6 7	9780553212455	Sherlock ...	Budi Santos...	29-12-2023	05-01-2024	08-01-2024	Rp 4,000
7 6	9780451524935	1984	Wulans Mur...	01-01-2024	12-01-2024	15-01-2024	Rp 4,000

Menu Data Buku yang belum dikembalikan bisa dibuka dengan memencet tombol (Riwayat peminjaman dan denda) di Form Data Buku Tampilan di atas digunakan untuk menampilkan riwayat data buku yang belum Kembali maka akan terkena denda

#### 14. Menu User



- Logout

Setelah tombol logout ditekan akan kembali ke form login

- Keluar

Setelah tombol keluar ditekan akan langsung keluar dari program

## **BAB III**

### **PENUTUP**

#### **Kesimpulan**

Berdasarkan pengembangan sistem manajemen perpustakaan menggunakan PyQt5 dan MySQL, dapat disimpulkan bahwa implementasi ini telah berhasil meningkatkan efisiensi operasional perpustakaan secara signifikan. Antarmuka pengguna yang responsif dan intuitif, bersama dengan fitur-fitur seperti pencarian anggota dan buku, peminjaman, pengembalian, serta riwayat transaksi, telah mempermudah pengelolaan dan akses informasi.

Keamanan data dan integritasnya tetap terjaga melalui penggunaan MySQL dengan parameterized query, yang mengurangi risiko potensial seperti SQL injection. Penerapan notifikasi otomatis untuk buku yang lewat waktu peminjaman menjadi salah satu aspek yang sangat berharga, memberikan pengingat kepada pengguna dan pengelola perpustakaan terkait tenggat waktu. Pembaruan stok buku yang terintegrasi dengan baik antara aplikasi dan basis data menyiratkan adanya manajemen inventaris yang efektif. Penggunaan teknologi PyQt5 turut mendukung kinerja dan responsivitas aplikasi secara keseluruhan.

Namun, untuk menjaga keberlanjutan dan menghadapi perubahan kebutuhan pengguna, penting untuk terus melibatkan pengguna dalam proses evaluasi dan perbaikan. Evaluasi rutin akan memastikan bahwa sistem tetap relevan dan dapat diandalkan dalam memenuhi tuntutan perpustakaan modern.

Kesimpulan laporan ini bahwa implementasi sistem manajemen perpustakaan ini memberikan dampak positif dan mendukung tujuan untuk meningkatkan efisiensi dan pengelolaan perpustakaan secara keseluruhan.



## DAFTAR PUSTAKA

- Dariyanto, Hidayat, A., & Irawan, D. (2020). APLIKASI PERPUSTAKAAN BERBASIS DESKTOP PADA SMK. 117-126.
- Herrmann, M. (n.d.). *PyQt5 tutorial 2023: Create a GUI with Python and Qt*. Retrieved Desember 29, 2023, from <https://build-system.fman.io/pyqt5-tutorial>
- Muslim, F. (2016, September 28). *Membuat Database Mysql Perpustakaan*. Retrieved Desember 29, 2023, from <https://fachrezamuslim.blogspot.com/2016/09/membuat-database-perpustakaan-mysql-cmd.html>
- Visual Paradigm Online*. (2022, Februari 9). Retrieved Desember 2023, 29, from Apa Enam Jenis Hubungan Dalam Diagram Kelas UML?: <https://blog.visual-paradigm.com/id/what-are-the-six-types-of-relationships-in-uml-class-diagrams/>

## Fungsi – Fungsi Dalam Class

File login.py

```
class Login(QMainWindow):
    def __init__(self):
        super().__init__()
        loadUi('form/form_login.ui', self)

        self.btnLogin.clicked.connect(self.userLogin)
        self.btnCancel.clicked.connect(self.userCancel)
        self.btnToggle.clicked.connect(self.toggle)

        # fungsi menampilkan pesan
    def tampilPesan(self, pesan):
        msgbox = QMessageBox()
        msgbox.setIconPixmap(QIcon('ICON/info.png').pixmap(40, 40))
        msgbox.setWindowTitle('Aplikasi Perpustakaan')
        msgbox.setWindowIcon(QIcon('ICON/library_logo.png'))
        msgbox.setText(pesan)
        msgbox.exec()

        # fungsi menghapus teks saat login
    def hapusTeks(self):
        self.editUsername.setText('')
        self.editPassword.setText('')

        # fungsi melakukan login
    def userLogin(self):
        try:
            username = self.editUsername.displayText()
            password = self.editPassword.text()

            if username != '' and password != '':
                # konek ke mysql
                con = mc.connect(
                    host='localhost',
                    user='root',
                    password='',
                    database='pboperpus',
                    port='3306')

                # login berdasarkan data yang tersimpan di database
                query = 'SELECT * FROM tbl_user WHERE username=%s AND
password=MD5(%s)'
                value = (username, password)
                cursor = con.cursor()
                cursor.execute(query, value)
                data_user = cursor.fetchall()

                if len(data_user) == 1:
```

```

        self.hapusTeks()
        tipe_user = data_user[0][4] # pengecekan tipe user
        form = FormMain(tipe_user)
        widget.addWidget(form)
        widget.showMaximized()
        widget.setWindowTitle('Aplikasi Perpustakaan')
        widget.setWindowIcon(QIcon('ICON/library_logo.png'))
        widget.setCurrentIndex(1)
    else:
        self.tampilPesan('Login gagal, silahkan masukkan ulang
username dan password anda')
        self.hapusTeks()

    else:
        self.tampilPesan('Username dan Password tidak boleh kosong,
silahkan masukkan terlebih dahulu')

    except:
        self.tampilPesan('Terjadi kesalahan saat login')

# fungsi untuk tombol show hidden password
def toggle(self):
    if self.editPassword.echoMode() == QLineEdit.Password:
        self.editPassword.setEchoMode(QLineEdit.Normal)
        self.btnToggle.setIcon(QIcon('ICON/hide.png'))
    else:
        self.editPassword.setEchoMode(QLineEdit.Password)
        self.btnToggle.setIcon(QIcon('ICON/show.png'))

# fungsi tombol cancel untuk keluar dari aplikasi
def userCancel(self):
    widget.close()

```

Penjelasan:

Class yang digunakan untuk menampilkan form login untuk masuk ke aplikasi utama.

File main.py

```

class FormMain(QMainWindow):
    def __init__(self, tipeUser):
        super().__init__()
        loadUi('form/form_main.ui', self)

        self.setWindowTitle('Aplikasi Perpustakaan Kelompok 12')
        self.setWindowIcon(QIcon('ICON/library_logo.png'))

        # membuat Mdi
        self.mdi = QMdiArea()
        self.setCentralWidget(self.mdi)

```

```

# menu bar
self.menu_bar = QMenuBar()

administrator = self.menu_bar.addMenu('Administrator')
data = self.menu_bar.addMenu('Data')
transaksi = self.menu_bar.addMenu('Transaksi')
user = self.menu_bar.addMenu('User')

# sub menu
dataUser = QAction(QIcon('ICON/account.png'), 'Registrasi
Pengguna', self)
administrator.addAction(dataUser)

dataMember = QAction(QIcon('ICON/member.png'), 'Data Anggota',
self)
dataBook = QAction(QIcon('ICON/book.png'), 'Data Buku', self)
data.addAction(dataMember)
data.addAction(dataBook)

dataLoan = QAction(QIcon('ICON/loan_transaction.png'), 'Transaksi
Peminjaman Pengembalian', self)
transaksi.addAction(dataLoan)

logout = QAction(QIcon('ICON/logout.png'), 'Logout', self)
exitApp = QAction(QIcon('ICON/exit.png'), 'Keluar', self)
user.addAction(logout)
user.addAction(exitApp)

self.setMenuBar(self.menu_bar)

# trigger
dataUser.triggered.connect(self.tampilFormRegistrasi)
dataMember.triggered.connect(self.tampilFormAnggota)
dataBook.triggered.connect(self.tampilFormBuku)
dataLoan.triggered.connect(self.tampilFormTransaksi)
logout.triggered.connect(self.logoutUser)
exitApp.triggered.connect(self.keluar)

# pengecekan tipe user
if tipeUser == 'User':
    dataUser.setEnabled(False)
    dataBook.setEnabled(False)
    dataMember.setEnabled(False)

# fungsi tampilan dialog konfirmasi
def jendelaACC(self, pesan):
    msgbox = QMessageBox()

```

```

msgbox.setWindowTitle('Aplikasi Perpustakaan')
msgbox.setIconPixmap(QIcon('ICON/question.png').pixmap(40, 40))
msgbox.setWindowIcon(QIcon('ICON/library_logo.png'))
msgbox.setText(pesan)
msgbox.setStandardButtons(QMessageBox.Ok | QMessageBox.Cancel)

return msgbox.exec()

# fungsi untuk menampilkan form data user
def tampilFormRegistrasi(self):
    form = Registrasi()
    sub_form = QMdiSubWindow()
    sub_form.setWidget(form)
    sub_form.setFixedSize(1080, 520)
    self.mdi.addSubWindow(sub_form)
    sub_form.show()

# fungsi menampilkan form data anggota
def tampilFormAnggota(self):
    form = Anggota()
    sub_form = QMdiSubWindow()
    sub_form.setWidget(form)
    sub_form.setFixedSize(1290, 550)
    self.mdi.addSubWindow(sub_form)
    sub_form.show()

# fungsi menampilkan form data buku
def tampilFormBuku(self):
    form = Buku()
    sub_form = QMdiSubWindow()
    sub_form.setWidget(form)
    sub_form.setFixedSize(1140, 570)
    self.mdi.addSubWindow(sub_form)
    sub_form.show()

def tampilFormTransaksi(self):
    form = Transaksi()
    sub_form = QMdiSubWindow()
    sub_form.setWidget(form)
    sub_form.setFixedSize(990, 600)
    self.mdi.addSubWindow(sub_form)
    sub_form.show()

# fungsi untuk logout
def logoutUser(self):
    msgbx = self.jendelaACC('Apakah anda yakin untuk logout ?')
    if msgbx == QMessageBox.Ok:
        widget = self.parentWidget()

```

```

        widget.setCurrentIndex(0)
        widget.showNormal()

# fungsi keluar dari aplikasi
def keluar(self):
    msgbx = self.jendelaACC('Apakah anda yakin untuk keluar dari
aplikasi ?')
    if msgbx == QMessageBox.Ok:
        self.parentWidget().close()

```

Penjelasan:

