Tugas 2: Praktikum 2 dan Praktikum Mandiri 2 Machine Learning

**Aria Kristallinacht Sundanis - 0110222076**

Teknik Informatika, STT Terpadu Nurul Fikri, Depok

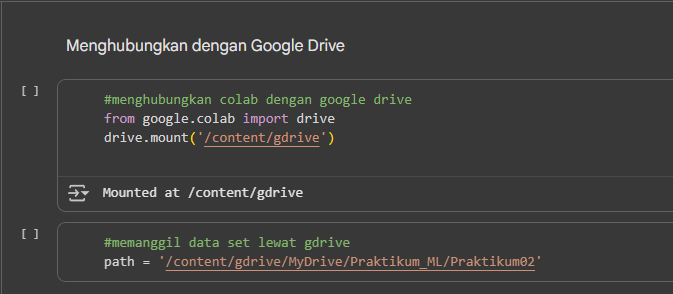
E-mail: [aria22076ti@student.nurulfikri.ac.id](mailto:aria22076ti@student.nurulfikri.ac.id)

**Praktikum 2**

Pada Praktikum kali ini kami diperkenalkan pada lingkungan pengembangan pemodelan machine learning menggunakna platform aplikasi Google Colab yang mana hal ini memungkinkan kami menulis dan mengeksekusi Python di browser, mendukung penggunaan GPU gratis, dan dapat dengan mudah dihubungkan dengan Google Drive. Hal ini mempermudah proses pengolahan data maupun penyimpanan file yang dibutuhkan selama praktikum.

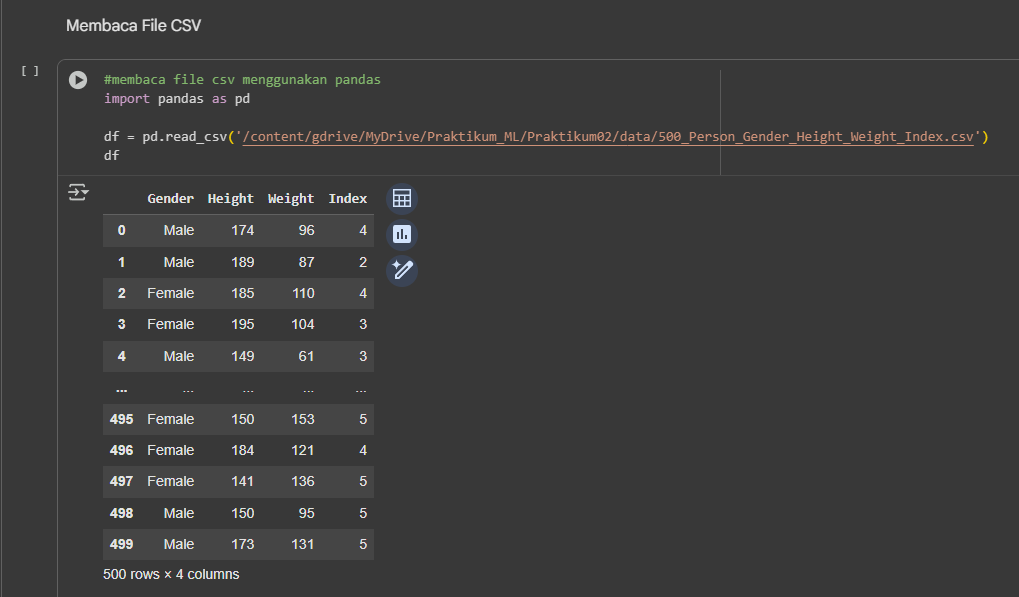
1. Tutorial Penggunaan Google Colab

* Membuat Notebook di Colab dan Menghubungkan dengan Google Drive



**Gambar 1.** Tahap pertama yang dilakukan adalah menghubungkan google colab dengan akun google drive. Tapi sebelum itu harus membuka google colab terlebih dahulu untuk membuat new notebook dan menyimpannya pada folder Praktikum\_ML/Praktikum02/notebooks/. Setelah itu jalankan kode seperti digambar, lalu akan muncul tautan otoritasi. Lalu memilih akun google untuk memberi izin akses ke google colab. Proses ini hanya dilakukan sekali setiap sesi.

