LAPORAN AKHIR PROYEK

SISTEM BASIS DATA

Sistem Perpustakaan Digital

**KELOMPOK-07**

**Anggota** :

|  |  |
| --- | --- |
| **Nama** | **NIM** |
| Winda V. Sembiring | 11323013 |
| Ricky J. Silaen | 11323028 |
| Vinci G. Baringbing | 11323051 |
| Febyanti Hutahaean | 11323055 |

**Prodi: D-III Teknologi Informasi 2023**

Fakultas Vokasi

Intitute Teknologi Del

2024/2025

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Proyek SBD 2024**  **Institut Teknologi Del** | | | |  |
|  | | |  | *Tanggal : 11-12-2024* | *Jumlah Halaman :* | |
|  | | |  |  |  | |

# DAFTAR ISI

[DAFTAR ISI 2](#_Toc184831009)

[1 PENDAHULUAN 4](#_Toc184831010)

[1.1 Latar Belakang 4](#_Toc184831011)

[1.2 Rumusan Masalah 4](#_Toc184831012)

[1.3 Tujuan 4](#_Toc184831013)

[1.4 Manfaat 4](#_Toc184831014)

[2 URAIAN PROYEK 5](#_Toc184831015)

[2.1 Tempat dan Waktu Pelaksanaan 5](#_Toc184831016)

[2.1.1 Tempat 5](#_Toc184831017)

[2.1.2 Waktu Pelaksanaan: 5](#_Toc184831018)

[2.2 Stack Teknologi 6](#_Toc184831019)

[2.3 Design ER-Diagram 6](#_Toc184831020)

[3 IMPLEMENTASI 7](#_Toc184831021)

[3.1 Authorization 7](#_Toc184831022)

[3.1.1 Role Admin 7](#_Toc184831023)

[3.1.2 Role Anggota 7](#_Toc184831024)

[3.2 Create Tabel User 8](#_Toc184831025)

[3.2.1 Admin 8](#_Toc184831026)

[3.2.2 Anggota 10](#_Toc184831027)

[3.3 Create Table 12](#_Toc184831028)

[3.3.1 Tabel Kategori 12](#_Toc184831029)

[3.3.2 Tabel Buku 13](#_Toc184831030)

[3.3.3 Tabel log\_peminjaman 13](#_Toc184831031)

[3.3.4 Tabel log\_denda 14](#_Toc184831032)

[3.4 Insert Data Menggunakan Crawling 15](#_Toc184831033)

[3.4.1 Import Library 15](#_Toc184831034)

[3.4.2 Koneksi ke database PostgreSQL 15](#_Toc184831035)

[3.4.3 Mengambil data buku dari Google Books API 16](#_Toc184831036)

[3.4.4 Insert Kategori Baru 16](#_Toc184831037)

[3.4.5 Insert data buku 17](#_Toc184831038)

[3.4.6 Eksport data ke file csv 17](#_Toc184831039)

[3.4.7 Mengambil data buku untuk setiap kategori 17](#_Toc184831040)

[3.4.8 Export data kategori dan buku ke file .csv 18](#_Toc184831041)

[3.4.9 Tutup Koneksi 18](#_Toc184831042)

[3.5 Create Trigger Insert Data 18](#_Toc184831043)

[3.5.1 Tabel kategori 18](#_Toc184831044)

[3.5.2 Tabel Anggota 20](#_Toc184831045)

[3.5.3 Tabel Buku 23](#_Toc184831046)

[3.6 Create Proses Peminjaman Buku menggunakan Trigger dan Fungsi 26](#_Toc184831047)

[3.6.1 Create Function 26](#_Toc184831048)

[3.6.2 Create Trigger 27](#_Toc184831049)

[3.6.3 Pengujian 27](#_Toc184831050)

[3.7 Create Proses Pengembalian Buku menggunakan Transaksi Manual 29](#_Toc184831051)

[3.7.1 Create Transaction 29](#_Toc184831052)

[3.7.2 Pengujian 30](#_Toc184831053)

[3.8 Proses Pembayaran Denda menggunakan Transaksi Manual 31](#_Toc184831054)

[3.8.1 Create transaction 31](#_Toc184831055)

[3.8.2 Pengujian 32](#_Toc184831056)

[3.9 Create Tabel Laporan 33](#_Toc184831057)

[3.9.1 Tabel laporan jumlah denda peminjaman setiap anggota 33](#_Toc184831058)

[3.9.2 Tabel laporan buku paling sering dipinjam per-tahun 34](#_Toc184831059)

[3.9.3 Tabel laporan buku paling sering dipinjam per-bulan 35](#_Toc184831060)

[3.9.4 Tabel laporan buku paling sering dipinjam per-minggu 36](#_Toc184831061)

[3.10 Stored Procedure untuk Laporan buku paling sering dibaca 36](#_Toc184831062)

[3.10.1 Create Stored Procedure 37](#_Toc184831063)

[3.10.2 Memanggil laporan buku terpopuler 37](#_Toc184831064)

[3.11 Cursor untuk Laporan buku paling sering dibaca 38](#_Toc184831065)

[3.11.1 Create cursor 38](#_Toc184831066)

[3.11.2 Memanggil laporan buku terpopuler 38](#_Toc184831067)

[3.12 Backup dan Restore 39](#_Toc184831068)

[3.12.1 Backup database 39](#_Toc184831069)

[3.12.2 Restore database 39](#_Toc184831070)

[3.13 Fungsi Mencari Buku 40](#_Toc184831071)

[3.13.1 Create Function 40](#_Toc184831072)

[3.13.2 Pengujian 41](#_Toc184831073)

[4 Kesimpulan dan Saran 42](#_Toc184831074)

[4.1 Kesimpulan 42](#_Toc184831075)

[4.2 Saran 42](#_Toc184831076)

[HASIL PRESENTASI 43](#_Toc184831077)

# PENDAHULUAN

## Latar Belakang

Buku dan informasi telah menjadi bagian penting dalam kehidupan manusia, baik sebagai sarana pembelajaran, hiburan, maupun pengembangan wawasan. Di era digital ini, perpustakaan digital menjadi solusi untuk memberikan akses mudah dan cepat ke berbagai koleksi secara online.

Sistem perpustakaan digital sederhana yang menggunakan sintaks SQL dirancang untuk memenuhi kebutuhan dasar pengelolaan koleksi, seperti pencatatan, dan peminjaman buku secara efisien. Dengan pendekatan ini, pengguna akan dapat mengakses dan mengelola data dengan lebih mudah melalui antarmuka sederhana yang berfokus pada fungsionalitas esensial. Sistem ini menjadi langkah awal yang penting dalam mewujudkan perpustakaan digital yang lebih terintegrasi dan canggih di masa depan.

## Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan, rumusan masalah yang diangkat dalam pengembangan sistem perpustakaan digital adalah sebagai berikut:

1. Bagaimana merancang sistem perpustakaan digital yang dapat mencatat log peminjaman buku secara efisien ?
2. Bagaimana sistem dapat menghitung dan mencatat log denda secara otomatis berdasarkan tanggal pengembalian yang terlambat?
3. Bagaimana sistem dapat menyimpan dan menampilkan laporan buku pengguna secara terstruktur dan mudah diakses?

## Tujuan

Tujuan utama dari pengembangan sistem perpustakaan digital adalah sebagai berikut:

1. Membangun fitur pencatatan log peminjaman yang dapat mencatat setiap transaksi peminjaman buku dengan akurat dan efisien.
2. Mengimplementasikan fitur untuk menghitung denda otomatis dan mencatat log denda berdasarkan peraturan yang telah ditentukan (misalnya, denda per hari keterlambatan).
3. Menyediakan laporan yang mudah diakses oleh petugas perpustakaan, baik laporan harian, mingguan, maupun tahunan, guna memudahkan pengelolaan dan pemantauan.

## Manfaat

Pengembangan sistem perpustakaan digital ini memberikan berbagai manfaat, baik bagi pengguna maupun pengelola perpustakaan, antara lain:

1. Memudahkan pemantauan dan analisis tren peminjaman melalui laporan otomatis harian, mingguan, atau tahunan.
2. Meningkatkan efisiensi dan akurasi pencatatan peminjaman, pengembalian, serta denda secara otomatis.

# URAIAN PROYEK

## Tempat dan Waktu Pelaksanaan

Proyek pengembangan Sistem Perpustakaan Digital ini dilaksanakan di **Institut Teknologi Del** selama periode perkuliahan, dengan rincian sebagai berikut:

### Tempat

* + Ruang kelas Gedung 5 dan lingkungan kampus Institut Teknologi Del.
  + Pengembangan dilakukan menggunakan fasilitas dan perangkat pribadi .

### Waktu Pelaksanaan:

Peiode waktu pelaksanaannya dimulai dari Minggu ke-14 perkuliahan (28 November 2024) hingga minggu ke-16 ( 10 Desember 2024) perkuliahan. Untuk melihat lebih jelas tahapan pelaksanaannya silahkan lihat **Tabel 1. Log Aktivitas** berikut.

**Tabel 1. Log Aktivitas**

|  |  |  |
| --- | --- | --- |
| **Minggu** | **Sesi** | **Aktivitas** |
| 14 | 1 | * + - 1. Design ER-Diagram |
| 2 | 1. Design ER-Diagram II 2. Create Tabel User |
| 3 | 1. Create Tabel Kategori, Buku, Log Peminjaman 2. Create Tabel log\_denda 3. Insert Dummy Data |
| 15 | 1 | 1. Create Authorization 2. Memberi Hak Akses 3. Create Trigger Insert Data |
| 2 | 1. Create Trigger Insert Data 2. Create Trigger dan Fungsi Proses Peminjaman 3. Create Transaksi Manual Proses Pengembalian |
| 3 | 1. Create Transaksi Manual Pembayaran Denda 2. Create View anggota\_peminjaman\_denda 3. Create View Laporan Buku Pertahun 4. Create View Laporan Buku Perbulan |
| 16 | 1 | 1. Create Stored Procedure Laporan Bulu Paling Sering dibaca 2. Create Cursor Laporan Buku paling sering dibaca |
| 2 | 1. Menambahkan data pada tabel buku dengan crawling 2. Menyusun Laporan Akhir. |
| 3 | 1. Lanjutan membuat crawling 2. Lanjutan Menyusun Laporan dan finalisasi laporan 3. Membuat slide presentasi 4. Backup dan Restore 5. Create View Cari Buku |

