**LAPORAN PROYEK KELOMPOK 5**

**SISTEM BASIS DATA**

***“Sistem Akademik Sederhana”***



|  |  |  |  |
| --- | --- | --- | --- |
| No | NIM | Nama | Prodi |
| 1. | 11323010 | Franklyn Aldo Ignatia Lumbantoruan | D3TI |
| 2. | 11323011 | Ferry Bastian Siagian | D3TI |
| 3. | 11323005 | Abeloisa Chelsea Pardosi | D3TI |
| 4. | 11323047 | Agnes Elyestra Sidabutar | D3TI |

**INSTITUT TEKNOLOGI DEL**

**FAKULTAS VOKASI 2024/2025**

Daftar Isi

[Bab 1 Pendahuluan 3](#_Toc184752100)

[1.1. Latar Belakang 3](#_Toc184752101)

[Bab 2. Rancangan dan Metode 4](#_Toc184752102)

[2.1 Pengumpulan Data 4](#_Toc184752103)

[Bab 3. Implementasi 5](#_Toc184752104)

[3.2. Create Table 6](#_Toc184752105)

[3.3. DUMMY DATA 7](#_Toc184752106)

[3.4. JOIN 8](#_Toc184752107)

[3.5. VIEW 9](#_Toc184752108)

[3.6. TRIGGER 12](#_Toc184752109)

[3.7. Authorization 12](#_Toc184752110)

[3.7. TRANSACTION 13](#_Toc184752111)

[3.8. STORED PROCEDURE 14](#_Toc184752112)

[3.9. CURSOR 15](#_Toc184752113)

[3.10. FITUR UTAMA 16](#_Toc184752114)

[3.11 Backup & Restore 19](#_Toc184752115)

[3.11.1. Backup 19](#_Toc184752116)

[3.11.2. Restore 19](#_Toc184752117)

[3.12. Crawling dan Scraping 20](#_Toc184752118)

[3.12.1. Crawling 20](#_Toc184752119)

[3.12.2. Scraping 24](#_Toc184752120)

# 

# **Bab 1 Pendahuluan**

## Latar Belakang

Sistem akademik adalah elemen dalam mendukung operasional institusi pendidikan, khususnya dalam mengelola berbagai data yang kompleks seperti jadwal perkuliahan, data mahasiswa, dosen, mata kuliah, serta evaluasi akademik. Pembuatan mini proyek sistem akademik sederhana menggunakan PostgreSQL bertujuan untuk memberikan solusi yang efisien dan terstruktur dalam pengelolaan data akademik. PostgreSQL dipilih sebagai basis utama sistem ini karena kehandalannya sebagai sistem manajemen basis data relasional open-source yang mendukung fitur-fitur canggih seperti transaksi ACID (Atomicity, Consistency, Isolation, Durability), keamanan tingkat tinggi, dan kemampuan menangani volume data yang besar. Sistem ini diharapkan dapat membantu institusi pendidikan, khususnya yang memiliki keterbatasan sumber daya, dalam mengotomatiskan proses administrasi, meminimalkan kesalahan manual, dan meningkatkan efisiensi kerja.

Proyek ini dirancang untuk menjadi sederhana namun mencakup fungsi-fungsi dasar yang esensial, seperti pencatatan data mahasiswa, nilai, mata kuliah, serta penyesuaian jadwal mata kuliah dari google kalender menggunakan API. Salah satu fitur dalam proyek ini dapat menampilkan IPK mahasiswa berdasarkan dengan nilai yang di input. Proyek ini juga dirancang untuk memberikan pengalaman langsung kepada mahasiswa dalam mengembangkan sistem basis data relasional, termasuk desain skema database, integrasi data, dan implementasi query SQL untuk kebutuhan akademik. Lebih jauh lagi, mini proyek ini tidak hanya berfungsi untuk memenuhi tugas proyek mata kuliah Sistem Basis Data tetapi juga berfungsi sebagai studi kasus yang dapat dikembangkan lebih lanjut menjadi sistem yang lebih kompleks dan terintegrasi. demikian, proyek ini sebagai media pembelajaran bagi mahasiswa untuk memahami bagaimana membangun dan mengimplementasikan sistem berbasis database yang praktis dan fungsional.

# **Bab 2. Rancangan dan Metode**

## **2.1 Pengumpulan Data**

Langkah awal yang dilakukan dalam pengerjaan proyek ini adalah menganalisis system akademik yang terjadi di dalam CIS. Dengan meninjau serta website CIS dan bertanya kepada Teaching Assiten, dan Dosen kami mendapatkan arahan dan pandangan untuk mengerjakan proyek ini.

**2.2 Tools Yang Digunakan**

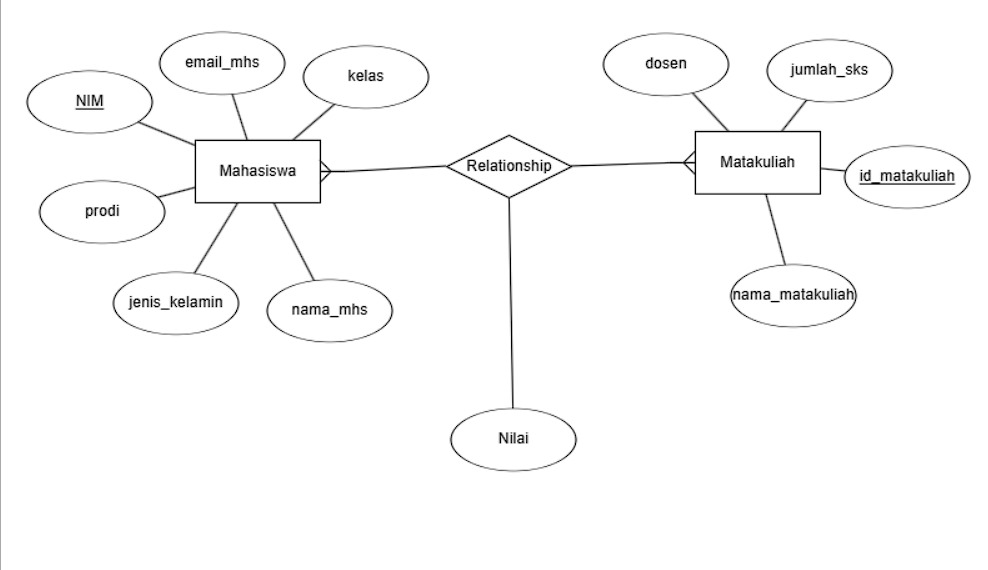
Pada pengerjaan proyek ini ada beberapa tools yang digunakan yaitu :

* ERD plus: Tools ini digunakan untuk merancang Entity-Relationship Diagram (ERD) pada sistem akademik ini. Dengan menggunakan ERD Plus, pengembang dapat membuat representasi visual dari struktur database yang mencakup entitas, atribut, dan relasi antar entitas. Hal ini membantu dalam memahami alur data, mengidentifikasi hubungan penting, dan memastikan desain database yang optimal sebelum implementasi.
* Pg Admin : Merupakan tools manajemen grafis untuk PostgreSQL yang digunakan untuk mengelola database pada sistem akademik ini. Dengan PgAdmin, pengembang dapat membuat, memodifikasi, dan memonitor database
* Google colab: Platform ini digunakan untuk menulis dan menjalankan kode Python yang terintegrasi dengan database PostgreSQL dan layanan API Google. Google Colab mempermudah pengembangan sistem akademik karena memungkinkan kolaborasi langsung, akses dari berbagai perangkat, serta mendukung integrasi pustaka-pustaka Python yang diperlukan. Dalam system ini Google colab digunakan untuk melakukan Scraping dan Crawling jadwal Matakuliah
* Google API : Layanan ini digunakan untuk mengintegrasikan fitur tambahan seperti Google Calendar ke dalam sistem akademik. Google API memungkinkan sistem untuk menambahkan jadwal perkuliahan secara otomatis ke Google Calendar, memberikan pengalaman pengguna yang lebih modern dan terintegrasi. Dengan fitur ini, jadwal yang diinput di sistem akademik dapat disinkronkan langsung ke Google Calenda
* Trello : : Trello digunakan untuk mengelola dan mengakomodasi pengerjaan proyek sistem akademik. Dengan Trello, mahasiswa dapat membuat jadwal dan daftar tugas yang terstruktur, seperti "To Do" (apa yang akan dikerjakan), "In Progress" (tugas yang sedang dikerjakan), dan "Done" (tugas yang telah selesai). Selain itu, Trello memungkinkan pembagian tugas yang jelas sesuai dengan peran dan tanggung jawab masing-masing anggota tim. Trello juga dapat diintegrasikan dengan berbagai layanan seperti GitHub untuk manajemen versi kode atau Google Drive untuk menyimpan dokumen dan hasil kerja.
* GitHub: GitHub digunakan sebagai platform manajemen kolaborasi untuk proyek sistem akademik ini. Dengan GitHub, mahasiswa dapat menyimpan, mengelola, dan melacak perubahan kode yang dibuat selama pengembangan sistem. Fitur version control di GitHub memungkinkan untuk melihat riwayat perubahan, mengidentifikasi siapa yang membuat perubahan tertentu, dan memulihkan versi sebelumnya jika diperlukan

# **Bab 3. Implementasi**

Pada pengerjaan proyek ini dilakukan beberapa implementasi yang terkait dengan materi Sistem Basis Data yaitu :

3.1. **Entity Relationship Diagram (ERD)**



Gambar 1 ERD Sistem akademik Sederhana

Gambar ini merupakan struktur yang menggambarkan bagaimana struktur data untuk system akademik sederhana yang akan dibuat, dimana pada gambar tersebut terdapat 2 entitas yaitu :

1. **Entitas Mahasiswa** dengan atribut :

* NIM (Nomor Induk Mahasiswa)
* nama\_mhs: Nama mahasiswa.
* jenis\_kelamin: Jenis kelamin mahasiswa.
* prodi: Program studi yang diambil oleh mahasiswa.
* email\_mhs: Email resmi mahasiswa.
* kelas: Kelas atau kelompok belajar mahasiswa

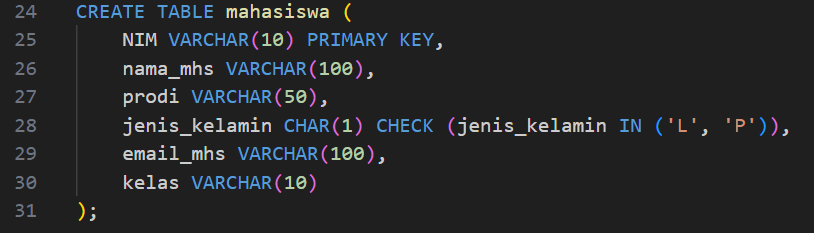
1. **Entitas Matakuliah** dengan atribut :