Class ini digunakan untuk menampilkan menu utama pada aplikasi

File registration.py

```

class Registrasi(QMainWindow):
    def __init__(self):
        super().__init__()
        loadUi('form/form_register.ui', self)
        self.setWindowTitle('User Registration')
        self.setWindowIcon(QIcon('ICON/library_logo.png'))

        self.tampilDataRegister()

# tombol
        self.btnSimpan.clicked.connect(self.simpan)
        self.btnEdit.clicked.connect(self.edit)
        self.btnHapus.clicked.connect(self.hapus)
        self.btnToggle.clicked.connect(self.toggle)
        self.btnUbahPass.clicked.connect(self.tampilDialogUbahPass)

        self.tblUser.clicked.connect(self.dataPilih)

# fungsi menampilkan pesan
def tampilPesan(self, pesan):
    msgbox = QMessageBox()
    msgbox.setIconPixmap(QIcon('ICON/info.png').pixmap(40, 40))
    msgbox.setWindowTitle('Data Perpustakaan')
    msgbox.setWindowIcon(QIcon('ICON/library_logo.png'))
    msgbox.setText(pesan)
    msgbox.exec()

# fungsi jendela konfirmasi
def jendelaACC(self, pesan):
    msgbox = QMessageBox()
    msgbox.setIcon(QMessageBox.Question)
    msgbox.setWindowTitle('Aplikasi Perpustakaan')
    msgbox.setWindowIcon(QIcon('ICON/library_logo.png'))
    msgbox.setIconPixmap(QIcon('ICON/question.png').pixmap(40, 40))
    msgbox.setText(pesan)

```

```

msgbox.setStandardButtons(QMessageBox.Ok | QMessageBox.Cancel)

return msgbox.exec()

# fungsi menghapus teks saat login
def hapusTeks(self):
    self.editID.setText('')
    self.editNamaDepan.setText('')
    self.editNamaTengah.setText('')
    self.editNamaBelakang.setText('')
    self.cmbTipeUser.setCurrentIndex(0)
    self.editUsername.setText('')
    self.editPassword.setText('')

# fungsi menampilkan data register
def tampilDataRegister(self):
    try:
        # konek ke mysql
        con = mc.connect(
            host='localhost',
            user='root',
            password='',
            database='pboperpus',
            port='3306')

        # akses data user pada tbl_user
        query = 'SELECT * FROM tbl_user'
        cursor = con.cursor()
        cursor.execute(query)
        data = cursor.fetchall()

        # menutup koneksi
        con.close()

        # menampilkan data user ke form tabel
        panjang = len(data)
        self.tblUser.setRowCount(panjang)
        baris = 0
        for i in data:
            self.tblUser.setItem(baris, 0, QTableWidgetItem(i[0]))
            self.tblUser.setItem(baris, 1, QTableWidgetItem(i[1]))
            self.tblUser.setItem(baris, 2, QTableWidgetItem(i[2]))
            self.tblUser.setItem(baris, 3, QTableWidgetItem(i[3]))
            self.tblUser.setItem(baris, 4, QTableWidgetItem(i[4]))
            self.tblUser.setItem(baris, 5, QTableWidgetItem(i[5]))
            baris += 1
    except:

```

```

        self.tampilPesan('Terjadi kesalahan saat menampilkan data
pengguna')

# fungsi untuk tombol show hidden password
def toggle(self):
    if self.editPassword.echoMode() == QLineEdit.Password:
        self.editPassword.setEchoMode(QLineEdit.Normal)
        self.btnToggle.setIcon(QIcon('ICON/hide.png'))
    else:
        self.editPassword.setEchoMode(QLineEdit.Password)
        self.btnToggle.setIcon(QIcon('ICON/show.png'))

# fungsi simpan data user
def simpan(self):
    try:
        idUser = self.editID.displayText()
        nDepan = self.editNamaDepan.displayText()
        nTengah = self.editNamaTengah.displayText()
        nBelakang = self.editNamaBelakang.displayText()
        tipeUser = self.cmbTipeUser.currentText()
        username = self.editUsername.displayText()
        password = self.editPassword.text()

        if idUser != '' and nDepan != '' and tipeUser != '' and username
!= '' and password != '':
            # konek ke mysql
            con = mc.connect(
                host='localhost',
                user='root',
                password='',
                database='pboperpus',
                port='3306')

            # menyimpan data tabel form ke data tbl_user
            query = '''
                INSERT INTO tbl_user(id_user, nama_depan,
nama_tengah, nama_belakang, tipe_user, username, password)
                VALUES(%s, %s, %s, %s, %s, %s, MD5(%s))
                ,,,
            '''
            value = (idUser, nDepan, nTengah, nBelakang, tipeUser,
username, password)
            cursor = con.cursor()
            cursor.execute(query, value)
            con.commit()

            # menutup koneksi
            con.close()

```



```

        # menampilkan pesan bila berhasil
        self.tampilPesan('Data berhasil di simpan')

        # hapus teks setelah disimpan
        self.hapusTeks()

        # menampilkan data user pada tabel dengan memanggil fungsi
        tampilDataRegister
        self.tampilDataRegister()
    else:
        self.tampilPesan('Data tidak boleh kosong, silahkan lengkapi
!')

except:
    self.tampilPesan('Terjadi kesalahan saat menyimpan data')

# fungsi menampilkan data user bila di pilih
def dataPilih(self):
    try:
        # mendapatkan data user
        items = self.tblUser.selectedItems()
        idUser = items[0].text()
        nDepan = items[1].text()
        nTengah = items[2].text()
        nBelakang = items[3].text()
        tipeUser = items[4].text()
        username = items[5].text()

        # menampilkan detail data ke form
        self.editID.setText(idUser)
        self.editNamaDepan.setText(nDepan)
        self.editNamaTengah.setText(nTengah)
        self.editNamaBelakang.setText(nBelakang)
        self.editUsername.setText(username)
        if tipeUser == 'Administrator':
            self.cmbTipeUser.setCurrentIndex(1)
        elif tipeUser == 'User':
            self.cmbTipeUser.setCurrentIndex(2)
        else:
            pass

    except:
        self.tampilPesan('Terjadi kesalahan saat memilih data')

# fungsi edit data user
def edit(self):
    try:
        # mengambil data form

```

```

idUser = self.editID.displayText()
nDepan = self.editNamaDepan.displayText()
nTengah = self.editNamaTengah.displayText()
nBelakang = self.editNamaBelakang.displayText()
tipeUser = self.cmbTipeUser.currentText()
username = self.editUsername.displayText()

if idUser != '' and nDepan != '' and tipeUser != '' and username
!= '':
    # konek ke mysql
    con = mc.connect(
        host='localhost',
        user='root',
        password='',
        database='pboperpus',
        port='3306')

    # query edit data
    query = '''
        UPDATE
            tbl_user
        SET
            nama_depan=%s,
            nama_tengah=%s,
            nama_belakang=%s,
            tipe_user=%s,
            username=%s
        WHERE
            id_user=%s
    '''

    value = (nDepan, nTengah, nBelakang, tipeUser, username,
idUser)
    cursor = con.cursor()
    cursor.execute(query, value)
    con.commit()

    # menutup koneksi
    con.close()

    # menampilkan pesan bila berhasil
    self.tampilPesan('Data berhasil di edit')

    # memanggil fungsi menampilkan data pada tabel
    self.tampilDataRegister()

    # memanggil fungsi hapusTeks agar form kosong setelah dihapus
    self.hapusTeks()

```

```

        else:
            self.tampilPesan('Data tidak boleh kosong, silahkan lengkapi
!')
    except:
        self.tampilPesan('Terjadi kesalahan saat mengedit data')

# fungsi data user
def hapus(self):
    try:
        # mengambil id user
        idUser = self.editID.displayText()

        if idUser != '':
            msgbox = self.jendelaACC('Apakah anda yakin ingin menghapus
data ini ?')

            if msgbox == QMessageBox.Ok:
                # konek ke mysql
                con = mc.connect(
                    host='localhost',
                    user='root',
                    password='',
                    database='pboperpus',
                    port='3306')

                # query hapus data
                query = 'DELETE FROM tbl_user WHERE id_user=%s'
                value = (idUser,)
                cursor = con.cursor()
                cursor.execute(query, value)
                con.commit()

                # menutup koneksi
                con.close()

                self.tampilPesan('Data berhasil di hapus')

                self.tampilDataRegister()

                # memanggil fungsi hapusTeks agar form kosong setelah dihapus
                self.hapusTeks()
            else:
                pass

    else:
        self.tampilPesan('Data kosong, silahkan pilih data terlebih
dahulu')

```

```

except:
    self.tampilPesan('Terjadi kesalahan saat menghapus data')

# fungsi menampilkan dialog ubah password
def tampilDialogUbahPass(self):
    try:
        idUser = self.editID.displayText()
        if idUser != '':
            form = UbahPass(idUser)
            form.exec()
        else:
            self.tampilPesan('Silahkan pilih user yang ingin di ubah passwordnya')
    except:
        self.tampilPesan('Terjadi kesalahan saat mengganti password')

```

Penjelasan:

Class ini digunakan untuk menampilkan form register yang berguna untuk menyimpan data pengguna yang nantinya berisi username dan password yang digunakan untuk login pada aplikasi.

