Tugas Mandiri 3 : Analisis Regresi Linear Pada Dataset Bike Sharing

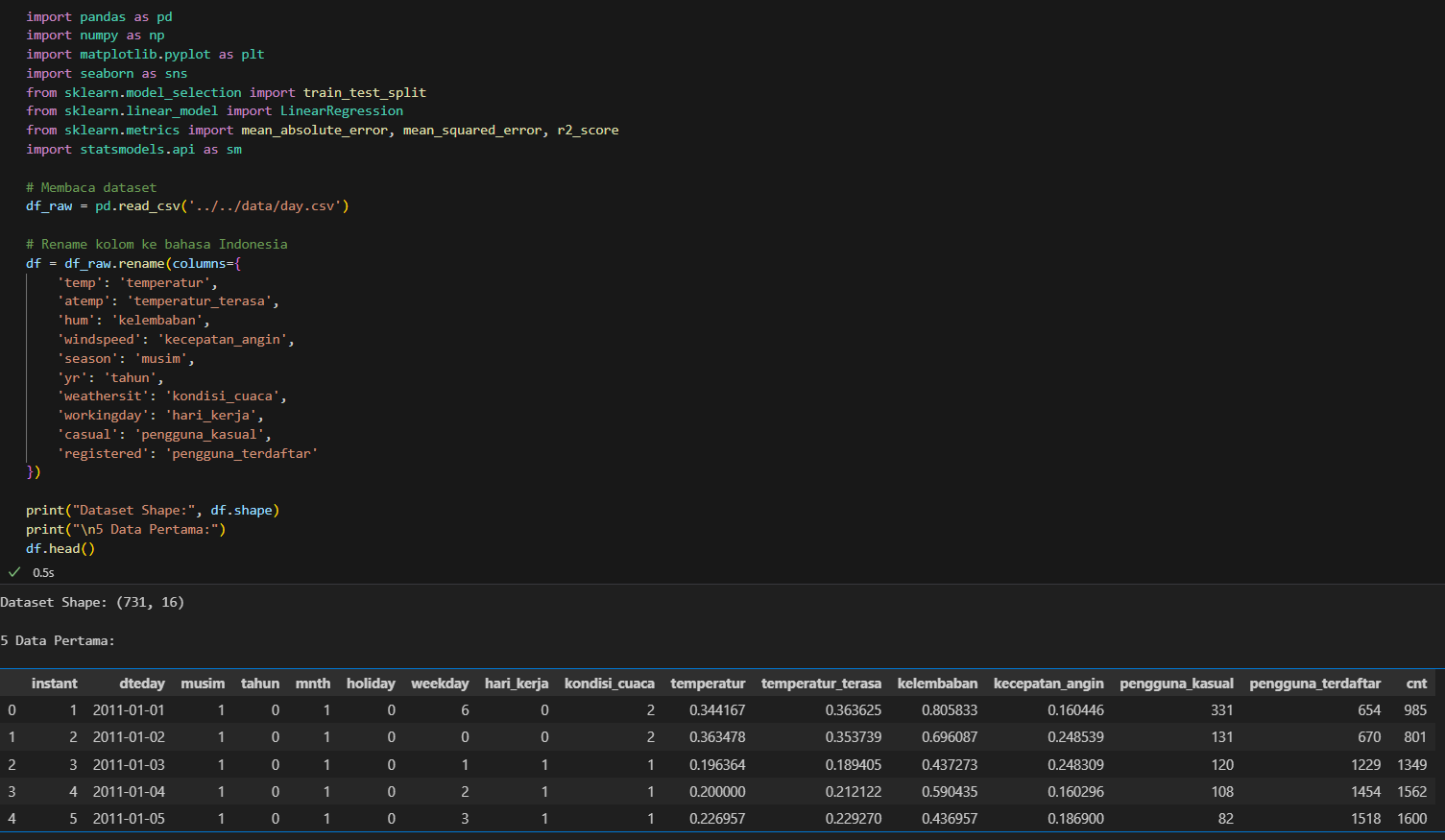
**Raffa Yuda Pratama - 0110224081**

Teknik Informatika, STT Terpadu Nurul Fikri, Depok E-mail: [0110224081@student.nurulfikri.ac.id](mailto:0110224081@student.nurulfikri.ac.id)

**Abstract.** Laporan ini menerapkan regresi linear sederhana dan berganda pada dataset Bike Sharing (day.csv) untuk memprediksi jumlah penyewaan (cnt). Pada regresi sederhana digunakan satu fitur (temperatur), sedangkan pada regresi berganda digunakan delapan fitur (temperatur, temperatur\_terasa, kelembaban, kecepatan\_angin, musim, tahun, kondisi\_cuaca, hari\_kerja). Evaluasi dilakukan dengan metrik R², MAE, MSE, dan RMSE. Hasil menunjukkan model regresi berganda memberikan R² lebih tinggi dan error lebih rendah dibanding regresi sederhana, sehingga lebih cocok untuk tugas prediksi sederhana ini. Nama kolom dalam notebook juga telah diperbarui ke bahasa Indonesia untuk memudahkan pembacaan.