## Stack Teknologi

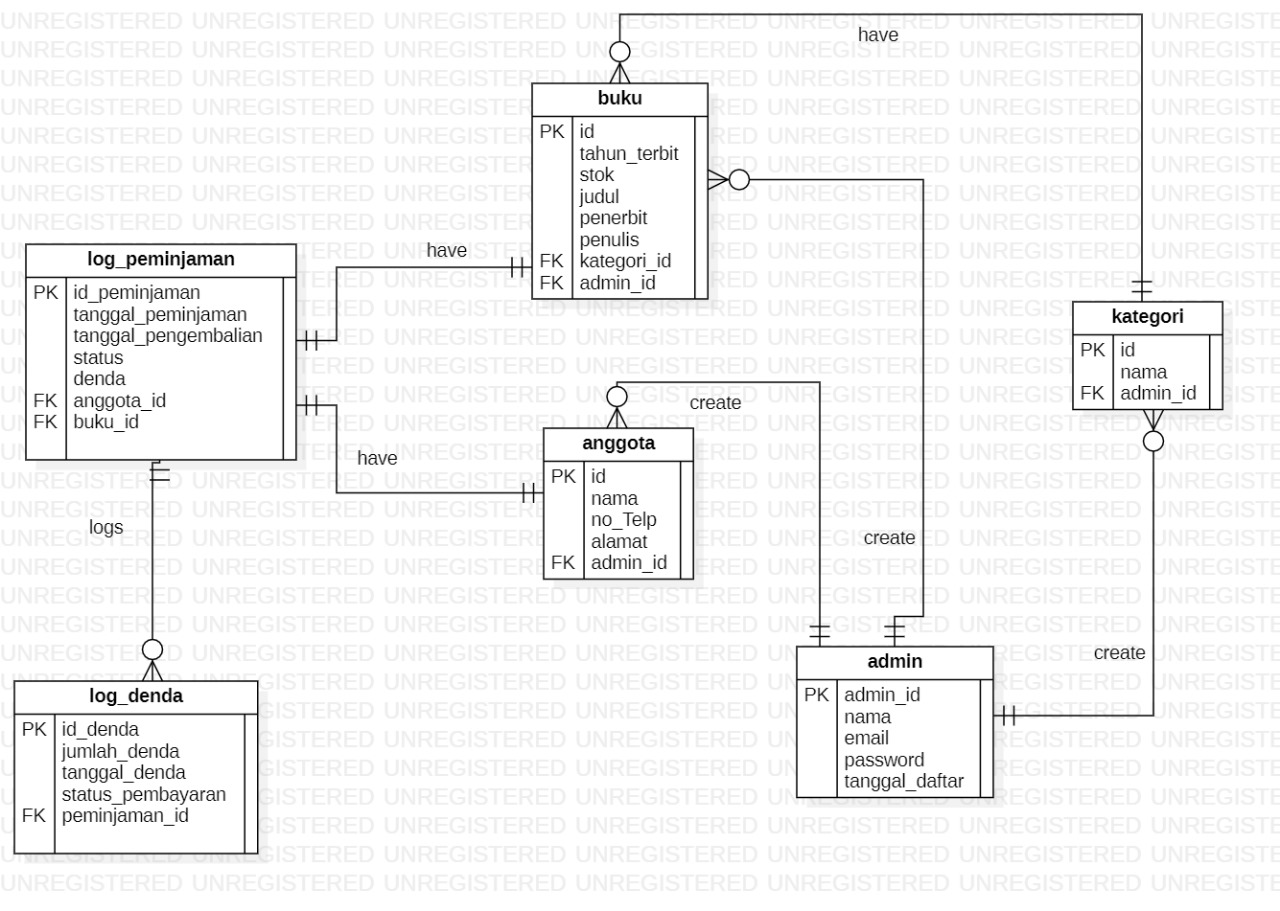
Berikut adalah detail teknologi, versi yang digunakan, jenis teknologi, serta kegunaannya dalam proyek pengembangan Sistem Perpustakaan Digital. Lihat **Tabel 2. Stack Teknologi** berikut.

**Tabel 2. Stack Teknologi**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nama Teknologi** | **Versi** | **Jenis** | **Kegunaan** |
| 1 | PostgreSQL | 16 | Database | Sebagai tempat untuk menyimpan data sistem. |
| 2 | Canva | N/A | Desain | Membuat desain presentasi atau poster. |
| 3 | StarUML | 5 | UML Modelling | Membuat ER-Diagram dan model UML. |
| 4 | Trello | N/A | Manajemen Proyek | Mengatur dan mengelola proyek secara efisien. |
| 5 | GitHub | N/A | Version Control | Tempat kolaborasi dan manajemen kode sumber. |

## Design ER-Diagram

Sistem Perpustakaan Digital\_07 memiliki dua peran utama, yaitu admin dan anggota. Admin memiliki hak penuh untuk melakukan CRUD (Create, Read, Update, Delete) pada tabel anggota, buku, dan kategori. Sementara itu, anggota hanya dapat mencari buku, meminjam buku, mengembalikan buku, serta membayar denda apabila pengembalian buku melewati batas waktu yang telah ditentukan. Sistem ini dirancang untuk mendukung pengelolaan perpustakaan secara efisien dan terstruktur.



# IMPLEMENTASI

## Authorization

Pada authorization bertujuan untuk mengelola hak akses pengguna dalam system

### Role Admin

Role admin memberikan hak akses penuh bagi pengguna untuk mengelola serta mengakseses semua data yang ada dalam system ,yang dimana admin dapat melakukan CRUD (Create,Update,Delete)

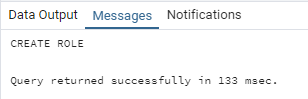
#### Create Role Admin

Langkah pertama untuk membuat role admin dengan cara membuat perintah CREATE ROLE ADMIN

* **Berikut merupakan syntaxnya :**



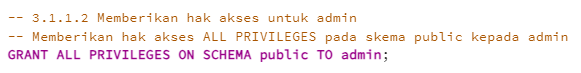
* **Output yang dihasilkan:**



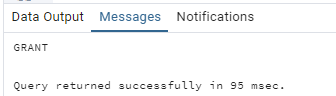
#### Memberikan hak akses untuk admin

Query ini memberikan hak akses penuh kepada role admin sehingga berfungsi untuk mengelola seluruh data seperti mengakses, mengubah, menambah, atau menghapus

* **Berikut merupakan syntaxnya :**



* **Output yang dihasilkan:**



### Role Anggota

Role anggota memiliki hak akses terbatas dalam system ini karena disini role anggota hanya berfungsi untuk melakukan log peminjaman untuk meminjam buku dan melakukan pembayaran denda apabila terjadi keterlambatan saat pengembalian buku .

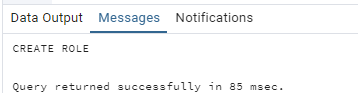
#### Create Role Anggota

Untuk membuat Role anggota dapat dilakukan dengan membuat perintah “Create Role Anggota”

* **Berikut merupakan syntaxnya:**



* **Output yang dihasilkan:**



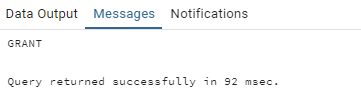
#### Memberikan hak akses untuk anggota

Query ini memberikan hak akses kepada role anggota untuk melihat, melakukan peminjaman, membayar denda tabel buku, log\_peminjaman, dan log\_denda. Dengan hak akses ini, anggota dapat mengakses daftar buku yang tersedia, riwayat peminjaman yang telah dilakukan, serta informasi terkait denda yang dikenakan akibat keterlambatan pengembalian buku. Namun, anggota hanya memiliki akses untuk membaca data tanpa kemampuan untuk mengubah, menambah data

* **Berikut merupakan syntaxnya:**



* **Output yang dihasilkan:**

****

## Create Tabel User

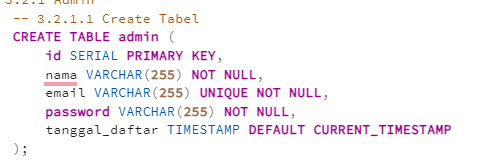
Tabel ini berfungsi untuk memisahkan data pengguna sesuai dengan peran serta hak akses yang dimiliki masing-masing.

### Admin

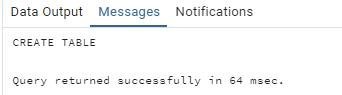
#### Create Tabel

Query ini berfungsi untuk membuat tabel yang menyimpan data admin dalam system.

* **Berikut merupakan syntaxnya:**



* **Output yang dihasilkan:**



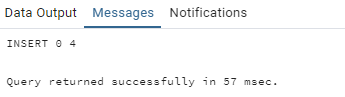
#### Insert Data

Berikut merupakan data yang akan ditambahkan pada table admin . Data tersebut mencakup beberapa informasi penting, seperti nama admin, email, dan password, yang dimasukkan untuk mengelola sistem

* **Berikut merupakan syntaxnya:**



* **Output yang dihasilkan:**

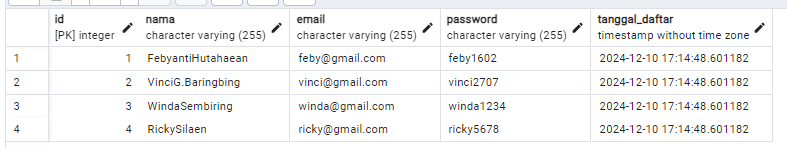
****

Setelah proses penambahan data selesai, dilakukan pemeriksaan untuk memastikan bahwa data telah berhasil disimpan ke dalam tabel.

* **Berikut merupakan syntaxnya :**



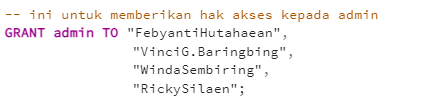
* **Output yang dihasilkan:**



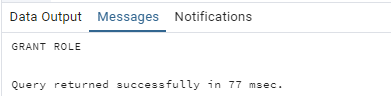
#### Menambahkan admin yang baru diinsert ke dalam role admin

Query ini bertujuan untuk menambahkan admin yang telah diinsert sebelumnya ke dalam role **admin**, sehingga memberikan mereka hak akses penuh untuk mengelola sistem

* **Berikut merupakan syntaxnya:**



* **Output yang dihasilkan:**

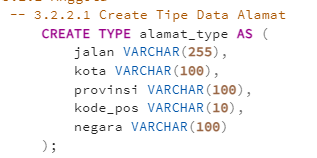


### Anggota

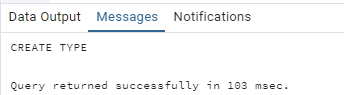
#### Create Tipe Data Alamat

Query ini bertujuan untuk menyimpan informasi alamat dengan data yang terstruktur .

* **Berikut merupakan syntaxnya:**



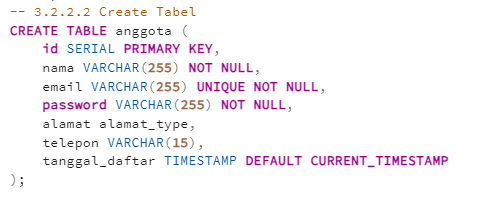
* **Output yang dihasilkan:**



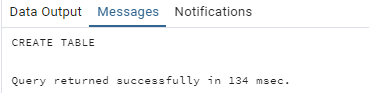
#### Create Tabel

Query ini berfungsi untuk membuat tabel yang menyimpan anggota dalam system.

* **Berikut merupakan syntaxnya :**



* **Output yang dihasilkan:**



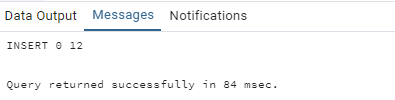
#### Insert Data

Berikut merupakan data yang akan ditambahkan pada table anggota . Data tersebut mencakup beberapa informasi penting, seperti nama admin, email, password, alamat ,telepon ,tanggal daftar yang dibuat untuk mencatat anngota.

* **Berikut merupakan syntaxnya :**



* **Output yang dihasilkan:**



Setelah proses penambahan data selesai, dilakukan pemeriksaan untuk memastikan bahwa data telah berhasil disimpan ke dalam tabel

* **Berikut merupakan syntaxnya :**



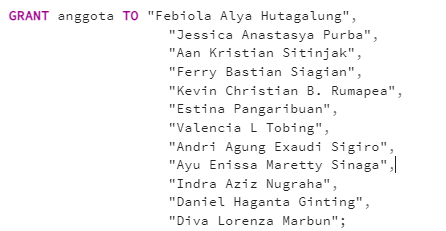
* **Output yang dihasilkan:**



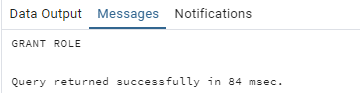
#### Menambahkan anggota yang baru diinsert ke dalam role anggota

Query ini bertujuan untuk memberikan hak akses kepada anggota yang baru dimasukkan ke dalam table anggota sehingga dengan menambahkan role tersebut anggota memiliki hak akses terbatas yang dimana hanya dapat melakukan log peminjaman untuk meminjam buku dan melakukan pembayaran denda apabila terjadi keterlambatan saat pengembalian buku.