* id\_matakuliah: Identitas unik untuk setiap mata kuliah.
* nama\_matakuliah: Nama mata kuliah.
* jumlah\_sks: Jumlah Satuan Kredit Semester (SKS) untuk mata kuliah.
* dosen: Nama dosen pengajar mata kuliah

pada gambar tersebut terdapat relasi menghubungkan **Mahasiswa** dengan **Matakuliah**. Ini menunjukkan bahwa setiap mahasiswa dapat mengambil beberapa mata kuliah, dan setiap mata kuliah dapat diikuti oleh banyak mahasiswa (hubungan **many-to-many**). Pada relasi terdapat atribut yaitu nilai karena merupakan informasi tambahan yang terkait dengan hubungan antara **Mahasiswa** dan **Matakuliah**

## **3.2. Create Table**

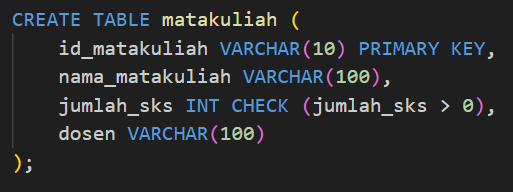
**a. table mahasiswa**



Gambar 2. Create Tabel Mahasiswa

Query ini digunakan untuk membuat table mahasiswa dimana pada table ini sudah di set nim sebagai primary key nya, pada table ini juga memiliki atribut lain nya seperti nama\_mhs, prodi, jenis kelamin, email dan kelas sesuai dengan tipe data nya masing masing. Pada table ini juga terdapat method check untuk mengechek jenis kelamin dari data mahasiswa yang akan di insert nanti

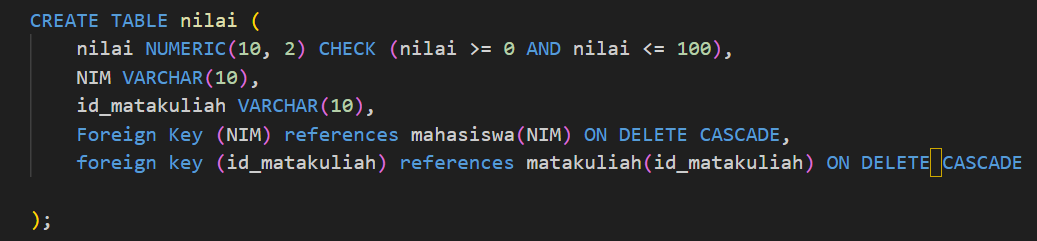
**b. Tabel matakuliah**



Gambar 3. Create Tabel Matakuliah

Query ini menambahkan data ke tabel matakuliah yang mencakup ID mata kuliah (primary key), nama mata kuliah, jumlah SKS, dan dosen pengampu beserta tipa data nya masing masing. Pada table ini juga terjadi pengecekan dimana pada saat memasukkan data nya nanti jumlah sks harus lebih dari 0

**c. table nilai**



Query ini memasukkan data ke tabel nilai yang menghubungkan mahasiswa (melalui NIM) dan mata kuliah (melalui id\_matakuliah) dengan nilai akhir. Nilai yang dimasukkan dibatasi antara 0 hingga 100.

## **3.3. DUMMY DATA**

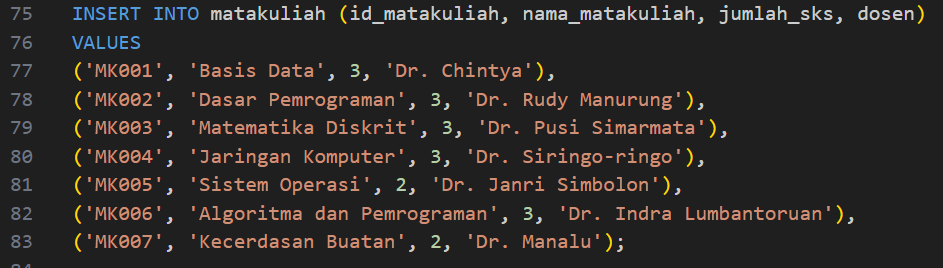
**a. Tabel mahasiswa**



Gambar 4. Dummy data Mahasiswa

Query ini digunakan untuk memasukkan Dummy data pada tabel mahasiswa sesuai dengan atribut dari table mahasiwa yang terbuat tadi dimana terdiri dari 22 entri mahasiswa dengan kolom NIM, nama\_mhs, prodi, jenis\_kelamin, email\_mhs, dan kelas.

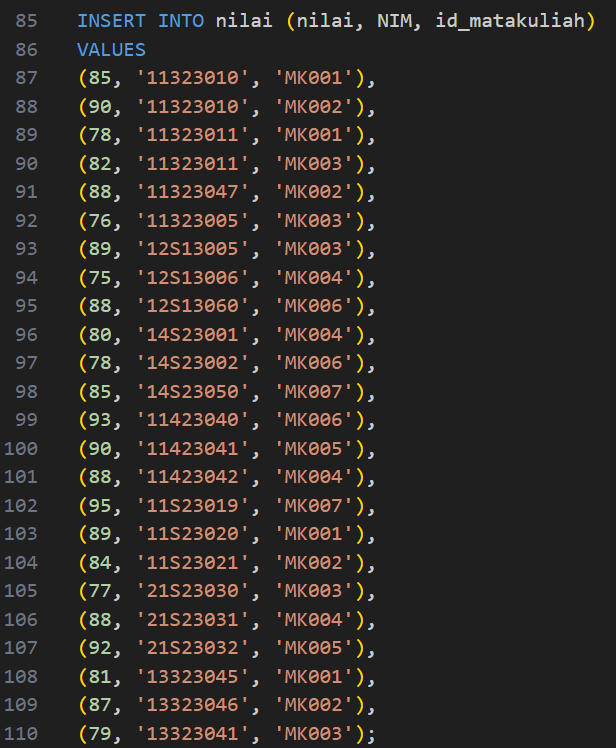
**b. Tabel matakuliah**



Gambar 5. Dummy Data Matakuliah

Query ini digunakan untuk memasukkan Dummy data pada tabel matakuliah sesuai dengan atribut dari table matakuliah yang dibuat tadi dimana mencakup 7 entri mata kuliah dengan kolom id\_matakuliah, nama\_matakuliah, jumlah\_sks, dan dosen. Setiap mata kuliah memiliki ID unik seperti 'MK001' untuk Basis Data atau 'MK007' untuk Kecerdasan Buatan. Data ini mencakup jumlah SKS (2 atau 3 SKS) yang menunjukkan bobot kredit akademik mata kuliah tersebut serta nama dosen pengampu, seperti Dr. Chintya untuk Basis Data dan Dr. Manalu untuk Kecerdasan Buatan.

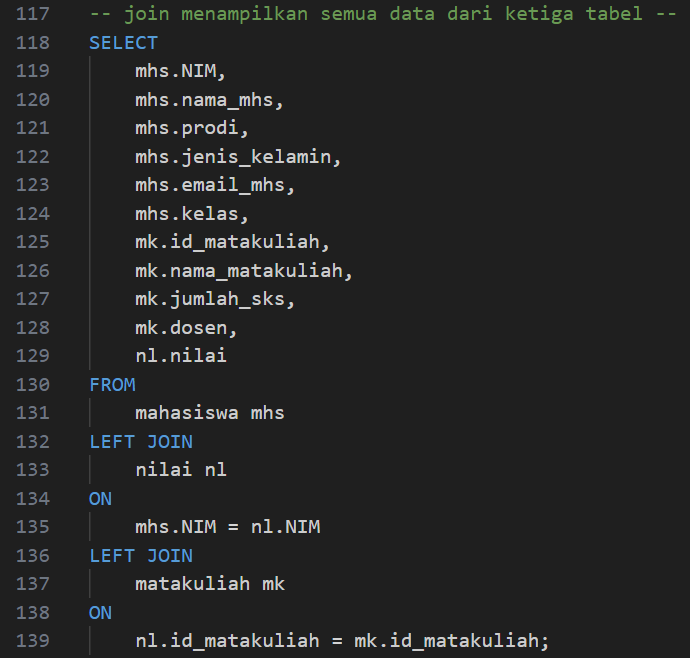
**c. Tabel nilai**



Gambar 6. Dummy data tabel nilai

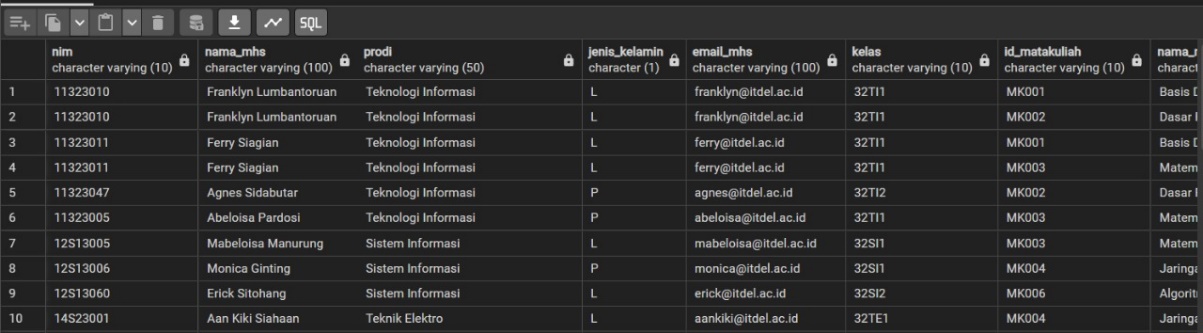
Query ini digunakan untuk memasukkan Dummy data pada tabel nilai sesuai dengan atribut dari table nilau yang terbuat tadi dimana terdiri dari 24 entri yang mencatat nilai akademik mahasiswa dalam mata kuliah tertentu. Kolom nilai berisi angka 0–100, menunjukkan performa mahasiswa dalam mata kuliah terkait. Setiap baris dihubungkan dengan NIM dari tabel mahasiswa dan id\_matakuliah dari tabel matakuliah sebagai foreign key. Contohnya, mahasiswa dengan NIM '11323010' memperoleh nilai 85 pada mata kuliah Basis Data (MK001) dan 90 pada Dasar Pemrograman (MK002). Data ini menghubungkan performa akademik dengan mahasiswa dan mata kuliah tertentu

## **3.4. JOIN**



Gambar 7. Join antara tabel

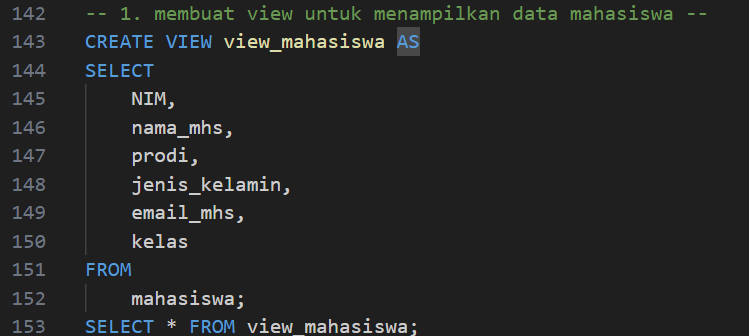
Query di atas merupakan operasi ***LEFT JOIN*** untuk menggabungkan tiga tabel, yaitu mahasiswa , nilai dan matakuliah Query ini bertujuan untuk menampilkan semua data dari tabel mahasiswa serta data yang relevan dari tabel nilai dan matakuliah. Pada query ini juga digunakan alias untuk melakukan join agar tidak terjadi kekeliruan Ketika menggabungkan table