# Persiapan Data dan Analisis

# Import Library dan Load Dataset

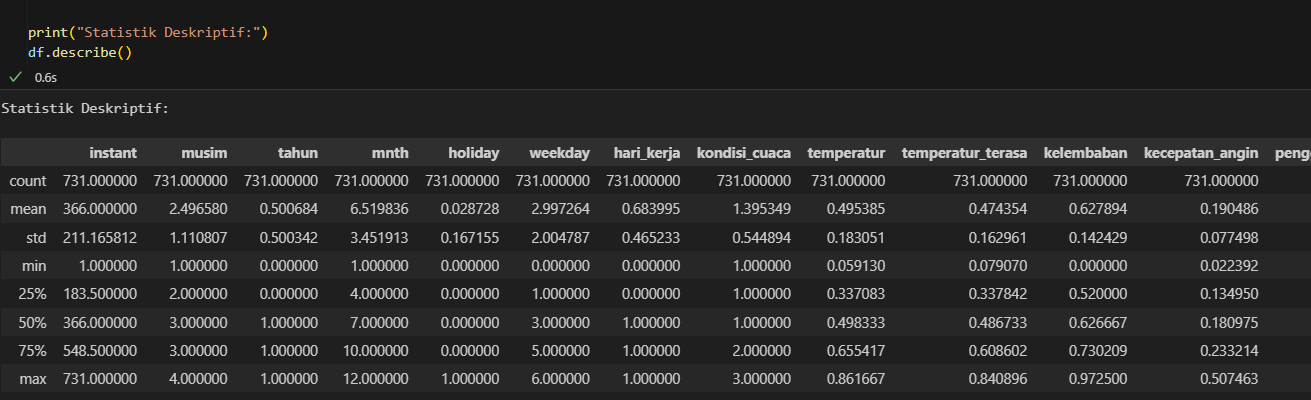
****

Gambar 1 Import Library dan Load Dataset

Bagian ini merupakan tahap persiapan awal dimana semua library yang diperlukan diimport dan dataset bike sharing dibaca dari file CSV dengan melakukan rename kolom ke bahasa Indonesia.

* **Import pandas dan numpy**: Library untuk manipulasi data dan operasi numerik
* **Import matplotlib dan seaborn**: Library untuk visualisasi data dalam bentuk grafik
* **Import sklearn modules**: Untuk split data, membuat model regresi, dan menghitung metrik evaluasi
* **Import statsmodels**: Untuk analisis regresi yang lebih detail dengan output statistik lengkap
* **Membaca dataset**: Menggunakan pd.read\_csv() untuk load file day.csv ke dalam df\_raw
* **Rename kolom**: Menggunakan .rename() untuk mengubah nama kolom ke bahasa Indonesia
  + temp → temperatur
  + atemp → temperatur\_terasa
  + hum → kelembaban
  + windspeed → kecepatan\_angin
  + season → musim
  + yr → tahun
  + weathersit → kondisi\_cuaca
  + workingday → hari\_kerja
  + casual → pengguna\_kasual
  + registered → pengguna\_terdaftar
* **Menampilkan shape**: Melihat dimensi dataset (731 baris dan jumlah kolom)
* **Menampilkan 5 data pertama**: Menggunakan df.head() untuk preview struktur dan isi dataset

## Eksplorasi Data dan Statistik Deskriptif

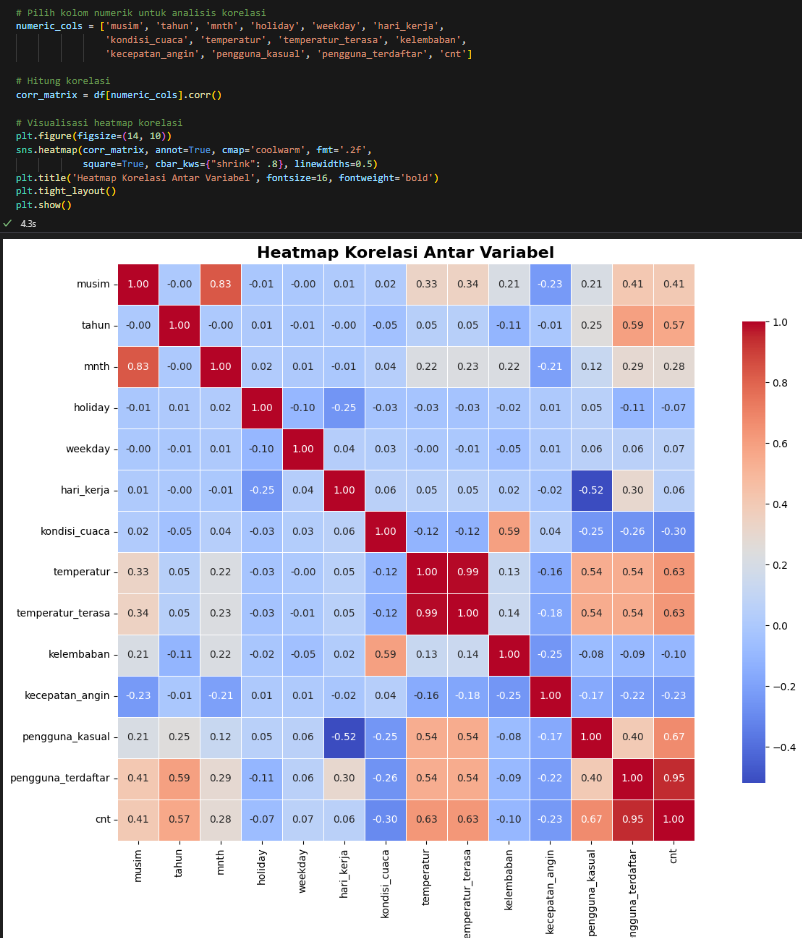


Gambar 2 Eksplorasi Data dan Statistik Deskriptif

Tahap ini melakukan eksplorasi awal untuk memahami karakteristik dan struktur dataset sebelum melakukan pemodelan.

