

Linear Transformation

Ahmad Ardra - 71486
Alice Shizuka - 71485
Nabila Karin - 71478

Latar Belakang

Bu Ani adalah seorang guru matematika di sebuah sekolah menengah. Ia ingin membuat sebuah sistem **prediksi** kelulusan siswa berbasis machine learning sederhana, agar ke depannya bisa mengidentifikasi siswa-siswa yang berisiko tidak lulus berdasarkan data objektif.

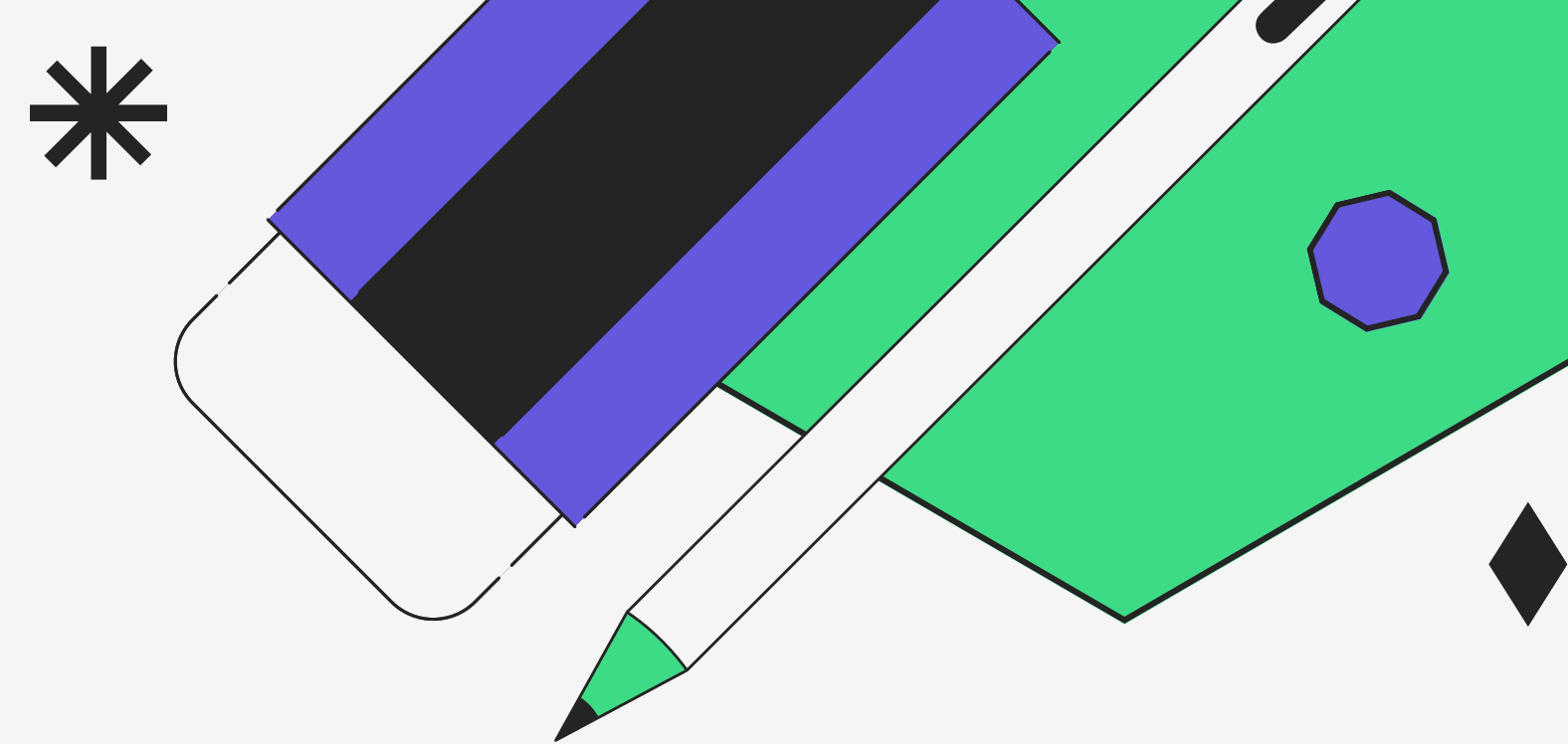
Dataset tersebut mencakup :

- **Nilai ujian akhir** dalam rentang 0–50 (berdasarkan jumlah soal benar).
- **Jumlah kehadiran** (maks 20 pertemuan).
- **Jumlah tugas yang dikerjakan** (maks 10 tugas).
- **Status kelulusan** (Lulus atau Tidak Lulus), ditentukan dari performa siswa.