File changedPass.py

```

class UbahPass(QDialog):
    def __init__(self, idUser):
        super().__init__()
        loadUi('dialog/dialog_changedPass.ui', self)

        self.idUser = idUser

        self.setWindowTitle('Changed Password')
        self.setWindowIcon(QIcon('ICON/library_logo.png'))

        self.accepted.connect(self.tombolOk)
        self.rejected.connect(self.tombolCancel)
        self.btnToggle.clicked.connect(self.toggle)

# fungsi menampilkan pesan
def tampilPesan(self, pesan):
    msgbox = QMessageBox()
    msgbox.setIconPixmap(QIcon('ICON/info.png').pixmap(40, 40))
    msgbox.setWindowTitle('Data Perpustakaan')
    msgbox.setWindowIcon(QIcon('ICON/library_logo.png'))
    msgbox.setText(pesan)
    msgbox.exec()

def tombolOk(self):
    try:

```

```

passBaru = self.editPassBaru.text()
idUser = self.idUser
# konek ke mysql
con = mc.connect(
    host='localhost',
    user='root',
    password='',
    database='pboperpus',
    port='3306')

# mengubah pass pada tbl_user di database
query = 'UPDATE tbl_user SET password = MD5(%s) WHERE id_user =
%s'

data = (passBaru, idUser)
cursor = con.cursor()
cursor.execute(query, data)
con.commit()

con.close()

self.tampilPesan('Password berhasil di ubah')

except:
    self.tampilPesan('Terjadi kesalahan saat mengubah password')

# fungsi untuk tombol show hidden password
def toggle(self):
    if self.editPassBaru.echoMode() == QLineEdit.Password:
        self.editPassBaru.setEchoMode(QLineEdit.Normal)
        self.btnToggle.setIcon(QIcon('ICON/hidden.png'))
    else:
        self.editPassBaru.setEchoMode(QLineEdit.Password)
        self.btnToggle.setIcon(QIcon('ICON/show.png'))

def tombolCancel(self):
    self.close()

```

Penjelasan:

Class ini digunakan untuk mengubah password dalam register user

File member.py

```

class Anggota(QMainWindow):
    def __init__(self):
        super().__init__()
        loadUi('form/form_member.ui', self)

        self.setWindowTitle('Member Data')
        self.setWindowIcon(QIcon('ICON/library_logo.png'))

```

```

self.tampilDataAnggota()

self.btnSimpan.clicked.connect(self.simpan)
self.btnEdit.clicked.connect(self.edit)
self.btnHapus.clicked.connect(self.hapus)
self.btnSearch.clicked.connect(self.cari)

self.tblDataAnggota.clicked.connect(self.dataPilih)

# fungsi menampilkan pesan
def tampilPesan(self, pesan):
    msgbox = QMessageBox()
    msgbox.setWindowTitle('Aplikasi Perpustakaan')
    msgbox.setIconPixmap(QIcon('ICON/info.png').pixmap(40, 40))
    msgbox.setWindowIcon(QIcon('ICON/library_logo.png'))
    msgbox.setText(pesan)
    msgbox.exec()

# fungsi tampilan dialog konfirmasi
def jendelaACC(self, pesan):
    msgbox = QMessageBox()
    msgbox.setWindowTitle('Aplikasi Perpustakaan')
    msgbox.setIconPixmap(QIcon('ICON/question.png').pixmap(40, 40))
    msgbox.setWindowIcon(QIcon('ICON/library_logo.png'))
    msgbox.setText(pesan)
    msgbox.setStandardButtons(QMessageBox.Ok | QMessageBox.Cancel)

    return msgbox.exec()

# fungsi menampilkan data anggota
def tampilDataAnggota(self):
    try:
        con = mc.connect(
            host='localhost',
            user='root',
            password='',
            database='pboperpus',
            port='3306')

        # akses data user pada tbl_anggota
        query = 'SELECT * FROM tbl_anggota'
        cursor = con.cursor()
        cursor.execute(query)
        data = cursor.fetchall()

        # menutup koneksi
        con.close()

```

```

# menampilkan data anggota ke form tabel
panjang = len(data)
self.tblDataAnggota.setRowCount(panjang)
baris = 0

for i in data:
    self.tblDataAnggota.setItem(baris, 0, QTableWidgetItem(i[0]))
    self.tblDataAnggota.setItem(baris, 1, QTableWidgetItem(i[1]))
    self.tblDataAnggota.setItem(baris, 2, QTableWidgetItem(i[2]))
    self.tblDataAnggota.setItem(baris, 3, QTableWidgetItem(i[3]))
    self.tblDataAnggota.setItem(baris, 4, QTableWidgetItem(i[4]))
    self.tblDataAnggota.setItem(baris, 5, QTableWidgetItem(i[5]))
    self.tblDataAnggota.setItem(baris, 6, QTableWidgetItem(i[6]))
    self.tblDataAnggota.setItem(baris, 7, QTableWidgetItem(i[7]))

    baris += 1

except:
    self.tampilPesan('Terjadi kesalahan saat menampilkan data
anggota')

# fungsi menghapus teks line edit
def hapusTeks(self):
    self.editID.setText('')
    self.editNamaDepan.setText('')
    self.editNamaTengah.setText('')
    self.editNamaBelakang.setText('')
    self.cmbJenisKel.setCurrentIndex(0)
    self.editNoTlp.setText('')
    self.editAlamat.setText('')
    self.editEmail.setText('')

# fungsi untuk memilih data
def dataPilih(self):
    try:
        # mendapatkan data user
        items = self.tblDataAnggota.selectedItems()
        idAnggota = items[0].text()
        nDepan = items[1].text()
        nTengah = items[2].text()
        nBelakang = items[3].text()
        jenisKel = items[4].text()
        noTlp = items[5].text()
        alamat = items[6].text()
        email = items[7].text()

        # menampilkan detail data ke form

```

```

self.editID.setText(idAnggota)
self.editNamaDepan.setText(nDepan)
self.editNamaTengah.setText(nTengah)
self.editNamaBelakang.setText(nBelakang)
self.editNoTlp.setText(noTlp)
self.editAlamat.setText(alamat)
self.editEmail.setText(email)

if jenisKel == 'Laki-laki':
    self.cmbJenisKel.setCurrentIndex(1)
elif jenisKel == 'Perempuan':
    self.cmbJenisKel.setCurrentIndex(2)
else:
    pass

except:
    self.tampilPesan('Terjadi kesalahan saat memilih data')

# fungsi untuk menyimpan data
def simpan(self):
    try:
        # mengambil data form
        idAnggota = self.editID.displayText()
        nDepan = self.editNamaDepan.displayText()
        nTengah = self.editNamaTengah.displayText()
        nBelakang = self.editNamaBelakang.displayText()
        jenisKel = self.cmbJenisKel.currentText()
        noTlp = self.editNoTlp.displayText()
        alamat = self.editAlamat.displayText()
        email = self.editEmail.displayText()

        if idAnggota!='' and nDepan!='' and jenisKel!='' and noTlp!=''
and alamat!='' and email!='':
            # konek ke mysql
            con = mc.connect(
                host='localhost',
                user='root',
                password='',
                database='pboperpus',
                port='3306')

            # query untuk memasukkan data
            query = '''
                INSERT INTO tbl_anggota (id_anggota, nama_depan,
nama_tengah, nama_belakang, jenis_kelamin, no_telp, alamat, email)
                VALUES (%s, %s, %s, %s, %s, %s, %s, %s)
            '''

```



```

        value = (idAnggota, nDepan, nTengah, nBelakang, jenisKel,
noTlp, alamat, email)
        cursor = con.cursor()
        cursor.execute(query, value)
        con.commit()

        # menutup koneksi
        con.close()

        self.tampilPesan('Data berhasil di simpan')

        # menampilkan data setelah penyimpanan
        self.tampilDataAnggota()

        # memanggil hapus agar form kosong setelah disimpan
        self.hapusTeks()
    else:
        self.tampilPesan('Data tidak boleh kosong, silahkan lengkapi
!')

except:
    self.tampilPesan('Terjadi kesalahan saat menyimpan data
anggota')

# fungsi edit data
def edit(self):
    try:
        # mengambil data form
        idAnggota = self.editID.displayText()
        nDepan = self.editNamaDepan.displayText()
        nTengah = self.editNamaTengah.displayText()
        nBelakang = self.editNamaBelakang.displayText()
        jenisKel = self.cmbJenisKel.currentText()
        noTlp = self.editNoTlp.displayText()
        alamat = self.editAlamat.displayText()
        email = self.editEmail.displayText()

        if idAnggota != '' and nDepan != '' and jenisKel != '' and noTlp
!= '' and alamat != '' and email != '':
            # konek ke mysql
            con = mc.connect(
                host='localhost',
                user='root',
                password='',
                database='pboperpus',
                port='3306')

            # query edit data

```

```

        query = '''
            UPDATE
                tbl_anggota
            SET
                nama_depan=%s,
                nama_tengah=%s,
                nama_belakang=%s,
                jenis_kelamin=%s,
                alamat=%s,
                email=%s
            WHERE
                id_anggota=%s
        '''

        value = (nDepan, nTengah, nBelakang, jenisKel, alamat, email,
idAnggota)
        cursor = con.cursor()
        cursor.execute(query, value)
        con.commit()

        # menutup koneksi
        con.close()

        # menampilkan pesan bila berhasil
        self.tampilPesan('Data berhasil di edit')

        # memanggil fungsi menampilkan data pada tabel
        self.tampilDataAnggota()

        # memanggil fungsi hapusTeks agar form kosong setelah dihapus
        self.hapusTeks()
    else:
        self.tampilPesan('Data tidak boleh kosong, silahkan lengkapi
!')

except:
    self.tampilPesan('Terjadi kesalahan saat mengedit data')

# fungsi menghapus data
def hapus(self):
    try:
        # mengambil id anggota
        idAnggota = self.editID.displayText()

        if idAnggota != '':
            msgbox = self.jendelaACC('Apakah anda yakin ingin menghapus
data ini ?')

```

```

if msgbox == QMessageBox.Ok:
    # konek ke mysql
    con = mc.connect(
        host='localhost',
        user='root',
        password='',
        database='pboperpus',
        port='3306')

    # query hapus data
    query = '''
        DELETE FROM tbl_anggota
        WHERE id_anggota=%s
    '''

    value = (idAnggota,)
    cursor = con.cursor()
    cursor.execute(query, value)
    con.commit()

    # menutup koneksi
    con.close()

    self.tampilPesan('Data berhasil di hapus')

    self.tampilDataAnggota()

    # memanggil fungsi hapus agar form kosong setelah dihapus
    self.hapusTeks()
else:
    pass

else:
    self.tampilPesan('Data kosong, silahkan pilih data terlebih
dahulu')

except:
    self.tampilPesan('Terjadi kesalahan saat menghapus data')

# fungsi pencarian data
def cari(self):
    try:
        search = self.editSearch.displayText()
        con = mc.connect(
            host='localhost',
            user='root',
            password='',
            database='pboperpus',

```

```

        port='3306')

    query =f'''
        SELECT * FROM tbl_anggota
        WHERE
            id_anggota LIKE "%{search}%" OR
            nama_depan LIKE "%{search}%" OR
            nama_tengah LIKE "%{search}%" OR
            nama_belakang LIKE "%{search}%"
        '''

    cursor = con.cursor()
    cursor.execute(query)
    data = cursor.fetchall()

    con.close()

    panjang = len(data)
    self.tblDataAnggota.setRowCount(panjang)
    baris = 0

    for i in data:
        self.tblDataAnggota.setItem(baris, 0, QTableWidgetItem(i[0]))
        self.tblDataAnggota.setItem(baris, 1, QTableWidgetItem(i[1]))
        self.tblDataAnggota.setItem(baris, 2, QTableWidgetItem(i[2]))
        self.tblDataAnggota.setItem(baris, 3, QTableWidgetItem(i[3]))
        self.tblDataAnggota.setItem(baris, 4, QTableWidgetItem(i[4]))
        self.tblDataAnggota.setItem(baris, 5, QTableWidgetItem(i[5]))
        self.tblDataAnggota.setItem(baris, 6, QTableWidgetItem(i[6]))
        self.tblDataAnggota.setItem(baris, 7, QTableWidgetItem(i[7]))
        baris += 1

    except:
        self.tampilPesan('Terjadi kesalahan saat pencarian data')

