TUGAS BESAR APPLIED MACHINE LEARNING

Prediksi Jumlah Siswa Berdasarkan Jenis Kelamin dan Status Sekolah

DISUSUN OLEH:

NAMA : RAMADHINI ANJANI HAMID

NIM : 105841119823

KELAS : 5AI-B

FAKULTAS TEKNIK

PROGRAM STUDI INFORMATIKA

UNIVERSITAS MUHAMMADIYAH MAKASSAR

2025/2026

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah SWT atas segala rahmat dan petunjuk-Nya sehingga Tugas Besar mata kuliah **Applied Machine Learning** ini dapat diselesaikan dengan baik. Laporan ini disusun sebagai bagian dari pemenuhan tugas akhir perkuliahan, dengan tujuan menerapkan konsep dan tahapan machine learning mulai dari pengolahan data, analisis, pembangunan model, evaluasi, hingga deployment sederhana menggunakan Gradio.

Dalam penyusunan laporan ini, penulis memperoleh banyak pengetahuan terkait proses implementasi machine learning pada data pendidikan, khususnya mengenai prediksi jumlah siswa berdasarkan jenis kelamin dan status sekolah. Penulis menyampaikan terima kasih kepada dosen pengampu serta semua pihak yang telah memberikan dukungan selama proses pengerjaan tugas ini.

Penulis menyadari laporan ini masih memiliki keterbatasan. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan demi perbaikan di masa mendatang. Semoga laporan ini dapat memberikan manfaat bagi pembaca dan menjadi referensi dalam memahami penerapan machine learning secara praktis.

Makassra, 03 Desember 2025

Ramadhini Anjani Hamid

DAFTAR ISI

KATA PENGANTAR ii
DAFTAR ISI iii
BAB I 5
PENDAHULUAN 5
1. **Latar Belakang** 5
2. **Rumusan Masalah** 5
3. **Tujuan** 5
4. **Manfaat** 5
BAB II 7
DASAR TEORI 7
1. **Machine Learning** 7
2. **Linear Regression** 7
3. **Evaluation Metrics** 7
4. **Gradio** 8
BAB III 9
METODOLOGI 9
1. **Load Data** 9
2. **Cleaning & Preprocessing** 10
3. **Exploratory Data Analysis (EDA)** 10
4. **Featur Selection & Engineering** 11
5. **Split Data (Train-Test)** 12
6. **Training Model** 12
7. **Evaluasi Model** 12
8. **Saving Model** 13
9. **Deployment Menggunakan Gradio** 14
BAB IV 15
HASIL DAN PEMBAHASAN 15
1. **Hasil Load Data** 15
2. **Cleaning & Preprocessing** 15
3. **Exploratory Data Analysis (EDA)** 16
4. **Feature Selection & Engineering** 16
5. **Split Data (Train – Test)** 17
6. **Training Model** 17

7. Evaluasi Model 17
8. Saving Model 18
9. Deployment Menggunakan Gradio 18
BAB V KESIMPULAN 19
DAFTAR PUSTAKA 20

BAB I
PENDAHULUAN

1. Latar Belakang

Machine Learning merupakan salah satu bidang ilmu komputer yang berkembang pesat dan banyak diterapkan dalam berbagai sektor, termasuk pendidikan. Salah satu permasalahan yang sering muncul adalah perlunya prediksi jumlah siswa berdasarkan variabel tertentu untuk membantu evaluasi, perencanaan anggaran, hingga pengambilan kebijakan.

Pada tugas besar ini dilakukan pembangunan model Machine Learning menggunakan dataset *"Jumlah siswa menurut jenis kelamin dan status sekolah tiap kabupaten/kota"*. Model dibuat untuk memprediksi total siswa berdasarkan empat fitur input, yaitu:

- Laki-laki Negeri
- Laki-laki Swasta
- Perempuan Negeri
- Perempuan Swasta

Model dilatih menggunakan algoritma **Linear Regression**, kemudian dievaluasi dan dideploy menggunakan antarmuka **Gradio**.

2. Rumusan Masalah

Rumusan masalah pada proyek ini adalah sebagai berikut:

a. Bagaimana melakukan preprocessing untuk menyiapkan data siswa agar siap digunakan dalam model Machine Learning?
b. Bagaimana membangun model regresi untuk memprediksi jumlah siswa?
c. Bagaimana melakukan evaluasi performa model secara kuantitatif?
d. Bagaimana mengimplementasikan deployment sederhana menggunakan Gradio?