* **df.describe()**: Menghitung statistik deskriptif (mean, std, min, max, quartiles) untuk semua kolom numerik
* **Identifikasi missing values**: Mengecek apakah ada data yang hilang atau null
* **Memahami rentang data**: Melihat nilai minimum dan maksimum setiap variabel
* **Mendeteksi potensi outlier**: Dari nilai min/max dan quartiles dapat mengidentifikasi data yang tidak normal

## Analisis Korelasi



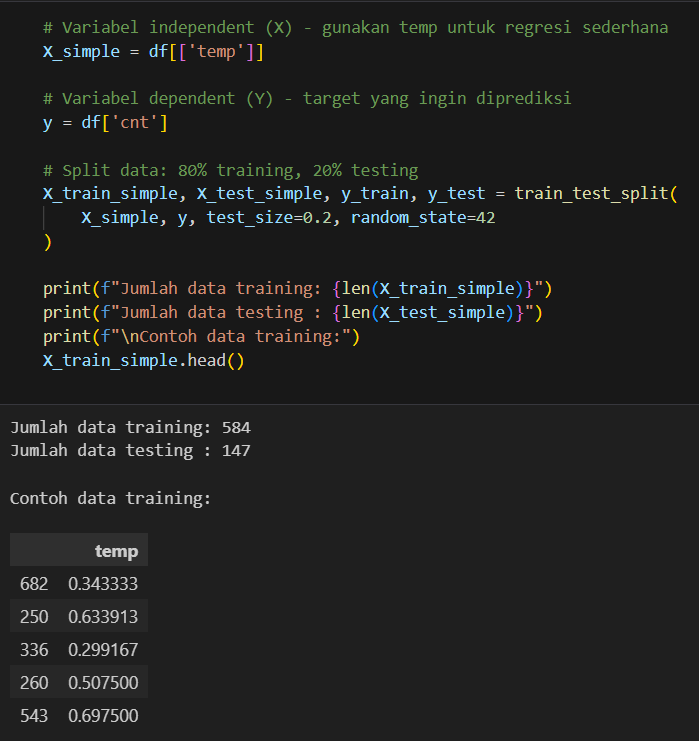
Gambar 3 Load dan Preview Data

Analisis korelasi sangat penting untuk menentukan variabel mana yang memiliki hubungan kuat dengan target prediksi (cnt).

* **Mendefinisikan kolom numerik**: Membuat list berisi nama-nama kolom yang bertipe numerik untuk analisis
* **Menghitung correlation matrix**: Menggunakan df.corr() untuk menghitung korelasi pearson antar semua variabel
* **Sorting korelasi dengan cnt**: Mengurutkan nilai korelasi dari tertinggi ke terendah untuk melihat variabel paling berpengaruh
* **Membuat heatmap**: Visualisasi matriks korelasi dengan warna, dimana merah = korelasi positif tinggi, biru = korelasi negatif
* **Set figure size**: Membuat ukuran plot 14x10 inch agar semua variabel terlihat jelas
* **Menambahkan anotasi**: Parameter annot=True menampilkan nilai korelasi di setiap cell
* **Interpretasi hasil**: Registered (0.95) dan casual (0.69) berkorelasi tinggi tapi tidak digunakan karena data leakage

# Regresi Linear Sederhana

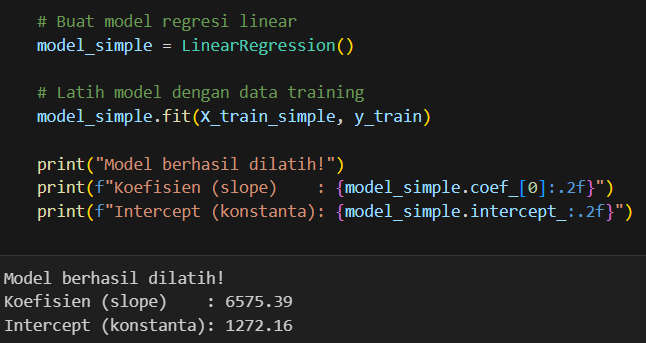
* 1. **Split Data untuk Regresi Linear Sederhana**



Persiapan data untuk model regresi linear sederhana dengan memilih satu variabel independent (temp) yang berkorelasi tinggi dengan target.

* **Mendefinisikan X\_simple**: Memilih kolom 'temp' sebagai variabel independent, ditulis dalam double bracket untuk mempertahankan format DataFrame
* **Mendefinisikan y**: Memilih kolom 'cnt' sebagai variabel dependent/target yang ingin diprediksi
* **train\_test\_split()**: Membagi dataset menjadi 80% training dan 20% testing
* **test\_size=0.2**: Parameter yang menentukan proporsi data testing sebesar 20%
* **random\_state=42**: Seed untuk memastikan hasil split konsisten setiap kali dijalankan
* **Print jumlah data**: Menampilkan jumlah data training dan testing untuk verifikasi pembagian
* **Menampilkan sample**: Menggunakan .head() untuk melihat contoh data training