```

Penjelasan:

Class ini digunakan untuk menampilkan form data anggota perpustakaan beserta simpan edit hapus datanya

File book.py

```

class Buku(QMainWindow):

    def __init__(self):
        super().__init__()
        loadUi('form/form_book.ui', self)

        self.setWindowTitle('Book Data')
        self.setWindowIcon(QIcon('ICON/library_logo.png'))

```

```

self.tampilDataGenre()
self.tampilDataPenulis()
self.tampilDataBuku()

self.tblBuku.clicked.connect(self.dataPilih)

self.btnAddGenre.clicked.connect(self.tampilDialogGenre)
self.btnAddPenulis.clicked.connect(self.tampilDialogPenulis)
self.btnSave.clicked.connect(self.simpan)
self.btnEdit.clicked.connect(self.edit)
self.btnHapus.clicked.connect(self.hapus)
self.btnSearch.clicked.connect(self.cari)

def tampilPesan(self, pesan):
    msgbox = QMessageBox()
    msgbox.setWindowTitle('Aplikasi Perpustakaan')
    msgbox.setIconPixmap(QIcon('ICON/info.png').pixmap(40, 40))
    msgbox.setWindowIcon(QIcon('ICON/library_logo.png'))
    msgbox.setText(pesan)
    msgbox.exec()

def jendelaACC(self, pesan):
    msgbox = QMessageBox()
    msgbox.setWindowTitle('Aplikasi Perpustakaan')
    msgbox.setIconPixmap(QIcon('ICON/question.png').pixmap(40, 40))
    msgbox.setWindowIcon(QIcon('ICON/library_logo.png'))
    msgbox.setText(pesan)
    msgbox.setStandardButtons(QMessageBox.Ok | QMessageBox.Cancel)

    return msgbox.exec()

def hapusTeks(self):
    self.editISBN.setText('')
    self.editJudul.setText('')
    self.cmbGenre.setCurrentIndex(0)
    self.cmbPenulis.setCurrentIndex(0)
    self.editPenerbit.setText('')
    self.dateTerbit.setDate(datetime.today())
    self.editStok.setText('')

def tampilDataGenre(self):
    try:
        con = mc.connect(
            host='localhost',
            user='root',
            password='',
            database='pboperpus',
            port='3306')

```

```

        query = 'SELECT * FROM tbl_genre'
        cursor = con.cursor()
        cursor.execute(query)
        data = cursor.fetchall()

        con.close()

        self.cmbGenre.clear()
        self.cmbGenre.addItem('')

        for i in data:
            self.cmbGenre.addItem(i[1])

    except:
        self.tampilPesan('Terjadi kesalahan saat menampilkan data genre')

def tampilDialogGenre(self):
    form = Genre()
    form.exec()
    self.tampilDataGenre()

def tampilDataPenulis(self):
    try:
        con = mc.connect(
            host='localhost',
            user='root',
            password='',
            database='pboperpus',
            port='3306')

        query = 'SELECT * FROM tbl_penulis'
        cursor = con.cursor()
        cursor.execute(query)
        data = cursor.fetchall()

        con.close()

        self.cmbPenulis.clear()
        self.cmbPenulis.addItem('')

        for i in data:
            self.cmbPenulis.addItem(i[1] + ' ' + i[2] + ' ' + i[3])

    except:
        self.tampilPesan('Terjadi kesalahan saat menampilkan data
penulis')

```

```

def tampilDialogPenulis(self):
    form = Penulis()
    form.exec()
    self.tampilDataPenulis()

# fungsi untuk mendapatkan semua data dari penulis
def getDataPenulis(self, flag, key):
    try:
        nDepan = ''
        nTengah = ''
        nBelakang = ''
        query = ''

        # flag 1 untuk mendapatkan nama penulis dan flag 2 untuk
mendapatkan id penulis
        if flag == 1:
            query = f'''
                SELECT
                    nama_depan,
                    nama_tengah,
                    nama_belakang
                FROM
                    tbl_penulis
                WHERE
                    id_penulis="{key}"
            '''
        elif flag == 2:
            nLengkap = key.split()

            if len(nLengkap) == 3:
                nDepan = nLengkap[0]
                nTengah = nLengkap[1]
                nBelakang = nLengkap[2]
            elif len(nLengkap) == 2:
                nDepan = nLengkap[0]
                nTengah = nLengkap[1]
                nBelakang = nLengkap[1]
            elif len(nLengkap) == 1:
                nDepan = nLengkap[0]

            query = f'''
                SELECT
                    id_penulis
                FROM
                    tbl_penulis
                WHERE
                    nama_depan LIKE "%{nDepan}%"
                    OR nama_tengah LIKE "%{nTengah}%"
            '''

```

```

        OR nama_belakang LIKE "%{nBelakang}%"
    '''

    con = mc.connect(
        host='localhost',
        user='root',
        password='',
        database='pboperpus',
        port='3306')

    cursor = con.cursor()
    cursor.execute(query)
    data = cursor.fetchall()

    con.close()

    # menyimpan nilai balik fungsi
    x = ''
    if flag == 1:
        x = data[0][0]+' '+data[0][1]+' '+data[0][2]
    elif flag == 2:
        x = data[0][0]

    return x

except:
    self.tampilPesan('Terjadi kesalahan saat mengakses data penulis')

# fungsi untuk mendapatkan semua data dari penulis
def getDataGenre(self, flag, key):
    try:
        query = ''
        if flag == 1:
            query = f'SELECT nama_genre FROM tbl_genre WHERE
id_genre="{key}"'
        elif flag == 2:
            query = f'SELECT id_genre FROM tbl_genre WHERE nama_genre LIKE
"%{key}%"'

        con = mc.connect(
            host='localhost',
            user='root',
            password='',
            database='pboperpus',
            port='3306')

        cursor = con.cursor()
        cursor.execute(query)

```



```

        data = cursor.fetchall()

        con.close()

        x = data[0][0]

        return x

    except:
        self.tampilPesan('Terjadi kesalahan saat mengakses data genre')

def tampilDataBuku(self):
    try:
        con = mc.connect(
            host='localhost',
            user='root',
            password='',
            database='pboperpus',
            port='3306')

        query = 'SELECT * FROM tbl_buku'
        cursor = con.cursor()
        cursor.execute(query)
        data = cursor.fetchall()

        con.close()

        panjang = len(data)
        self.tblBuku.setRowCount(panjang)
        baris = 0

        for i in data:
            genre = self.getDataGenre(1, i[1])
            penulis = self.getDataPenulis(1, i[2])

            self.tblBuku.setItem(baris, 0, QTableWidgetItem(i[0]))
            self.tblBuku.setItem(baris, 1, QTableWidgetItem(i[3]))
            self.tblBuku.setItem(baris, 2, QTableWidgetItem(genre))
            self.tblBuku.setItem(baris, 3, QTableWidgetItem(penulis))
            self.tblBuku.setItem(baris, 4, QTableWidgetItem(i[4]))
            self.tblBuku.setItem(baris, 5,
QTableWidgetItem(i[5].strftime('%d-%m-%Y')))
            self.tblBuku.setItem(baris, 6, QTableWidgetItem(str(i[6])))

            baris += 1

    except:
        self.tampilPesan('Terjadi kesalahan saat menampilkan data buku')

```

```

def dataPilih(self):
    try:
        items = self.tblBuku.selectedItems()
        isbn = items[0].text()
        judul = items[1].text()
        genre = items[2].text()
        penulis = items[3].text()
        penerbit = items[4].text()
        terbit = items[5].text()
        stok = items[6].text()

        self.editISBN.setText(isbn)
        self.editJudul.setText(judul)
        self.cmbGenre.setCurrentText(genre)
        self.cmbPenulis.setCurrentText(penulis)
        self.editPenerbit.setText(penerbit)
        self.dateTerbit.setDate(datetime.strptime(terbit, '%d-%m-%Y'))
        self.editStok.setText(stok)

    except:
        self.tampilPesan('Terjadi kesalahan saat memilih data')

def simpan(self):
    try:
        isbn = self.editISBN.displayText()
        judul = self.editJudul.toPlainText()
        genre = self.cmbGenre.currentText()
        penulis = self.cmbPenulis.currentText()
        penerbit = self.editPenerbit.displayText()
        terbit = self.dateTerbit.date()
        stok = self.editStok.displayText()

        if isbn!='' and judul!='' and genre!='' and penulis!='' and
penerbit!='' and terbit!='' and stok!='':
            idGenre = self.getDataGenre(2, genre)
            idPenulis = self.getDataPenulis(2, penulis)

            con = mc.connect(
                host='localhost',
                user='root',
                password='',
                database='pboperpus',
                port='3306')

            query = '''
                INSERT INTO tbl_buku (isbn, judul, id_genre,
id_penulis, penerbit, tanggal_publicasi, jumlah_stok)

```

```

        VALUES (%s, %s, %s, %s, %s, %s, %s)
        '''

        value = (isbn, judul, idGenre, idPenulis, penerbit,
        terbit.toString('yyyy-MM-dd'), int(stok))
        cursor = con.cursor()
        cursor.execute(query, value)
        con.commit()

        con.close()

        self.tampilPesan('Data berhasil di simpan')

        self.hapusTeks()

        self.tampilDataBuku()

    else:
        self.tampilPesan('Data tidak boleh kosong, silahkan lengkapi
!')
except:
    self.tampilPesan('Terjadi kesalahan saat menyimpan data')

def edit(self):
    try:
        isbn = self.editISBN.displayText()
        judul = self.editJudul.toPlainText()
        genre = self.cmbGenre.currentText()
        penulis = self.cmbPenulis.currentText()
        penerbit = self.editPenerbit.displayText()
        terbit = self.dateTerbit.date()
        stok = self.editStok.displayText()

        if isbn!= '' and judul!= '' and genre!= '' and penulis!= '' and
penerbit!= '' and stok!= '':
            # konek ke mysql
            con = mc.connect(
                host='localhost',
                user='root',
                password='',
                database='pboperpus',
                port='3306')

            # mendapatkan id genre dan penulis
            idGenre = self.getDataGenre(2, genre)
            idPenulis = self.getDataPenulis(2, penulis)

            # query edit data