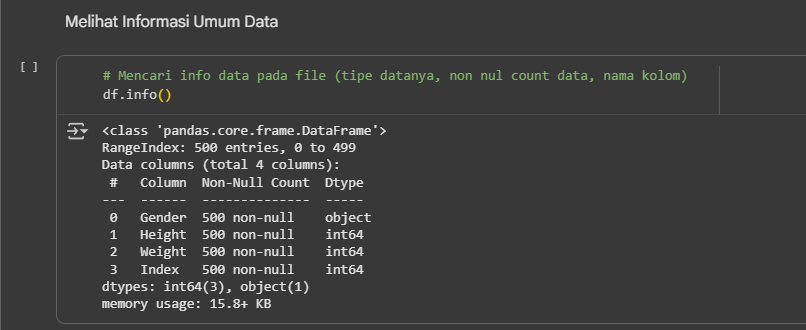
* Membaca File CSV



**Gambar 2.** Pada tahap ini menggunakan library pandas untuk membaca dataset. Lokasi file 500\_Person\_Gender\_Height\_Weight\_Index.csv yang sudah disimpan di Google Drive dipanggil melalui sebuah variabel path. Setelah itu, file tersebut dibaca dengan fungsi **pd.read\_csv()** dan hasilnya disimpan ke dalam sebuah DataFrame dengan nama **df**, sehingga data bisa langsung diolah dan dianalisis lebih lanjut. Untuk memastikan data terbaca dengan benar, kami menampilkan beberapa baris awal dengan df.head(), yang menunjukkan 4 kolom utama yaitu **Gender, Height, Weight, dan Index**.

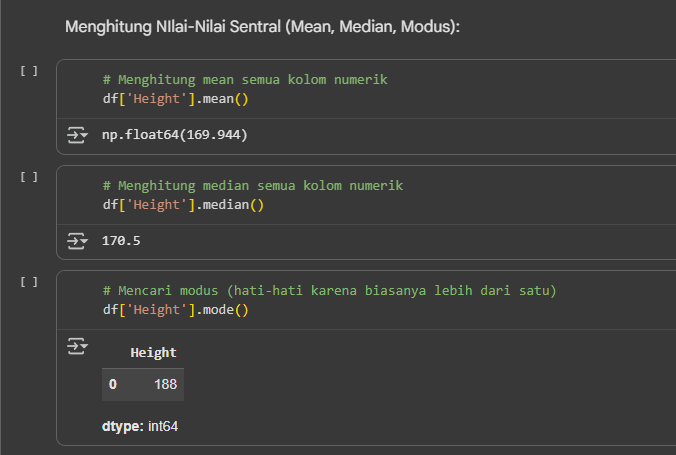
1. Analisi Statistik Deskriptif

* Melihat Informasi Umum Data



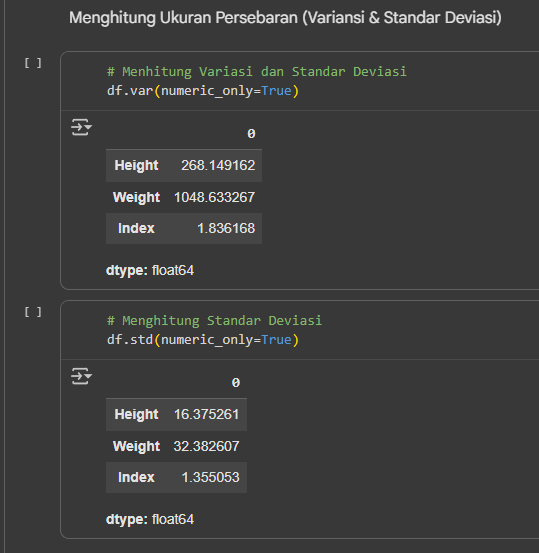
**Gambar 3.** Sebagai langkah awal analisis, menggunakan fungsi untuk melihat informasi umum dari dataset. Dari hasil yang ditampilkan, terlihat bahwa data terdiri dari 500 baris dengan 4 kolom. Selain itu, muncul juga nama setiap kolom, jumlah data yang terisi (non-null) pada masing-masing kolom, serta tipe data yang digunakan, misalnya **int64** untuk angka dan **object** untuk data berupa teks.