## Pemodelan Regresi Linear Sederhana

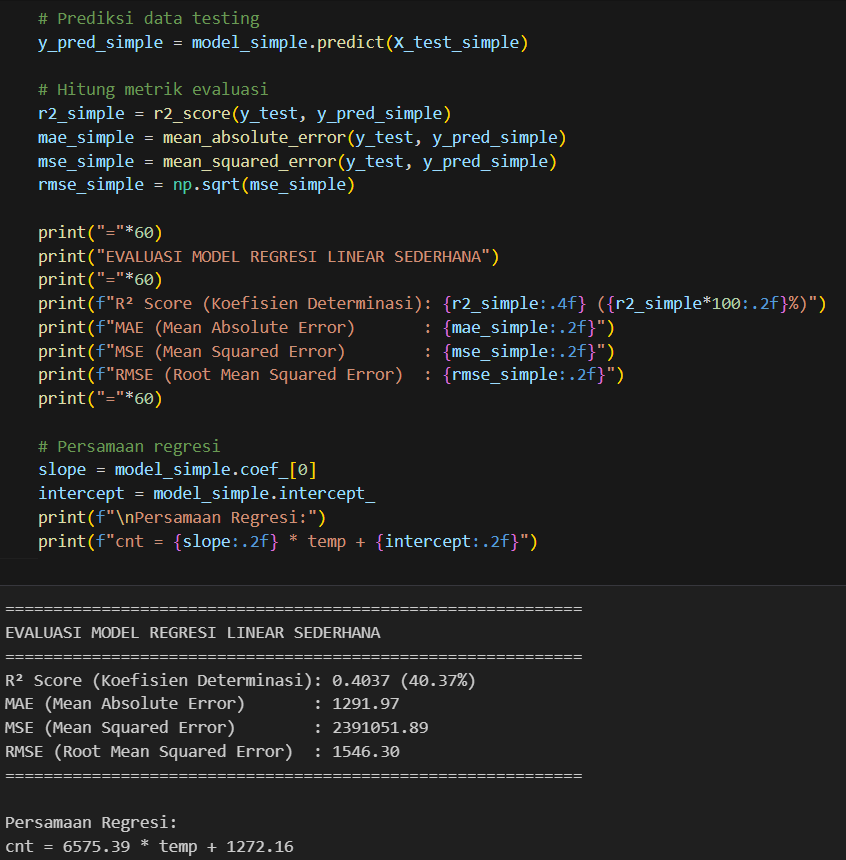


Gambar 4 Pemodelan Regresi Linear Sederhana

Membuat dan melatih model regresi linear sederhana menggunakan sklearn untuk memprediksi cnt berdasarkan temperatur.

* **Membuat object model**:Instantiate kelas LinearRegression() dari sklearn
* **Melatih model**:Menggunakan model.fit() dengan parameter X\_train\_simple dan y\_train
* **Proses fitting**:Model mencari nilai koefisien dan intercept optimal yang meminimalkan error
* **Menampilkan koefisien**:Nilai model.coef\_[0] menunjukkan perubahan cnt untuk setiap unit perubahan temp
* **Menampilkan intercept**:Nilai model.intercept\_ adalah nilai cnt ketika temp = 0
* **Interpretasi koefisien positif**:Jika koefisien positif berarti peningkatan temperatur meningkatkan jumlah penyewaan

## Evaluasi Model Regresi Linear Sederhana



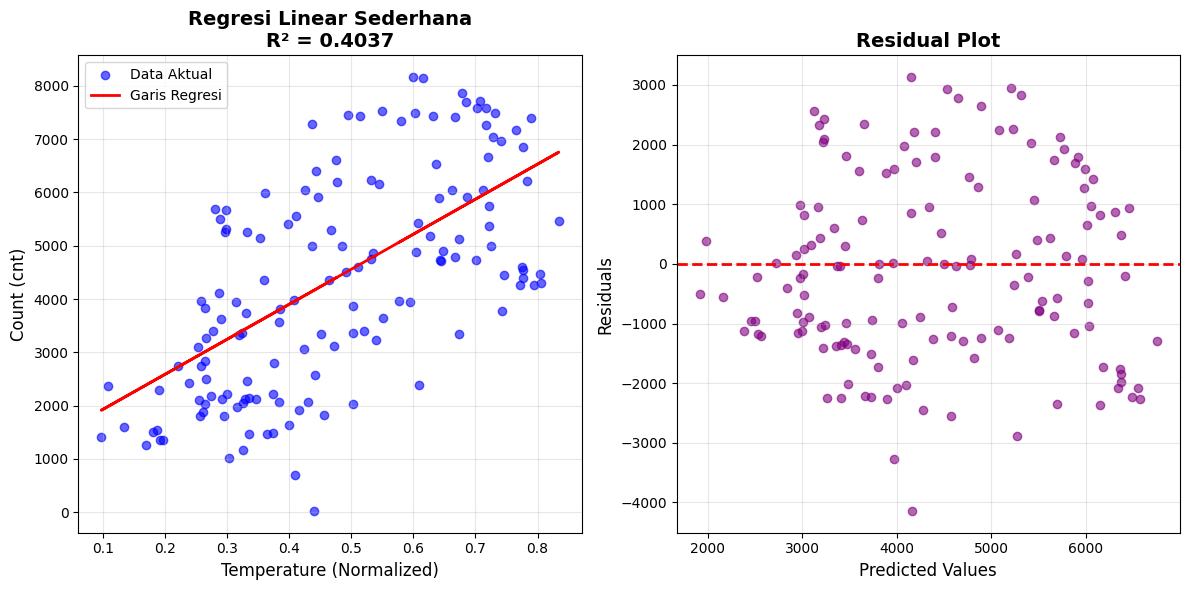
Gambar 5 Menghitung Nilai Sentral

Mengukur performa model pada data testing menggunakan berbagai metrik evaluasi untuk menilai akurasi prediksi.