* **Berikut merupakan syntaxnya :**



* **Output yang dihasilkan:**

****

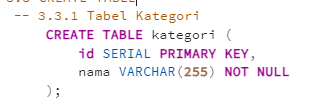
## Create Table

Query ini berfungsi untuk membuat tabel yang menyimpan kategori dalam system , khususnya untuk mengelompokkan buku-buku yang ada di dalam perpustakaan berdasarkan jenis

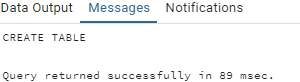
### Tabel Kategori

Pada table kategori fungsinya untuk menyimpan informasi mengenai kategori-kategori buku yang tersedia dalam sistem perpustakaan

* **Berikut merupakan syntaxnya :**



* **Output yang dihasilkan:**

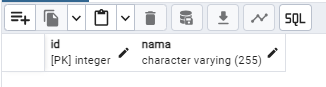


Setelah pembuatan table maka dilakukan pengecekan untuk memeriksa data yang ada dalam tabel

* **Berikut merupakan syntaxnya :**

****

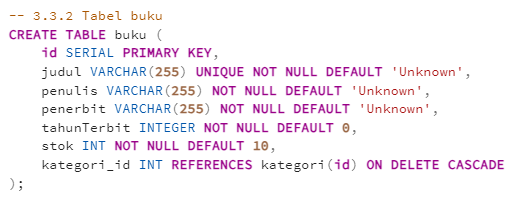
* **Output yang dihasilkan:**

****

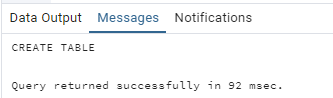
### Tabel Buku

Query ini digunakan untuk membuat table buku yang menyimpan informasi mengenai buku-buku dalam sistem perpustakaan, termasuk data seperti judul, penulis, penerbit, tahun terbit, stok, dan kategori.

* **Berikut merupakan syntaxnya :**



* **Output yang dihasilkan:**

****

Setelah pembuatan table maka dilakukan pengecekan untuk memeriksa data yang ada dalam tabel

* **Berikut merupakan syntaxnya :**

****

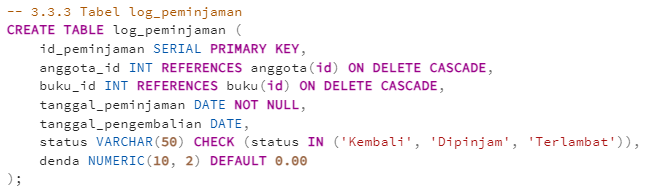
* **Output yang dihasilkan:**

****

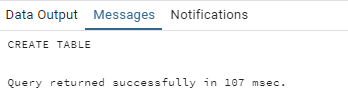
### Tabel log\_peminjaman

Query ini digunakan untuk melakukan penyimpanan informasi tentang transaksi peminjaman buku di perpustakaan termasuk informasi tentang anggota, buku yang dipinjam, tanggal peminjaman, tanggal pengembalian, status peminjaman, dan denda jika ada keterlambatan

* **Berikut merupakan syntaxnya :**



* **Output yang dihasilkan:**



Setelah pembuatan table maka dilakukan pengecekan untuk memeriksa data yang ada dalam tabel

* **Berikut merupakan syntaxnya :**

****

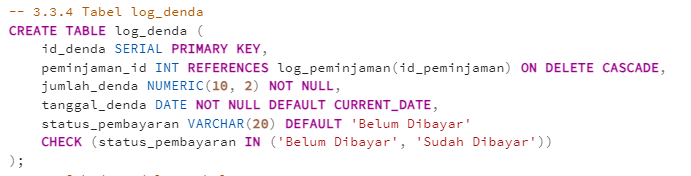
* **Output yang dihasilkan:**

****

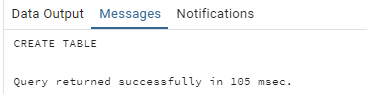
### Tabel log\_denda

Query ini digunakan untuk membuat table log\_denda yang menyimpan data denda terkait transaksi peminjaman buku, termasuk informasi mengenai peminjaman yang terkait, jumlah denda, tanggal denda, dan status pembayaran denda. Tabel ini menghubungkan data dengan table log\_peminjaman yang mengatur penghapusan data secara otomatis sehinnga dapat memeriksa status pembayaran apakah 'Belum Dibayar' atau 'Sudah Dibayar'

* **Berikut merupakan syntaxnya :**



* **Output yang dihasilkan:**

****

Setelah pembuatan table maka dilakukan pengecekan untuk memeriksa data yang ada dalam tabel

* **Berikut merupakan syntaxnya :**

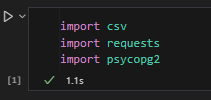
****

* **Output yang dihasilkan:**

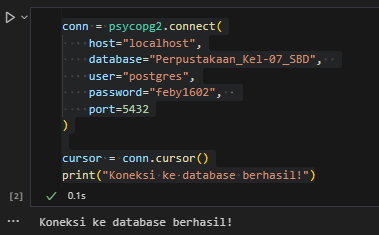
****

## Insert Data Menggunakan Crawling

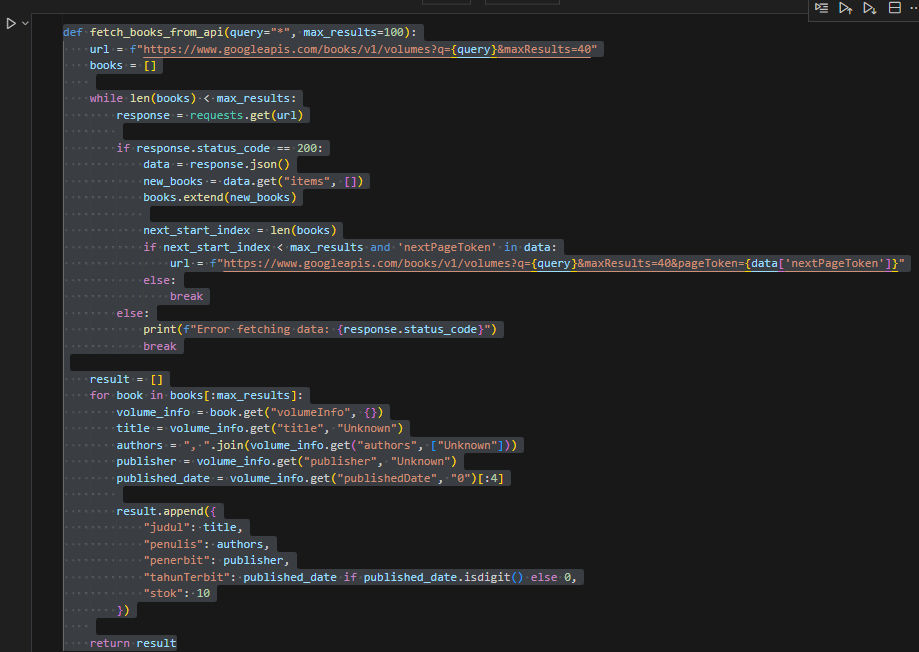
### Import Library



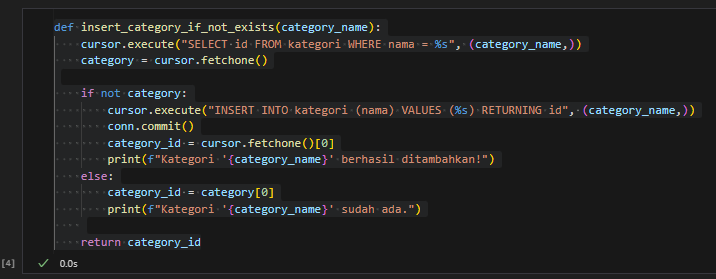
### Koneksi ke database PostgreSQL



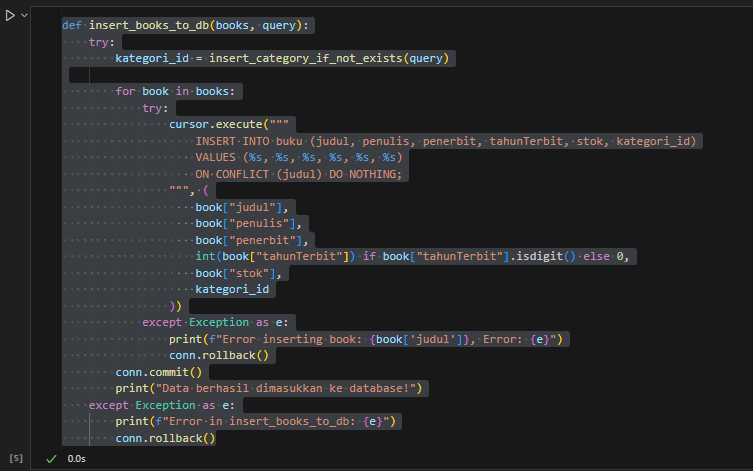
### Mengambil data buku dari Google Books API



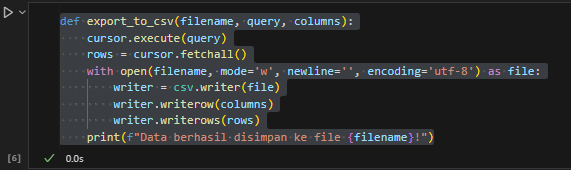
### Insert Kategori Baru



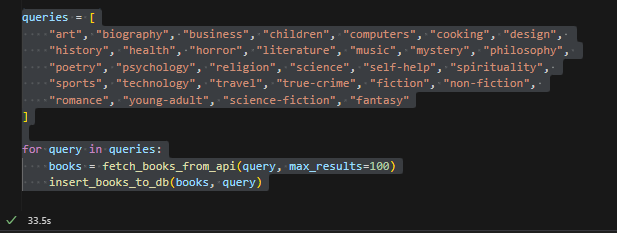
### Insert data buku



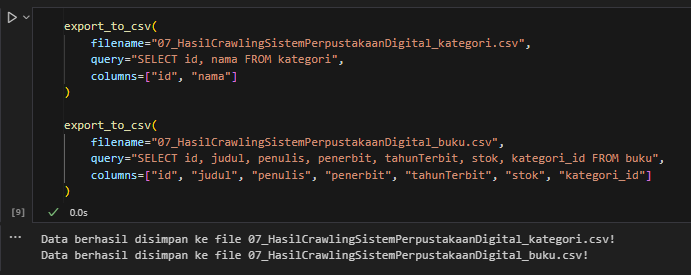
### Eksport data ke file csv



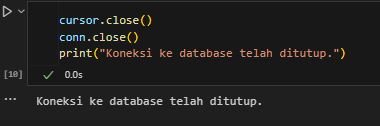
### Mengambil data buku untuk setiap kategori



### Export data kategori dan buku ke file .csv



### Tutup Koneksi



## Create Trigger Insert Data

Membuat trigger untuk memvalidasi data yang akan ditambahkan ke dalam table kategori

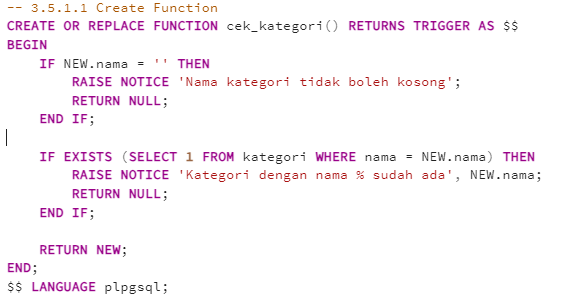
### Tabel kategori

Tabel kategori berfungsi untuk menyimpan informasi mengenai kategori buku yang tersedia di perpustakaan.sehingga memudahkan anngota mengakses buku berdasarkan jenis atau topiknya,