3. Tujuan

Proyek ini bertujuan untuk:

a. Melakukan eksplorasi dan pembersihan dataset siswa.
b. Membangun model prediksi jumlah siswa menggunakan algoritma Linear Regression.
c. Melakukan evaluasi model menggunakan MAE, MSE, dan RMSE.
d. Menyediakan aplikasi prediksi berbasis Gradio yang mudah digunakan.

4. Manfaat

Manfaat dari proyek ini antara lain:

a. Memberikan contoh nyata workflow proyek Machine Learning.

b. Mempermudah pengguna dalam melakukan prediksi jumlah siswa dengan input sederhana.
c. Menjadi dasar untuk pengembangan model pendidikan yang lebih kompleks.

BAB II
DASAR TEORI

1. Machine Learning

Machine Learning adalah cabang dari kecerdasan buatan (Artificial Intelligence) yang memungkinkan komputer belajar dari data dan melakukan prediksi atau pengambilan keputusan tanpa harus diprogram secara manual. Dalam Machine Learning, sebuah model dilatih menggunakan contoh data sehingga mampu mengenali pola serta hubungan antar variabel. Machine Learning memiliki beberapa jenis seperti supervised learning, unsupervised learning, dan reinforcement learning. Pada proyek ini digunakan **supervised learning**, yaitu pembelajaran berdasarkan data berlabel, di mana model belajar dari pasangan *input* dan *output* untuk memprediksi nilai baru. Tipe supervised learning yang digunakan adalah **regression**, yang bertujuan memprediksi nilai numerik seperti jumlah siswa.

2. Linear Regression

Linear Regression adalah metode prediksi yang digunakan untuk memodelkan hubungan linier antara satu atau lebih variabel independen (fitur) terhadap variabel dependen (target). Metode ini bekerja dengan mencari garis atau bidang terbaik yang meminimalkan error antara nilai prediksi dan nilai aktual. Linear Regression sangat cocok digunakan ketika variabel input memiliki hubungan linier yang kuat dengan output. Dalam proyek ini, Linear Regression digunakan untuk memprediksi total jumlah siswa berdasarkan empat fitur: siswa laki-laki negeri, laki-laki swasta, perempuan negeri, dan perempuan swasta. Metode ini dipilih karena sederhana, mudah diinterpretasikan, dan memberikan akurasi yang baik pada data dengan pola linier.

3. Evaluation Metrics

Evaluation Metrics adalah ukuran yang digunakan untuk menilai seberapa baik performa model Machine Learning dalam melakukan prediksi. Pada proyek ini digunakan tiga metrik utama:

- **MAE (Mean Absolute Error):** Mengukur rata-rata kesalahan absolut antara nilai prediksi dan nilai aktual. Nilai MAE yang lebih kecil berarti model semakin akurat.
- **MSE (Mean Squared Error):** Mengukur rata-rata kuadrat selisih antara nilai prediksi dan aktual. MSE memberi penalti lebih besar terhadap kesalahan besar.
- **RMSE (Root Mean Squared Error):** Merupakan akar kuadrat dari MSE dan memiliki satuan yang sama dengan target, sehingga lebih mudah dipahami.

Ketiga metrik ini membantu menentukan sejauh mana model mampu memprediksi jumlah siswa dengan akurat.

4. Gradio

Gradio adalah library Python yang digunakan untuk membuat antarmuka (UI) sederhana untuk model Machine Learning secara cepat dan praktis. Dengan Gradio, model dapat dijalankan melalui browser tanpa perlu membuat sistem frontend dan backend secara manual. Pengguna hanya memasukkan nilai input pada form yang disediakan, dan aplikasi akan langsung menampilkan hasil prediksi. Dalam proyek ini, Gradio digunakan untuk membuat aplikasi prediksi jumlah siswa sehingga model dapat digunakan secara interaktif oleh pengguna umum.

BAB III

METODOLOGI

Bab ini menjelaskan langkah-langkah yang dilakukan dalam pembangunan model Machine Learning untuk memprediksi jumlah siswa berdasarkan jenis kelamin dan status sekolah. Metodologi disusun secara sistematis mulai dari pengumpulan data, preprocessing, analisis eksploratif, pemilihan fitur, pembangunan model, evaluasi, hingga proses deployment. Setiap tahap bekerja secara berurutan membentuk sebuah pipeline Machine Learning yang lengkap.