* Menghitung Nilai-Nilai Sentral (Mean, Median, Modus):



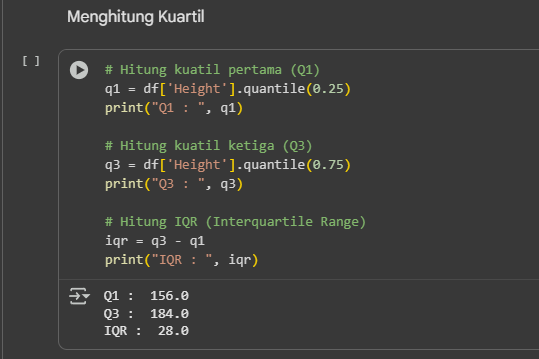
**Gambar 4.** Pada tahap ini kami menghitung nilai-nilai pusat dari data, yaitu **mean, median, dan modus**. Nilai mean memberikan rata-rata dari setiap kolom numerik, median menunjukkan angka yang berada tepat di tengah, sedangkan modus menampilkan nilai yang paling sering muncul pada data.

* Menghitung Ukuran Persebaran (Variansi & Standar Deviasi):

l

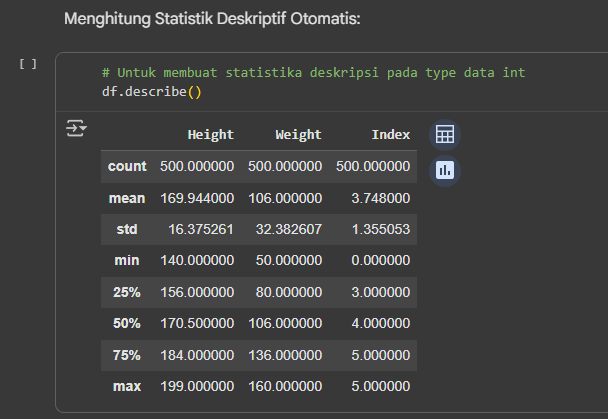
**Gambar 5.** Selanjutnya menghitung ukuran persebaran data dengan melihat nilai **variansi** dan **standar deviasi**. Variansi digunakan untuk mengetahui seberapa jauh penyebaran data terhadap nilai rata-rata, sedangkan standar deviasi menunjukkan besar penyimpangan data dalam satuan yang sama dengan data aslinya. Dengan cara ini, bisa memahami seberapa bervariasi data yang ada.

* Menghitung Kuartil



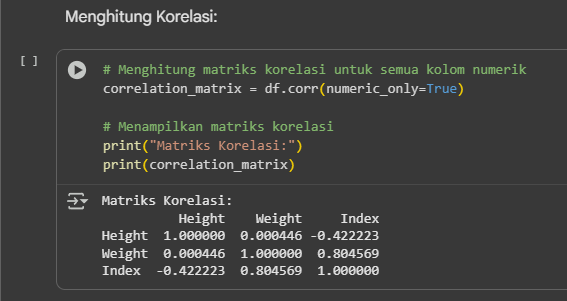
**Gambar 6.** Pada bagian ini menghitung **kuartil** untuk melihat penyebaran data lebih detail. Kuartil pertama (Q1) diperoleh dari perhitungan quantile(0.25), sedangkan kuartil ketiga (Q3) dari quantile(0.75). Selisih antara Q3 dan Q1 disebut **IQR**, yang menunjukkan rentang sebaran data di tengah.

* Menghitung Statistik Deskriptif Otomatis



**Gambar 7.** Untuk mempermudah analisis, disini menggunakan metode **.describe()** yang secara otomatis menampilkan ringkasan statistik deskriptif dari setiap kolom numerik. Dari hasilnya, bisa melihat jumlah data yang terisi, nilai rata-rata, standar deviasi, nilai minimum, hingga kuartil pertama, median, kuartil ketiga, serta nilai maksimum. Dengan cara ini, langsung mendapatkan gambaran umum mengenai distribusi data tanpa harus menghitung satu per satu.