Gambar 8. Output Join

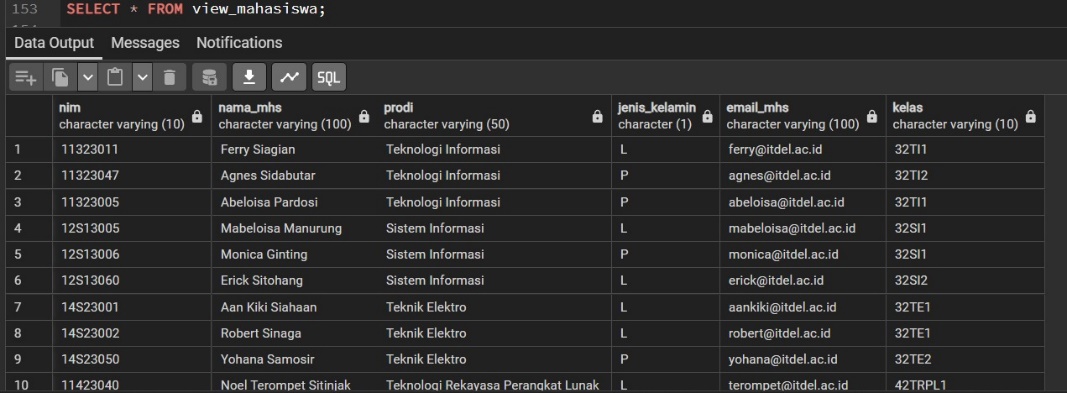
## **3.5. VIEW**

**a. View mahasiswa**



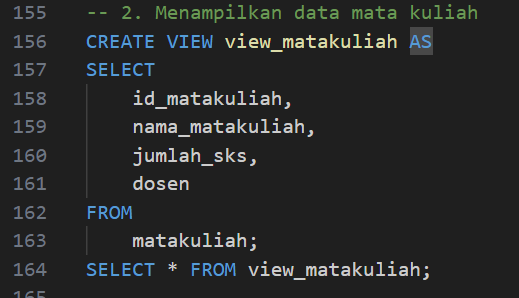
Gambar 9. View Mahasiwa

View ini digunakan untuk menampilkan data mahasiswa secara sederhana. Data yang ditampilkan adalah atribut penting seperti NIM, nama\_mhs, prodi, jenis\_kelamin, email\_mhs, dan kelas dari tabel mahasiswa. View ini memudahkan akses data mahasiswa tanpa harus mengakses seluruh tabel mahasiswa, sehingga lebih efisien untuk operasi tertentu, seperti laporan. Select View mahasiswa yang dibuat maka akan tampil



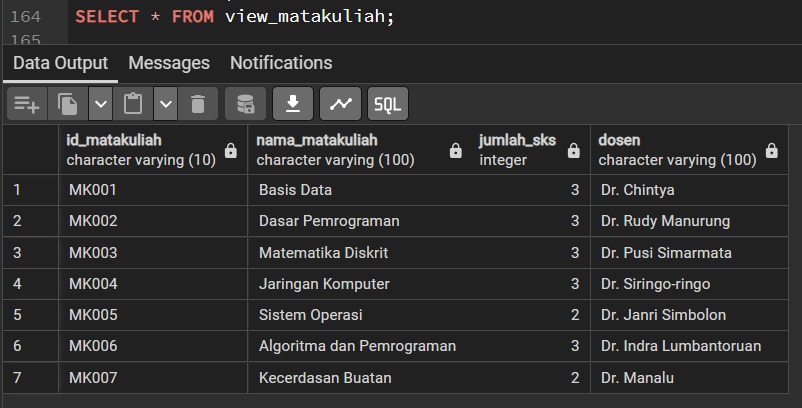
Gambar 10. Output View Mahasiswa

b. **View matakuliah**

****

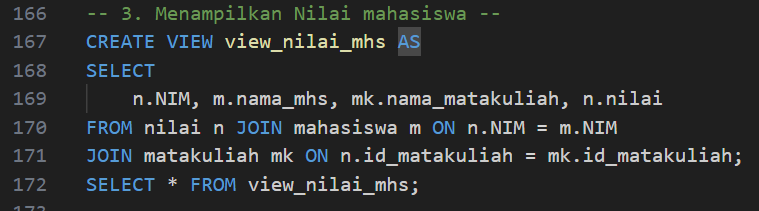
Gambar 11. View Matakuliah

View ini digunakan untuk menampilkan data mata kuliah. Informasi yang disediakan meliputi id\_matakuliah, nama\_matakuliah, jumlah\_sks, dan nama dosen. Dengan view ini, pengguna dapat melihat daftar mata kuliah beserta detailnya secara langsung tanpa harus melakukan query dari tabel matakuliah setiap saat.



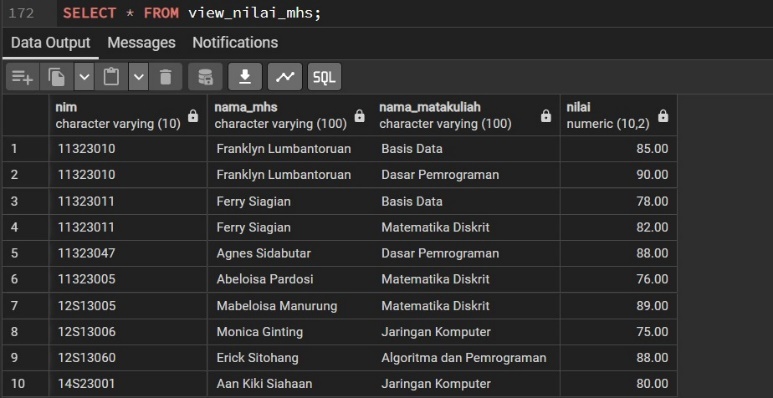
Gambar 12. Output View Matakuliah

**c. View nilai\_mahasiswa**

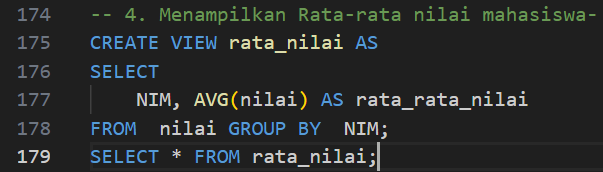
****

Gambar 13. View Nilai Mahasiswa

View ini digunakan untuk menampilkan informasi nilai mahasiswa yang digabungkan dari tiga tabel, yaitu nilai, mahasiswa, dan matakuliah. View ini menunjukkan NIM, nama\_mhs, nama\_matakuliah, dan nilai. Dengan adanya view ini, pengguna dapat dengan mudah melihat nilai setiap mahasiswa untuk setiap mata kuliah tanpa harus menulis query join yang kompleks.

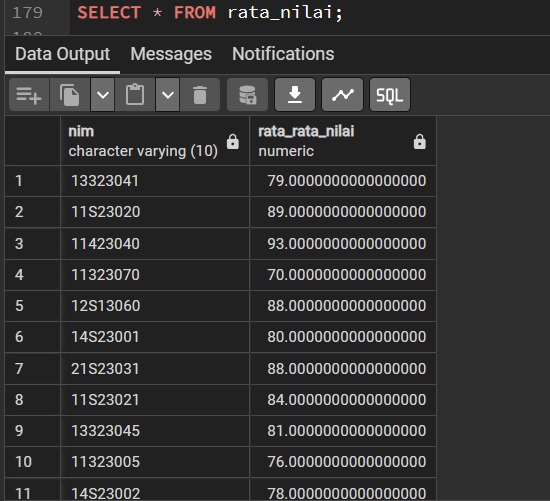


**d. View rata rata nilai mahasiwa**

****

Gambar 14. View Rata- Rata Nilai mahasiswa

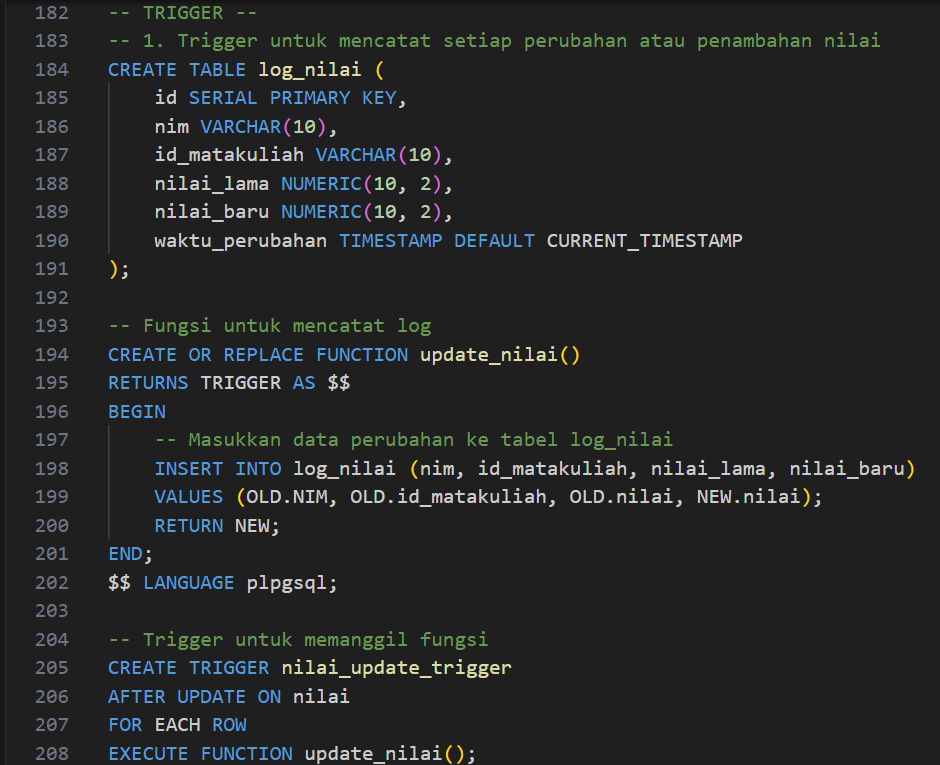
View ini dibuat untuk menghitung rata-rata nilai (AVG(nilai)) setiap mahasiswa berdasarkan NIM. Data yang dihasilkan meliputi NIM mahasiswa dan rata-rata nilai mereka. Dengan view ini, pengguna dapat dengan mudah memperoleh informasi tentang performa akademik mahasiswa secara ringkas,