```

```

        query = 'UPDATE tbl_buku SET judul=%s, id_genre=%s,
id_penulis=%s, penerbit=%s, tanggal_publicasi=%s, jumlah_stok=%s WHERE
isbn=%s'
        value = (judul, idGenre, idPenulis, penerbit,
terbit.toString('yyyy-MM-dd'), stok, isbn)
        cursor = con.cursor()
        cursor.execute(query, value)
        con.commit()

        # menutup koneksi
        con.close()

        # menampilkan pesan bila berhasil
        self.tampilPesan('Data berhasil di edit')

        # memanggil fungsi menampilkan data pada tabel
        self.tampilDataBuku()

        # memanggil fungsi hapusTeks agar form kosong setelah dihapus
        self.hapusTeks()
    else:
        self.tampilPesan('Data tidak boleh kosong, silahkan lengkapi
!')

except:
    self.tampilPesan('Terjadi kesalahan saat mengedit data')

def hapus(self):
    try:
        isbn = self.editISBN.displayText()

        if isbn != '':
            msgbox = self.jendelaACC('Apakah anda yakin ingin menghapus
data ini ?')

            if msgbox == QMessageBox.Ok:
                # konek ke mysql
                con = mc.connect(
                    host='localhost',
                    user='root',
                    password='',
                    database='pboperpus',
                    port='3306')

                # query hapus data
                query = 'DELETE FROM tbl_buku WHERE isbn=%s'
                value = (isbn,)
                cursor = con.cursor()

```

```

        cursor.execute(query, value)
        con.commit()

        con.close()

        self.tampilPesan('Data berhasil di hapus')

        self.tampilDataBuku()

        self.hapusTeks()
    else:
        pass

    else:
        self.tampilPesan('Data kosong, silahkan pilih data terlebih
dahulu')

    except:
        self.tampilPesan('Terjadi kesalahan saat menghapus data')

def cari(self):
    try:
        search = self.editSearch.displayText()
        con = mc.connect(
            host='localhost',
            user='root',
            password='',
            database='pboperpus',
            port='3306')

        query = f'''
            SELECT
                tbl_buku.isbn,
                tbl_buku.judul,
                tbl_genre.nama_genre,
                tbl_penulis.nama_depan,
                tbl_penulis.nama_tengah,
                tbl_penulis.nama_belakang,
                tbl_buku.penerbit,
                tbl_buku.tanggal_publicasi,
                tbl_buku.jumlah_stok
            FROM
                ((tbl_buku
                INNER JOIN tbl_penulis ON tbl_buku.id_penulis =
tbl_penulis.id_penulis)
                INNER JOIN tbl_genre ON tbl_buku.id_genre =
tbl_genre.id_genre)
            WHERE

```

```

        tbl_buku.judul LIKE "%{search}%"
        OR tbl_penulis.nama_depan LIKE "%{search}%"
        OR tbl_penulis.nama_tengah LIKE "%{search}%"
        OR tbl_penulis.nama_belakang LIKE "%{search}%"
    '''

    cursor = con.cursor()
    cursor.execute(query)
    data = cursor.fetchall()

    con.close()

    panjang = len(data)
    self.tblBuku.setRowCount(panjang)
    baris = 0

    for i in data:
        nama = i[3]+' '+'+i[4]+' '+'+i[5]

        self.tblBuku.setItem(baris, 0, QTableWidgetItem(i[0]))
        self.tblBuku.setItem(baris, 1, QTableWidgetItem(i[1]))
        self.tblBuku.setItem(baris, 2, QTableWidgetItem(i[2]))
        self.tblBuku.setItem(baris, 3, QTableWidgetItem(nama))
        self.tblBuku.setItem(baris, 4, QTableWidgetItem(i[6]))
        self.tblBuku.setItem(baris, 5,
QTableWidgetItem(i[7].strftime('%d-%m-%Y')))
        self.tblBuku.setItem(baris, 6, QTableWidgetItem(str(i[8])))

        baris += 1

    except:
        self.tampilPesan('Terjadi kesalahan saat pencarian data')

```

Penjelasan:

Class ini digunakan untuk menampilkan form data buku perpustakaan beserta simpan edit hapus datanya.

File author.py

```

class Penulis(QDialog):

    def __init__(self):
        super().__init__()
        loadUi('dialog/dialog_author.ui', self)

        self.setWindowTitle('Author Data')
        self.setWindowIcon(QIcon('ICON/library_logo.png'))

        self.tampilDataPenulis()

        self.btnSimpan.clicked.connect(self.simpan)

```

```

self.btnEdit.clicked.connect(self.edit)
self.btnHapus.clicked.connect(self.hapus)
self.btnSearch.clicked.connect(self.cari)

self.tblPenulis.clicked.connect(self.dataPilih)

def tampilPesan(self, pesan):
    msgbox = QMessageBox()
    msgbox.setWindowTitle('Aplikasi Perpustakaan')
    msgbox.setIconPixmap(QIcon('ICON/info.png').pixmap(40, 40))
    msgbox.setWindowIcon(QIcon('ICON/library_logo.png'))
    msgbox.setText(pesan)
    msgbox.exec()

def jendelaACC(self, pesan):
    msgbox = QMessageBox()
    msgbox.setWindowTitle('Aplikasi Perpustakaan')
    msgbox.setIconPixmap(QIcon('ICON/question.png').pixmap(40, 40))
    msgbox.setWindowIcon(QIcon('ICON/library_logo.png'))
    msgbox.setText(pesan)
    msgbox.setStandardButtons(QMessageBox.Ok | QMessageBox.Cancel)

    return msgbox.exec()

def hapusTeks(self):
    self.editID.setText('')
    self.editNamaDepan.setText('')
    self.editNamaTengah.setText('')
    self.editNamaBelakang.setText('')

def tampilDataPenulis(self):
    try:
        con = mc.connect(
            host='localhost',
            user='root',
            password='',
            database='pboperpus',
            port='3306')

        query = 'SELECT * FROM tbl_penulis'
        cursor = con.cursor()
        cursor.execute(query)
        data = cursor.fetchall()

        con.close()

        # menampilkan data penulis ke form tabel
        panjang = len(data)

```

```

        self.tblPenulis.setRowCount(panjang)
        baris = 0

        for i in data:
            self.tblPenulis.setItem(baris, 0, QTableWidgetItem(i[0]))
            self.tblPenulis.setItem(baris, 1, QTableWidgetItem(i[1]))
            self.tblPenulis.setItem(baris, 2, QTableWidgetItem(i[2]))
            self.tblPenulis.setItem(baris, 3, QTableWidgetItem(i[3]))

            baris += 1

    except:
        self.tampilPesan('Terjadi kesalahan saat menampilkan data
pengguna')

def dataPilih(self):
    try:
        # mengambil data penulis
        items = self.tblPenulis.selectedItems()
        idPenulis = items[0].text()
        nDepan = items[1].text()
        nTengah = items[2].text()
        nBelakang = items[3].text()

        # menampilkan detail data ke form
        self.editID.setText(idPenulis)
        self.editNamaDepan.setText(nDepan)
        self.editNamaTengah.setText(nTengah)
        self.editNamaBelakang.setText(nBelakang)

    except:
        self.tampilPesan('Terjadi kesalahan saat memilih data')

def simpan(self):
    try:
        idPenulis = self.editID.displayText()
        nDepan = self.editNamaDepan.displayText()
        nTengah = self.editNamaTengah.displayText()
        nBelakang = self.editNamaBelakang.displayText()

        if idPenulis != '' and nDepan != '':
            con = mc.connect(
                host='localhost',
                user='root',
                password='',
                database='pboperpus',
                port='3306')

```



```

        query = '''
            INSERT INTO tbl_penulis(id_penulis, nama_depan,
nama_tengah, nama_belakang)
            VALUES (%s, %s, %s, %s)
        '''

        value = (idPenulis, nDepan, nTengah, nBelakang)
        cursor = con.cursor()
        cursor.execute(query, value)
        con.commit()

        con.close()

        self.tampilPesan('Data berhasil di simpan')

        self.hapusTeks()

        self.tampilDataPenulis()
    else:
        self.tampilPesan('Data tidak boleh kosong, silahkan lengkapi
!')

except:
    self.tampilPesan('Terjadi kesalahan saat menyimpan data')

def edit(self):
    try:
        # mengambil data form
        idPenulis = self.editID.displayText()
        nDepan = self.editNamaDepan.displayText()
        nTengah = self.editNamaTengah.displayText()
        nBelakang = self.editNamaBelakang.displayText()

        if idPenulis != '' and nDepan != '':
            con = mc.connect(
                host='localhost',
                user='root',
                password='',
                database='pboperpus',
                port='3306')

            query = '''
                UPDATE
                tbl_penulis
                SET
                nama_depan=%s,
                nama_tengah=%s,
                nama_belakang=%s
            '''

```

```

        WHERE
            id_penulis=%s
    '''

    value = (nDepan, nTengah, nBelakang, idPenulis)
    cursor = con.cursor()
    cursor.execute(query, value)
    con.commit()

    con.close()

    self.tampilPesan('Data berhasil di edit')

    self.tampilDataPenulis()

    self.hapusTeks()

else:
    self.tampilPesan('Data tidak boleh kosong, silahkan lengkapi
!')

except:
    self.tampilPesan('Terjadi kesalahan saat mengedit data')

def hapus(self):
    try:
        # mengambil id penulis
        idPenulis = self.editID.displayText()

        if idPenulis != '':
            msgbox = self.jendelaACC('Apakah anda yakin ingin menghapus
data ini ?')

            if msgbox == QMessageBox.Ok:
                con = mc.connect(
                    host='localhost',
                    user='root',
                    password='',
                    database='pboperpus',
                    port='3306')

                query = 'DELETE FROM tbl_penulis WHERE id_penulis=%s'
                value = (idPenulis,)
                cursor = con.cursor()
                cursor.execute(query, value)
                con.commit()

                con.close()

```

```

        self.tampilPesan('Data berhasil di hapus')

        self.tampilDataPenulis()

        self.hapusTeks()
    else:
        pass

    else:
        self.tampilPesan('Data kosong, silahkan pilih data terlebih
dahulu')

except:
    self.tampilPesan('Terjadi kesalahan saat menghapus data')

def cari(self):
    try:
        search = self.editSearch.displayText()
        con = mc.connect(
            host='localhost',
            user='root',
            password='',
            database='pboperpus',
            port='3306')

        query = f'''
            SELECT *
            FROM tbl_penulis
            WHERE
                id_penulis LIKE "%{search}%" OR
                nama_depan LIKE "%{search}%" OR
                nama_tengah LIKE "%{search}%" OR
                nama_belakang LIKE "%{search}%"
            '''

        cursor = con.cursor()
        cursor.execute(query)
        data = cursor.fetchall()

        con.close()

        panjang = len(data)
        self.tblPenulis.setRowCount(panjang)
        baris = 0

        for i in data:
            self.tblPenulis.setItem(baris, 0, QTableWidgetItem(i[0]))

```

```

        self.tblPenulis.setItem(baris, 1, QTableWidgetItem(i[1]))
        self.tblPenulis.setItem(baris, 2, QTableWidgetItem(i[2]))
        self.tblPenulis.setItem(baris, 3, QTableWidgetItem(i[3]))

        baris += 1

    except:
        self.tampilPesan('Terjadi kesalahan saat pencarian data')

```

Penjelasan:

Class ini digunakan untuk menampilkan dialog penulis yang nantinya dihubungkan ke form data buku.