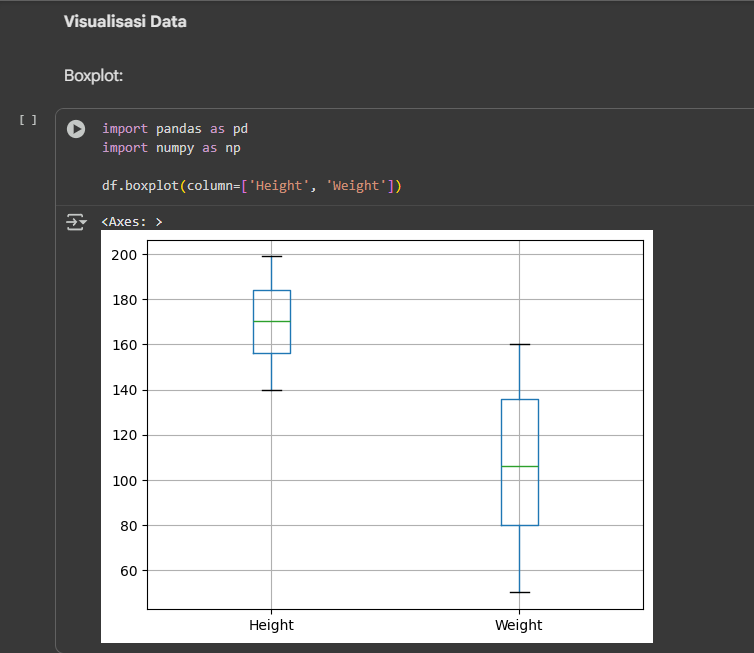
* Menghitung korelasi



**Gambar 8.** Tahap ini menghitung korelasi Pearson untuk melihat seberapa kuat hubungan antar variabel numerik di dataset. Dari hasil perhitungannya bisa diketahui apakah dua variabel bergerak searah, berlawanan, atau bahkan tidak punya hubungan linear sama sekali, dengan nilai korelasi yang berkisar antara -1 sampai 1.

