

Tugas 12 : LAPORAN PRAKTIKUM MANDIRI

Syauqi Rabbani - 0110224208 ^{1*}

¹ Teknik Informatika, STT Terpadu Nurul Fikri, Depok

² Sistem Informasi, STT Terpadu Nurul Fikri, Depok

*E-mail: syauqi.rabbani36@gmail.com

1. Load Dataset

```
[13] df = pd.read_csv(f"{path}/data.csv")
df.head()

id diagnosis radius_mean texture_mean perimeter_mean area_mean smoothness_mean compactness_mean concavity_mean concave
points_mean ... texture_worst
0 842302 M 17.99 10.38 122.80 1001.0 0.11840 0.27760 0.3001 0.14710 ... 17.33
1 842517 M 20.57 17.77 132.90 1326.0 0.08474 0.07864 0.0869 0.07017 ... 23.41
2 84300993 M 19.69 21.25 130.00 1203.0 0.10960 0.15990 0.1974 0.12790 ... 25.53
3 84348301 M 11.42 20.38 77.58 386.1 0.14250 0.28390 0.2414 0.10520 ... 26.50
4 84358402 M 20.29 14.34 135.10 1297.0 0.10030 0.13280 0.1980 0.10430 ... 16.67
5 rows × 33 columns
```

Dataset medis dimuat menggunakan fungsi `pd.read_csv("data.csv")`. Dataset ini berisi data karakteristik sel kanker dengan jumlah fitur yang cukup banyak. Proses pemuatan data bertujuan agar seluruh data dapat diakses dan diolah secara sistematis, sehingga memudahkan proses preprocessing dan analisis PCA pada tahap selanjutnya..

2. Preprocessing Data

```
[5] df = df.drop(columns=["id", "Unnamed: 32"])
```

Pada tahap preprocessing, dilakukan pembersihan data dengan menghapus kolom `id` dan `Unnamed: 32`. Kedua kolom tersebut tidak memiliki makna statistik dan tidak berkontribusi terhadap proses analisis, sehingga perlu dihilangkan. Setelah itu, data difokuskan pada fitur numerik yang relevan agar PCA dapat bekerja secara optimal dalam menangkap variasi data.

3. Standarisasi Data

```
[7] X = df.drop(columns=["diagnosis"])
y = df["diagnosis"]

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

Sebelum PCA diterapkan, seluruh fitur distandarisasi menggunakan `StandardScaler`. Standarisasi dilakukan dengan mengubah nilai data agar memiliki rata-rata nol dan standar deviasi

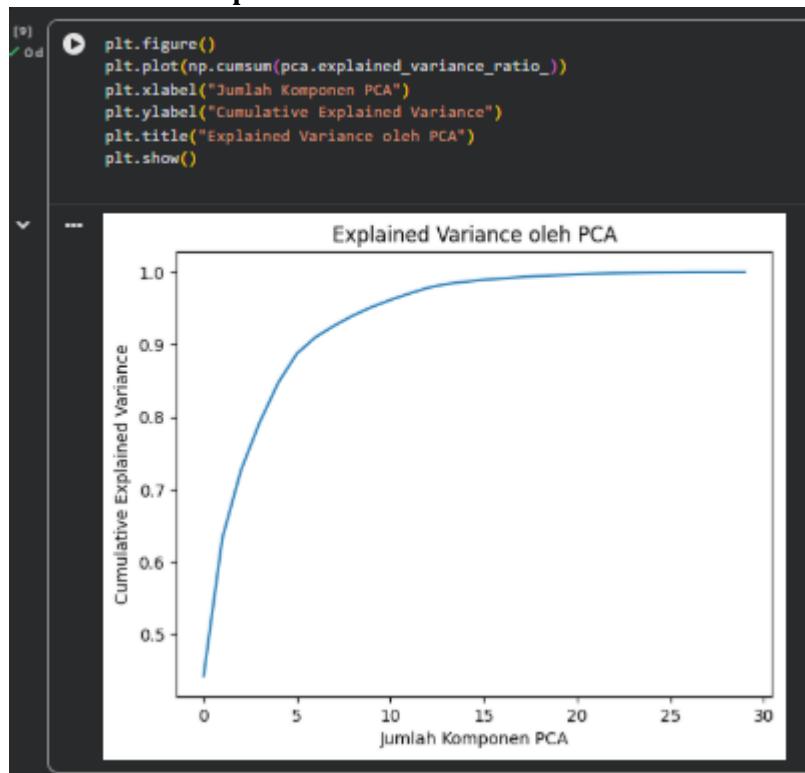
satu. Langkah ini sangat penting karena PCA sensitif terhadap perbedaan skala antar fitur, sehingga standarisasi memastikan setiap fitur memiliki pengaruh yang seimbang dalam pembentukan principal component.

4. PCA tanpa Menentukan Jumlah Komponen