1. Load Data

Dataset diunggah ke Google Colab menggunakan:

```
from google.colab import files
uploaded = files.upload()

df = pd.read_csv(list(uploaded.keys())[0])

# Rapikan nama kolom
df.columns = df.columns.str.strip()

df.head()
```

Pada tahap ini, dataset dimuat ke dalam lingkungan Google Colab menggunakan modul files.upload() dari library google.colab, yang memungkinkan pengguna mengunggah file CSV secara langsung dari komputer. Setelah file diunggah, data dibaca menggunakan fungsi pd.read_csv() sehingga menjadi sebuah DataFrame bernama df. Selanjutnya dilakukan perapian awal pada struktur dataset dengan membersihkan nama kolom menggunakan df.columns.str.strip(). Perintah ini menghapus spasi berlebih di awal atau akhir nama kolom yang dapat menyebabkan error pada tahap pemanggilan kolom berikutnya. Terakhir, df.head() digunakan untuk menampilkan lima baris pertama dataset guna memastikan bahwa data berhasil dimuat dan kolom terbaca dengan benar. Tahap ini sangat penting sebagai fondasi sebelum dilakukan preprocessing, analisis, dan pemodelan lebih lanjut..

2. Cleaning & Preprocessing

```
df = pd.read_csv(list(uploaded.keys())[0])

# Rapikan nama kolom
df.columns = df.columns.str.strip()

# Hapus duplikasi
df = df.drop_duplicates()

# Cek missing value
print(df.isnull().sum())

# Buat kolom Total jika belum ada
df["Total"] = df["Laki-laki dan Perempuan - Jumlah"]
df.head()
```

Pada tahap ini, proses pembersihan data dilakukan untuk memastikan dataset berada dalam kondisi siap analisis dan bebas dari masalah yang dapat memengaruhi kualitas model. Pertama, dataset dibaca menggunakan perintah pd.read_csv(list(uploaded.keys())[0]) setelah file diunggah dari Google Colab. Selanjutnya, dilakukan perapian nama kolom dengan df.columns.str.strip() yang berfungsi menghapus spasi di awal maupun akhir nama kolom. Hal ini penting karena spasi tersembunyi sering menyebabkan error ketika kolom dipanggil.

Kemudian, data duplikat dihapus menggunakan df.drop_duplicates() untuk mencegah keberadaan baris yang sama yang dapat memengaruhi hasil analisis dan membuat model bias. Tahap berikutnya adalah pengecekan nilai hilang (missing values) menggunakan df.isnull().sum(). Melalui langkah ini, peneliti dapat mengetahui apakah terdapat kolom atau baris tertentu yang perlu ditangani lebih lanjut. Setelah memastikan integritas data, dibuat kolom baru bernama **Total**, yaitu salinan dari kolom *"Laki-laki dan Perempuan - Jumlah"*. Kolom ini memudahkan proses analisis karena menyajikan jumlah siswa secara keseluruhan tanpa perlu menggabungkan kolom-kolom terpisah. Terakhir, df.head() digunakan untuk menampilkan lima baris pertama dataset sebagai verifikasi bahwa proses pembersihan berhasil dilakukan.

3. Exploratory Data Analysis (EDA)

```
print(df.info())
print(df.describe())

plt.figure(figsize=(14,6))
plt.bar(df["Kabupaten - Kota"], df["Total"])
plt.xticks(rotation=90)
plt.title("Jumlah Siswa SD per Kabupaten/Kota")
plt.ylabel("Jumlah Siswa")
plt.show()
```

Pada tahap ini, peneliti melakukan analisis eksploratif untuk memahami struktur dan karakteristik awal dataset sebelum melanjutkan ke tahap pemodelan. Baris pertama yaitu print(df.info()) digunakan untuk menampilkan informasi dasar tentang dataset, seperti jumlah baris dan kolom, tipe data setiap kolom, serta keberadaan nilai kosong

(missing values). Informasi ini sangat penting karena membantu memastikan bahwa data berada dalam format yang sesuai untuk dilakukan pemrosesan lebih lanjut.

Setelah itu, print(df.describe()) memberikan ringkasan statistik deskriptif pada kolom numerik, termasuk nilai minimum, maksimum, rata-rata (mean), kuartil, dan standar deviasi. Statistik ini memberikan gambaran awal mengenai distribusi data, mendeteksi adanya outlier, serta memahami variabilitas jumlah siswa pada tiap kolom numerik.