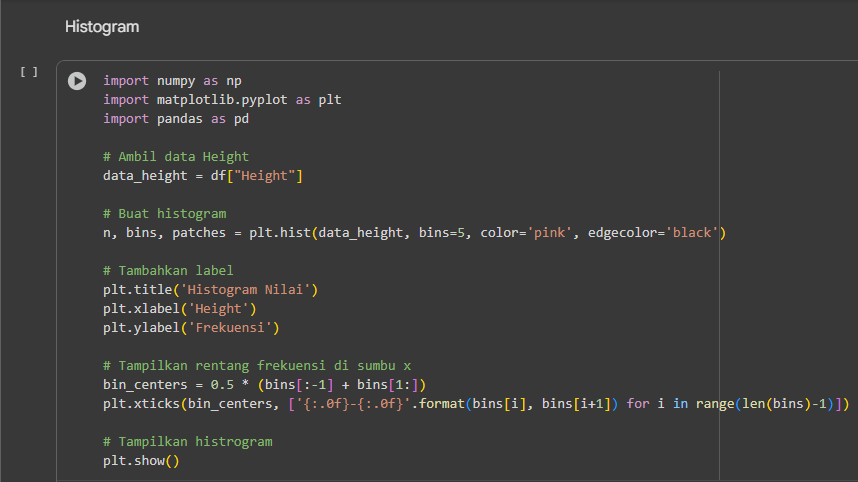
1. Visualisasi Data

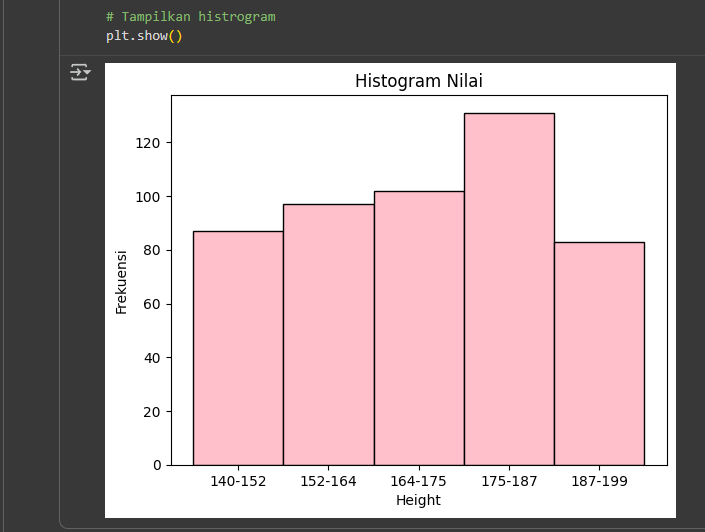
* Boxplot



**Gambar 9.** Pada tahap visualisasi ini menggunakan fungsi **df.boxplot()** untuk membuat diagram boxplot pada kolom **Height** dan **Weight**. Dari grafik yang dihasilkan, garis tengah di dalam kotak menunjukkan nilai median, sementara ukuran kotak menggambarkan rentang antar kuartil (IQR). Garis horizontal di bagian ujung atau *whiskers* menandakan batas data yang dianggap normal. Selain itu, saya juga menambahkan judul utama pada plot menggunakan fungsi **plt.suptitle()** agar hasil visualisasi lebih jelas.