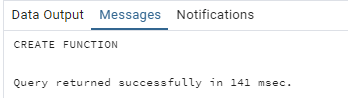
#### Create Function

Query ini berfungsi untuk memvalidasi data yang akan dimasukkan ke dalam table kategori yang dimana fungsi ini akan akan dipanggil oleh trigger sebelum data baru disisipkan ke dalam tabel.dan memeriksa apakah kolom nama pada data yang akan dimasukan (New=nama)kosongmaka sistem akan menampilkan pesan pemberitahuan "Nama kategori tidak boleh kosong" dan membatalkan proses insert dengan mengembalikan.Selanjutnya fungsi memeriksa apakah nama kategori yang sama sudah ada di dalam table kategori Jika ditemukan kategori dengan nama yang sama, fungsi akan menampilkan pesan "Kategori dengan nama [nama kategori] sudah ada" dan kembali membatalkan proses insert. Jika kedua kondisi di atas tidak ditemukan, fungsi akan mengembalikan yang berarti data kategori yang baru dapat disimpan ke dalam tabel

* **Berikut merupakan syntaxnya :**



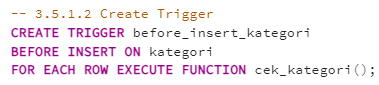
* **Output yang dihasilkan:**



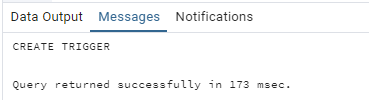
#### Create Trigger

Query ini digunakan untuk membuat trigger yang akan dijalankan sebelum data di insert ke dalam table kategori , trigger ini akan memanggil fungsi cek kategori untuk memeriksa apakah nama kategori yang akan dimasukkan valid atau tidak. Jika nama kategori kosong atau sudah ada dalam tabel, fungsi akan membatalkan proses insert dan memberikan pemberitahuan.

* **Berikut merupakan syntaxnya :**



* **Output yang dihasilkan:**

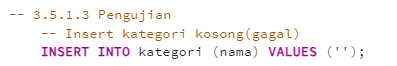


#### Pengujian

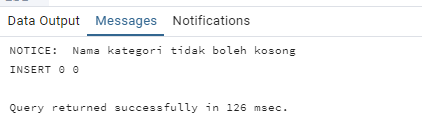
Pengujian I :

Query ini memasukkan data dengan nama kategori kosong ke dalam kategori sehinnga dengan adaanya fungsi trigger yang akan memeriksa apakah nama kategori kosong, maka query ini akan gagal. Sistem akan memberikan pemberitahuan "Nama kategori tidak boleh kosong" dan data tidak akan disisipkan ke dalam tabel

* **Berikut merupakan syntaxnya :**



* **Output yang dihasilkan:**



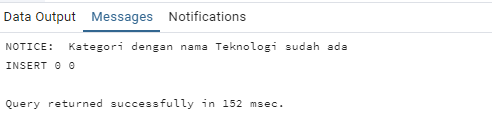
Pengujian II :

Query ini dapat memasukkan kategori dengan nama 'Teknologi', yang diasumsikan sudah ada dalam table kategori Fungsi trigger akan memeriksa apakah kategori dengan nama yang sama sudah ada. Jika ada, maka query ini akan gagal, dan sistem akan memberi pemberitahuan "Kategori dengan nama Teknologi sudah ada". Data tidak akan disisipkan

* **Berikut merupakan syntaxnya :**



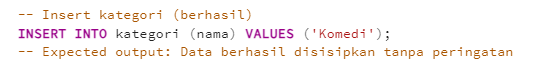
* **Output yang dihasilkan:**



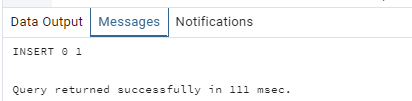
Pengujian III :

Query ini memasukkan kategori dengan nama 'Komedi'. Karena nama kategori ini belum ada sebelumnya dan tidak kosong, query ini akan berhasil. Data kategori baru akan disisipkan ke dalam tabel tanpa masalah

* **Berikut merupakan syntaxnya :**



* **Output yang dihasilkan:**



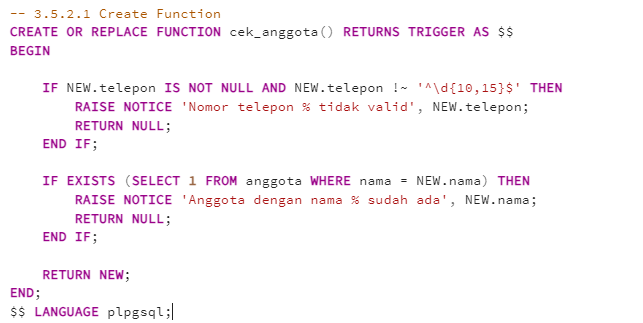
### Tabel Anggota

Pada anggota memiliki hak akses terbatas dalam system ini karena disini anggota hanya berfungsi untuk melakukan log peminjaman untuk meminjam buku dan melakukan pembayaran denda apabila terjadi keterlambatan saat pengembalian buku

#### Create Function

Query ini digunakan untuk memvalidasi data anggota sebelum dimasukkan ke dalam tabel. Fungsi ini pertama-tama memeriksa apakah nomor telepon yang dimasukkan valid, yaitu terdiri dari 10 hingga 15 digit angka. Jika nomor telepon tidak valid, fungsi akan membatalkan penyisipan dan memberikan pemberitahuan. Selain itu, fungsi juga memastikan bahwa tidak ada duplikasi nama anggota dengan memeriksa apakah nama yang sama sudah ada dalam tabel. Jika ada duplikasi, penyisipan juga dibatalkan

* **Berikut merupakan syntaxnya :**



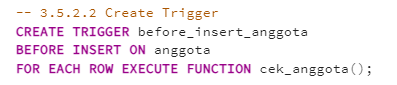
* **Output yang dihasilkan:**



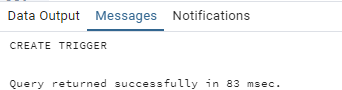
#### Create Trigger

Query ini dibuat untuk menjalankan fungsi cek\_anggota sebelum data dimasukkan ke table anggota . Fungsi ini akan memeriksa apakah nomor telepon valid dan apakah nama anggota sudah ada di dalam sistem. Jika ada masalah, seperti nomor telepon tidak valid atau nama sudah terdaftar, data tidak akan dimasukkan dan akan muncul pemberitahuan

* **Berikut merupakan syntaxnya :**



* **Output yang dihasilkan:**

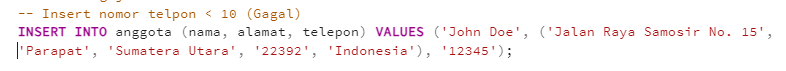


#### Pengujian

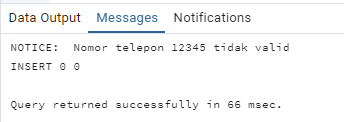
Pengujian I :

Pada pengujian ini dapat memasukkan data anggota dengan nomor telepon yang kurang dari 10 digit, yang tidak memenuhi format yang valid sesuai dengan aturan yang ditetapkan dalam fungsi Karena nomor telepon tersebut tidak valid, trigger akan memblokir proses insert dan menampilkan pesan pemberitahuan bahwa nomor telepon tidak valid. Dengan demikian, data tidak akan dimasukkan ke dalam tabel

* **Anggota Cek\_anngota .Berikut merupakan syntaxnya :**



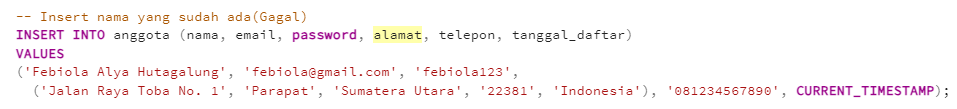
* **Output yang dihasilkan:**



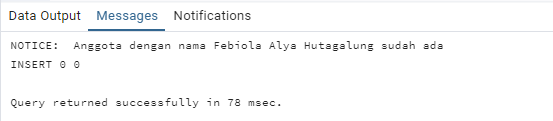
Pengujian II :

Pada pengujian ini, dilakukan upaya untuk menambahkan data anggota dengan nama **Febiola Alya Hutagalung** yang sudah ada dalam table anggota Fungsi cek\_anggota yang terkait dengan trigger akan memeriksa keberadaan nama tersebut di tabel sebelum proses insert dilakukan. Karena nama tersebut sudah ada, fungsi akan memblokir proses insert dan memunculkan pemberitahuan bahwa anggota dengan nama tersebut sudah ada. Dengan demikian, data baru tidak akan dimasukkan, dan tabel anggota tetap tidak mengalami perubahan.

* **Syntax :**



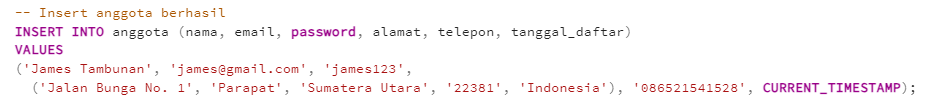
* **Output :**



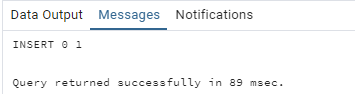
Pengujian III :

Pada pengujian ini, dilakukan penambahan data anggota baru dengan nama **James Tambunan**. Karena data yang dimasukkan valid—baik nama, email, password, alamat, nomor telepon, maupun tanggal pendaftaran—fungsi cek\_ anggota tidak menemukan pelanggaran aturan. Nama anggota tidak duplikat, dan nomor telepon memenuhi format yang ditentukan. Oleh karena itu, proses insert berhasil dilakukan, dan data anggota baru ditambahkan ke table anggota tanpa masalah.

* **Syntax :**



* **Output :**



Lakukan pengecekan data :

* **Syntax :**



* **Output :**



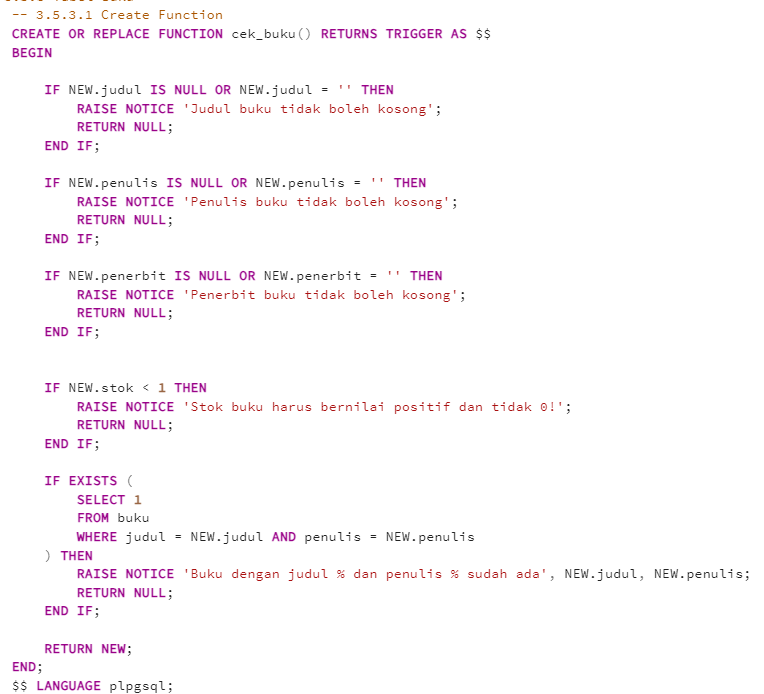
### Tabel Buku

Pada table ini digunakan untuk membuat table buku yang menyimpan informasi mengenai buku-buku dalam sistem perpustakaan, termasuk data seperti judul, penulis, penerbit, tahun terbit, stok, dan kategori.

