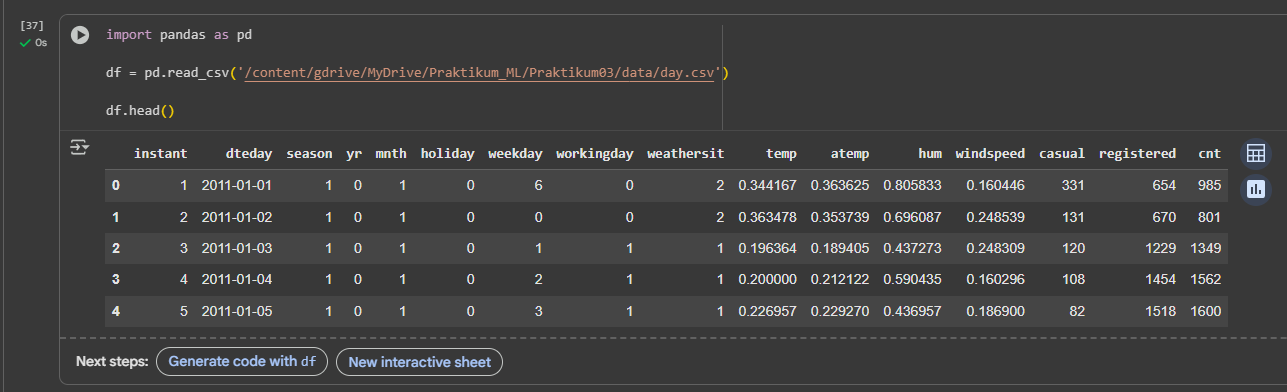
**Tugas 3: Praktikum Mandiri 3 Machine Learning**

**Aria Kristallinacht Sundanis - 0110222076**

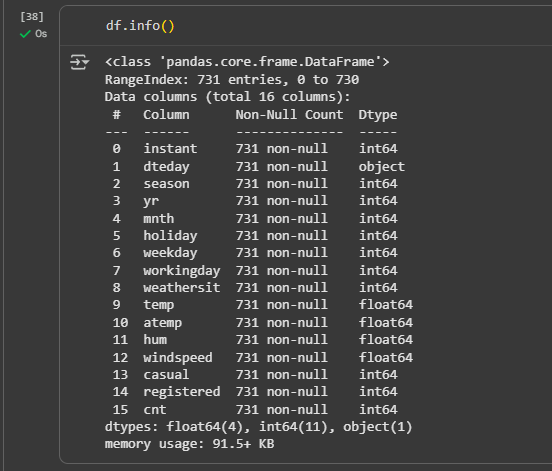
Teknik Informatika, STT Terpadu Nurul Fikri, Depok

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

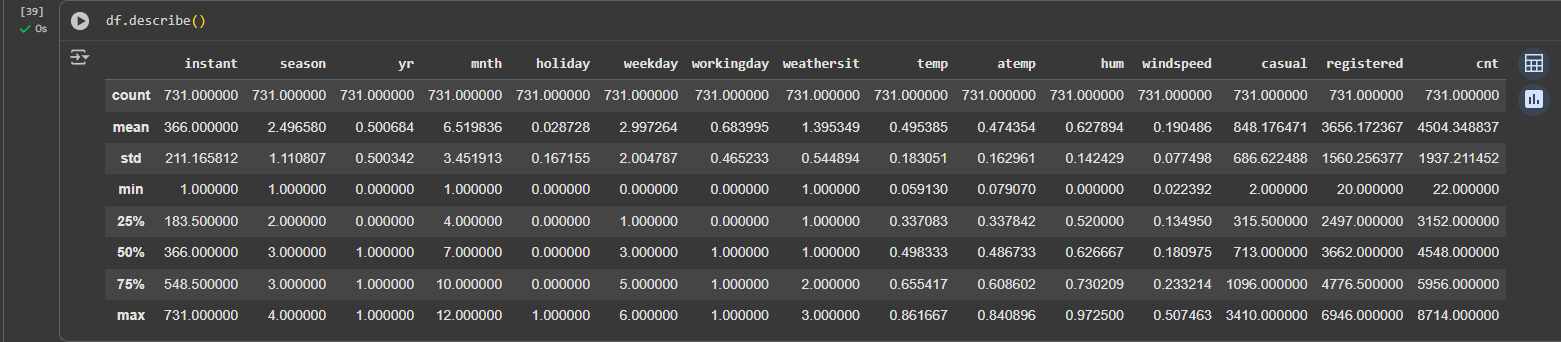
**Praktikum Mandiri 3**



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**, untuk memastikan data berhasil dimuat, digunakan perintah df.head() yang menampilkan lima baris pertama dari dataset. Secara keseluruhan, dataset ini berisi 731 baris dan 16 kolom yang merepresentasikan data penyewaan sepeda.