File genre.py

```

class Genre(QDialog):

    def __init__(self):
        super().__init__()
        loadUi('dialog/dialog_genre.ui', self)

        self.setWindowTitle('Genre Data')
        self.setWindowIcon(QIcon('ICON/library_logo.png'))

        self.tampilDataGenre()

        self.btnSimpan.clicked.connect(self.simpan)
        self.btnEdit.clicked.connect(self.edit)
        self.btnHapus.clicked.connect(self.hapus)

        self.tblGenre.clicked.connect(self.dataPilih)

    def tampilPesan(self, pesan):
        msgbox = QMessageBox()
        msgbox.setWindowTitle('Aplikasi Perpustakaan')
        msgbox.setIconPixmap(QIcon('ICON/info.png').pixmap(40, 40))
        msgbox.setWindowIcon(QIcon('ICON/library_logo.png'))
        msgbox.setText(pesan)
        msgbox.exec()

    def jendelaACC(self, pesan):
        msgbox = QMessageBox()
        msgbox.setWindowTitle('Aplikasi Perpustakaan')
        msgbox.setIconPixmap(QIcon('ICON/question.png').pixmap(40, 40))
        msgbox.setWindowIcon(QIcon('ICON/library_logo.png'))
        msgbox.setText(pesan)
        msgbox.setStandardButtons(QMessageBox.Ok | QMessageBox.Cancel)

        return msgbox.exec()

```

```

def hapusTeks(self):
    self.editID.setText('')
    self.editNamaGenre.setText('')

def tampilDataGenre(self):
    try:
        con = mc.connect(
            host='localhost',
            user='root',
            password='',
            database='pboperpus',
            port='3306')

        query = 'SELECT * FROM tbl_genre'
        cursor = con.cursor()
        cursor.execute(query)
        data = cursor.fetchall()

        con.close()

        panjang = len(data)
        self.tblGenre.setRowCount(panjang)
        baris = 0

        for i in data:
            self.tblGenre.setItem(baris, 0, QTableWidgetItem(i[0]))
            self.tblGenre.setItem(baris, 1, QTableWidgetItem(i[1]))

            baris += 1

    except:
        self.tampilPesan('Terjadi kesalahan saat menampilkan data genre')

def dataPilih(self):
    try:
        items = self.tblGenre.selectedItems()
        idGenre = items[0].text()
        nGenre = items[1].text()

        self.editID.setText(idGenre)
        self.editNamaGenre.setText(nGenre)

    except:
        self.tampilPesan('Terjadi kesalahan saat memilih data')

def simpan(self):
    try:
        idGenre = self.editID.displayText()

```

```

nGenre = self.editNamaGenre.displayText()

if idGenre != '' and nGenre != '':
    con = mc.connect(
        host='localhost',
        user='root',
        password='',
        database='pboperpus',
        port='3306')

    query = '''
        INSERT INTO tbl_genre(id_genre, nama_genre)
        VALUES (%s, %s)
        '''

    value = (idGenre, nGenre)
    cursor = con.cursor()
    cursor.execute(query, value)
    con.commit()

    con.close()

    self.tampilPesan('Data berhasil di simpan')

    self.hapusTeks()

    self.tampilDataGenre()
else:
    self.tampilPesan('Data tidak boleh kosong, silahkan lengkapi
!')

except:
    self.tampilPesan('Terjadi kesalahan saat menyimpan data')

def edit(self):
    try:
        idGenre = self.editID.displayText()
        nGenre = self.editNamaGenre.displayText()

        if idGenre != '' and nGenre != '':
            con = mc.connect(
                host='localhost',
                user='root',
                password='',
                database='pboperpus',
                port='3306')

            query = '''

```

```

        UPDATE
            tbl_genre
        SET
            nama_genre=%s
        WHERE
            id_genre=%s
    '''

    value = (nGenre, idGenre)
    cursor = con.cursor()
    cursor.execute(query, value)
    con.commit()

    con.close()

    self.tampilPesan('Data berhasil di edit')

    self.tampilDataGenre()

    self.hapusTeks()

    else:
        self.tampilPesan('Data tidak boleh kosong, silahkan lengkapi
!')

    except:
        self.tampilPesan('Terjadi kesalahan saat mengedit data')

def hapus(self):
    try:
        idGenre = self.editID.displayText()

        if idGenre != '':
            msgbox = self.jendelaACC('Apakah anda yakin ingin menghapus
data ini ?')

            if msgbox == QMessageBox.Ok:
                con = mc.connect(
                    host='localhost',
                    user='root',
                    password='',
                    database='pboperpus',
                    port='3306')

                query = 'DELETE FROM tbl_genre WHERE id_genre=%s'
                value = (idGenre,)
                cursor = con.cursor()
                cursor.execute(query, value)

```

```

        con.commit()

        con.close()

        self.tampilPesan('Data berhasil di hapus')

        self.tampilDataGenre()

        self.hapusTeks()
    else:
        pass

    else:
        self.tampilPesan('Data kosong, silahkan pilih data terlebih
dahulu')

    except:
        self.tampilPesan('Terjadi kesalahan saat menghapus data')

```

Penjelasan:

Class ini digunakan untuk menampilkan dialog genre yang nantinya dihubungkan ke form data buku.

File transaction.py

```

class Transaksi(QMainWindow):
    def __init__(self):
        super().__init__()
        loadUi('form/form_transaction.ui', self)

        self.setWindowTitle('Transaction Data')
        self.setWindowIcon(QIcon('ICON/library_logo.png'))

        # mengatur tanggal saat ini
        self.datePinjam.setDate(QDate.currentDate())
        self.dateTempo.setDate(QDate.currentDate())
        self.dateKembali.setDate(QDate.currentDate())

        # mengatur tanggal 14 hari setelah hari ini
        self.dateTempo.setDate(datetime.today() + timedelta(days=14))
        self.dateKembali.setDate(datetime.today() + timedelta(days=14))

        self.tblPeminjaman.clicked.connect(self.dataPilih)

        # tombol-tombol
        self.btnBorrow.clicked.connect(self.pinjam)
        self.btnReturn.clicked.connect(self.kembali)
        self.btnCancel.clicked.connect(self.batal)
        self.btnSearchMember.clicked.connect(self.tampilDialogAnggota)
        self.btnSearchBook.clicked.connect(self.tampilDialogBuku)

```



```

        self.btnNotBackYet.clicked.connect(self.tampilDialogBukuBelumKembali)
        self.btnHistory.clicked.connect(self.tampilDialogRiwayat)

    def tampilDialogAnggota(self):
        form = CariAnggota()
        form.exec()
        self.editIDAnggota.setText(form.id_member)
        self.editNamaAnggota.setText(form.name)

        self.tampilDataPeminjaman(form.id_member)

    def tampilDialogBuku(self):
        form = CariBuku()
        form.exec()
        self.editISBN.setText(form.isbn)
        self.editJudul.setText(form.title)

    def tampilDialogBukuBelumKembali(self):
        form = BukuBelumKembali()
        form.exec()

    def tampilDialogRiwayat(self):
        form = Riwayat()
        form.exec()

    # fungsi menampilkan pesan
    def tampilPesan(self, pesan):
        msgbox = QMessageBox()
        msgbox.setWindowTitle('Aplikasi Perpustakaan')
        msgbox.setIconPixmap(QIcon('ICON/info.png').pixmap(40, 40))
        msgbox.setWindowIcon(QIcon('ICON/library_logo.png'))
        msgbox.setText(pesan)
        msgbox.exec()

    # fungsi tampilan dialog konfirmasi
    def jendelaACC(self, pesan):
        msgbox = QMessageBox()
        msgbox.setWindowTitle('Aplikasi Perpustakaan')
        msgbox.setIconPixmap(QIcon('ICON/question.png').pixmap(40, 40))
        msgbox.setWindowIcon(QIcon('ICON/library_logo.png'))
        msgbox.setText(pesan)
        msgbox.setStandardButtons(QMessageBox.Ok | QMessageBox.Cancel)

        return msgbox.exec()

    def hapusTeks(self):
        self.editISBN.setText('')

```

```

self.editJudul.setText('')
self.datePinjam.setDate(datetime.today())
self.dateTempo.setDate(datetime.today() + timedelta(days=14))
self.dateKembali.setDate(datetime.today() + timedelta(days=14))

def tampilDataPeminjaman(self, idAnggota):
    try:
        con = mc.connect(
            host='localhost',
            user='root',
            password='',
            database='pboperpus',
            port='3306')

        query = f'''
            SELECT
                tbl_peminjaman.id_peminjaman,
                tbl_buku.isbn, tbl_buku.judul,
                tbl_peminjaman.tanggal_pinjam,
                tbl_peminjaman.tanggal_tempo,
                tbl_peminjaman.tanggal_kembali
            FROM
                (tbl_peminjaman
                 INNER JOIN tbl_buku ON
tbl_peminjaman.isbn=tbl_buku.isbn)
            WHERE
                tbl_peminjaman.id_anggota="{idAnggota}" AND
tbl_peminjaman.tanggal_kembali IS NULL
            '''

        cursor = con.cursor()
        cursor.execute(query)
        data = cursor.fetchall()

        con.close()

        panjang = len(data)
        self.tblPeminjaman.setRowCount(panjang)
        baris = 0

        for i in data:
            self.tblPeminjaman.setItem(baris, 0,
            QTableWidgetItem(str(i[0])))
            self.tblPeminjaman.setItem(baris, 1, QTableWidgetItem(i[1]))
            self.tblPeminjaman.setItem(baris, 2, QTableWidgetItem(i[2]))
            self.tblPeminjaman.setItem(baris, 3,
            QTableWidgetItem(i[3].strftime('%d-%m-%Y')))
            self.tblPeminjaman.setItem(baris, 4,
            QTableWidgetItem(i[4].strftime('%d-%m-%Y')))