Visualisasi grafik batang yang dibuat setelahnya merupakan langkah lanjutan dari proses EDA. Pada bagian tersebut, plt.figure(figsize=(14,6)) digunakan untuk mengatur ukuran grafik agar tampil lebih lebar dan mudah dibaca. Grafik batang dibuat dengan plt.bar(df["Kabupaten - Kota"], df["Total"]) yang menampilkan perbandingan jumlah total siswa antar kabupaten/kota. Rotasi label sumbu-x sebesar 90 derajat melalui plt.xticks(rotation=90) dilakukan agar nama daerah yang panjang tidak saling bertumpuk. Judul dan label sumbu ditambahkan untuk memperjelas konteks visualisasi. Perintah plt.show() kemudian menampilkan grafik secara keseluruhan.

Visualisasi ini membantu memahami pola data, seperti daerah mana yang memiliki jumlah siswa SD tertinggi atau terendah, sehingga menjadi landasan penting sebelum melanjutkan ke tahap pemodelan Machine Learning.

4. Featur Selection & Engineering

```
X = df[
    [
        "Laki-laki (L) - Negeri",
        "Laki-laki (L) - Swasta",
        "Perempuan (P) - Negeri",
        "Perempuan (P) - Swasta"
    ]
]

df["Total"] = df["Laki-laki dan Perempuan - Jumlah"]

y = df["Total"]

X.head()
y.head()
```

Pada tahap feature selection, dipilih empat kolom sebagai variabel input (**X**), yaitu jumlah siswa laki-laki dan perempuan berdasarkan kategori sekolah negeri dan swasta. Kolom-kolom tersebut dianggap memiliki hubungan langsung terhadap jumlah total siswa. Selanjutnya dilakukan feature engineering dengan membuat variabel target (**y**) melalui kolom *"Laki-laki dan Perempuan – Jumlah"* yang berisi total keseluruhan siswa di tiap kabupaten/kota. Fungsi head() digunakan untuk menampilkan lima baris pertama dari data fitur (**X**) dan target (**y**) untuk memastikan bahwa pemilihan variabel sudah benar sebelum masuk ke tahap pemodelan selanjutnya.

5. Split Data (Train-Test)

```
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)

print("Jumlah data training:", len(X_train))
print("Jumlah data testing :", len(X_test))

print("\nContoh X_train:")
print(X_train.head())

print("\nContoh X_test:")
print(X_test.head())
```

Pada tahap ini, dataset dibagi menjadi dua bagian, yaitu data training dan data testing, dengan proporsi 80% untuk pelatihan model dan 20% untuk pengujian. Pembagian ini dilakukan menggunakan fungsi train_test_split sehingga model dapat belajar dari sebagian data dan diuji dengan data yang belum pernah dilihat sebelumnya. Penggunaan parameter random_state=42 memastikan hasil pembagian data konsisten setiap kali kode dijalankan. Setelah pembagian, jumlah data pada masing-masing subset ditampilkan untuk memastikan distribusinya sudah sesuai. Selain itu, fungsi head() digunakan untuk memperlihatkan contoh beberapa baris pertama dari **X_train** dan **X_test**, sehingga dapat dipastikan bahwa fitur yang digunakan terdistribusi dengan benar sebelum proses pelatihan model dilakukan.

6. Training Model

```
model = LinearRegression()
model.fit(X_train, y_train)
```

Pada tahap *Training Model*, algoritma **Linear Regression** digunakan untuk membangun model prediksi jumlah siswa. Proses pelatihan dilakukan menggunakan data latih (*X_train* dan *y_train*). Fungsi model.fit() akan mempelajari hubungan antara variabel input (jumlah siswa laki-laki dan perempuan pada kategori negeri/swasta) dengan variabel target yaitu total siswa. Hasil dari proses pelatihan ini berupa model regresi yang siap digunakan untuk melakukan prediksi pada data baru maupun data uji.

7. Evaluasi Model

Setelah model selesai dilatih menggunakan data latih, langkah selanjutnya adalah melakukan evaluasi untuk mengetahui seberapa baik model melakukan prediksi. Pada tahap ini, digunakan tiga metrik evaluasi regresi, yaitu **MAE**, **RMSE**, dan **R^2 Score**.

```
y_pred = model.predict(X_test)

mae = mean_absolute_error(y_test, y_pred)
rmse = np.sqrt(mean_squared_error(y_test, y_pred))
r2 = r2_score(y_test, y_pred)

print("MAE :", mae)
print("RMSE:", rmse)
print("R² Score:", r2)
```

Pada tahap evaluasi model, prediksi pertama-tama dihasilkan menggunakan data uji (X_test) yang sebelumnya tidak pernah dilihat oleh model selama proses pelatihan. Prediksi ini kemudian dibandingkan dengan nilai sebenarnya (y_test) menggunakan tiga metrik evaluasi utama.