Dataset

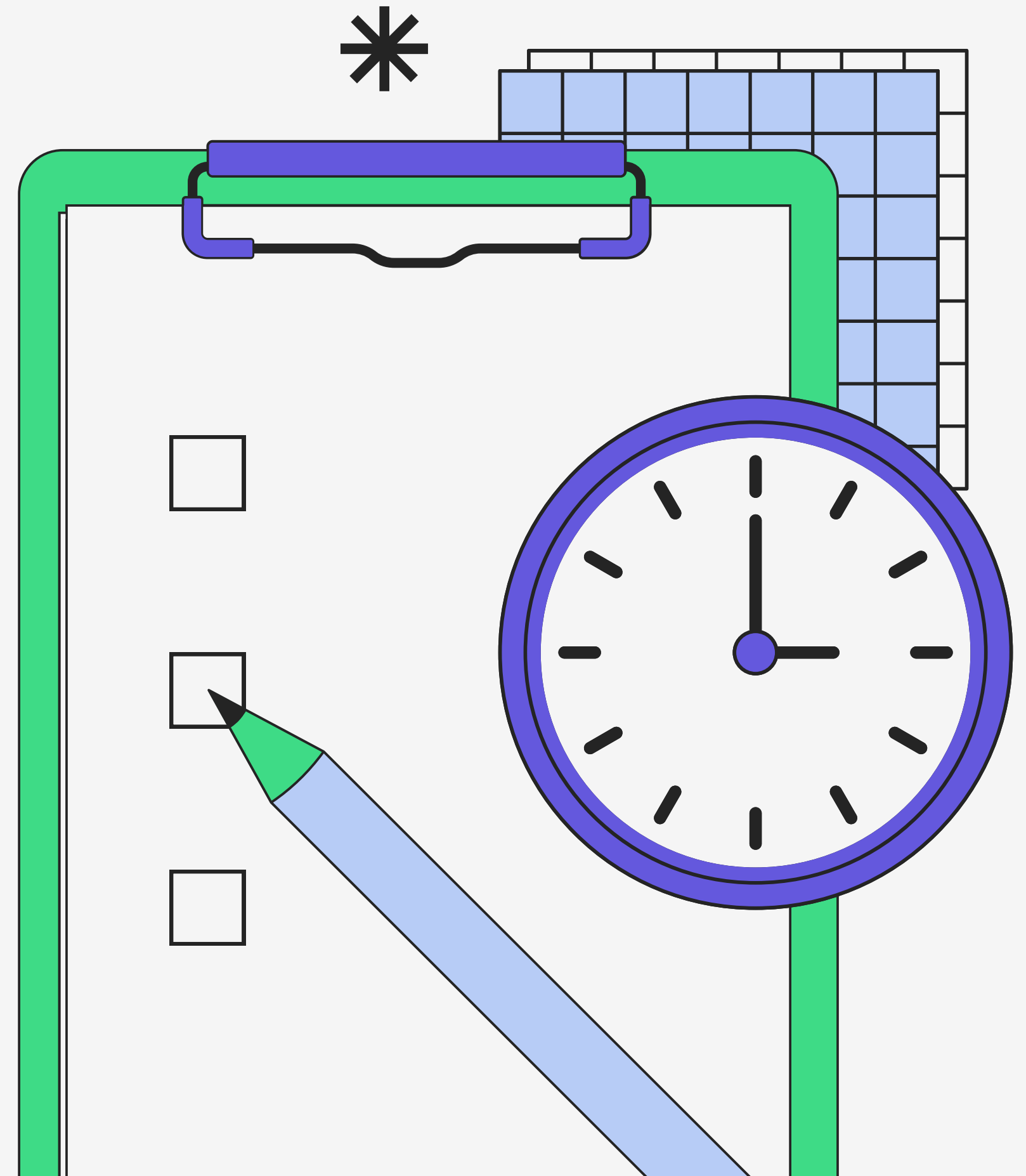
Before Linear Transformation



A <input type="checkbox"/> Nama	# Nilai_50	# Kehadiran	# Tugas	# Lulus
Ani	35	18	8	1
Budi	40	20	10	1
Cici	45	19	9	1
Deni	20	12	5	0
Eka	25	15	6	0
Fani	30	17	7	0
Gilang	50	20	10	1
Hana	10	8	3	0
Indra	15	10	4	0
Joko	43	19	9	1

Implementasi Transformasi Linear pada Pandas

- df.mul()
- Min-Max Scaling
- df.corr()



df.mul() + df.add()

Mengubah suatu nilai numerik dengan rumus matematis.