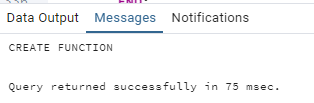
#### Create Function

Query ini digunakan untuk memvalidasi data sebelum disisipkan ke table buku. Fungsi ini memastikan bahwa data yang dimasukkan memenuhi kriteria tertentu. Pertama, judul, penulis, dan penerbit tidak boleh kosong atau bernilai NULL. Selanjutnya, stok buku harus bernilai positif dan tidak boleh 0, untuk memastikan buku yang tersedia memiliki jumlah yang valid. Fungsi ini juga memeriksa apakah ada buku dengan kombinasi judul dan penulis yang sama untuk mencegah duplikasi data. Jika semua kondisi terpenuhi, data baru akan diterima; jika tidak, fungsi akan memberikan notifikasi kesalahan dan membatalkan proses insert. Fungsi ini menjaga konsistensi dan integritas data dalam sistem perpustakaan.

* **Syntax :**



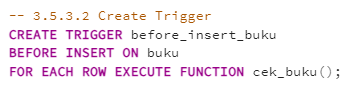
* **Output :**



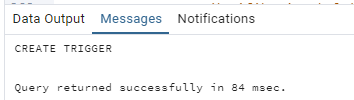
#### Create Trigger

Pada query ini digunakan untuk table buku agar validasi data dilakukan secara otomatis setiap kali ada upaya memasukkan data baru. Trigger ini memanfaatkan fungsi cek\_buku() untuk memastikan data memenuhi aturan yang telah ditentukan. Dengan deklarasi BEFORE INSERT, trigger dijalankan sebelum proses INSERT dilakukan. Setiap baris data baru akan diperiksa melalui fungsi tersebut, sehingga hanya data valid yang dapat dimasukkan ke tabel buk

* **Syntax :**



* **Output :**

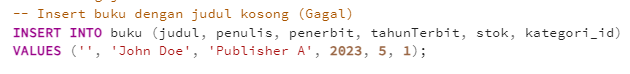


#### Pengujian

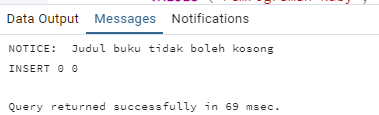
Pengujian I :

Pengujian ini dilakukan dengan mencoba menyisipkan data ke tabel buku menggunakan nilai judul kosong (''). Berdasarkan logika dalam fungsi **cek\_buku**, jika kolom judul tidak diisi atau bernilai kosong, maka sistem akan menampilkan pesan pemberitahuan **"Judul buku tidak boleh kosong"** dan proses penyisipan data akan dihentikan. Hal ini menunjukkan bahwa mekanisme validasi untuk mencegah data tidak valid berfungsi dengan baik, sehingga integritas data pada tabel buku tetap terjaga.

* **Syntax :**

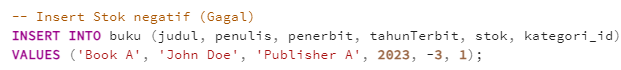


* **Output :**

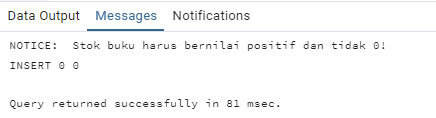


Pengujian II :

* **Syntax :**

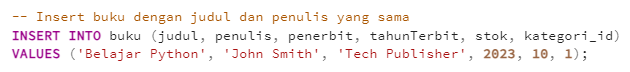


* **Output :**

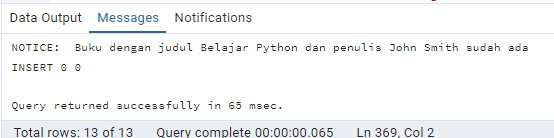


Pengujian III :

* **Syntax :**

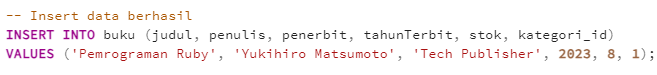


* **Output :**

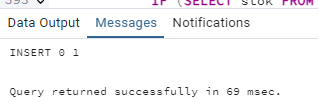


Pengujian IV :

* **Syntax :**



* **Output :**

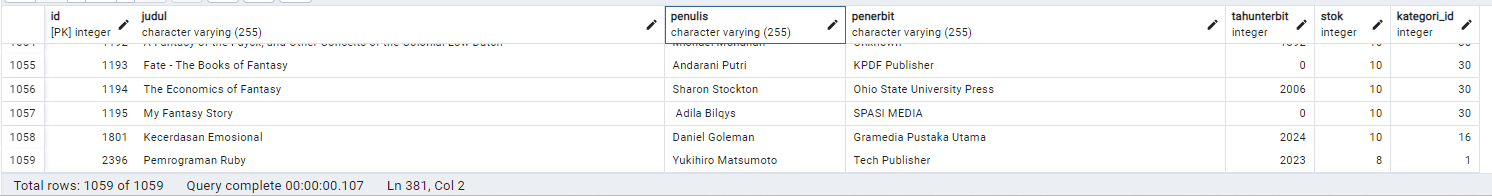


Lakukan pengecekan data :

* **Syntax :**



* **Output :**

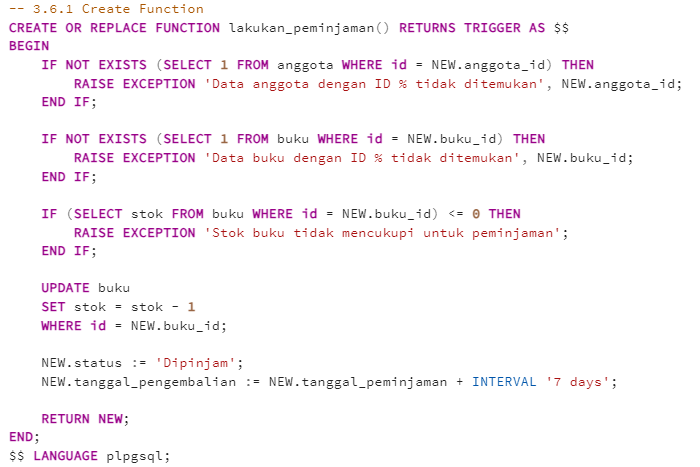


## Create Proses Peminjaman Buku menggunakan Trigger dan Fungsi

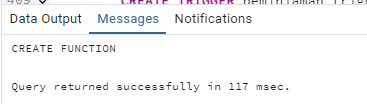
### Create Function

Proses peminjaman buku dalam sistem perpustakaan ini melibatkan penggunaan fungsi dan trigger untuk memastikan bahwa peminjaman hanya dilakukan jika semua kondisi valid. Fungsi lakukan\_peminjaman() pertama-tama memvalidasi keberadaan anggota dan buku yang akan dipinjam dengan memeriksa ID anggota dan ID buku di database. Jika ID anggota atau ID buku tidak ditemukan, maka fungsi akan menampilkan pesan kesalahan yang sesuai. Selain itu, fungsi ini juga memeriksa apakah stok buku mencukupi untuk peminjaman. Jika stok buku kurang dari atau sama dengan 0, peminjaman dibatalkan dan muncul pesan bahwa stok tidak mencukupi. Jika semua syarat tersebut terpenuhi, fungsi akan mengurangi stok buku sebanyak satu dan menetapkan status peminjaman buku serta tanggal pengembalian yang ditentukan tujuh hari setelah peminjaman. Trigger peminjaman\_trigger kemudian dipasang untuk memastikan fungsi ini dijalankan setiap kali ada upaya untuk memasukkan data peminjaman baru ke dalam tabel log\_peminjaman. Pengujian dilakukan dalam beberapa skenario, antara lain peminjaman dengan ID anggota yang tidak terdaftar, peminjaman dengan buku yang tidak tersedia, serta peminjaman dengan stok buku yang tidak mencukupi. Semua pengujian yang gagal menghasilkan pesan kesalahan yang sesuai, sementara peminjaman yang berhasil tercatat dengan benar dalam log\_peminjaman. Sistem ini memastikan bahwa proses peminjaman berjalan sesuai dengan aturan dan menjaga akurasi data di dalam sistem perpustakaan.

* **Syntax :**

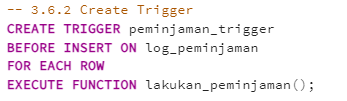


* **Output :**

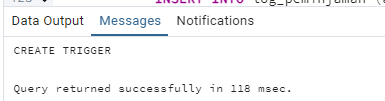


### Create Trigger

* **Syntax :**



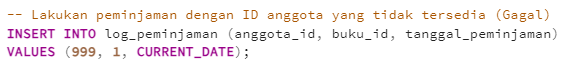
* **Output :**



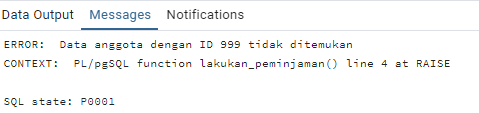
### Pengujian

Pengujian I :

* **Syntax :**

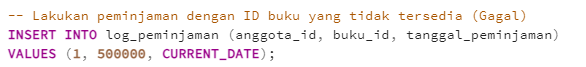


* **Output :**

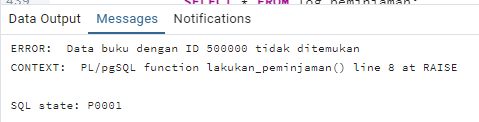


Pengujian II :

* **Syntax :**

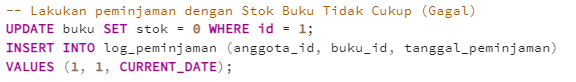


* **Output :**

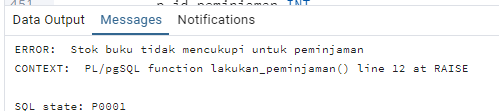


Pengujian III :

* **Syntax :**

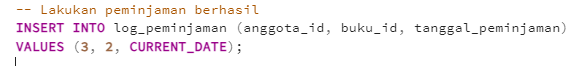


* **Output :**

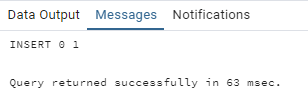


Pengujian IV :

* **Syntax :**



* **Output :**



Lakukan pengecekan data nya

* **Syntax :**



* **Output :**



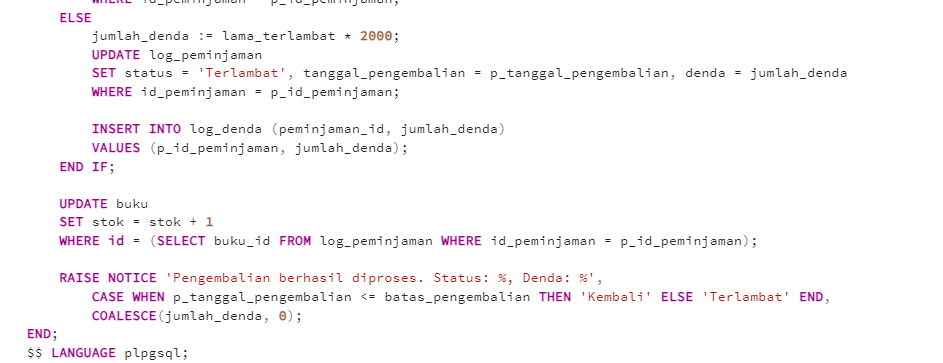
## Create Proses Pengembalian Buku menggunakan Transaksi Manual

### Create Transaction

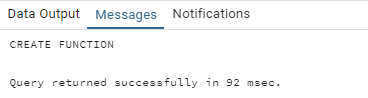
Fungsi proses\_pengembalian\_transaksi bertujuan untuk memproses pengembalian buku yang dipinjam oleh anggota, dengan mempertimbangkan apakah pengembalian dilakukan tepat waktu atau terlambat. Fungsi ini menerima dua parameter: p\_id\_peminjaman (ID peminjaman) dan p\_tanggal\_pengembalian (tanggal pengembalian buku). Pertama, fungsi memeriksa status peminjaman pada tabel log\_peminjaman. Jika peminjaman sudah dikembalikan atau terlambat, fungsi akan mengeluarkan pemberitahuan dan tidak melanjutkan proses pengembalian. Selanjutnya, fungsi menghitung batas waktu pengembalian (yaitu 7 hari setelah tanggal peminjaman) dan menghitung berapa lama keterlambatan, jika ada. Jika pengembalian dilakukan tepat waktu, status peminjaman diperbarui menjadi "Kembali" dan tanggal pengembalian diset. Jika pengembalian terlambat, status diperbarui menjadi "Terlambat" dan denda dihitung berdasarkan jumlah hari keterlambatan, kemudian dimasukkan ke dalam tabel log\_denda. Setelah itu, jumlah stok buku pada tabel buku akan ditambahkan satu, menandakan bahwa buku telah kembali. Fungsi ini diujikan dengan tiga kondisi: pengembalian lebih awal, tepat waktu, dan terlambat, untuk memeriksa apakah pengembalian diproses dengan benar.