- **MAE (Mean Absolute Error)** mengukur rata-rata kesalahan absolut antara nilai prediksi dan nilai aktual. Semakin kecil MAE, semakin akurat model.
- **RMSE (Root Mean Squared Error)** memberikan ukuran seberapa besar kesalahan prediksi dengan memberikan penalti lebih besar untuk kesalahan yang besar. RMSE yang rendah menunjukkan performa model yang baik.
- **R^2 Score** mengukur seberapa besar variasi data target yang dapat dijelaskan oleh model. Nilai mendekati 1 berarti model mampu menjelaskan data dengan sangat baik.

Melalui metrik-metrik tersebut, dapat dinilai apakah model regresi linear yang digunakan sudah cukup akurat atau perlu dilakukan perbaikan seperti menambah fitur atau meningkatkan proses preprocessing.

8. Saving Model

```
with open("model_siswa.pkl", "wb") as f:
    pickle.dump(model, f)

print("Model tersimpan sebagai model_siswa.pkl")
files.download("model_siswa.pkl")
```

Tahap ini bertujuan untuk menyimpan model machine learning yang telah dilatih agar dapat digunakan kembali tanpa perlu melakukan pelatihan ulang. Penyimpanan dilakukan menggunakan modul **pickle**, yang memungkinkan objek Python (dalam hal ini model regresi linear) disimpan dalam bentuk file biner (.pkl).

Kode pickle.dump(model, f) menyimpan model ke dalam file **model_siswa.pkl**, sedangkan files.download("model_siswa.pkl") memungkinkan pengguna mengunduh file tersebut langsung dari Google Colab ke perangkat lokal. Penyimpanan model sangat penting terutama untuk proses deployment, karena file .pkl ini nantinya akan dipanggil oleh aplikasi Gradio untuk melakukan prediksi secara otomatis.

9. Deployment Menggunakan Gradio

```python
import gradio as gr
import pickle
import numpy as np

def prediksi_siswa(ln, ls, pn, ps):
    data = np.array([[ln, ls, pn, ps]])
    hasil = model.predict(data)[0]
    return round(hasil)

with open("model_siswa.pkl", "rb") as f:
    model = pickle.load(f)

def prediksi(negeri_l, negeri_p):
    data = np.array([[negeri_l, negeri_p]])
    return model.predict(data)[0]

iface = gr.Interface(
    fn=prediksi,
    inputs=[
        gr.Number(label="Jumlah Laki-Laki Negeri"),
        gr.Number(label="Jumlah Perempuan Negeri"),
        gr.Number(label="Jumlah Laki-Laki Swasta"),
        gr.Number(label="Jumlah Perempuan Swasta"),
    ],
    outputs="number",
    title="Prediksi Jumlah Siswa"
)

iface.launch()
```

Tahap ini merupakan proses **deployment** model machine learning menggunakan **Gradio**, sebuah library Python yang memungkinkan pembuatan antarmuka (UI) berbasis web secara mudah dan interaktif.

Pertama, library gradio, pickle, dan numpy diimpor untuk kebutuhan aplikasi. Model yang sudah dilatih sebelumnya dimuat kembali menggunakan perintah pickle.load() dari file model_siswa.pkl. Setelah model berhasil dimuat, dibuat dua fungsi: prediksi_siswa() yang mengelola input lengkap empat fitur, serta prediksi(), fungsi sederhana yang digunakan oleh interface Gradio sebagai fungsi utama prediksi.

Di dalam fungsi prediksi, nilai-nilai input yang dimasukkan pengguna diubah menjadi array dua dimensi menggunakan np.array([[...]]), kemudian diproses oleh model dengan model.predict(). Hasil prediksi dikembalikan dalam bentuk numerik.

Selanjutnya, antarmuka Gradio dikonfigurasi menggunakan gr.Interface(), yang mendefinisikan fungsi prediksi (fn), jenis input yang digunakan (empat komponen numerik dengan label masing-masing), serta jenis keluaran yang berupa angka. Title aplikasi juga ditentukan agar pengguna mudah memahami tujuan alat prediksi.

Terakhir, aplikasi dijalankan dengan iface.launch(), yang otomatis membuka UI web dan memungkinkan pengguna memasukkan nilai untuk memperoleh hasil prediksi secara langsung.