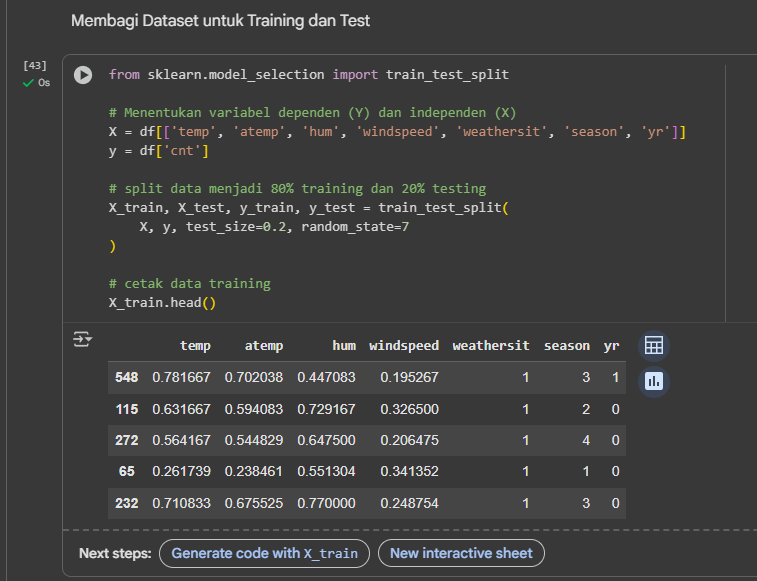


Perintah df.info() digunakan untuk menampilkan informasi umum mengenai struktur dataset. Dari hasil tersebut terlihat bahwa dataset memiliki 731 baris dan 16 kolom, dengan semua kolom berisi data lengkap, tidak ada missing value. Tipe data yang digunakan terdiri dari int64, float64, dan object (untuk kolom tanggal).

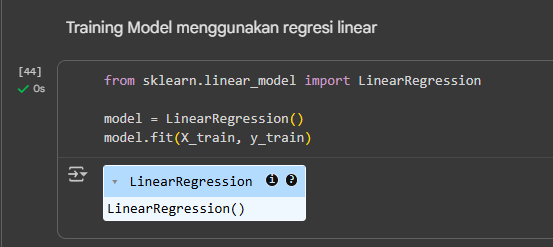


Perintah df.describe() digunakan untuk menampilkan statistik deskriptif dari setiap kolom numerik dalam dataset. Dari hasilnya terlihat bahwa:

* Dataset memiliki 731 data pada tiap kolom.
* Nilai rata-rata penyewaan sepeda (cnt) adalah sekitar 4504 dengan nilai maksimum 8714.
* Kolom temp, atemp, hum, dan windspeed menunjukkan nilai rata-rata kondisi cuaca harian.
* Statistik seperti mean, min, max, dan quartile (25%, 50%, 75%) membantu memahami sebaran serta rentang nilai dari masing-masing variabel sebelum dilakukan analisis lebih lanjut.



Pada tahap ini digunakan fungsi train\_test\_split dari library sklearn.model\_selection untuk membagi dataset menjadi dua bagian, yaitu data training dan data testing. Variabel X diisi dengan beberapa kolom fitur seperti temp, atemp, hum, windspeed, weathersit, season, dan yr, sedangkan variabel y berisi kolom cnt yang merupakan jumlah penyewaan sepeda dan menjadi target yang ingin diprediksi. Proses pembagian data dilakukan dengan proporsi 80% untuk data training dan 20% untuk data testing menggunakan random\_state=7 agar hasil pembagian tetap konsisten. Setelah itu, ditampilkan beberapa baris awal dari data training menggunakan X\_train.head() untuk memastikan data sudah terbagi dengan benar.



Selanjutnya melatih model dengan menggunakan regresi linear. Untuk menerapkan regresi linear kita perlu menggunakan LinearRegression milik scikit-learn. Lalu mengiput data train milik variabel dependen dan independen.