- 🔵 **df.mul()** → mengalikan setiap elemen dengan skalar
- 🔵 **df.add()** → menambahkan nilai konstan ke setiap elemen

Konversi nilai ujian ke skala 100

```
# 1. TL Manual: Ubah nilai ke skala 100  
df['Nilai_100'] = df['Nilai_50'].mul(2).add(0)
```

$$\text{Nilai}_{100} = 2 \cdot \text{Nilai}_{50}$$

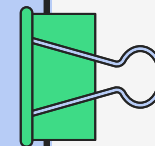
- $a = 2$
- $b = 0$

- menunjukkan bahwa setiap nilai dikalikan dua ($a = 2$)
- tidak ada penambahan nilai tetap ($b = 0$)



Rumus Transformasi Linear (Umum)

$$y = ax + b.$$



Min-Max Scaling

Scaling ini gunanya untuk menjaga hubungan proporsional data dan memudahkan algoritma dalam mengenali pola. Hal ini mempermudah proses pelatihan model karena model bisa konvergen lebih cepat dan tidak bias terhadap fitur dengan skala besar.

Setelah dikonversi ke skala 100, nilai dinormalisasi ke rentang [0, 1] menggunakan rumus:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

```
# 2. TL Scaling: Min-max scaling pada Nilai_100
```

```
df['Nilai_100_Scaled'] = (df['Nilai_100'] - df['Nilai_100'].min()) / (df['Nilai_100'].max() - df['Nilai_100'].min())
```



$$\text{Nilai_100_Scaled} = \frac{\text{Nilai_100} - \min(\text{Nilai_100})}{\max(\text{Nilai_100}) - \min(\text{Nilai_100})}$$

Bentuk TL-nya bisa ditulis:

$$y = \frac{1}{\max - \min} \cdot x - \frac{\min}{\max - \min}$$

- $a = \frac{1}{\max - \min}$
- $b = -\frac{\min}{\max - \min}$

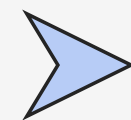
$$y = ax + b$$

df.corr()

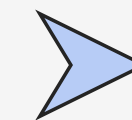
Adalah fungsi di Pandas untuk menghitung korelasi antar kolom numerik dalam sebuah DataFrame. df.corr() bukan merupakan transformasi linear. Tetapi Z-score adalah transformasi Linear

Rumus Z-score dalam bentuk Transformasi Linear:

$$z = \frac{x - \mu}{\sigma}$$



$$z = a \cdot x + b \quad \text{dengan} \quad a = \frac{1}{\sigma}, \quad b = -\frac{\mu}{\sigma}$$