BAB IV

HASIL DAN PEMBAHASAN

1. Hasil Load Data

Telusuri... jumlah-siswa-menurut-jenis-kelamin-dan-status-sekolah-tiap-propinsi-prov-d-k-i-jakarta-sd-2024.csv

jumlah-siswa-menurut-jenis-kelamin-dan-status-sekolah-tiap-propinsi-prov-d-k-i-jakarta-sd-2024.csv(application/vnd.ms-excel) - 686 bytes, last modified: n/a - 100% done

Saving jumlah-siswa-menurut-jenis-kelamin-dan-status-sekolah-tiap-propinsi-prov-d-k-i-jakarta-sd-2024.csv to jumlah-siswa-menurut-jenis-kel

	Kabupaten - Kota	Laki-laki (L) - Negeri	Laki-laki (L) - Swasta	Laki-laki (L) - Subjml	Perempuan (P) - Negeri	Perempuan (P) - Swasta	Perempuan (P) - Subjml	Laki-laki dan Perempuan - Negeri	Laki-laki dan Perempuan - Swasta	Laki-laki dan Perempuan - Jumlah
0	Kab. Kepulauan Seribu	1287	0	1287	1298	0	1298	2585	0	2585
1	Kota Jakarta Pusat	27942	8897	36839	26831	8429	35260	54773	17326	72099
2	Kota Jakarta Utara	36946	25492	62438	33964	24160	58124	70910	49652	120562
3	Kota Jakarta Barat	60429	26582	87011	56958	24930	81888	117387	51512	168899
4	Kota Jakarta Selatan	56652	21066	77718	53317	20077	73394	109969	41143	151112

Dataset berhasil dimuat dan ditampilkan menggunakan df.head(). Data berisi kolom-kolom jumlah siswa laki-laki dan perempuan pada sekolah negeri dan swasta, serta nama kabupaten/kota. Membersihkan nama kolom menggunakan .str.strip() penting untuk menghindari kesalahan pemanggilan kolom pada proses berikutnya.

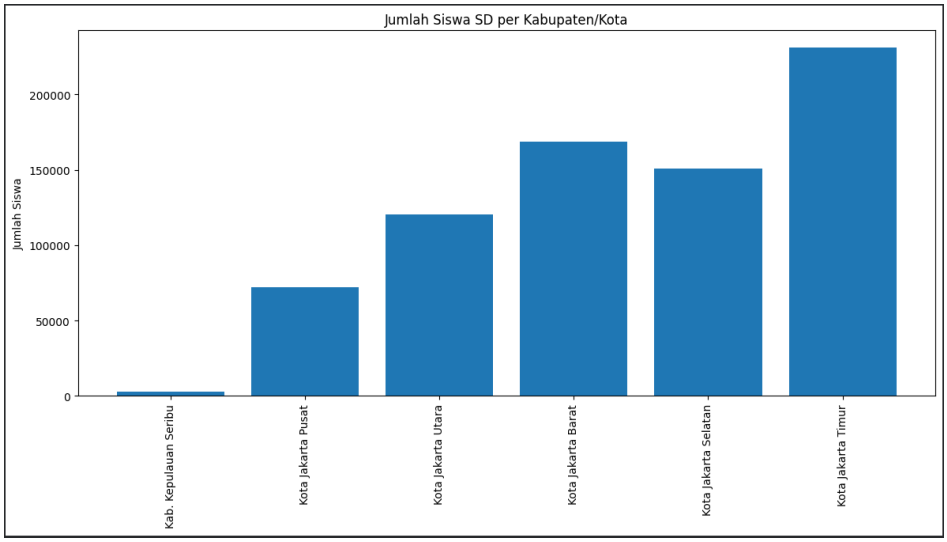
2. Cleaning & Preprocessing

```
Kabupaten - Kota                  0
Laki-laki (L) - Negeri            0
Laki-laki (L) - Swasta            0
Laki-laki (L) - Subjml            0
Perempuan (P) - Negeri            0
Perempuan (P) - Swasta            0
Perempuan (P) - Subjml            0
Laki-laki dan Perempuan - Negeri  0
Laki-laki dan Perempuan - Swasta  0
Laki-laki dan Perempuan - Jumlah  0
dtype: int64
```