* **Syntax :**





* **Output :**



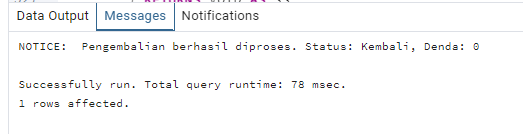
### Pengujian

Pengujian I :

* **Syntax :**



* **Output :**

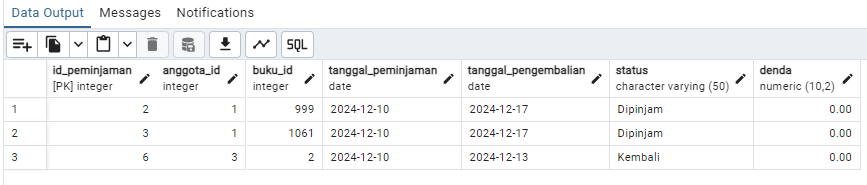


Pengujian II :

* **Syntax :**

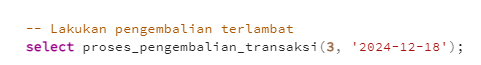


* **Output :**

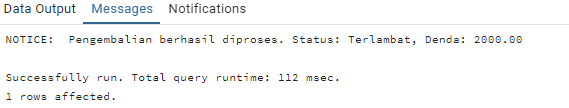


Pengujian III :

* **Syntax :**



* **Output :**

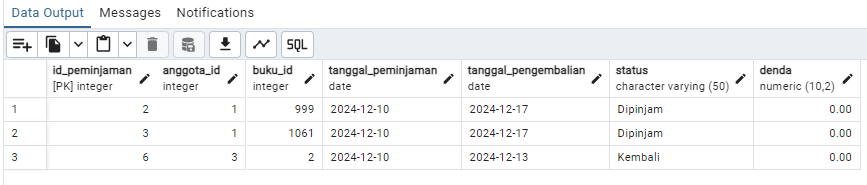


Lakukan pengecekan terhadap data :

* **Syntax :**



* **Output :**

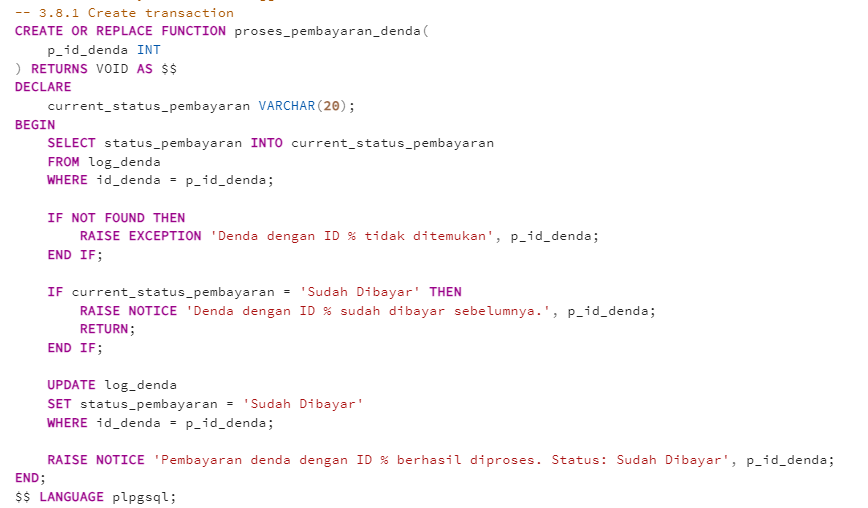


## Proses Pembayaran Denda menggunakan Transaksi Manual

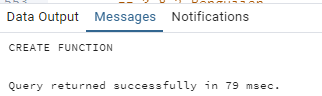
Fungsi proses\_pembayaran\_denda bertujuan untuk memproses pembayaran denda berdasarkan ID yang diberikan. Pertama, fungsi memeriksa apakah denda dengan ID tersebut ada di dalam tabel log\_denda. Jika tidak ditemukan, fungsi akan menampilkan pesan kesalahan. Jika denda sudah dibayar sebelumnya, maka fungsi akan memberikan pemberitahuan bahwa denda tersebut sudah dibayar, dan proses akan dihentikan. Jika denda belum dibayar, status pembayaran akan diperbarui menjadi "Sudah Dibayar", dan pemberitahuan akan ditampilkan bahwa pembayaran denda berhasil. Pengujian dilakukan untuk dua kondisi: pertama dengan ID denda yang tidak ada, yang menyebabkan kegagalan, dan kedua dengan ID denda yang valid, yang berhasil memperbarui status pembayaran.

### Create transaction

* **Syntax :**

****

* **Output :**



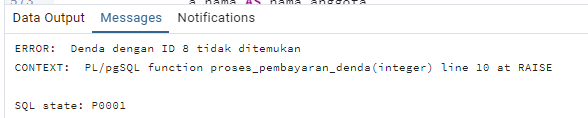
### Pengujian

Pengujian I :

* **Syntax :**

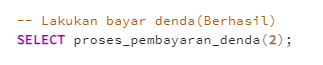


* **Output :**

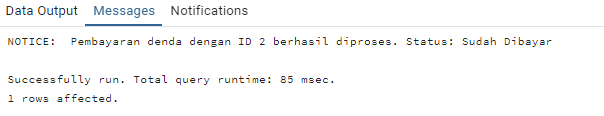


Pengujian II :

* **Syntax :**



* **output :**

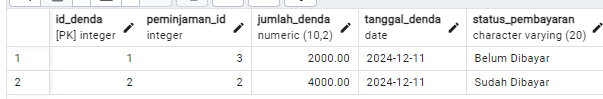


Lakukan pengecekan data :

* **Syntax :**



* **Output :**



## Create Tabel Laporan

### Tabel laporan jumlah denda peminjaman setiap anggota

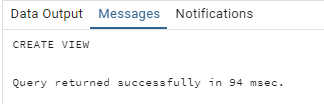
Fungsi anggota\_peminjaman\_denda adalah sebuah **view** yang menampilkan laporan peminjaman dan denda setiap anggota. View ini menggabungkan data dari beberapa tabel: anggota, log\_peminjaman, buku, dan log\_denda. Data yang ditampilkan mencakup ID anggota, nama anggota, judul buku, tanggal peminjaman, tanggal pengembalian, status peminjaman, jumlah denda (jika ada), dan status pembayaran denda. Jika denda tidak ada, maka jumlah denda akan bernilai 0 dan status pembayaran akan bernilai "Belum Dibayar".

#### Create View

* **Syntax :**



* **Output :**

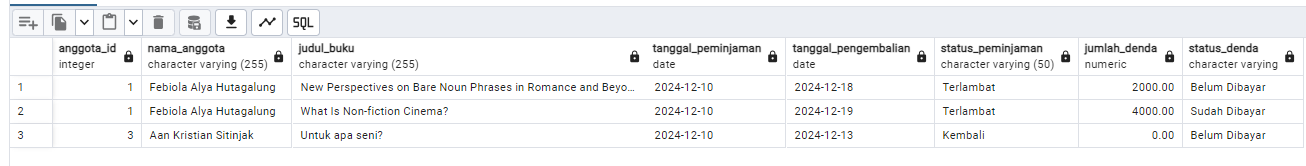


#### Menampilkan hasil tabel laporan buku

* **Syntax :**



* **Output :**

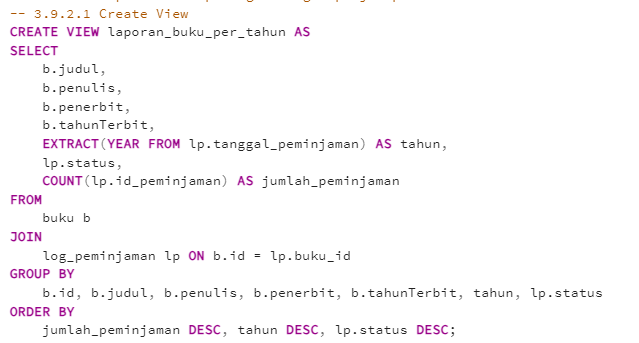


### Tabel laporan buku paling sering dipinjam per-tahun

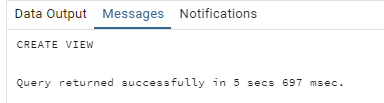
Query ini membuat view bernama laporan\_buku\_per\_tahun, yang menghasilkan laporan mengenai buku yang paling sering dipinjam setiap tahunnya. Data yang ditampilkan meliputi judul, penulis, penerbit, tahun terbit, status peminjaman, dan jumlah peminjaman untuk setiap buku. Query ini menggabungkan data dari tabel buku dan log\_peminjaman dengan melakukan perhitungan jumlah peminjaman per buku, dikelompokkan berdasarkan tahun peminjaman. Hasilnya diurutkan berdasarkan jumlah peminjaman terbanyak, tahun, dan status peminjaman, untuk memudahkan analisis buku yang paling populer tiap tahun.

#### Create View

* **Syntax :**



* **Output :**

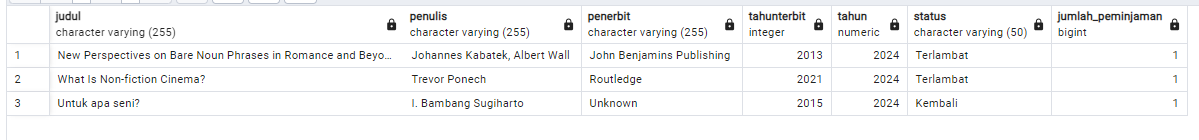


#### Menampilkan hasil tabel laporan

* **Syntax :**



* **Output :**

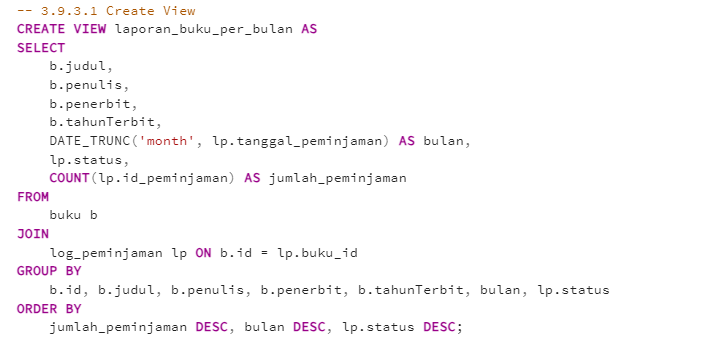


### Tabel laporan buku paling sering dipinjam per-bulan

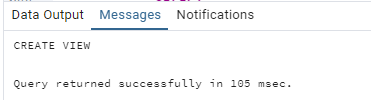
Querry ini ini membuat view bernama laporan\_buku\_per\_bulan, yang menghasilkan laporan mengenai buku yang paling sering dipinjam setiap bulan. Data yang ditampilkan meliputi judul, penulis, penerbit, tahun terbit, bulan peminjaman, status peminjaman, dan jumlah peminjaman per buku. Query ini menggabungkan data dari tabel buku dan log\_peminjaman dengan menghitung jumlah peminjaman setiap buku, dikelompokkan berdasarkan bulan peminjaman. Hasilnya diurutkan berdasarkan jumlah peminjaman terbanyak, bulan, dan status peminjaman untuk memudahkan analisis buku yang paling populer tiap bulan.