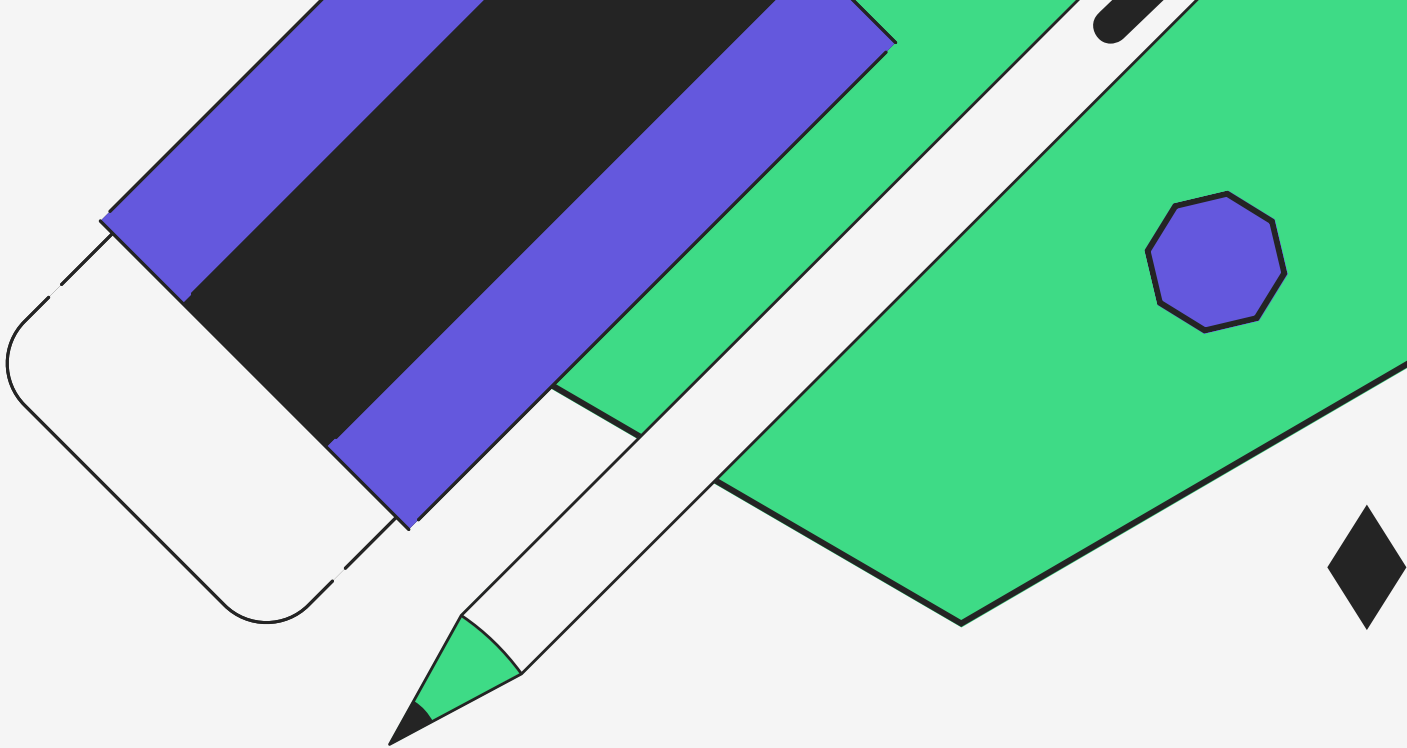
$$y = ax + b$$

```
# 3. Korelasi terhadap kelulusan
corr_matrix = df[['Nilai_100', 'Nilai_100_Scaled', 'Kehadiran', 'Tugas', 'Lulus']].corr()

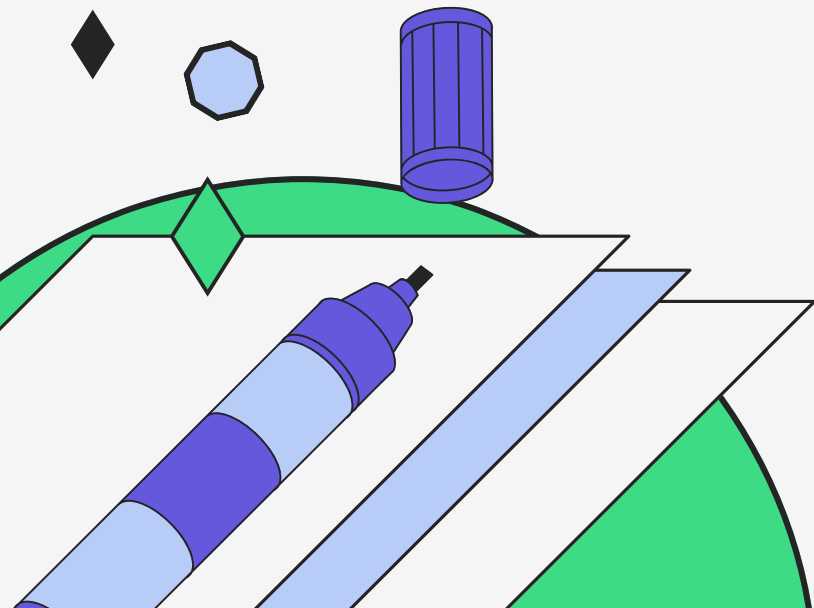
print(df)
print("\nKorelasi terhadap Kelulusan:")
print(corr_matrix['Lulus'].sort_values(ascending=False))
```