```

```

        baris += 1

except:
    self.tampilPesan('Terjadi kesalahan saat menampilkan data')

def dataPilih(self):
    try:
        items = self.tblPeminjaman.selectedItems()
        self.idPeminjaman = items[0].text()
        isbn = items[1].text()
        judul = items[2].text()
        tglPinjam = items[3].text()
        tglTempo = items[4].text()

        self.editISBN.setText(isbn)
        self.editJudul.setText(judul)
        self.datePinjam.setDate(datetime.strptime(tglPinjam, '%d-%m-%Y'))
        self.dateTempo.setDate(datetime.strptime(tglTempo, '%d-%m-%Y'))

    except:
        self.tampilPesan('Terjadi kesalahan saat memilih data')

def pinjam(self):
    try:
        idAnggota = self.editIDAnggota.displayText()
        namaAnggota = self.editNamaAnggota.displayText()
        isbn = self.editISBN.displayText()
        tglPinjam = self.datePinjam.date()
        tglTempo = self.dateTempo.date()
        pinjam = -1 # mengurangi stok saat dipinjam

        if idAnggota!='' and isbn!='':
            con = mc.connect(
                host='localhost',
                user='root',
                password='',
                database='pboperpus',
                port='3306')

            # mengurangi stok buku
            query = 'SELECT jumlah_stok from tbl_buku WHERE isbn=%s'
            value = (isbn,)
            cursor = con.cursor()
            cursor.execute(query, value)
            stok = cursor.fetchone()[0]

            # validasi jika stok buku habis

```

```

        if stok > 0:
            query = '''
                INSERT INTO tbl_peminjaman(isbn, id_anggota,
tanggal_pinjam, tanggal_tempo)
                VALUES(%s, %s, %s, %s)
            '''

            value = (isbn, idAnggota, tglPinjam.toString('yyyy-MM-dd'),
tglTempo.toString('yyyy-MM-dd'))
            cursor = con.cursor()
            cursor.execute(query, value)
            con.commit()

            # update stok buku
            query = 'UPDATE tbl_buku SET jumlah_stok = jumlah_stok + %s
WHERE isbn=%s'
            value = (pinjam, isbn)
            cursor = con.cursor()
            cursor.execute(query, value)
            con.commit()

            self.tampilPesan('Peminjaman buku berhasil')
        else:
            self.tampilPesan('Maaf, stok buku yang ingin di pinjam sudah
habis')

        con.close()

        self.tampilDataPeminjaman(idAnggota)

        self.hapusTeks()

    else:
        self.tampilPesan('Data tidak boleh kosong, silahkan lengkapi
!')

    except:
        self.tampilPesan('Terjadi kesalahan saat melakukan peminjaman')

    def kembali(self):
        try:
            idPeminjaman = self.idPeminjaman
            idAnggota = self.editIDAnggota.displayText()
            isbn = self.editISBN.displayText()
            tglKembali = self.dateKembali.date()
            kembali = 1 # menambahkan stok saat dikembalikan

            if idPeminjaman!='' and idAnggota!='' and isbn!='':

```

```

        con = mc.connect(
            host='localhost',
            user='root',
            password='',
            database='pboperpus',
            port='3306')

        query = 'UPDATE tbl_peminjaman SET tanggal_kembali=%s WHERE
id_peminjaman=%s'
        value = (tglKembali.toString('yyyy-MM-dd'), int(idPeminjaman))
        cursor = con.cursor()
        cursor.execute(query, value)
        con.commit()

        # update stok buku
        query = 'UPDATE tbl_buku SET jumlah_stok = jumlah_stok + %s
WHERE isbn=%s'
        value = (kembali, isbn)
        cursor = con.cursor()
        cursor.execute(query, value)
        con.commit()

        con.close()

        self.tampilPesan('Pengembalian buku berhasil')

        self.tampilDataPeminjaman(idAnggota)

        self.hapusTeks()
    else:
        self.tampilPesan('Data tidak boleh kosong, silahkan lengkapi
!')

except:
    self.tampilPesan('Terjadi kesalahan saat melakukan pengembalian')

def batal(self):
    try:
        idPeminjaman = self.idPeminjaman
        idAnggota = self.editIDAnggota.displayText()
        isbn = self.editISBN.displayText()
        kembali = 1

        if idPeminjaman!='' and idAnggota!='' and isbn!='':
            msgbox = self.jendelaACC('Apakah anda yakin ingin membatalkan
pinjaman ?')

            if msgbox == QMessageBox.Ok:

```

```

        con = mc.connect(
            host='localhost',
            user='root',
            password='',
            database='pboperpus',
            port='3306')

        query = 'DELETE FROM tbl_peminjaman WHERE id_peminjaman=%s'
        value = (int(idPeminjaman),)
        cursor = con.cursor()
        cursor.execute(query, value)

        # update stok buku
        query = 'UPDATE tbl_buku SET jumlah_stok = jumlah_stok + %s
WHERE isbn=%s'
        value = (kembali, isbn)
        cursor = con.cursor()
        cursor.execute(query, value)

        con.commit()

        con.close()

        self.tampilDataPeminjaman(idAnggota)

        self.hapusTeks()

    except:
        self.tampilPesan('Terjadi kesalahan saat membatalkan peminjaman')

```

Penjelasan:

Class ini digunakan untuk transaksi peminjaman dan pengembalian buku terdapat fitur pinjam, kembali, dan batal.

File searchMember.py

```

class CariAnggota(QDialog):
    id_member = ''
    name = ''

    def __init__(self):
        super().__init__()
        loadUi('dialog/dialog_member.ui', self)

        self.setWindowTitle('Member Data')
        self.setWindowIcon(QIcon('ICON/library_logo.png'))

        self.tampilDataAnggota()
        self.btnSearch.clicked.connect(self.cari)
        self.tblAnggota.doubleClicked.connect(self.dataPilih)

```

```

def tampilPesan(self, pesan):
    msgbox = QMessageBox()
    msgbox.setWindowTitle('Aplikasi Perpustakaan')
    msgbox.setIconPixmap(QIcon('ICON/info.png').pixmap(40, 40))
    msgbox.setWindowIcon(QIcon('ICON/library_logo.png'))
    msgbox.setText(pesan)
    msgbox.exec()

def tampilDataAnggota(self):
    try:
        con = mc.connect(
            host='localhost',
            user='root',
            password='',
            database='pboperpus',
            port='3306')

        # akses data user pada tbl_anggota
        query = 'SELECT * FROM tbl_anggota'
        cursor = con.cursor()
        cursor.execute(query)
        data = cursor.fetchall()

        # menutup koneksi
        con.close()

        # menampilkan data anggota ke form tabel
        panjang = len(data)
        self.tblAnggota.setRowCount(panjang)
        baris = 0
        for i in data:
            self.tblAnggota.setItem(baris, 0, QTableWidgetItem(i[0]))
            self.tblAnggota.setItem(baris, 1, QTableWidgetItem(i[1]))
            self.tblAnggota.setItem(baris, 2, QTableWidgetItem(i[2]))
            self.tblAnggota.setItem(baris, 3, QTableWidgetItem(i[3]))
            self.tblAnggota.setItem(baris, 4, QTableWidgetItem(i[4]))
            self.tblAnggota.setItem(baris, 5, QTableWidgetItem(i[5]))
            self.tblAnggota.setItem(baris, 6, QTableWidgetItem(i[6]))
            self.tblAnggota.setItem(baris, 7, QTableWidgetItem(i[7]))
            baris += 1
        except:
            self.tampilPesan('Terjadi kesalahan saat menampilkan data anggota')

def cari(self):
    try:
        search = self.editSearch.displayText()

```

```

con = mc.connect(
    host='localhost',
    user='root',
    password='',
    database='pboperpus',
    port='3306')

query = f'''
    SELECT
        *
    FROM
        tbl_anggota
    WHERE
        id_anggota LIKE "%{search}%"
        OR nama_depan LIKE "%{search}%"
        OR nama_tengah LIKE "%{search}%"
        OR nama_belakang LIKE "%{search}%"
    '''

cursor = con.cursor()
cursor.execute(query)
data = cursor.fetchall()

con.close()

panjang = len(data)
self.tblAnggota.setRowCount(panjang)
baris = 0

for i in data:
    self.tblAnggota.setItem(baris, 0, QTableWidgetItem(i[0]))
    self.tblAnggota.setItem(baris, 1, QTableWidgetItem(i[1]))
    self.tblAnggota.setItem(baris, 2, QTableWidgetItem(i[2]))
    self.tblAnggota.setItem(baris, 3, QTableWidgetItem(i[3]))
    self.tblAnggota.setItem(baris, 4, QTableWidgetItem(i[4]))
    self.tblAnggota.setItem(baris, 5, QTableWidgetItem(i[5]))
    self.tblAnggota.setItem(baris, 6, QTableWidgetItem(i[6]))
    self.tblAnggota.setItem(baris, 7, QTableWidgetItem(i[7]))
    baris += 1

except:
    self.tampilPesan('Terjadi kesalahan saat pencarian data')

def dataPilih(self):
    items = self.tblAnggota.selectedItems()
    name = items[1].text()+' '+items[2].text()+' '+items[3].text()
    self.id_member = items[0].text()
    self.name = name

```



```
self.close()
```

Penjelasan:

Class ini digunakan untuk menampilkan dialog daftar anggota yang nantinya digunakan untuk memilih anggota yang ingin meminjam atau mengembalikan buku, class ini dihubungkan dengan transaction.py