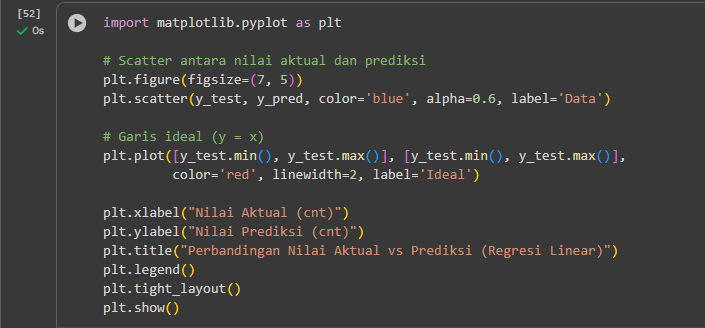
Pada tahap ini dilakukan proses pembuatan dan evaluasi model regresi linear menggunakan library scikit-learn. Pertama, model LinearRegression() diinisialisasi dan dilatih menggunakan data training (X\_train, y\_train) agar model bisa mempelajari hubungan antara variabel independen (seperti suhu, kelembapan, kecepatan angin, dan lainnya) dengan variabel target yaitu cnt (jumlah penyewaan sepeda). Setelah model selesai dilatih, data testing digunakan untuk menguji kemampuan model dalam memprediksi nilai cnt melalui model.predict(X\_test).

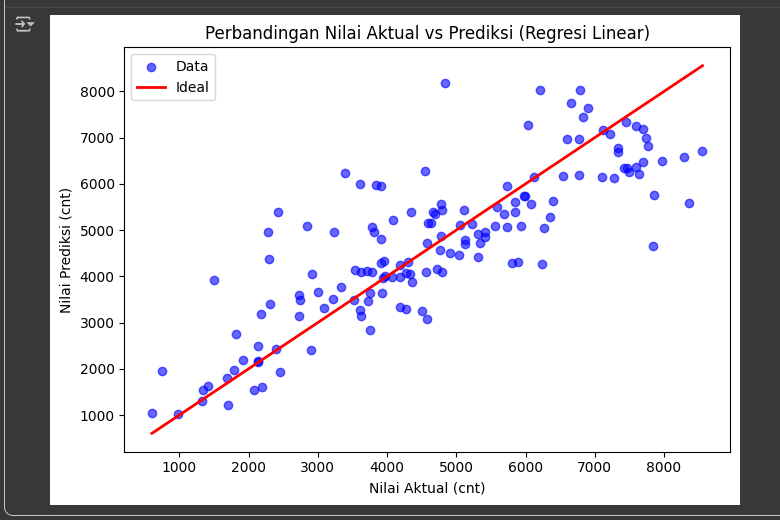
Kemudian, beberapa metrik evaluasi dihitung:

* **R²**: seberapa bagus model menjelaskan data.
* **MAE**: rata-rata selisih absolut antara prediksi dan data asli.
* **MSE:** rata-rata dari kuadrat selisih antara nilai aktual dan prediksi
* **RMSE**: seberapa jauh hasil prediksi dari nilai sebenarnya.