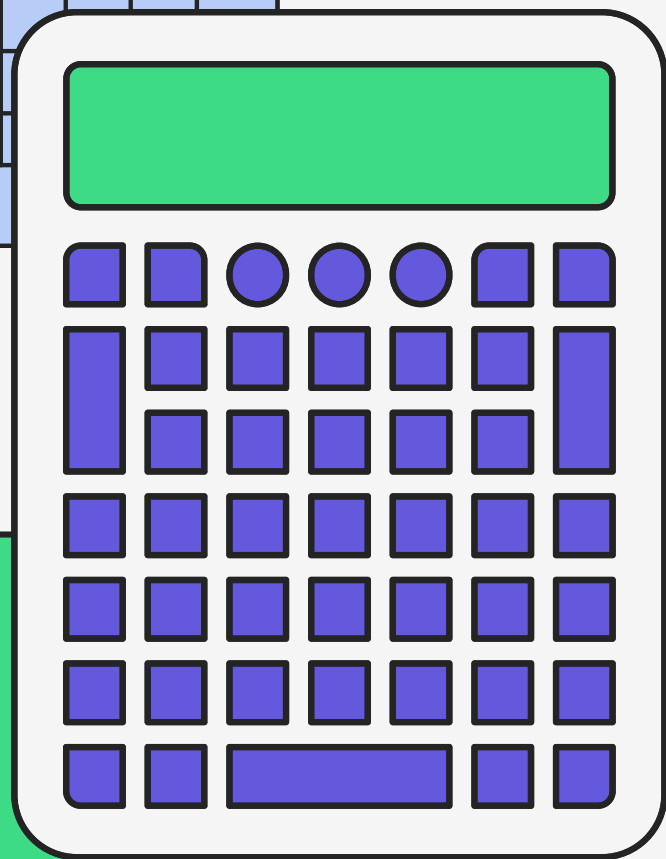
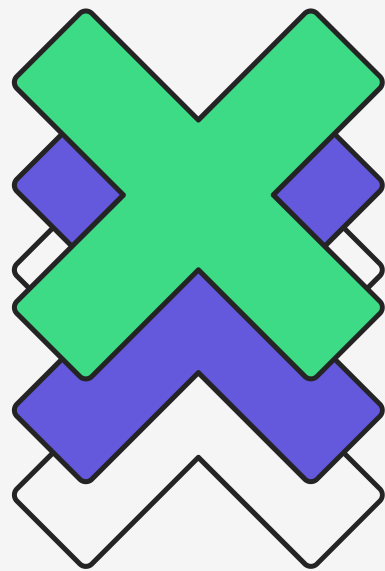
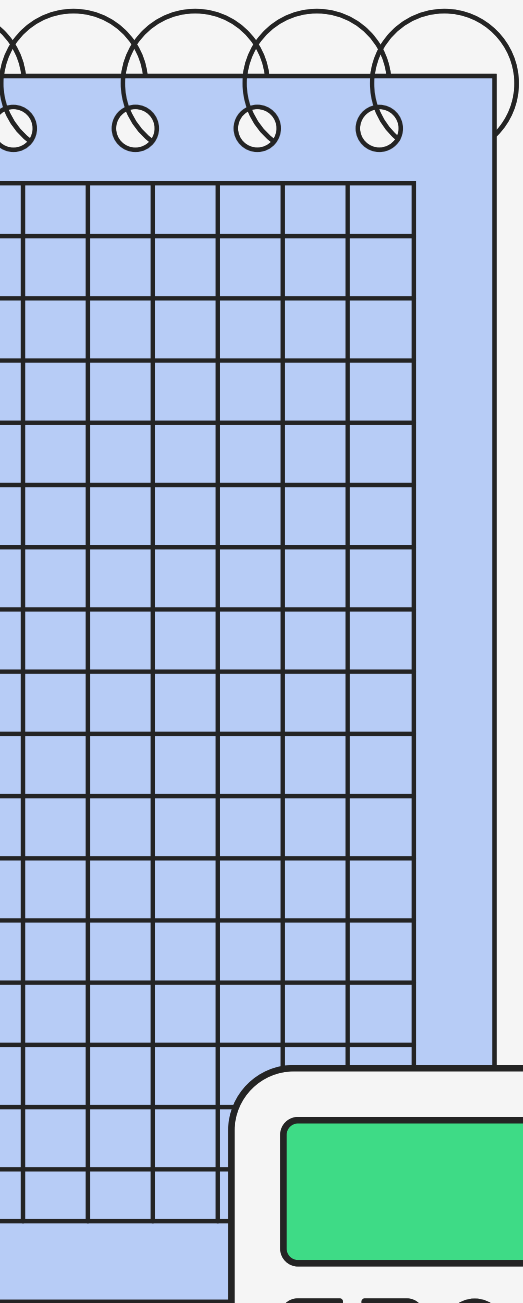
Dataset

After Linear Transformation



Nama	# Nilai_50	# Kehadiran	# Tugas	# Lulus	# Nilai_100	# Nilai_100_Scaled
Ani	35	18	8	1	70	0.625
Budi	40	20	10	1	80	0.75
Cici	45	19	9	1	90	0.875
Deni	20	12	5	0	40	0.25
Eka	25	15	6	0	50	0.375
Fani	30	17	7	0	60	0.5
Gilang	50	20	10	1	100	1.0
Hana	10	8	3	0	20	0.0
Indra	15	10	4	0	30	0.125
Joko	43	19	9	1	86	0.825





THANK
YOU