File searchBook.py

```
class CariBuku(QDialog):
    isbn = ''
    title = ''

    def __init__(self):
        super().__init__()
        loadUi('dialog/dialog_book.ui', self)

        self.setWindowTitle('Book Data')
        self.setWindowIcon(QIcon('ICON/library_logo.png'))

        self.tampilDataBuku()
        self.btnSearch.clicked.connect(self.cari)
        self.tblBuku.doubleClicked.connect(self.dataPilih)

    def tampilPesan(self, pesan):
        msgbox = QMessageBox()
        msgbox.setWindowTitle('Aplikasi Perpustakaan')
        msgbox.setIconPixmap(QIcon('ICON/info.png').pixmap(40, 40))
        msgbox.setWindowIcon(QIcon('ICON/library_logo.png'))
        msgbox.setText(pesan)
        msgbox.exec()

    def tampilDataBuku(self):
        try:
            con = mc.connect(
                host='localhost',
                user='root',
                password='',
                database='pboperpus',
                port='3306')

            query = f'''
                SELECT
                    tbl_buku.isbn,
                    tbl_buku.judul,
                    tbl_genre.nama_genre,
                    tbl_penulis.nama_depan,
                    tbl_penulis.nama_tengah,
                    tbl_penulis.nama_belakang,
```

```

        tbl_buku.penerbit,
        tbl_buku.tanggal_publicasi,
        tbl_buku.jumlah_stok
    FROM
        ((tbl_buku
            INNER JOIN tbl_penulis ON tbl_buku.id_penulis =
tbl_penulis.id_penulis)
            INNER JOIN tbl_genre ON tbl_buku.id_genre =
tbl_genre.id_genre)
    '''

    cursor = con.cursor()
    cursor.execute(query)
    data = cursor.fetchall()

    con.close()

    panjang = len(data)
    self.tblBuku.setRowCount(panjang)
    baris = 0

    for i in data:
        nama = i[3]+' '+'+i[4]+' '+'+i[5]

        self.tblBuku.setItem(baris, 0, QTableWidgetItem(i[0]))
        self.tblBuku.setItem(baris, 1, QTableWidgetItem(i[1]))
        self.tblBuku.setItem(baris, 2, QTableWidgetItem(i[2]))
        self.tblBuku.setItem(baris, 3, QTableWidgetItem(nama))
        self.tblBuku.setItem(baris, 4, QTableWidgetItem(i[6]))
        self.tblBuku.setItem(baris, 5,
QTableWidgetItem(i[7].strftime('%d-%m-%Y')))
        self.tblBuku.setItem(baris, 6, QTableWidgetItem(str(i[8])))

        baris += 1

    except:
        self.tampilPesan('Terjadi kesalahan saat pencarian data')

def cari(self):
    try:
        search = self.editSearch.displayText()
        con = mc.connect(
            host='localhost',
            user='root',
            password='',
            database='pboperpus',
            port='3306')

        query = f'''

```

```

        SELECT
            tbl_buku.isbn,
            tbl_buku.judul,
            tbl_genre.nama_genre,
            tbl_penulis.nama_depan,
            tbl_penulis.nama_tengah,
            tbl_penulis.nama_belakang,
            tbl_buku.penerbit,
            tbl_buku.tanggal_publicasi,
            tbl_buku.jumlah_stok
        FROM
            ((tbl_buku
                INNER JOIN tbl_penulis ON tbl_buku.id_penulis =
tbl_penulis.id_penulis)
                INNER JOIN tbl_genre ON tbl_buku.id_genre =
tbl_genre.id_genre)
        WHERE
            tbl_buku.judul LIKE "%{search}%"
            OR tbl_buku.isbn LIKE "%{search}%"
        ...

    cursor = con.cursor()
    cursor.execute(query)
    data = cursor.fetchall()

    con.close()

    panjang = len(data)
    self.tblBuku.setRowCount(panjang)
    baris = 0

    for i in data:
        nama = i[3]+' '+i[4]+' '+i[5]

        self.tblBuku.setItem(baris, 0, QTableWidgetItem(i[0]))
        self.tblBuku.setItem(baris, 1, QTableWidgetItem(i[1]))
        self.tblBuku.setItem(baris, 2, QTableWidgetItem(i[2]))
        self.tblBuku.setItem(baris, 3, QTableWidgetItem(nama))
        self.tblBuku.setItem(baris, 4, QTableWidgetItem(i[6]))
        self.tblBuku.setItem(baris, 5,
QTableWidgetItem(i[7].strftime('%d-%m-%Y')))
        self.tblBuku.setItem(baris, 6, QTableWidgetItem(str(i[8])))

        baris += 1

    except:
        self.tampilPesan('Terjadi kesalahan saat pencarian data')

def dataPilih(self):

```

```

items = self.tblBuku.selectedItems()
self.isbn = items[0].text()
self.title = items[1].text()

self.close()

```

Penjelasan:

Class ini digunakan untuk menampilkan dialog daftar buku yang nantinya digunakan untuk memilih buku yang ingin dipinjam atau dikembalikan, class ini dihubungkan dengan transaction.py

File notBackYet.py

```

class BukuBelumKembali(QDialog):

    def __init__(self):
        super().__init__()
        loadUi('dialog/dialog_notYetBack.ui', self)

        self.setWindowTitle('Book Data That Has Not Been Returned')
        self.setWindowIcon(QIcon('ICON/library_logo.png'))

        self.tampilData()

    def tampilPesan(self, pesan):
        msgbox = QMessageBox()
        msgbox.setWindowTitle('Aplikasi Perpustakaan')
        msgbox.setIconPixmap(QIcon('ICON/info.png').pixmap(40, 40))
        msgbox.setWindowIcon(QIcon('ICON/library_logo.png'))
        msgbox.setText(pesan)
        msgbox.exec()

    def tampilData(self):
        try:
            con = mc.connect(
                host='localhost',
                user='root',
                password='',
                database='pboperpus',
                port='3306')

            query = f'''
                SELECT
                    tbl_peminjaman.id_peminjaman,
                    tbl_buku.judul,
                    tbl_anggota.nama_depan,
                    tbl_anggota.nama_tengah,
                    tbl_anggota.nama_belakang,
                    tbl_peminjaman.tanggal_pinjam,
                    tbl_peminjaman.tanggal_tempo,

```

```

        tbl_peminjaman.tanggal_kembali
    FROM
        ((tbl_peminjaman
            INNER JOIN tbl_buku ON
tbl_peminjaman.isbn=tbl_buku.isbn)
            INNER JOIN tbl_anggota ON
tbl_peminjaman.id_anggota=tbl_anggota.id_anggota)
        WHERE
            tbl_peminjaman.tanggal_kembali IS NULL
    '''

    cursor = con.cursor()
    cursor.execute(query)
    data = cursor.fetchall()

    con.close()

    panjang = len(data)
    self.tblBookNotYetBack.setRowCount(panjang)
    baris = 0

    for i in data:
        nama = i[2]+' '+i[3]+' '+i[4]

        self.tblBookNotYetBack.setItem(baris, 0,
    QTableWidgetItem(str(i[0])))
        self.tblBookNotYetBack.setItem(baris, 1,
    QTableWidgetItem(i[1]))
        self.tblBookNotYetBack.setItem(baris, 2,
    QTableWidgetItem(nama))
        self.tblBookNotYetBack.setItem(baris, 3,
    QTableWidgetItem(i[5].strftime('%d-%m-%Y')))
        self.tblBookNotYetBack.setItem(baris, 4,
    QTableWidgetItem(i[6].strftime('%d-%m-%Y')))
        baris += 1

    except:
        self.tampilPesan('Terjadi kesalahan saat menampilkan data')

```

Penjelasan:

Class ini untuk menampilkan dialog daftar buku yang belum dikembalikan

File history.py

```

class Riwayat(QDialog):

    def __init__(self):
        super().__init__()
        loadUi('dialog/dialog_history.ui', self)

        self.setWindowTitle('History Data')

```

```

self.setWindowIcon(QIcon('ICON/library_logo.png'))

self.tampilData()

def tampilPesan(self, pesan):
    msgbox = QMessageBox()
    msgbox.setWindowTitle('Aplikasi Perpustakaan')
    msgbox.setIconPixmap(QIcon('ICON/info.png').pixmap(40, 40))
    msgbox.setWindowIcon(QIcon('ICON/library_logo.png'))
    msgbox.setText(pesan)
    msgbox.exec()

def tampilData(self):
    try:
        con = mc.connect(
            host='localhost',
            user='root',
            password='',
            database='pboperpus',
            port='3306')

        query = '''
            SELECT
                tbl_peminjaman.id_peminjaman,
                tbl_buku.isbn, tbl_buku.judul,
                tbl_anggota.nama_depan,
                tbl_anggota.nama_tengah,
                tbl_anggota.nama_belakang,
                tbl_peminjaman.tanggal_pinjam,
                tbl_peminjaman.tanggal_tempo,
                tbl_peminjaman.tanggal_kembali,
                CONCAT('Rp ',
                    FORMAT(
                        CASE
                            WHEN tbl_peminjaman.tanggal_kembali IS NOT NULL
AND tbl_peminjaman.tanggal_kembali > tbl_peminjaman.tanggal_tempo THEN
                                -- Menghitung denda hanya untuk hari-hari
kerja
                                    DATEDIFF(tbl_peminjaman.tanggal_kembali,
tbl_peminjaman.tanggal_tempo) * 2000
                                -
                                -- Mengurangkan denda untuk hari Minggu
                                    (DATEDIFF(tbl_peminjaman.tanggal_kembali,
tbl_peminjaman.tanggal_tempo) DIV 7) * 2000
                                -
                                -- Mengurangkan denda untuk hari-hari libur
                                    (SELECT COUNT(*) FROM tbl_tgl_merah

```

```

        WHERE tbl_tgl_merah.tgl_merah BETWEEN
tbl_peminjaman.tanggal_tempo AND tbl_peminjaman.tanggal_kembali) * 2000
        ELSE
            0
        END,
    0)
    ) AS denda
FROM
    ((tbl_peminjaman
        INNER JOIN tbl_buku ON
tbl_peminjaman.isbn=tbl_buku.isbn)
        INNER JOIN tbl_anggota ON
tbl_peminjaman.id_anggota=tbl_anggota.id_anggota)
    WHERE
        tbl_peminjaman.tanggal_kembali IS NOT NULL
    ORDER BY
        tbl_peminjaman.tanggal_kembali;
'''

cursor = con.cursor()
cursor.execute(query)
data = cursor.fetchall()

con.close()

panjang = len(data)
self.tblHistory.setRowCount(panjang)
baris = 0

for i in data:
    nama = i[3]+' '+i[4]+' '+i[5]

    self.tblHistory.setItem(baris, 0, QTableWidgetItem(str(i[0])))
    self.tblHistory.setItem(baris, 1, QTableWidgetItem(i[1]))
    self.tblHistory.setItem(baris, 2, QTableWidgetItem(i[2]))
    self.tblHistory.setItem(baris, 3, QTableWidgetItem(nama))
    self.tblHistory.setItem(baris, 4,
QTableWidgetItem(i[6].strftime('%d-%m-%Y')))
    self.tblHistory.setItem(baris, 5,
QTableWidgetItem(i[7].strftime('%d-%m-%Y')))
    self.tblHistory.setItem(baris, 6,
QTableWidgetItem(i[8].strftime('%d-%m-%Y')))
    self.tblHistory.setItem(baris, 7, QTableWidgetItem(str(i[9])))

    baris += 1

except:
    self.tampilPesan('Terjadi kesalahan saat menampilkan data')

```

Penjelasan:

Class ini untuk menampilkan dialog daftar Riwayat buku yang sudah dikembalikan beserta

dendanya jika ada.