Gambar 15. Output View rata rata nilai

## **3.6. TRIGGER**

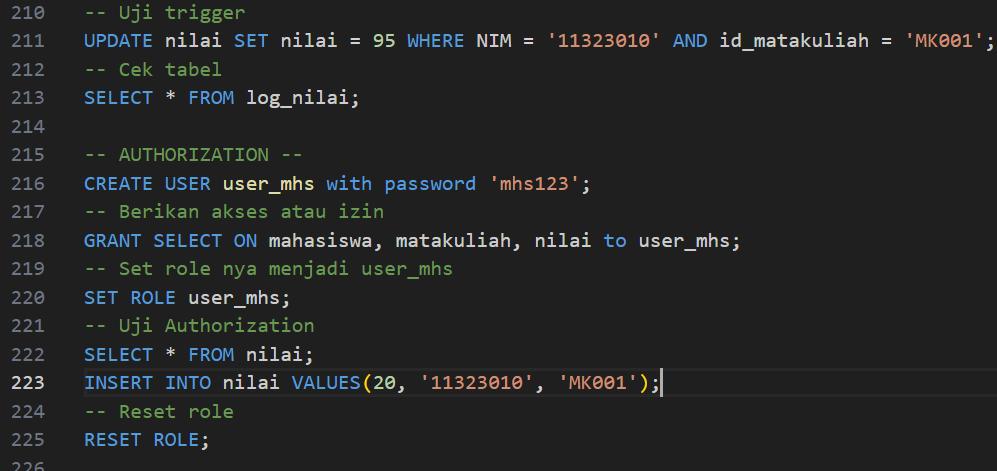
a. Trigger untuk mencatat setiap Perubahan atau penambahan nilai



Gambar 16.Trigger untuk mencatat setiap Perubahan atau penambahan nilai

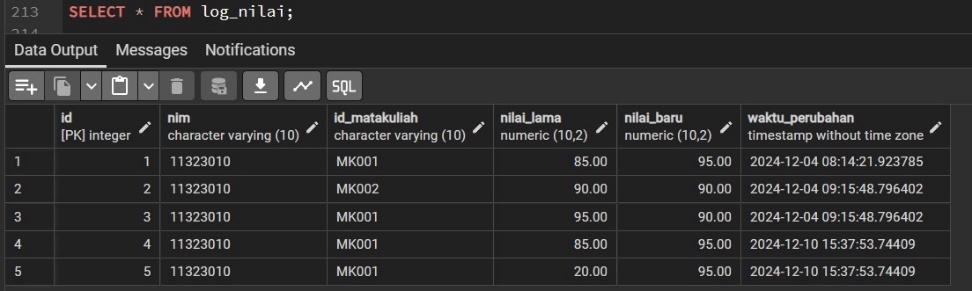
Trigger di atas digunakan untuk mencatat setiap perubahan nilai mahasiswa di tabel nilai ke dalam tabel log bernama log\_nilai. Pertama, tabel log\_nilai dibuat untuk menyimpan data log, yang mencakup nim, id\_matakuliah, nilai\_lama, nilai\_baru, dan waktu perubahan (waktu\_perubahan). Kemudian, sebuah fungsi bernama update\_nilai didefinisikan untuk mencatat data perubahan ke tabel log\_nilai setiap kali ada pembaruan data nilai pada tabel nilai. Fungsi ini dipanggil oleh trigger bernama nilai\_update\_trigger, yang diatur untuk berjalan setelah operasi UPDATE dilakukan pada tabel nilai

**Uji Trigger**



Gambar 17.Uji Trigger

Untuk menguji trigger, kita dapat melakukan update pada table nilai, dimana update tersebut otomatis akan masuk kedalam table log nilai



Gambar 18.Output Uji Trigger

## **3.7. Authorization**

a. Membuat user



Gambar 19.Membuat User

Query ini digunakan untuk membuat user bernama user\_mhs beserta dengan password nya

b. Memberikan akses

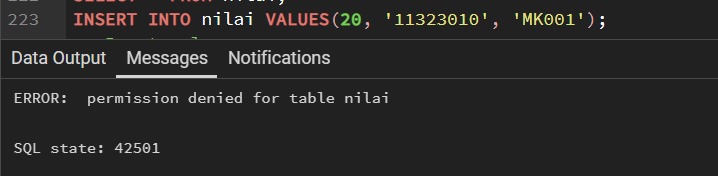


Gambar 20.Memberi akses

Query ini digunakan untuk memberikan akses tertentu kepada user

**Uji authorization**

Untuk melakukan ini kita coba menginsert nilai mahasiswa

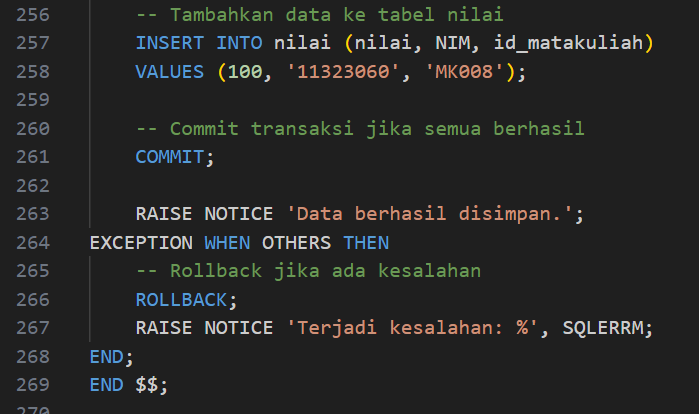
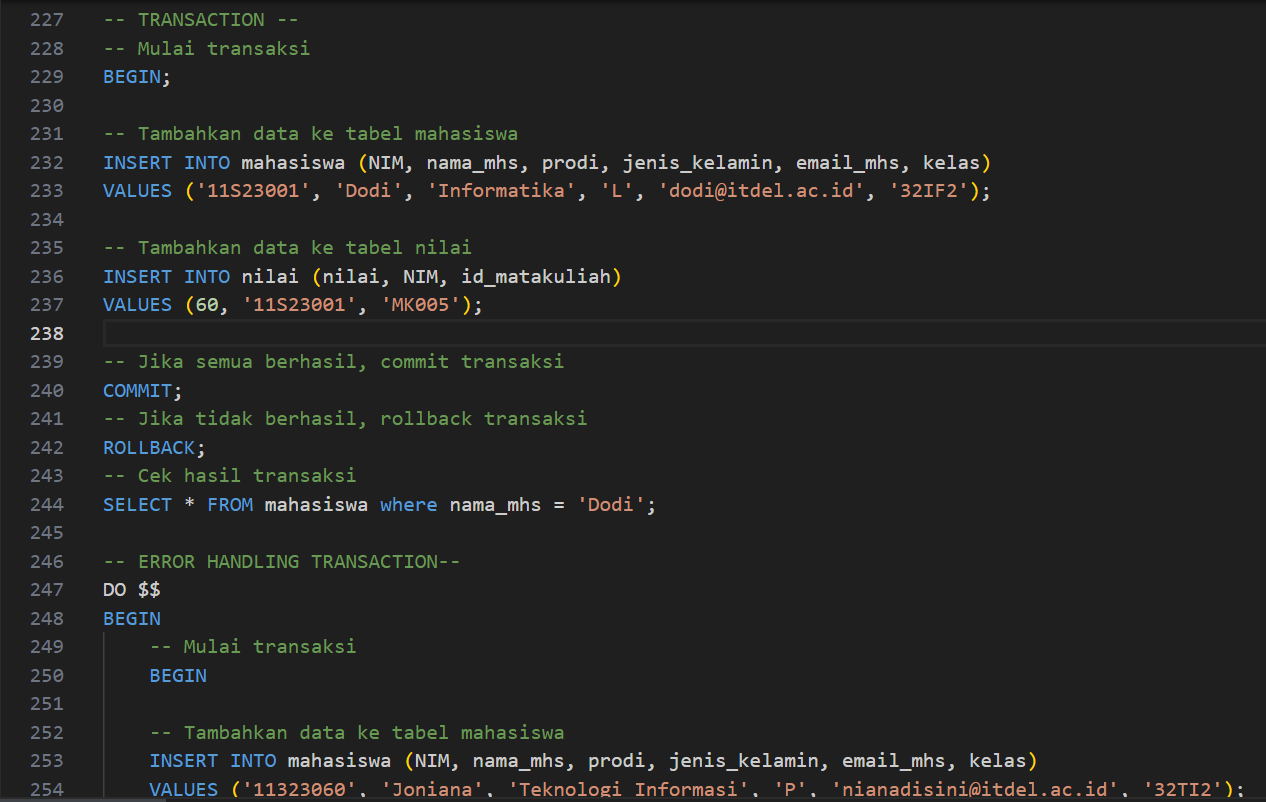


Gambar 21.Uji Authorization

Terjadi eror karena user mahasiswa hanya boleh melakukan select saja tidak dengan insert

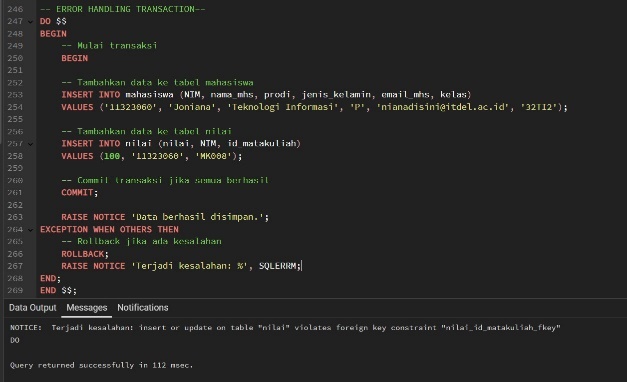
## **3.7. TRANSACTION**

a.Transaction untuk melakukan insert data



Gambar 22.Transaction untuk melakukan insert data

Pada transaksi pertama, BEGIN digunakan untuk memulai transaksi, kemudian data baru ditambahkan ke tabel mahasiswa dan nilai. Jika semua perintah berhasil, transaksi diselesaikan dengan COMMIT; jika terjadi kesalahan, transaksi dibatalkan dengan ROLLBACK sehingga tidak ada perubahan yang disimpan. Pada transaksi in juga terdapat penggunaan raise notice untuk memunculkan notifikasi apakah data tersebut berhasil dimasukkan atau tidak

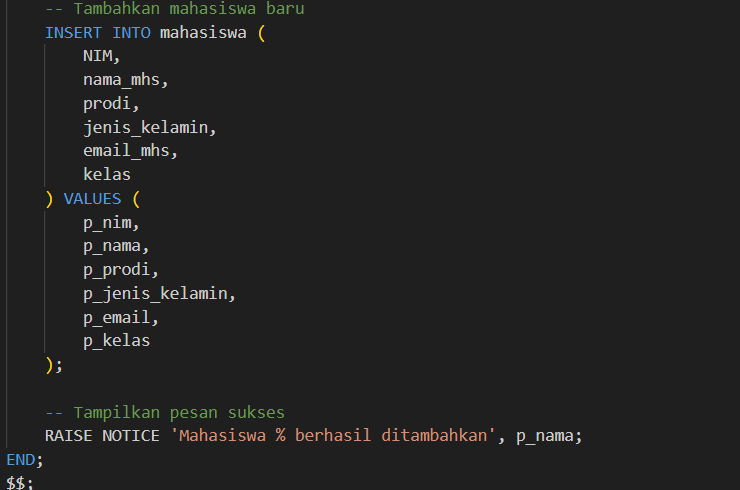
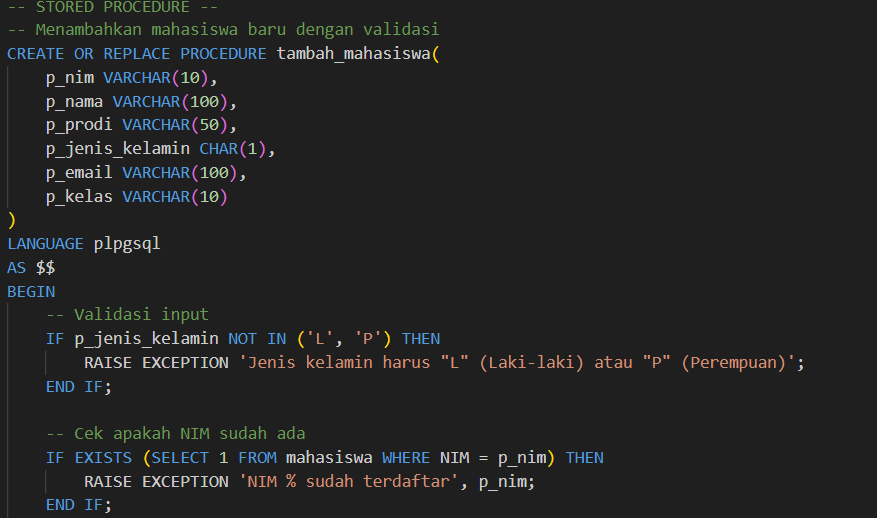