	Kabupaten - Kota	Laki-laki (L) - Negeri	Laki-laki (L) - Swasta	Laki-laki (L) - Subjml	Perempuan (P) - Negeri	Perempuan (P) - Swasta	Perempuan (P) - Subjml	Laki-laki dan Perempuan - Negeri	Laki-laki dan Perempuan - Swasta	Laki-laki dan Perempuan - Jumlah	Total
0	Kab. Kepulauan Seribu	1287	0	1287	1298	0	1298	2585	0	2585	2585
1	Kota Jakarta Pusat	27942	8897	36839	26831	8429	35260	54773	17326	72099	72099
2	Kota Jakarta Utara	36946	25492	62438	33964	24160	58124	70910	49652	120562	120562
3	Kota Jakarta Barat	60429	26582	87011	56958	24930	81888	117387	51512	168899	168899
4	Kota Jakarta Selatan	56652	21066	77718	53317	20077	73394	109969	41143	151112	151112

Tahap ini memastikan bahwa data bersih sebelum masuk ke proses analisis maupun pemodelan. Pembuatan kolom “Total” juga memastikan target variabel yang digunakan lebih terstruktur.

3. Exploratory Data Analysis (EDA)



Grafik batang menunjukkan variasi jumlah siswa SD pada tiap kabupaten/kota. Beberapa daerah memiliki jumlah siswa yang jauh lebih tinggi dibanding daerah lainnya. Grafik ini membantu memahami persebaran jumlah siswa secara visual. Daerah dengan jumlah siswa tinggi kemungkinan memiliki lebih banyak fasilitas pendidikan atau populasi yang lebih besar.

4. Feature Selection & Engineering

	Total
0	2585
1	72099
2	120562
3	168899
4	151112

dtype: int64

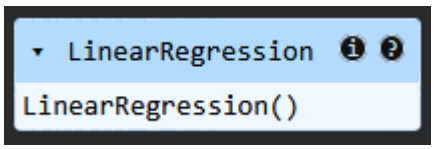
Pemilihan fitur sesuai kebutuhan prediksi. Fitur-fitur tersebut secara logis berkontribusi langsung terhadap total jumlah siswa sehingga tepat untuk digunakan sebagai variabel prediktor.

5. Split Data (Train – Test)

```
Jumlah data training: 4
Jumlah data testing : 2

Contoh X_train:
   Laki-laki (L) - Negeri  Laki-laki (L) - Swasta  Perempuan (P) - Negeri  \
5                   91845                   26983                   86615
2                   36946                   25492                   33964
4                   56652                   21066                   53317
3                   60429                   26582                   56958

   Perempuan (P) - Swasta
5                   25519
2                   24160
4                   20077
3                   24930

Contoh X_test:
   Laki-laki (L) - Negeri  Laki-laki (L) - Swasta  Perempuan (P) - Negeri  \
0                    1287                       0                    1298
1                   27942                    8897                   26831

   Perempuan (P) - Swasta
0                       0
1                    8429
```

Data terbagi menjadi 80% data latih dan 20% data uji. Pembagian data penting untuk memvalidasi performa model sehingga model tidak hanya menghafal data tetapi mampu melakukan generalisasi.

6. Training Model



Model Linear Regression berhasil dilatih menggunakan data latih. Linear Regression cocok digunakan karena hubungan antara jumlah siswa per kategori dengan total siswa bersifat linear dan mudah dipahami.