* **Membuat prediksi**: **model.predict(X\_test\_simple)** untuk memprediksi nilai cnt pada data testing
* **Menghitung R² Score**:Mengukur proporsi variansi yang dijelaskan model, nilai mendekati 1 = model bagus
* **Menghitung MAE**: Mean Absolute Error, rata-rata selisih absolut antara prediksi dan aktual
* **Menghitung MSE**: Mean Squared Error, rata-rata kuadrat error, memberikan penalti lebih besar untuk error besar
* **Menghitung RMSE**: Akar dari MSE, memiliki satuan yang sama dengan target (jumlah penyewaan)
* **Menampilkan hasil**: Print semua metrik evaluasi dengan format yang rapi
* **Membuat persamaan regresi**: Menampilkan formula cnt = slope × temp + intercept untuk interpretasi
  1. **Visualisasi Regresi Linear Sederhana**



Gambar 6 Syntax visuliasasi Regresi Linear Sederhan

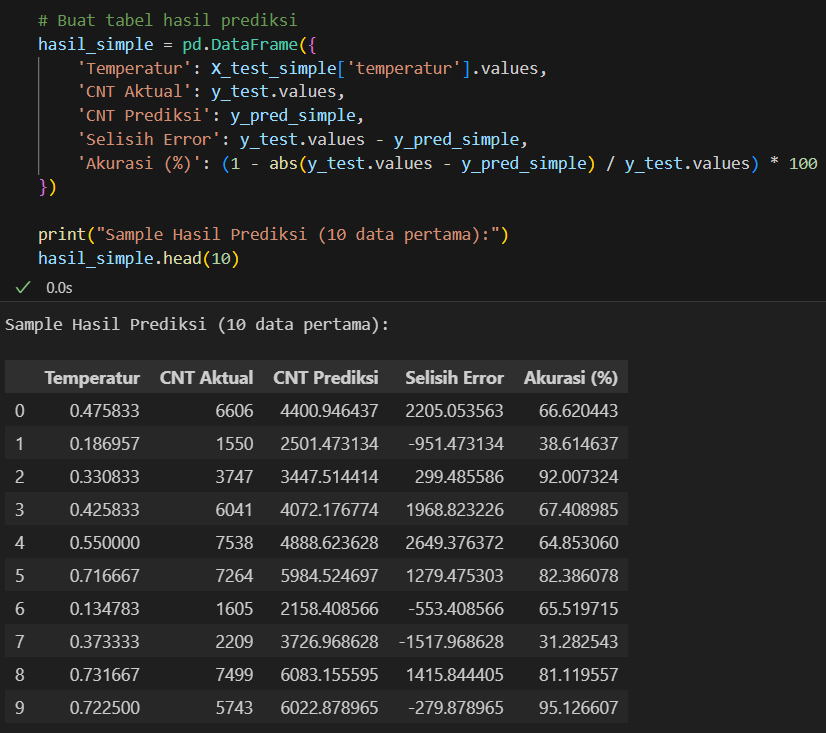


Gambar 7 Visualisasi Regresi Linear Sederhana

Membuat visualisasi untuk memahami hubungan antara temperatur dan jumlah penyewaan serta kualitas prediksi model.

* **Membuat figure dengan 2 subplot**: plt.subplots(1, 2) untuk menampilkan 2 grafik berdampingan
* **Subplot 1 - Scatter plot**: Menampilkan data aktual sebagai titik biru dan garis regresi merah
* **Plot garis regresi**: Menggunakan plt.plot() untuk menggambar garis prediksi model
* **Menambahkan label sumbu**: xlabel untuk temperatur, ylabel untuk count
* **Menampilkan R² di title**: Menunjukkan nilai R² Score sebagai indikator kualitas model
* **Subplot 2 - Residual plot**: Scatter plot antara nilai prediksi dan residual (error)
* **Garis horizontal y=0**: Referensi untuk melihat apakah residual tersebar random di sekitar nol
* **Menambahkan grid**: Parameter grid=True untuk memudahkan pembacaan grafik
* **tight\_layout()**: Mengatur spacing antar subplot agar tidak overlap

## Tabel Hasil Prediksi Regresi Sederhana



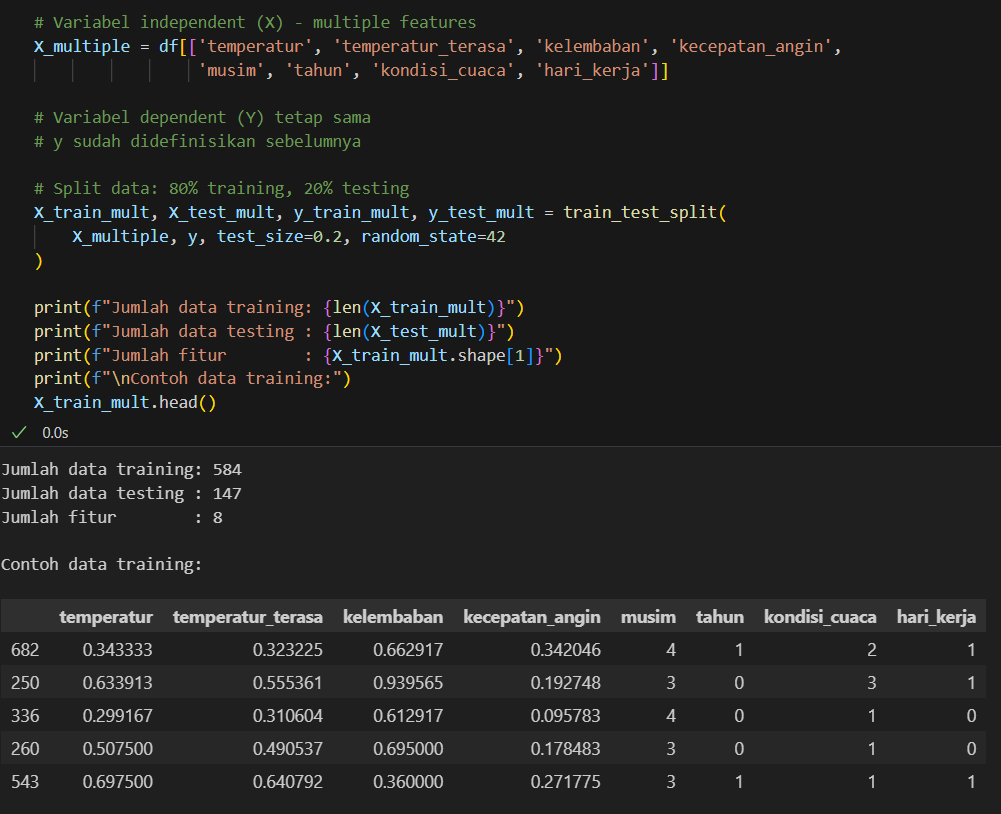
Gambar 8 Tabel Hasil Prediksi Regresi Sederhana