Gambar 23.Pesan error

Pada transaksi ini muncul pesan eror karena susai dengan query nya tadi jika ingin menginsert data harus sesuai dengan ketentuan yang ada pada saat ini di insert data yang tidak sesuai maka mncul lah eror

## **3.8. STORED PROCEDURE**

a. Stored menambahkan mahasiwa baru dengan validasi

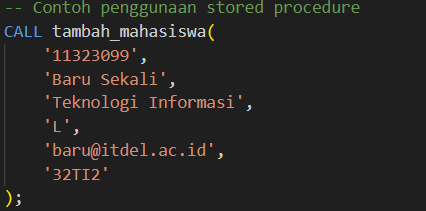


Gambar 24.Stored menambahkan mahasiwa baru dengan validasi

Query ini adalah sebuah prosedur penyimpanan bernama tambah\_mahasiswa, yang bertujuan untuk menambahkan data mahasiswa baru dengan melakukan validasi terlebih dahulu. Prosedur ini menerima beberapa parameter, yaitu NIM, nama, program studi, jenis kelamin, email, dan kelas mahasiswa. Pada procedure ini juga terdapat beberapa kondisi, Jika semua kondisi berhasil, data mahasiswa baru akan dimasukkan ke dalam tabel, dan sistem akan menampilkan pesan sukses yang menyatakan bahwa mahasiswa berhasil ditambahkan

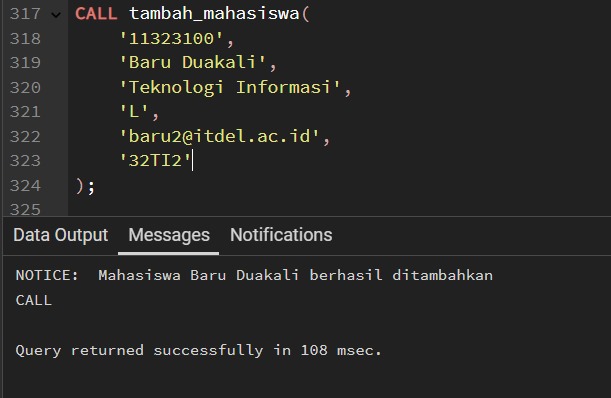
**Pemanggilan Stored Procedure**

Untuk mengecek apakah procedure berhasil atau tidak maka perlu dilakukan pemanggilan

****

Gambar 25.Pemanggilan Stored Procedure

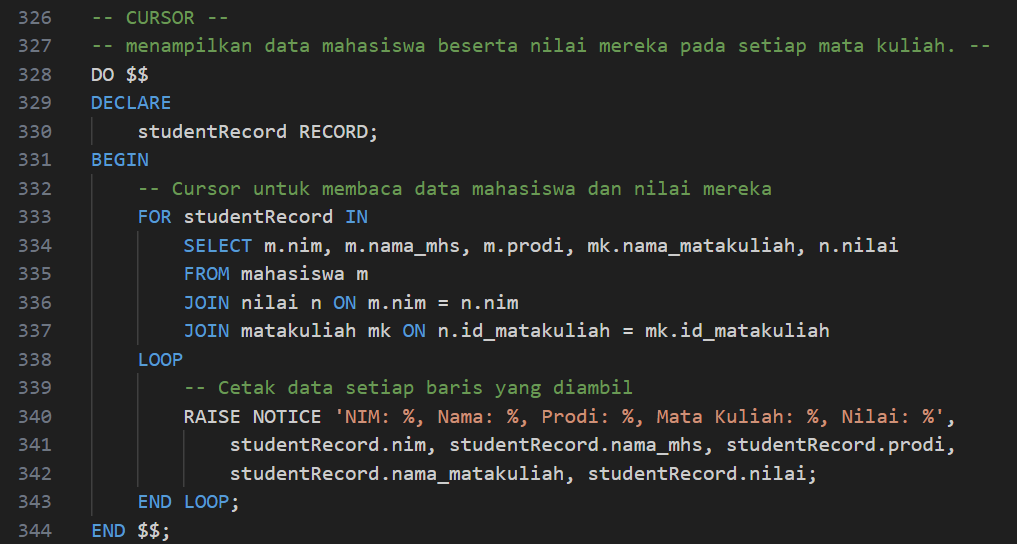
Jika berhasil maka akan tampil seperti ini



Gambar 26.Output pemanggilan Stored Procedure jika berhasil

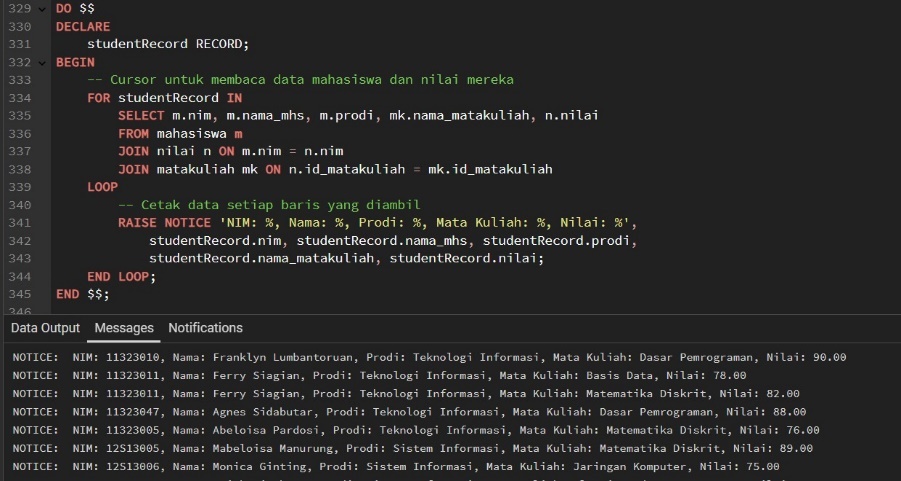
## **3.9. CURSOR**

a. Menampilkan data mahasiswa beserta nilai pada setiap mata kuliah



Gambar 27.Cursor menampilkan data mahasiswa beserta nilai pada setiap mata kuliah

Query ini digunakan untuk menampilkan data mahasiswa beserta nilai mereka pada setiap mata kuliah. Prosedur ini mendeklarasikan sebuah cursor bernama studentRecord, yang digunakan untuk membaca data mahasiswa dan nilai mereka melalui query SQL yang melakukan join antara tabel mahasiswa dan tabel nilai. Dalam bagian loop, setiap baris hasil dari cursor diambil dan ditampilkan menggunakan perintah RAISE NOTICE, yang mencetak informasi seperti NIM, nama, program studi, dan nilai mata kuliah. Maka akan ditampilkan



Gambar 28.Output data mahasiswa beserta nilai pada setiap mata kuliah

## **3.10. FITUR UTAMA**

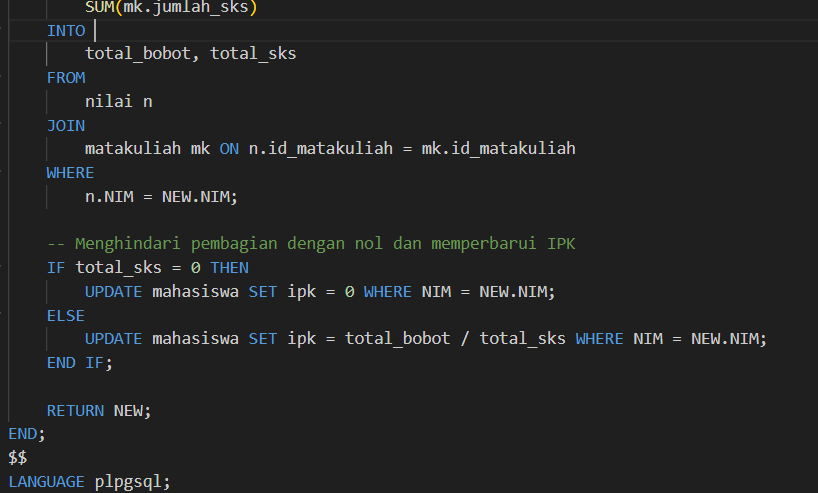
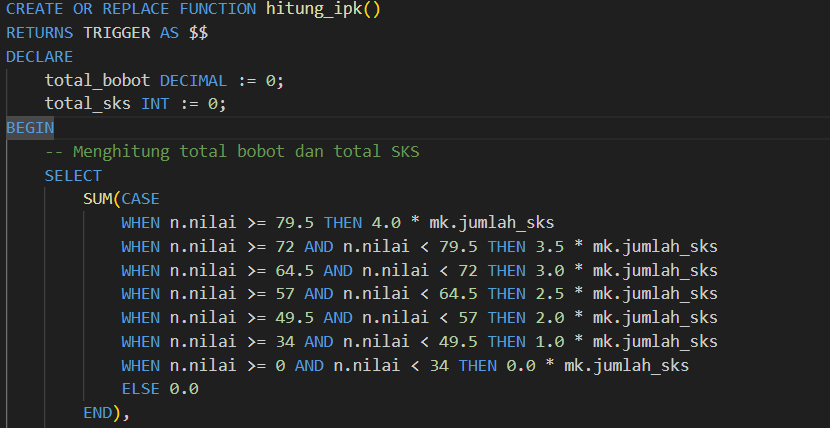
***a. Fungsi menghitung IPK otomatis***

untuk melakukam fungsi ini, terlebih dahulu kita menambahkan atribut ipk kedalam table mahasiswa dengan alter table



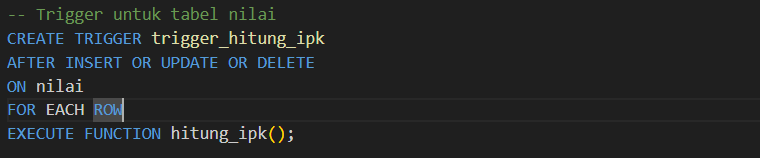
Gambar 29.Fungsi menghitung IPK otomatis

Selanjutnya tambahkan la fungsi nya



Gambar 30.Fungsi untuk menghitung Indeks Prestasi Kumulatif (IPK) secara otomatis