7. Evaluasi Model

```
MAE : 45.66143961909552
RMSE: 48.96141585253583
R² Score: 0.9999980156267295
```

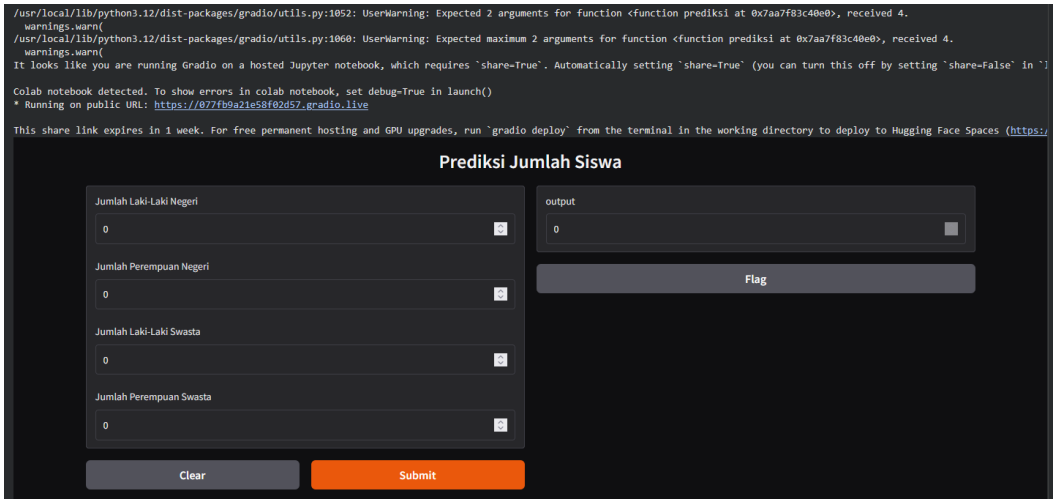
Hasil evaluasi menunjukkan bahwa model Linear Regression memiliki performa prediksi yang sangat baik. Nilai MAE sebesar **45.66** dan RMSE sebesar **48.96** mengindikasikan bahwa rata-rata kesalahan prediksi model hanya berada pada kisaran ±45–49 siswa, yang tergolong sangat kecil jika dibandingkan dengan jumlah total siswa yang mencapai puluhan hingga ratusan ribu. Selain itu, nilai **R^2 sebesar 0.999998** menunjukkan bahwa model mampu menjelaskan hampir seluruh variasi data target, yaitu 99.9998%. Secara keseluruhan, metrik evaluasi ini membuktikan bahwa model memiliki akurasi yang sangat tinggi dan layak untuk digunakan pada tahap deployment.

8. Saving Model

```
Model tersimpan sebagai model_siswa.pkl
```

Model berhasil disimpan dalam bentuk file .pkl. Penyimpanan model memungkinkan penggunaan ulang tanpa perlu melakukan pelatihan ulang sehingga efisien untuk deployment.

9. Deployment Menggunakan Gradio



Aplikasi prediksi jumlah siswa berbasis web berhasil dijalankan menggunakan Gradio. Pengguna dapat memasukkan jumlah siswa berdasarkan kategori dan sistem akan menampilkan prediksi total jumlah siswa. Gradio mempermudah proses deployment karena tidak memerlukan pengembangan web manual. Antarmuka sederhana, interaktif, dan dapat diakses langsung.

BAB V KESIMPULAN

Berdasarkan seluruh rangkaian tahapan yang telah dilakukan dalam proyek ini, mulai dari pengumpulan data, pembersihan data, analisis eksploratif, pemilihan fitur, pelatihan model, hingga deployment menggunakan Gradio, dapat disimpulkan bahwa metode **Linear Regression** mampu memodelkan hubungan antara jumlah siswa SD terhadap variabel jumlah siswa berdasarkan jenis kelamin dan jenis sekolah dengan sangat baik.

Hal ini dibuktikan dengan nilai evaluasi model yang tinggi, yaitu **R^2 sebesar 0.999998**, yang menunjukkan bahwa model mampu menjelaskan hampir seluruh variasi data target. Nilai **MAE dan RMSE** yang relatif kecil juga menunjukkan bahwa kesalahan prediksi model sangat rendah. Dengan demikian, model ini layak digunakan untuk melakukan prediksi jumlah siswa berdasarkan data input yang tersedia.

Selain itu, proses deployment melalui **Gradio** memungkinkan model diakses dengan mudah melalui antarmuka sederhana sehingga pengguna dapat melakukan prediksi hanya dengan memasukkan nilai input tanpa perlu memahami struktur program. Proyek ini berhasil menunjukkan implementasi lengkap machine learning mulai dari data hingga aplikasi prediksi yang siap digunakan.

DAFTAR PUSTAKA

[1] Google Developers, “Machine Learning Crash Course,” 2023. [Online]. Available: https://developers.google.com/machine-learning

[2] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, 3rd ed. Waltham, MA: Elsevier, 2012.

[3] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, New York: Springer, 2009.

[4] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning with Applications in R*, New York: Springer, 2013.

[5] F. Pedregosa et al., “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[6] Gradio Team, “Gradio Documentation,” 2023. [Online]. Available: https://gradio.app

[7] Pandas Development Team, “Pandas Documentation,” 2023. [Online]. Available: https://pandas.pydata.org/[8] NumPy Developers, “NumPy Documentation,” 2023. [Online]. Available: https://numpy.org/doc/[9] Dinas Pendidikan Provinsi DKI Jakarta, “Dataset Jumlah Siswa SD Berdasarkan Jenis Kelamin,” 2025. [Online]. Available: https://data.jakarta.go.id/