Membuat tabel detail yang menampilkan hasil prediksi untuk setiap sampel data testing beserta metrik akurasinya.

* **Membuat DataFrame hasil**: Menggunakan pd.DataFrame() dengan dictionary berisi berbagai kolom
* **Kolom Temperatur**: Nilai temperatur dari X\_test\_simple menggunakan .values untuk convert ke numpy array
* **Kolom CNT Aktual**: Nilai jumlah penyewaan aktual dari y\_test yang merupakan ground truth
* **Kolom CNT Prediksi**: Hasil prediksi model (y\_pred\_simple) untuk setiap sampel
* **Kolom Selisih Error**: Dihitung sebagai y\_test - y\_pred\_simple (positif = overpredict, negatif = underpredict)
* **Kolom Akurasi (%)**: Dihitung dengan formula (1 - |error|/actual) × 100% untuk akurasi per-baris
* **Menampilkan 10 data pertama**: Menggunakan .head(10) untuk melihat sample hasil prediksi
* **Interpretasi tabel**: Memudahkan identifikasi kasus dimana model prediksi sangat akurat atau meleset jauh

# Multiple Linear Regression

* 1. **Split Data untuk Multiple Linear Regression**

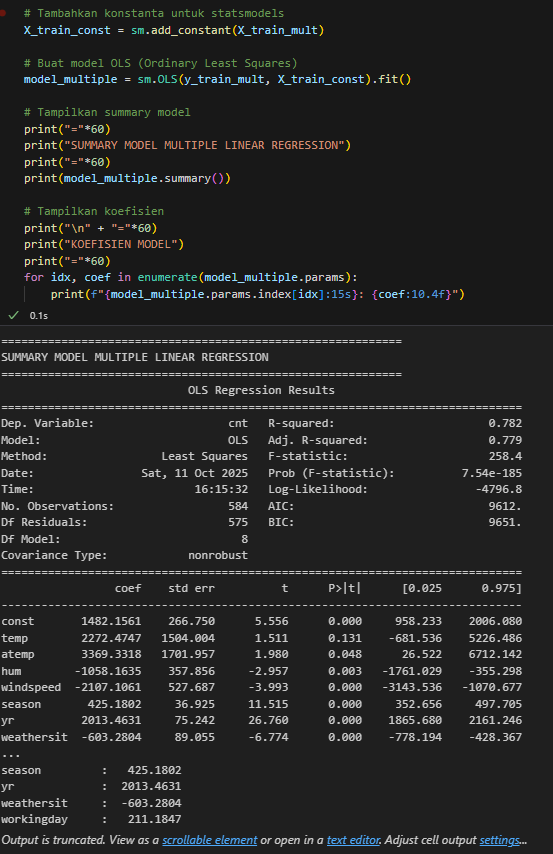


Gambar 9 method describe()

Mempersiapkan data untuk model regresi berganda dengan menggunakan 8 variabel independent yang berkorelasi signifikan dengan target.

* **Mendefinisikan X\_multiple**: Memilih 8 kolom fitur (temp, atemp, hum, windspeed, season, yr, weathersit, workingday)
* **Pemilihan fitur berdasarkan korelasi**: Variabel yang dipilih memiliki korelasi signifikan dengan cnt
* **Menghindari data leakage**: Tidak menggunakan 'registered' dan 'casual' meskipun korelasinya tinggi
* **Variabel y tetap sama**: Target prediksi masih cnt (jumlah penyewaan sepeda)
* **Split data 80:20**: Membagi data menjadi training dan testing dengan proporsi yang sama
* **random\_state=42**: Menggunakan seed yang sama untuk konsistensi hasil
* **Print informasi**: Menampilkan jumlah data training, testing, dan jumlah fitur yang digunakan
* **Verifikasi data**: Menampilkan 5 baris pertama data training untuk memastikan split berhasil

## Pemodelan Multiple Linear Regression



Gambar 10 Pemodelan Multiple Linear Regression

Membuat model regresi berganda menggunakan statsmodels OLS untuk mendapatkan informasi statistik yang lebih detail.