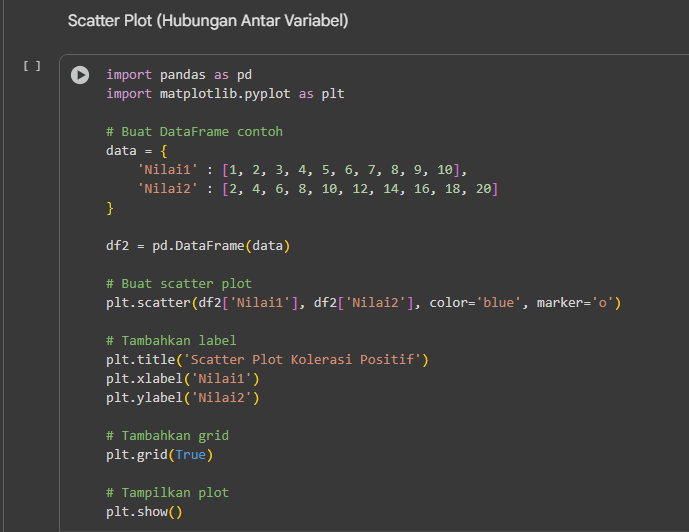
* Histogram

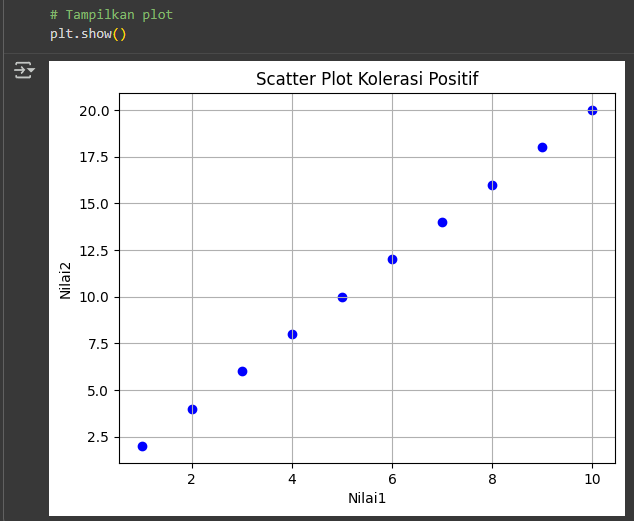


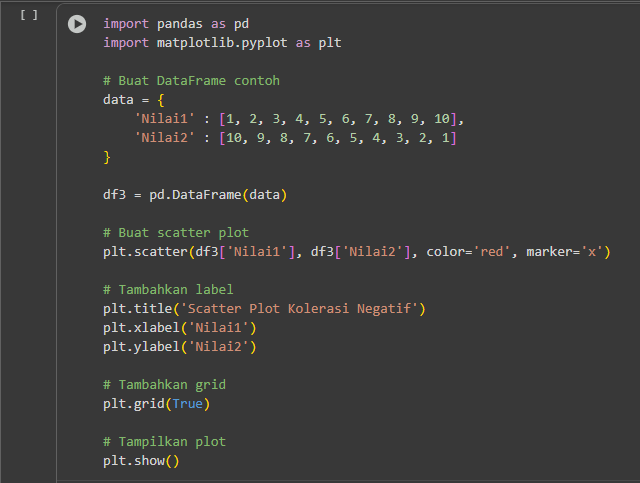


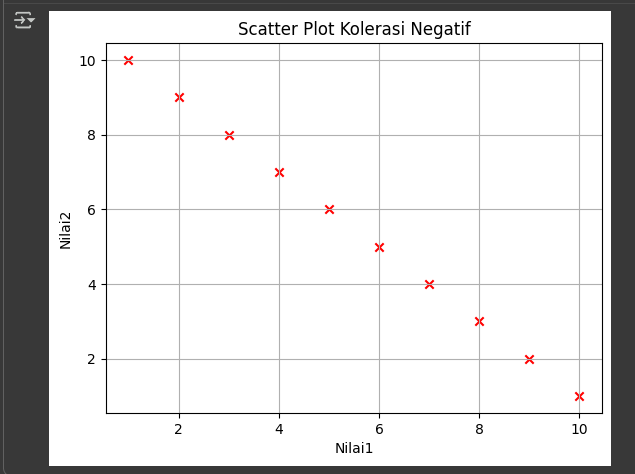
**Gambar 11 dan 12.** Pada tahap histogram disini membuat kolom untuk **Height**. Grafik histogram digunakan untuk melihat distribusi frekuensi data, sehingga terlihat seberapa banyak nilai yang muncul dalam rentang tertentu. Hasilnya berupa histogram dengan lima batang (*bins=5*), yang menunjukkan jumlah data pada setiap interval tinggi badan.

* Scatter Plot (Hubungan antar Variable)









**Gambar 13, 14, 15, dan 16.** Pada tahap scatter plot disini membuat diagram untuk melihat hubungan antara dua variabel numerik. Disini membuat dua plot terpisah, di mana plot pertama menggunakan data Nilai1 dan Nilai2 yang menunjukkan korelasi positif, sedangkan plot kedua menggambarkan korelasi negatif antara kedua variabel.

Lalu hasil dari visualisasi scatter plot tersebut Adalah terlihat dua pola yang berbeda. Pada plot dengan korelasi positif, titik-titik data membentuk garis yang cenderung naik, menandakan bahwa ketika nilai satu variabel meningkat, variabel lainnya juga ikut meningkat. Sebaliknya, pada plot dengan korelasi negatif, pola titik justru menurun, sehingga dapat disimpulkan bahwa kenaikan pada satu variabel diikuti oleh penurunan pada variabel lainnya.

**Praktikum Mandiri 2**

Buat program untuk membagi dataset day.csv menjadi tiga bagian, yaitu:

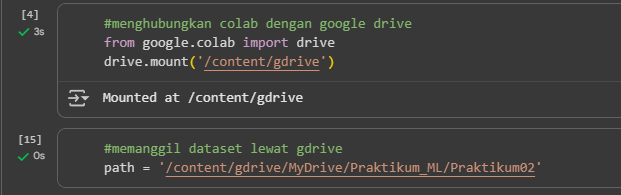
(a) Data Training: 80% dari total dataset

(b) Data Validation: 10% dari data training

(c) Data Testing: 20% dari total dataset

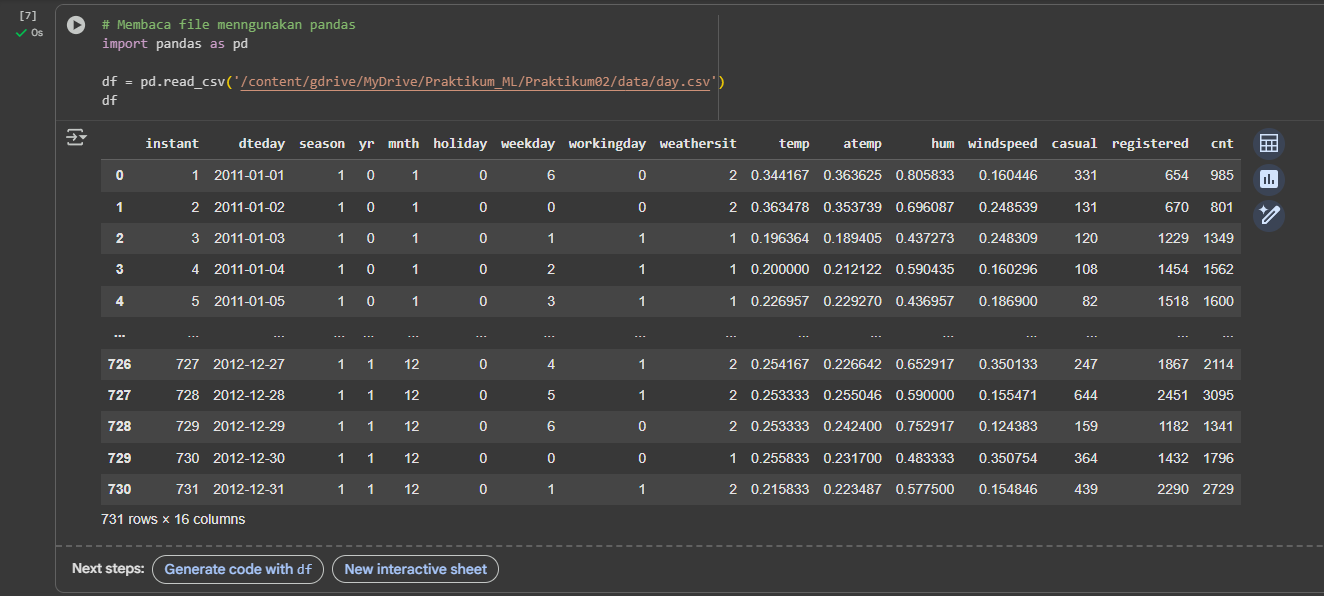
Tampilkan jumlah data dan 5 baris data teratas untuk setiap set (Training, Validation, dan Testing) sebagai bukti pengerjaa