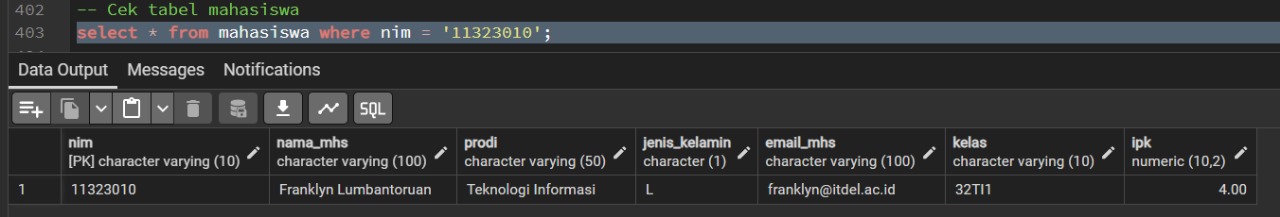
Query merupakan fungsi untuk menghitung Indeks Prestasi Kumulatif (IPK) secara otomatis dalam sistem basis data mahasiswa. Pertama, kolom ipk ditambahkan ke tabel mahasiswa untuk menyimpan nilai IPK. Fungsi hitung\_ipk didefinisikan untuk menghitung total bobot dan total SKS berdasarkan nilai mahasiswa yang tersimpan di tabel nilai. Fungsi ini menggunakan query SQL dengan kondisi untuk mengonversi nilai ke bobot sesuai dengan rentang nilai yang ditentukan



Gambar 31.Trigger\_hitung\_IPK

Selanjutnya, trigger bernama trigger\_hitung\_ipk dibuat untuk memastikan bahwa fungsi hitung\_ipk dieksekusi setiap kali terjadi perubahan (insert, update, atau delete) pada tabel nilai

Kita lakukan pengujian untuk menguji apakah fungsi tersebut berhasil dilakukan atau tidak

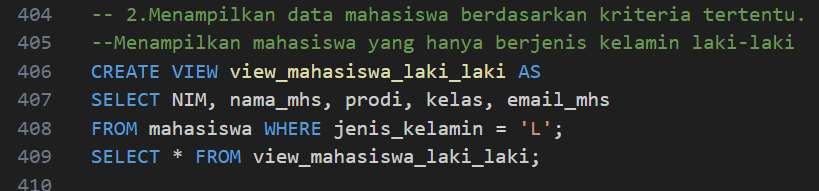


Gambar 32.Output Fungsi menghitung IPK otomatis

b. **Fitur Menampilkan data mahasiswa berdasarkan kriteria tertentu**

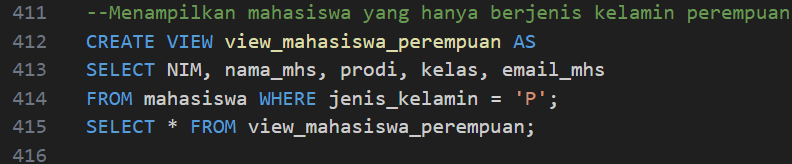
* ***Menampilkan Data Mahasiswa Berdasarkan Jenis Kelamin Laki-laki***

Fitur ini membuat dua *view* untuk mempermudah menampilkan data mahasiswa berdasarkan jenis kelamin



Gambar 33.Menampilkan Data Mahasiswa Berdasarkan Jenis Kelamin Laki-laki

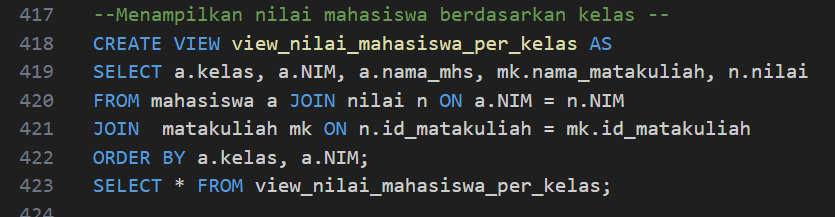
Fungsi ini membuat *view* view\_mahasiswa\_laki\_laki yang menyajikan data mahasiswa laki-laki dari tabel mahasiswa. Hanya data mahasiswa dengan kolom jenis\_kelamin bernilai 'L' yang ditampilkan.

* ***Menampilkan Mahasiswa yang Berjenis Kelamin Perempuan***

Gambar 34.Menampilkan Mahasiswa yang Berjenis Kelamin Perempuan

Fungsi ini membuat *view* view\_mahasiswa\_perempuan untuk menampilkan data mahasiswa perempuan dari tabel mahasiswa. Hanya mahasiswa dengan kolom jenis\_kelamin bernilai 'P' yang akan muncul dalam hasil *view*. View ini dirancang untuk mempermudah akses data mahasiswa berdasarkan gender perempuan.

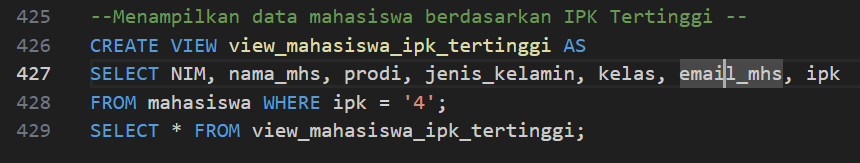
* ***Menampilkan Nilai Mahasiswa Berdasarkan Kelas***



Gambar 35.Menampilkan Nilai Mahasiswa Berdasarkan Kelas

Fungsi ini membuat *view* view\_nilai\_mahasiswa\_per\_kelas yang menggabungkan data dari tabel mahasiswa, nilai, dan matakuliah. View ini menampilkan nilai mahasiswa berdasarkan kelas mereka, Data dalam *view* ini diurutkan berdasarkan kelas dan NIM untuk mempermudah analisis nilai berdasarkan kelompok kelas tertentu.

* ***Menampilkan Data Mahasiswa Berdasarkan IPK Tertinggi***



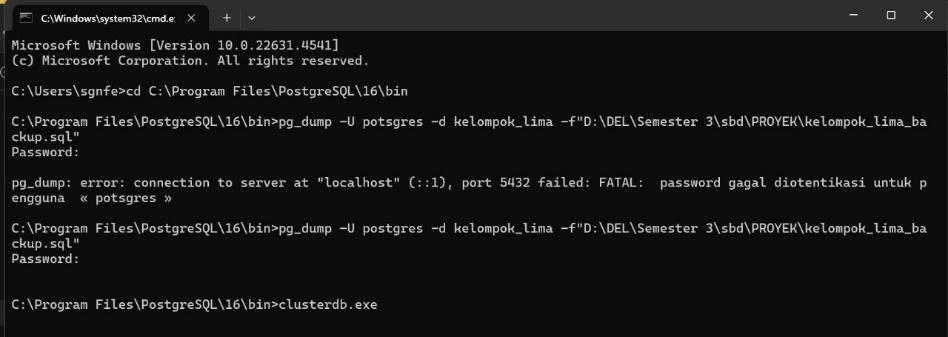
Gambar 36.Menampilkan Data Mahasiswa Berdasarkan IPK Tertinggi

Fungsi ini membuat *view* view\_mahasiswa\_ipk\_tertinggi untuk menyaring data mahasiswa dengan IPK tertinggi (bernilai 4.0). Data yang ditampilkan meliputi NIM, nama mahasiswa, program studi, jenis kelamin, kelas, email, dan IPK. View ini mempermudah identifikasi mahasiswa berprestasi tanpa perlu menulis ulang query untuk mencari mahasiswa dengan IPK maksimal. Fitur ini sangat relevan untuk keperluan penghargaan atau pelaporan prestasi akademik.

## **3.11 Backup & Restore**

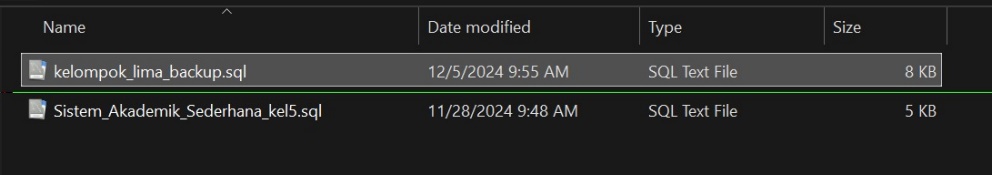
### **3.11.1. Backup**

Proses backup ini bertujuan untuk melindungi data dari kehilangan atau kerusakan yang tidak terduga, serta memastikan bahwa informasi penting tetap dapat diakses dan dipulihkan jika diperlukan.



Gambar 37.Backup

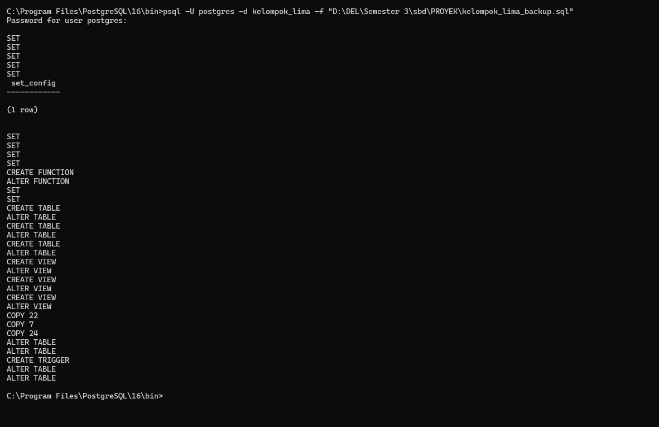
Ketika syntax tersebut dijalankan maka akan otomatis terdownload file backup dengan nama sesuai dengan yang diminta



Gambar 38.File Backup

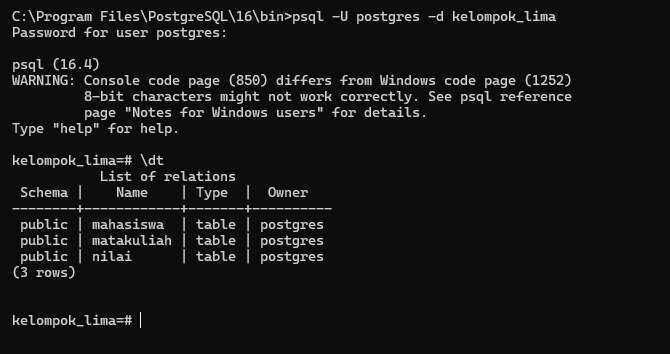
### **3.11.2. Restore**

Pada proyek ini juga dibuktikan berhasil nya backup yang dilakukan dengan melakukan penghapusan dari database tersebut dan merestore nya untuk mengembalikan data tersebut