* **Menambahkan konstanta**: sm.add\_constant() menambahkan kolom konstanta (intercept) ke data training
* **Membuat model OLS**: Ordinary Least Squares, metode standar untuk regresi linear
* **Fitting model**: .fit() melatih model dengan data training yang sudah ditambah konstanta
* **Menampilkan summary**: Output lengkap berisi R², Adjusted R², F-statistic, koefisien, p-value, confidence interval
* **Interpretasi p-value**: Nilai p < 0.05 menunjukkan variabel signifikan secara statistik
* **Cek multikolinearitas**: Dari condition number di summary dapat mendeteksi multikolinearitas
* **Loop koefisien**: Menampilkan semua koefisien model dalam format yang lebih mudah dibaca
* **Analisis koefisien**: Koefisien positif = meningkatkan cnt, negatif = menurunkan cnt

## Persamaan Regresi Berganda

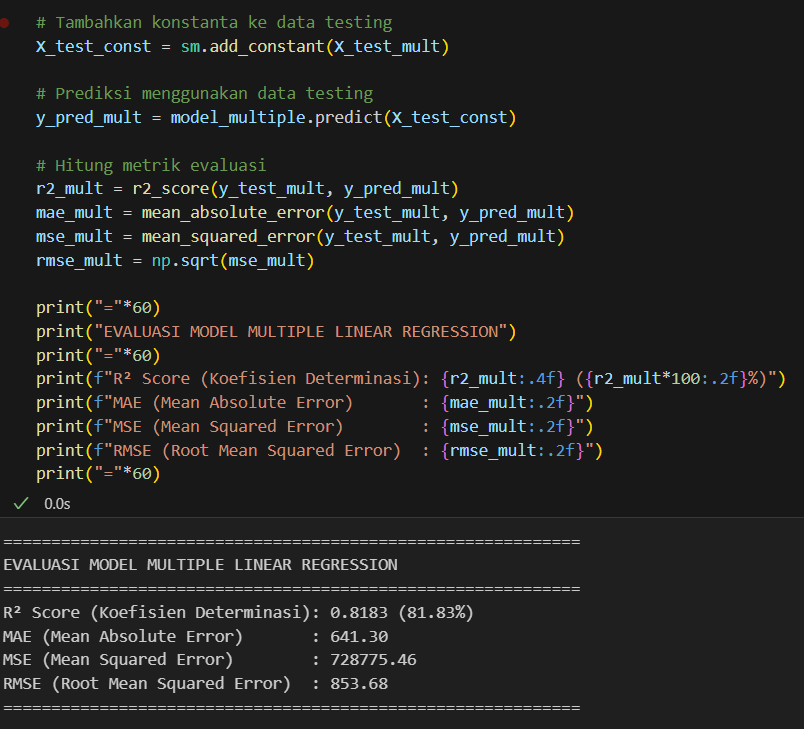


Gambar 11 Persamaan Regresi Berganda

Membuat persamaan matematis dari model regresi berganda untuk memudahkan interpretasi dan penggunaan model.

* **Ekstrak konstanta**: Mengambil nilai konstanta dari model\_multiple.params['const']
* **Ekstrak koefisien variabel**: Menggunakan .drop('const') untuk mendapatkan koefisien semua variabel tanpa konstanta
* **Inisialisasi equation**: Memulai string persamaan dengan nilai konstanta
* **Loop semua variabel**: Iterasi setiap feature dan koefisiennya untuk membangun persamaan lengkap
* **Menentukan tanda**: Menggunakan "+" untuk koefisien positif dan "-" untuk negatif
* **Format koefisien**: Menggunakan abs() dan format .2f untuk menampilkan 2 desimal
* **Print persamaan lengkap**: Menampilkan persamaan dalam format cnt = const + coef1×var1 + ... + coef8×var8
* **Kegunaan persamaan**: Dapat digunakan untuk membuat prediksi manual atau memahami kontribusi setiap variable