#### Create View

* **Syntax :**



* **Output :**

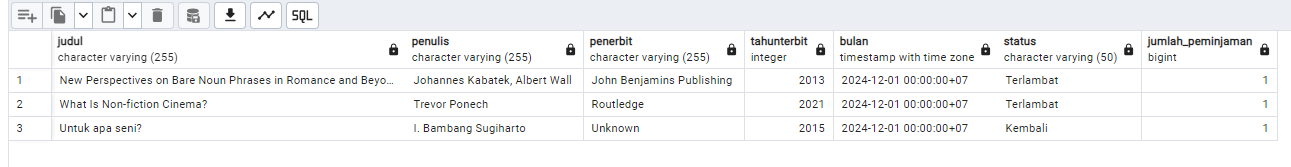


#### Menampilkan hasil tabel laporan

* **Syntax :**



* **Output :**

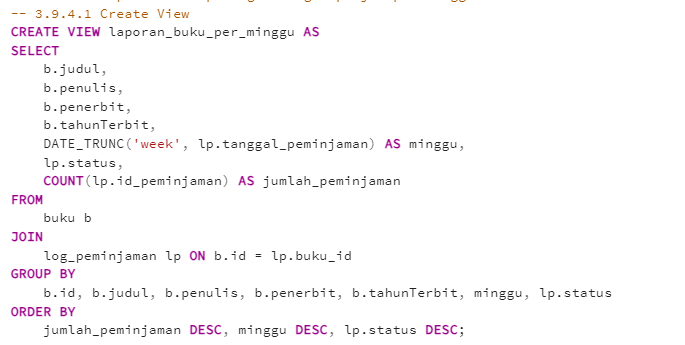


### Tabel laporan buku paling sering dipinjam per-minggu

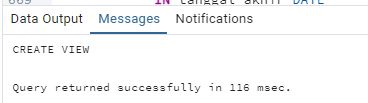
Querry ini membuat view bernama laporan\_buku\_per\_minggu, yang menyajikan laporan buku yang paling sering dipinjam per minggu. Laporan ini mencakup informasi judul buku, penulis, penerbit, tahun terbit, minggu peminjaman, status peminjaman, serta jumlah peminjaman setiap buku. Dengan menggunakan fungsi DATE\_TRUNC('week', ...), tanggal peminjaman dipotong menjadi minggu, kemudian hasilnya dikelompokkan berdasarkan buku dan minggu peminjaman. Hasil laporan diurutkan berdasarkan jumlah peminjaman tertinggi, minggu peminjaman, dan status peminjaman, untuk mempermudah identifikasi tren peminjaman buku dalam periode mingguan.

#### Create View

* **Syntax :**



* **Output :**

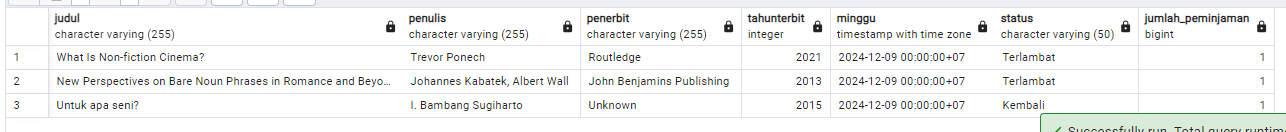


#### Menampilkan hasil tabel laporan buku

* **Syntax :**



* **Output :**

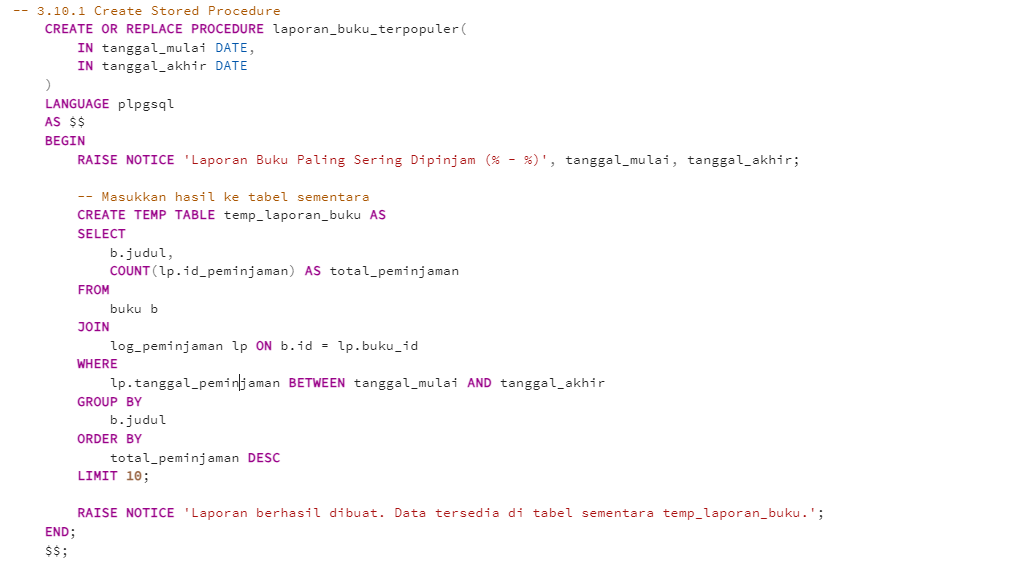


## Stored Procedure untuk Laporan buku paling sering dibaca

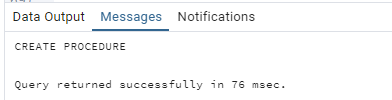
Stored procedure laporan\_buku\_terpopuler ini dibuat untuk menghasilkan laporan buku yang paling sering dipinjam dalam periode waktu tertentu. Procedure ini menerima dua parameter: tanggal\_mulai dan tanggal\_akhir, yang digunakan untuk menentukan rentang waktu peminjaman. Di dalam procedure, sebuah tabel sementara (temp\_laporan\_buku) dibuat untuk menyimpan hasil laporan, yang mencakup judul buku dan jumlah peminjaman dalam rentang waktu tersebut, diurutkan berdasarkan jumlah peminjaman terbanyak. Setelah selesai, laporan disimpan di tabel sementara, dan informasi tentang laporan yang berhasil dibuat ditampilkan.

### Create Stored Procedure

* **Syntax :**



* **Output :**

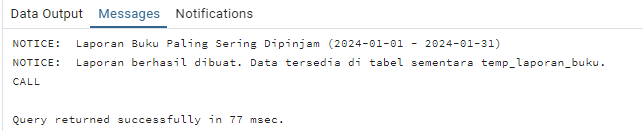


### Memanggil laporan buku terpopuler

* **Syntax :**



* **Output :**

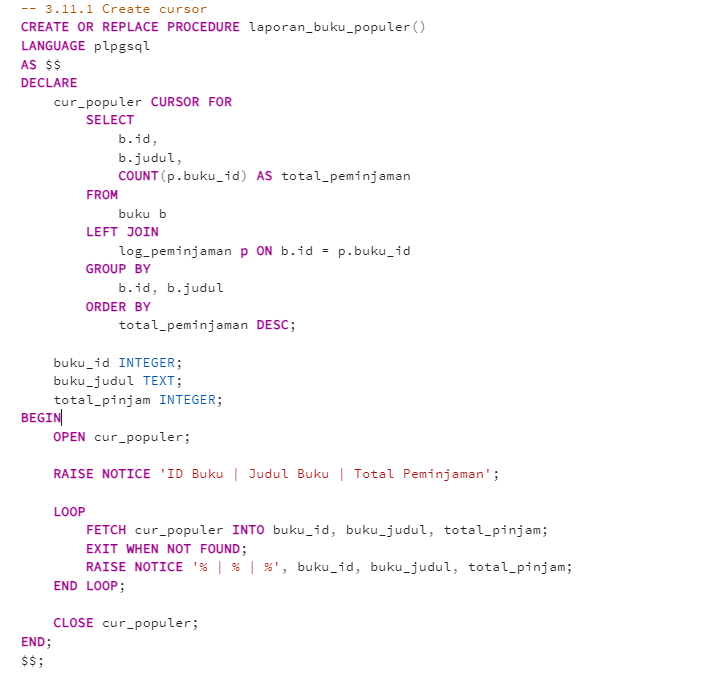
****

## Cursor untuk Laporan buku paling sering dibaca

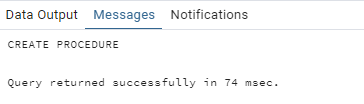
Stored procedure laporan\_buku\_populer menggunakan cursor (cur\_populer) untuk mengambil data buku yang paling sering dipinjam. Pertama, query SELECT mengambil ID buku, judul, dan jumlah peminjaman dari tabel buku dan log\_peminjaman, mengurutkannya berdasarkan jumlah peminjaman terbanyak. Setelah itu, cursor dibuka dan data yang diambil dimasukkan ke dalam variabel (buku\_id, buku\_judul, total\_pinjam) yang kemudian ditampilkan menggunakan RAISE NOTICE. Prosedur ini berjalan dalam loop hingga semua data buku ditampilkan, lalu cursor ditutup.

### Create cursor

* **Syntax :**



* **Output :**

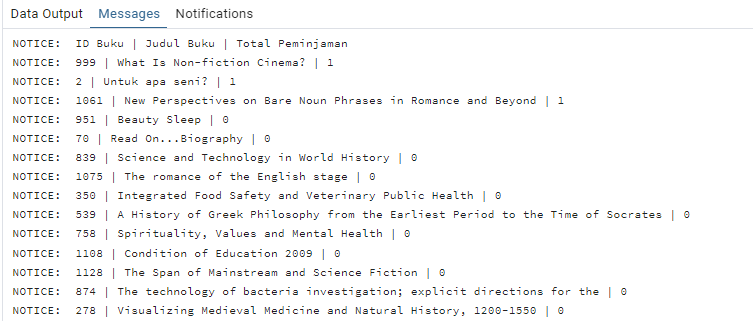


### Memanggil laporan buku terpopuler

* **Syntax :**



* **Output :**

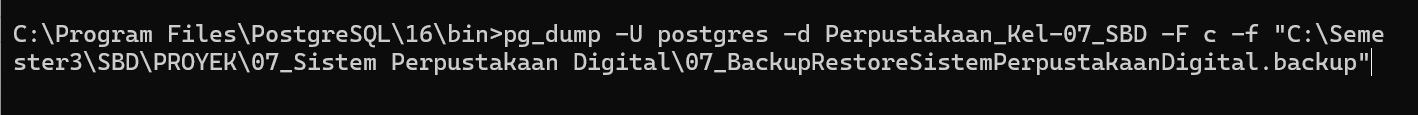


## Backup dan Restore

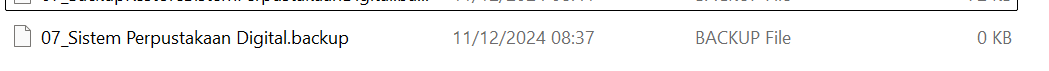
### Backup database

Backup adalah adalah proses membuat salinan data dari database yang ada. Backup dilakukan untuk memastikan bahwa data dapat dipulihkan jika terjadi kehilangan data.