1. Menghubungkan Google Drive ke Google Colab



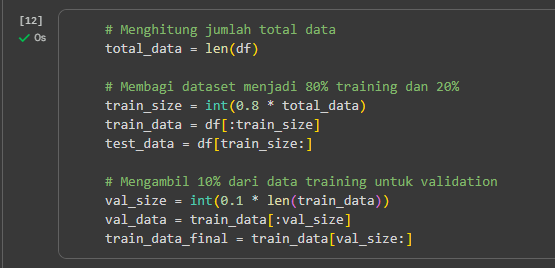
**Gambar 1.** Tahap pertama yang dilakukan adalah menghubungkan google colab dengan akun google drive. Setelah itu jalankan kode seperti digambar, lalu akan muncul tautan otoritasi. Lalu memilih akun google untuk memberi izin akses ke google colab. Proses ini hanya dilakukan sekali setiap sesi.

1. Membaca File CSV



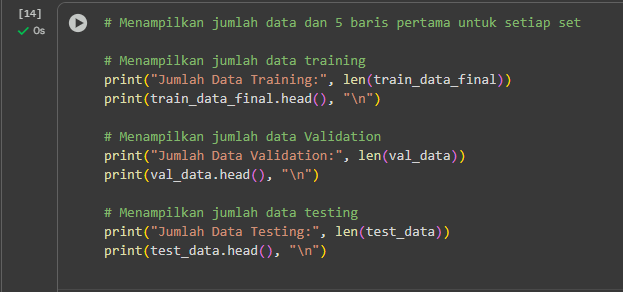
**Gambar 2.** Pada tahap ini menggunakan library pandas untuk membaca dataset. Lokasi day.csv yang sudah disimpan di Google Drive dipanggil melalui sebuah variabel path. Setelah itu, file tersebut dibaca dengan fungsi **pd.read\_csv()** dan hasilnya disimpan ke dalam sebuah DataFrame dengan nama **df**, sehingga data bisa langsung diolah dan dianalisis lebih lanjut. Secara keseluruhan, data ini memiliki **731 baris** dan **16 kolom** (variabel informasi), yang merupakan langkah awal dalam proyek **analisis atau prediksi data**.

1. Membagi Data Menjadi Training (80%) dan Testing (20%)



**Gambar 3.** Langkah berikutnya adalah membagi dataset menjadi beberapa bagian. Tahap pertama yaitu membagi dataset menggunakan kode **df[:train\_size]** yang artinya kode ini mengambil baris dari awal sampai batas 80%untuk training dan **df[train\_size:]** mengambil sisanya untuk testing yaitu 20%. Tapi dari data training yang didapat disini tidak dipakai semuanya. Data harus dibagi lagi untuk validation, dengan kode **val\_size** yang menunjukan jumlah baris yang diambil lalu kode **train\_data[:val\_size]** untuk mengambil sebagian kecil di awal data training sebagai validation, sementara sisanya **(train\_data[val\_size:])** dipakai sebagai data training final. Dengan begitu, saya memiliki tiga bagian dataset: training, validation, dan testing.