Gambar 39.Restore

Maka setelah di lakukan restore table yang ada akan Kembali lagi

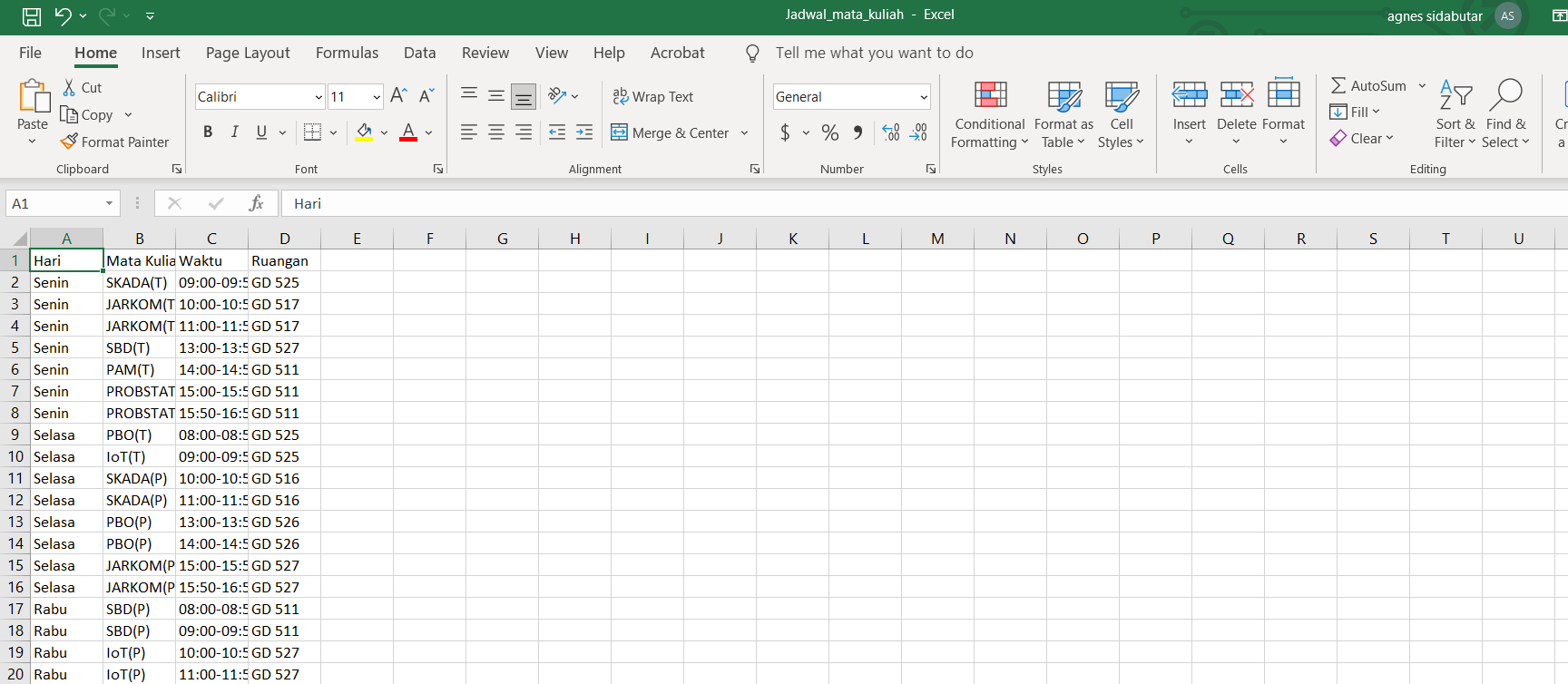
****

Gambar 40.Output dari Restore

## **3.12. Crawling dan Scraping**

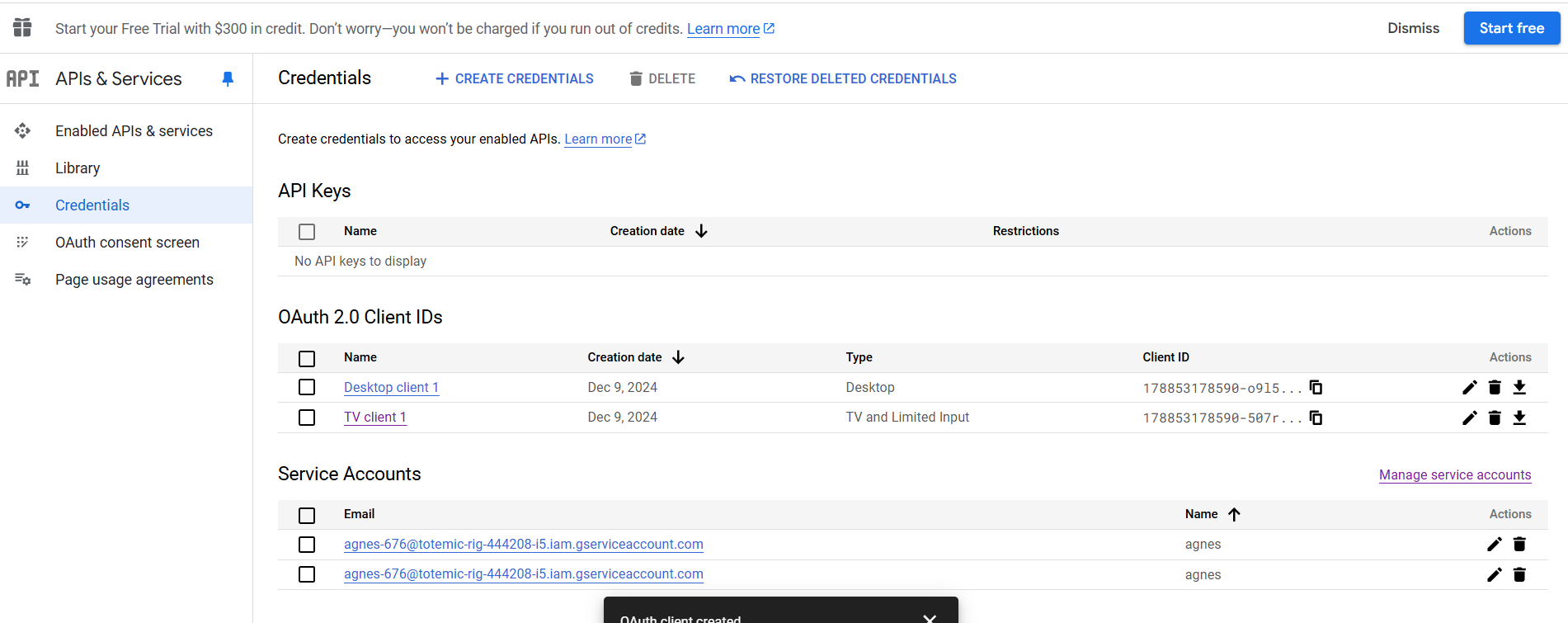
### **3.12.1. Crawling**

Pada proyek ini terdapat Crawling jadwal kelas denga mengambil data jadwal dari Google Calendar. Crawling ini dilakukan dengan sumber data (seperti Google Calendar) dengan menggunakan API yang disediakan. Adapun tahap yang dilakukan yaitu kita terlebih dahulu membuat file csv berisi jadwal matakuliah



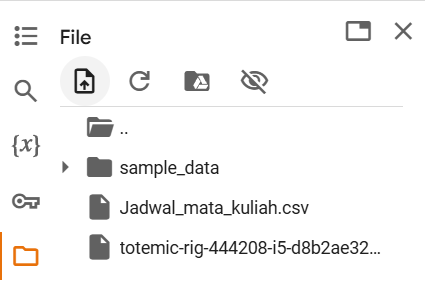
Gambar 41.CSV Jadwal Update

Selanjut nya kita akan membuat proyek di dalam google cloud dengan menerapkan google calender kemudian mendownload file autentikasi yang akan digunakan pada google colab

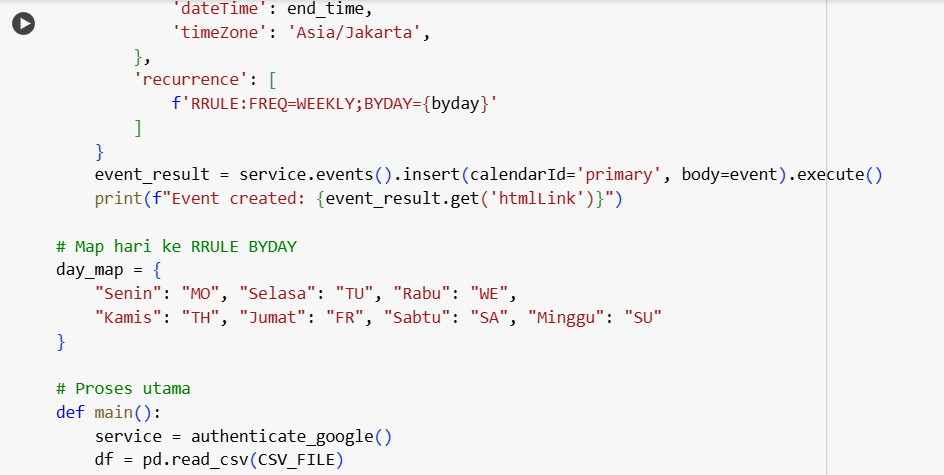
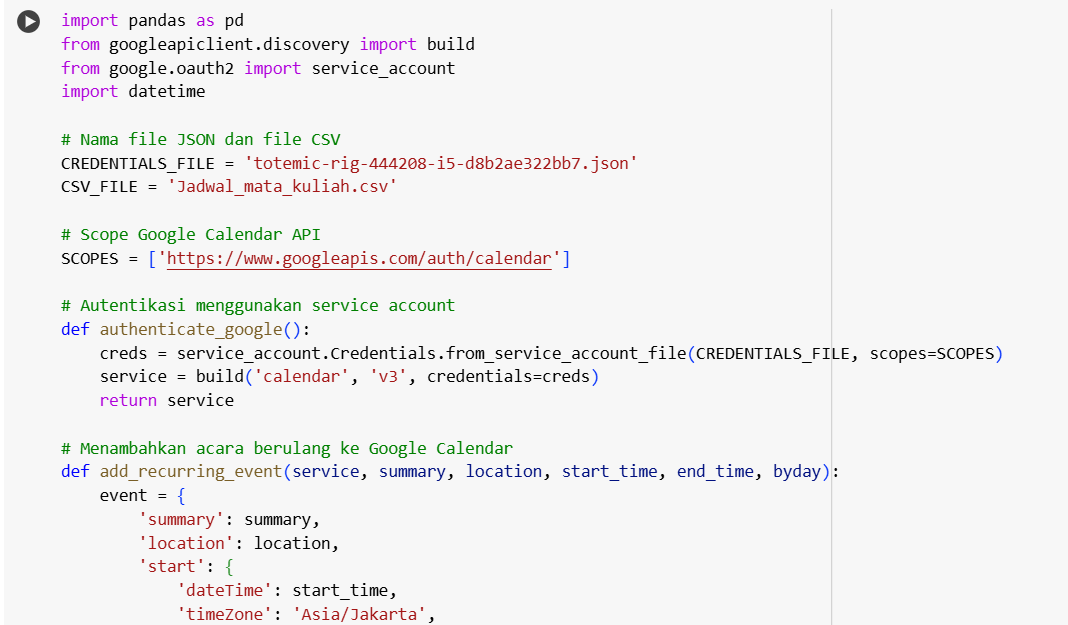