* **Syntax :**



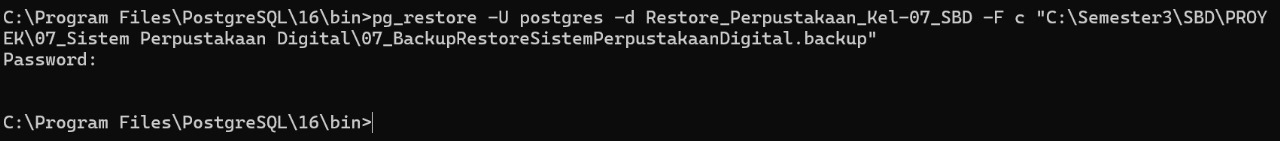
* **Output :**



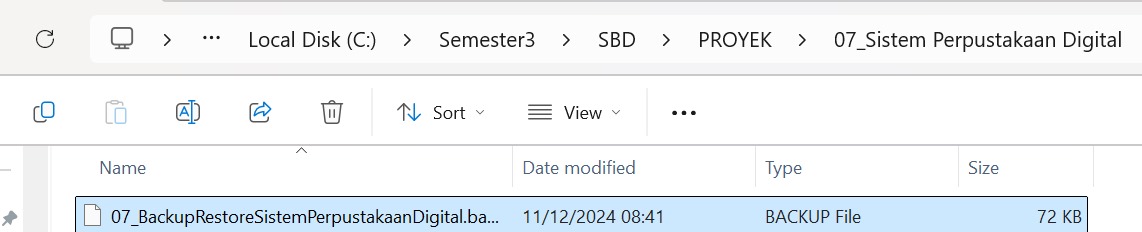
### Restore database

Restore adalah proses mengembalikan database dari salinan backup. Proses ini dilakukan ketika database mengalami kerusakan atau jika ada kebutuhan untuk memuat ulang data ke dalam sistem baru.

* **Syntax :**

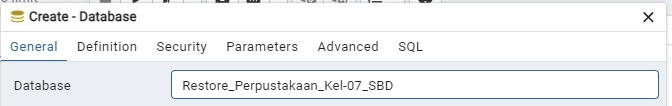


* **Output :**

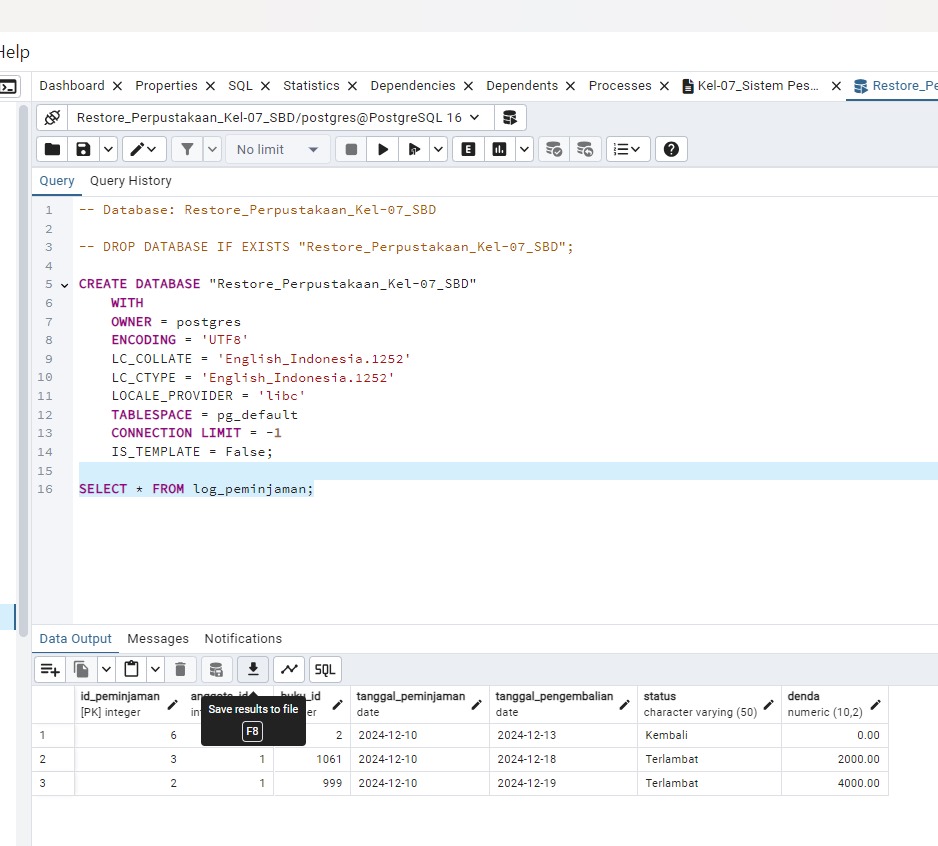


Lakukan pengecekan pada database yang sudah di create dengan nama ‘07\_BackupRestoreSistemPerpustakaanDigital.backup’ sebagai berikut :

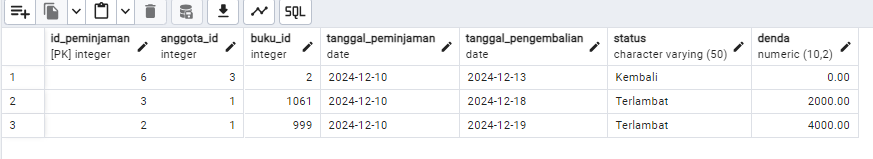
1. **Create database baru dengan nama Restore\_Perpustakaan\_Kel-07\_SBD**



1. **Jika sudah, klik Create Script -> lakukan selecct \* from log\_peminjaman.**



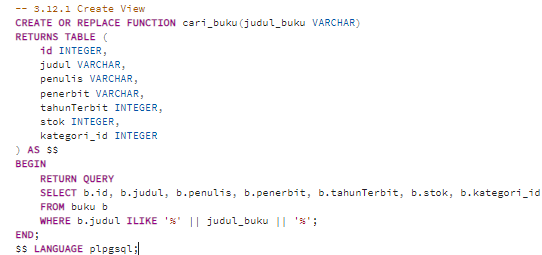
1. **Nanti akan menghasilkan output :**



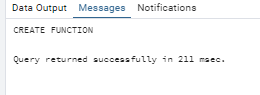
## Fungsi Mencari Buku

### Create Function

* **Syntax :**



* **Output :**

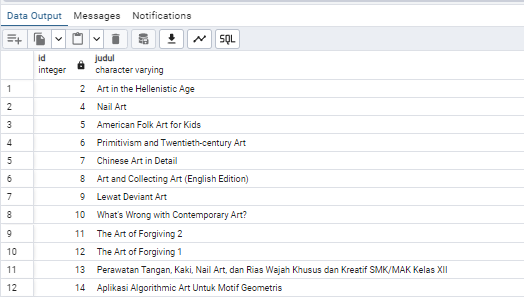


### Pengujian

* **Syntax :**



* **Output :**



# Kesimpulan dan Saran

## Kesimpulan

Sistem Perpustakaan Digital yang telah dikembangkan oleh Kelompok-07 berhasil memenuhi kebutuhan utama perpustakaan modern, seperti pengelolaan koleksi digital, peminjaman, pengembalian, dan pembayaran denda secara online. Sistem ini memberikan kemudahan akses bagi pengguna dan mengotomatiskan proses manajemen perpustakaan. Namun, sistem masih memiliki keterbatasan pada jumlah koleksi buku yang hanya sebanyak 1059 item dan tergantung pada koneksi internet untuk menjalankan fitur crawling.

## Saran

1. **Penambahan Koleksi Buku**  
   Sistem dapat ditingkatkan dengan menambahkan lebih banyak koleksi buku untuk memenuhi kebutuhan pengguna yang beragam.
2. **Optimalisasi Crawling**  
   Usahakan untuk mengoptimalkan proses crawling agar tetap efisien, bahkan pada koneksi internet yang kurang stabil.
3. **Pengembangan Fitur Tambahan**  
   Tambahkan fitur seperti rekomendasi buku berbasis preferensi pengguna atau integrasi dengan platform pembelajaran.
4. **Pengujian Lebih Lanjut**  
   Lakukan pengujian sistem secara menyeluruh untuk memastikan performa, keamanan, dan konsistensi data.
5. **Pengembangan Sistem Offline**  
   Pertimbangkan untuk menyediakan beberapa fitur dasar yang dapat diakses secara offline untuk mendukung pengguna dengan kendala internet.

Saran-saran ini diharapkan dapat membantu meningkatkan kualitas dan efisiensi Sistem Perpustakaan Digital.

# HASIL PRESENTASI

**Tabel 3. Jawaban Pertanyaan**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Pertanyaan** | **11323013** | **11323028** | **11323051** | **11323055** |
| 1 | Apakah keseluruhan implementasi fitur tercapai dan relevan dengan tujuannya? | Ya, Seluruh fitur kami berjalan dengan sempurna. | Ya, dari semua fitur yang kami tampilkan berjalan dengan baik. | Ya, secara keseluruhan fitur yang tersedia sudah berjalan dengan baik | Ya, secara keseluruhan semua fitur sudah terimplementasi dengan baik. |
| 2 | Apa yang lebih dan apa yang kurang untuk diperbaiki (kalau ada)? | Kami menambahkan beberapa fitur tambahan selain fitur utama yang sudah ditentukan, yaitu fitur proses pembayaran denda, laporan peminjaman (tahunan, bulanan, mingguan), laporan semua anggota yang meminjam, serta fitur pencarian. | Selain fitur utama yang telah ditetapkan, kami juga mengembangkan fitur tambahan, seperti proses pembayaran denda, laporan peminjaman (tahunan, bulanan, mingguan), laporan semua anggota peminjam, dan fitur pencarian data. | Fitur utama yang ditentukan telah kami penuhi, namun kami juga menambahkan fitur lainnya, yaitu proses pembayaran denda, laporan peminjaman (berdasarkan periode), laporan semua anggota peminjam, serta fitur pencarian. | Kami melengkapi fitur utama dengan beberapa tambahan, seperti proses bayar denda, laporan peminjaman (tahunan, bulanan, mingguan), laporan peminjaman semua anggota, dan fitur pencarian untuk memudahkan pengguna. |
| 3 | Menurutmu apakah kontribusimu sudah maksimal dan kamu paham dengan yang kamu implementasikan? | Kontribusi yang sudah saya lakukan dapat saya pahami dan dapat saya implementasikan ,serta pengerjaaan yang saya lakukan dapat saya pertanggung jawabkan | Kontribusi yang saya lakukan dapat saya pahami dan dapat implementasikan dengan baik dengan pekerjaan yang saya lakukan. | Menurut saya kontribusi yang sudah saya lakukan sudah cukup baik dalam tim ini, dan setelah saya mengerjakan proyek ini saya juga menjadi lebih paham mengenai pengimplementasian setiap sintax. | Menurut saya, kontribusi saya sudah maksimal karena saya terlibat dalam penyusunan laporan dan pembuatan query bersama kelompok. Saya juga memahami implementasi yang dilakukan, sehingga dapat berkontribusi secara optimal. |

**Tabel 4. Penilaian**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Jenis Penilaian** | **11323013** | **11323028** | **11323051** | **11323055** |
| 1 | Hasil Kelompok | 96 | 96 | 97 | 98 |
| 2 | Diri Sendiri | 95 | 95 | 97 | 97 |