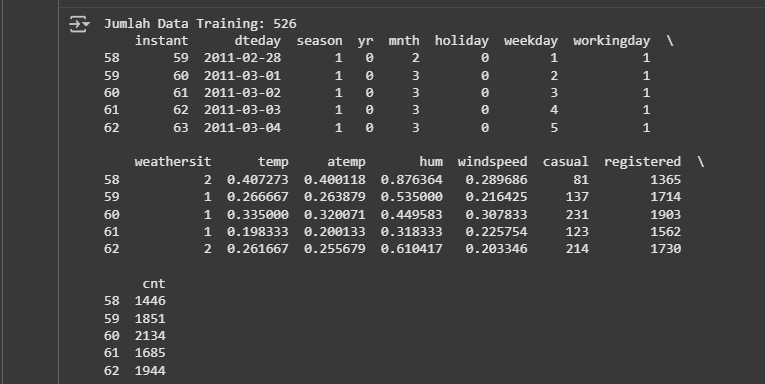
1. Menampilkan Informasi Data



**Gambar 4.** Langkah terakhir adalah menampilkan jumlah data dan beberapa baris pertama dari tiap set untuk memastikan pembagian sudah sesuai. Perintah **len()** dipakai untuk menghitung jumlah baris, sedangkan **head()** digunakan untuk menampilkan lima baris pertama sebagai sampel.

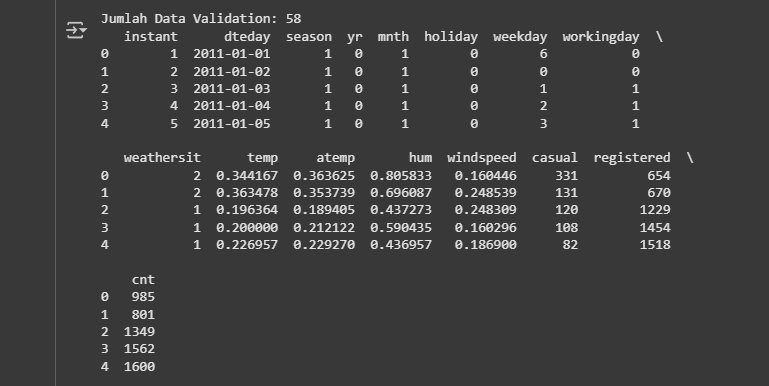
1. Hasil Jumlah data dari setiap set

* Data Training



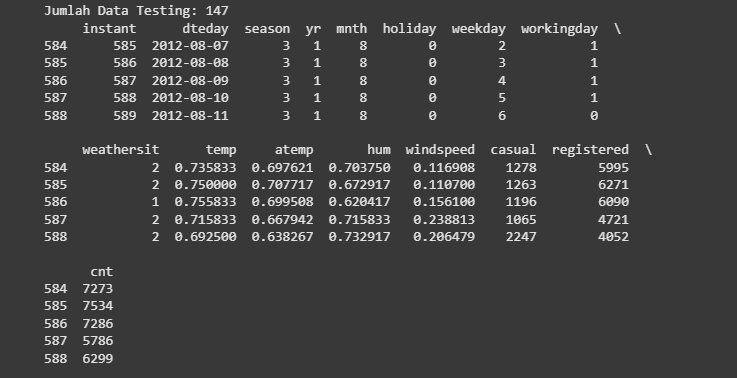
**Gambar 4.** Pada bagian data training, jumlah baris yang didapat adalah 526 baris. Data ini dibuat paling berast porsinya karena memang digunakan sebagai bahan utama untuk melatih model. Model akan mempelajari data tersebut untuk menemukan pola.

* Data Validation



**Gambar 4.** Pada bagian data validation, jumlah baris yang didapat hanya Sebagian kecil yaitu 58 baris. Hal ini karena data validation berfungsi sebagai alat evaluasi sementara. Agar bisa mencek bagaimana performa model selama ptoses training.

* Data Testing



**Gambar 5.** Pada bagian data testing terdiri dari 147 baris. Data ini sengaja dipisahkan dari awal dan tidak digunakan sama sekali selama proses training maupun validation. Karena data ini akan digunakan untuk menguji performa model pada data yang belum pernah dilihat sebelumnya.

Link GitHub :