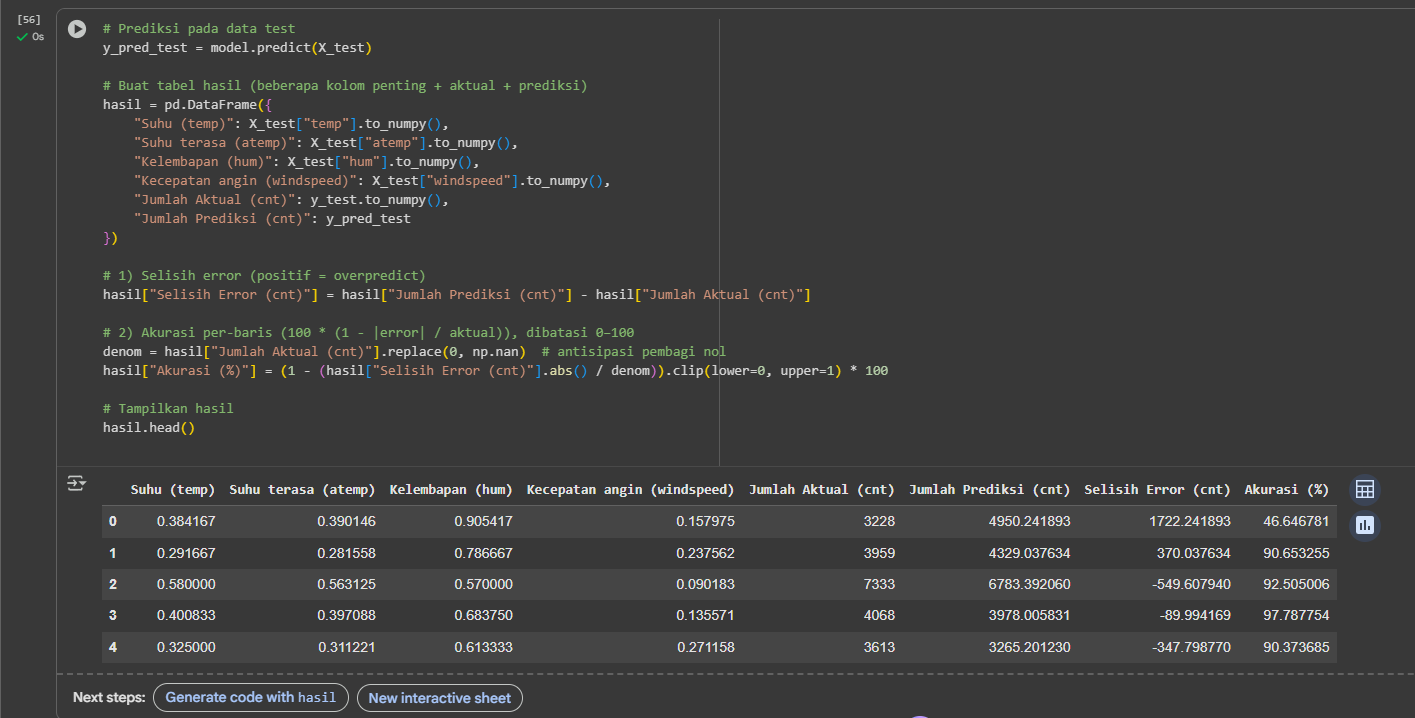


Kode di atas digunakan untuk menampilkan persamaan regresi linear berganda berdasarkan hasil pelatihan model. Pertama, nilai intercept diambil dari model.intercept\_, yang menunjukkan nilai dasar cnt ketika semua variabel X bernilai nol. Lalu, nilai koefisien dari masing-masing variabel independen diambil menggunakan model.coef\_, di mana setiap koefisien menunjukkan seberapa besar pengaruh variabel tersebut terhadap jumlah penyewaan sepeda (cnt). Setelah itu, kode membuat string persamaan dimulai dari nilai intercept, lalu menambahkan setiap variabel dan koefisiennya satu per satu melalui perulangan for. Hasil akhirnya adalah persamaan regresi berganda yang ditampilkan dalam format yang mudah dibaca, misalnya: cnt = 120.345 + (85.231 × temp) + (42.876 × hum) + ..., yang menggambarkan hubungan antara variabel cuaca dan jumlah penyewaan sepeda.





Kode di atas digunakan untuk membuat visualisasi perbandingan antara nilai aktual dan nilai prediksi dari model regresi linear. Pertama, fungsi plt.scatter() digunakan untuk menampilkan sebaran titik antara y\_test (nilai aktual) dan y\_pred (nilai hasil prediksi), di mana titik-titik berwarna biru menunjukkan seberapa jauh hasil prediksi dari nilai aslinya. Lalu, ditambahkan garis merah menggunakan plt.plot() yang merepresentasikan garis ideal y = x, yaitu kondisi di mana prediksi model benar-benar sama dengan nilai aktual. Jika titik-titik biru berada dekat dengan garis merah, berarti model memiliki akurasi yang baik. Terakhir, diberi label sumbu, judul, dan legenda agar grafik lebih mudah dibaca, lalu ditampilkan dengan plt.show().



Kode ini digunakan untuk menampilkan hasil prediksi model regresi dalam bentuk tabel yang membandingkan antara nilai aktual dan nilai prediksi dari variabel cnt. Pertama, model melakukan prediksi terhadap data uji dengan model.predict(X\_test) dan hasilnya disimpan ke variabel y\_pred\_test. Lalu dibuat sebuah DataFrame bernama hasil yang berisi beberapa kolom penting, seperti suhu (temp), suhu terasa (atemp), kelembapan (hum), kecepatan angin (windspeed), serta nilai aktual dan nilai prediksi dari jumlah penyewaan sepeda (cnt). Setelah itu, dihitung selisih error antara hasil prediksi dan nilai sebenarnya untuk melihat seberapa jauh prediksi model dari kenyataan. Kemudian ditambahkan kolom “Akurasi (%)” yang dihitung dengan rumus (1 - |error| / aktual) × 100, di mana nilainya dibatasi antara 0 sampai 100 agar tidak berlebihan. Terakhir, hasil.head() digunakan untuk menampilkan lima baris pertama dari tabel hasil perbandingan tersebut.

Link GitHub :