## Evaluasi Model Multiple Linear Regression

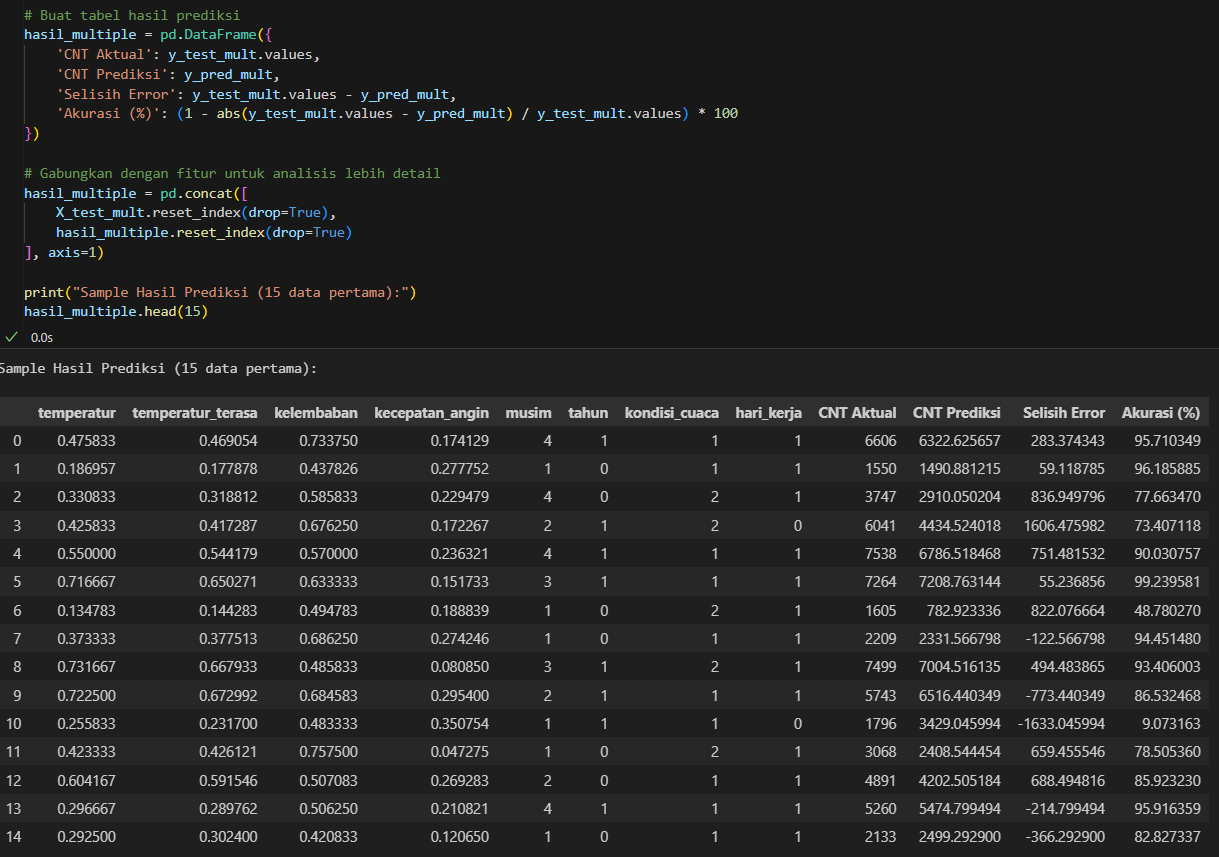


Gambar 12 Evaluasi Model Multiple Linear Regression

Mengukur performa model multiple regression pada data testing menggunakan berbagai metrik evaluasi.

* **Menambahkan konstanta ke data testing**: sm.add\_constant(X\_test\_mult) agar konsisten dengan data training
* **Membuat prediksi**: model\_multiple.predict() untuk memprediksi cnt pada data testing
* **Menghitung R² Score**: Mengukur seberapa baik model menjelaskan variansi data testing
* **Menghitung MAE**: Rata-rata error absolut dalam satuan jumlah penyewaan
* **Menghitung MSE**: Mean squared error untuk mengukur error dengan penalti kuadrat
* **Menghitung RMSE**: Root mean squared error, lebih interpretable karena satuannya sama dengan target
* **Perbandingan dengan simple regression**: R² lebih tinggi menunjukkan model lebih baik
* **Print hasil evaluasi**: Menampilkan semua metrik dengan format yang rapi dan mudah dibaca

## Tabel Hasil Prediksi Multiple Regression



Gambar 13 Tabel Hasil Prediksi Multiple Regression

Membuat tabel komprehensif yang menggabungkan semua fitur input dengan hasil prediksi dan metrik evaluasinya.

* **Membuat DataFrame hasil**: Berisi actual, predicted, error, dan akurasi
* **Kolom Actual\_cnt**: Nilai jumlah penyewaan aktual dari data testing
* **Kolom Predicted\_cnt**: Hasil prediksi model multiple regression
* **Kolom Selisih\_Error**: Dihitung sebagai actual - predicted untuk melihat arah error
* **Kolom Akurasi (%)**: Persentase akurasi dihitung dengan (1 - |error|/actual) × 100%
* **Menggabungkan dengan fitur**: pd.concat() untuk menggabungkan X\_test\_mult dengan hasil prediksi
* **Reset index**: Memastikan index konsisten setelah penggabungan DataFrame
* **Menampilkan 15 data**: .head(15) untuk melihat sample yang cukup representatif
* **Analisis detail**: Tabel ini memungkinkan analisis pengaruh kombinasi fitur terhadap akurasi prediksi

## Kesimpulan

Model regresi berganda lebih baik daripada regresi sederhana dalam konteks dataset ini: R² meningkat dan nilai MAE/MSE/RMSE menurun, artinya prediksi lebih akurat ketika menggunakan beberapa fitur. Temperatur terbukti berpengaruh positif terhadap jumlah penyewaan, namun fitur seperti pengguna\_terdaftar dan pengguna\_kasual tidak digunakan karena menyebabkan data leakage. Untuk perbaikan selanjutnya bisa dilakukan feature engineering, pemeriksaan multikolinearitas, atau penggunaan regularisasi agar generalisasi lebih baik. Secara praktis, model ini sudah layak untuk latihan forecasting sederhana dan dapat membantu estimasi kebutuhan sepeda berdasarkan kondisi cuaca dan temporal.

Link Github Praktikum : [https://github.com/raffayuda/MachineLearning/blob/main/praktikum1/Notebook/praktikum01.ipy](https://github.com/raffayuda/MachineLearning/blob/main/praktikum1/Notebook/praktikum01.ipynb) [nb](https://github.com/raffayuda/MachineLearning/blob/main/praktikum1/Notebook/praktikum01.ipynb)

Link Github Praktikum Mandiri : [https://github.com/raffayuda/Machine-](https://github.com/raffayuda/Machine-Learning/blob/main/praktikum1/Notebook/praktikum01_mandiri.ipynb) [Learning/blob/main/praktikum1/Notebook/praktikum01\_mandiri.ipynb](https://github.com/raffayuda/Machine-Learning/blob/main/praktikum1/Notebook/praktikum01_mandiri.ipynb)