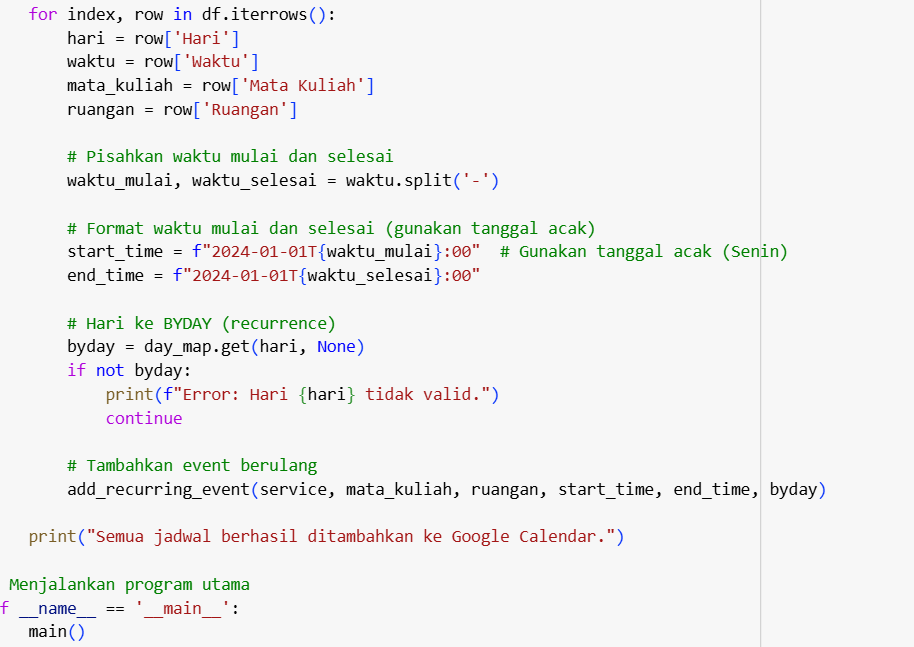
Gambar 42.Autentikasi google api

Selanjutnya pada google colab upload file csv dan file authentikasi nya yang kita gunakan untuk mengupload jadwal mata kuliah kita kedalam google kalender



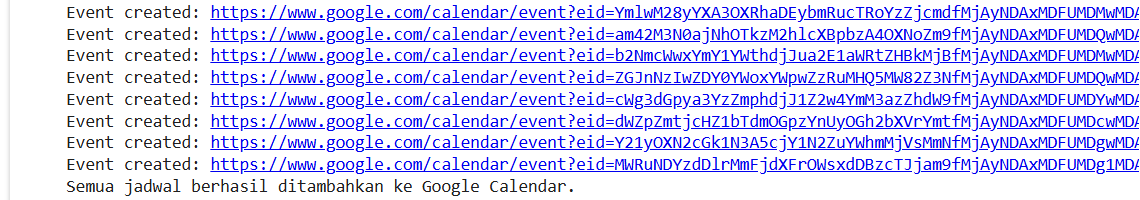
Gambar 43.Google colab upload file csv dan file authentikasi





Gambar 44.Menambahkan jadwal mata kuliah ke Google Calendar

Syntax ini digunakan untuk menambahkan jadwal mata kuliah ke Google Calendar secara otomatis menggunakan Google Calendar API. Program ini dimulai dengan mengimpor beberapa library yang diperlukan, seperti pandas untuk memanipulasi data dari file CSV, dan googleapiclient untuk mengakses Google Calendar. Setelah mendeklarasikan nama file JSON untuk autentikasi dan file CSV untuk jadwal, program menetapkan scope akses ke Google Calendar. Setelah berhasil akan muncul seperti ini,

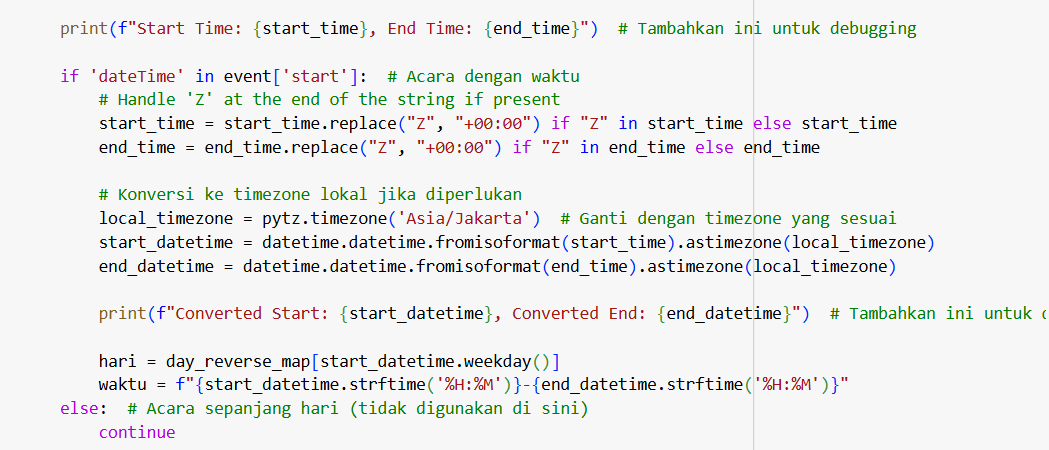


Gambar 45.CSV jadwal matakuliah berhasil di tambahkan kedalam google calender

Dimana pada gambar ini ditunjukkan bahwa semua data yang ada pada csv jadwal matakuliah berhasil di tambahkan kedalam google calender, selanjutnya kita akan melakukan crawling untuk mengambil jadwal nya tadi



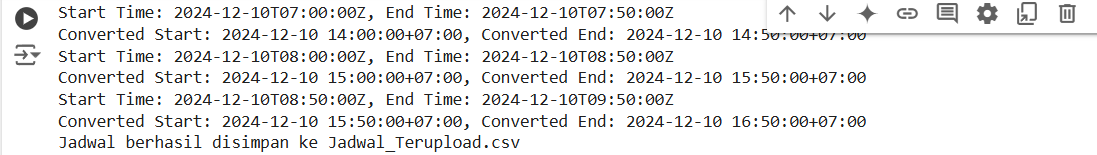






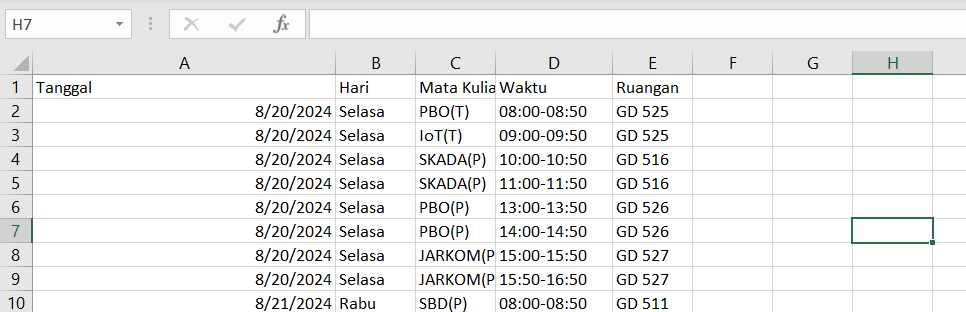
Gambar 46.Mengambil jadwal kelas dari Google Calendar menggunakan akun layanan (service account) dan menyimpannya ke dalam file CSV

Syntax diatas digunakan untuk mengambil jadwal kelas dari Google Calendar menggunakan akun layanan (service account) dan menyimpannya ke dalam file CSV. Proses ini mencakup autentikasi dengan Google Calendar API, pengambilan data acara dalam rentang waktu tertentu, serta pemrosesan dan penyimpanan data tersebut ke dalam format yang mudah dibaca. Setelah di run maka akan muncul



Gambar 47.Semua data berhasil di ambil dan disimpan

Ini berarti bahwa semua data yang ada di dalam google calender berhasil diambil dan disimpan dalam file csv baru yang langsung terdownload yang berisikan hasil gari google calender



Gambar 48.Jadwal yang sudah berhasil

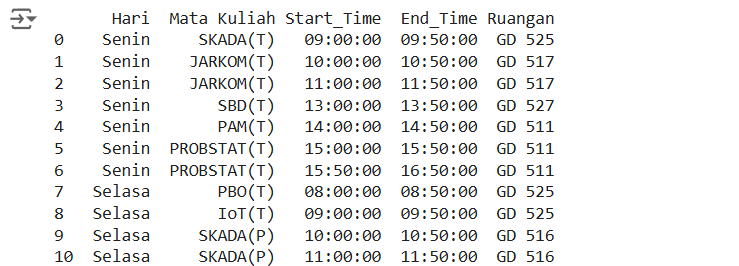
Dimana tanggal pada jadwal ini sudah mengikuti tanggal yang ada pada google calender

### **3.12.2. Scraping**



Gambar 49.Fungsi scrape\_class\_schedule

Fungsi scrape\_class\_schedule yang didefinisikan dalam ini bertujuan untuk memproses jadwal mata kuliah dari file CSV. Fungsi ini menerima satu parameter, yaitu file\_path, yang merupakan jalur menuju file CSV. akhirnya, fungsi mengembalikan DataFrame yang telah diproses ini yaitu



Gambar 50.Hasil scrape\_class\_schedule

**HASIL PRESENTASI**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Pertanyaan** | **11323010** | **11323011** | **11323005** | **11323047** |
|  | Apakah  keseluruhan  implementasi  fitur tercapai  dan relevan  dengan  tujuannya? | Implementasi fitur sudah tercapai dan relevan dengan tujuan. | Implementasi sudah tercapai dengan isi sesuai dengan apa yang telah dikerjakan pada praktikum sebelumnya. | Menurut saya sudah, karena pada proyek ini fitur" utama sudah dapat di jalankan dengan baik. | Iya sudah dilakukan implementasi dan fitur utama berhasil di jalankan. |
|  | Apa yang lebih dan apa yang  kurang untuk  diperbaiki  (kalau ada)? | Kami terlalu lebih membuat query sederhana dan kurang eksplorasi dan jumlah tabel ataupun entity dan role nya juga masih terlalu sedikit. | Pada erd ada baiknya menambah sebuah tabel baru untuk jadwal (dari hasil scrapping). | ERD yang dibuat untuk proyek kelompok 5 masih belum memenuhi, ada baiknya di tambahkan dan melakukan eksplorasi lebih banyak lagi di kemudian hari | Semakin melakukan eksplorasi lagi agar dapat mengembangkan sistem yang lebih baik lagi. |
|  | Menurutmu  apakah  kontribusimu  sudah maksimal  dan kamu  paham dengan  yang kamu  implementasika  n? | Kontribusi saya sudah maksimal dan saya sudah paham dengan yang saya buat. | Untuk kontribusi sudah bagus, untuk query yang tersedia pada proyek pun sudah terimplementasi. | Implementasi dan kontribusi saya masih kurang untuk pembelajaran ini, karena saya belum mengerti dan belum terlalu paham. | Kurang, karena masih mengimplementasi kan yang sederhana. |

**PENILAIAN**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Jenis Penilaian** | **11323010** | **11323011** | **11323005** | **11323047** |
| 1. | Hasil Kelompok | 90 | 100 | 100 | 100 |
| 2. | Diri Sendiri | 90 | 100 | 100 | 90 |