```
[8]  pca = PCA()
    X_pca = pca.fit_transform(X_scaled)
```

PCA pertama kali diterapkan tanpa menentukan jumlah komponen menggunakan `pca()`. Tujuan dari langkah ini adalah untuk menghitung seluruh principal component yang mungkin dan mengetahui seberapa besar variasi data yang dapat dijelaskan oleh masing-masing komponen. Hasil ini menjadi dasar untuk menentukan jumlah komponen yang paling optimal digunakan pada tahap berikutnya.

5. Visualisasi Explained Variance



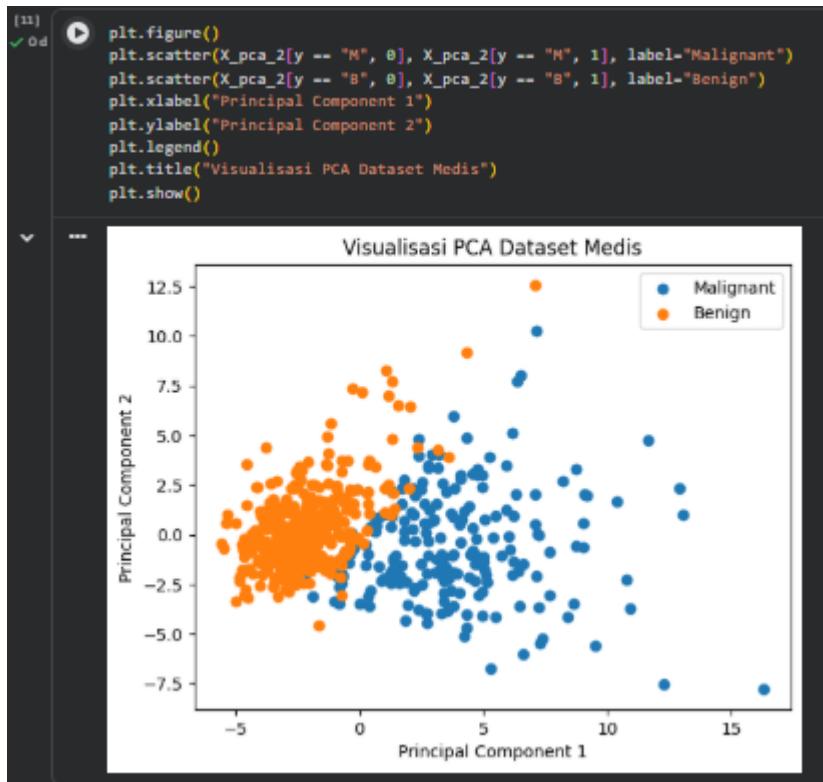
Nilai `explained_variance_ratio_` dari hasil PCA digunakan untuk membuat grafik cumulative explained variance. Grafik ini menunjukkan akumulasi proporsi variansi data yang dijelaskan oleh sejumlah principal component. Dengan melihat grafik ini, dapat ditentukan jumlah komponen PCA yang mampu mempertahankan sebagian besar informasi data, umumnya sekitar 90–95% dari total variansi.

6. PCA dengan 2 komponen

```
[10]  pca_2 = PCA(n_components=2)
    X_pca_2 = pca_2.fit_transform(X_scaled)
```

Setelah mengetahui distribusi variansi data, PCA diterapkan kembali dengan menentukan `n_components=2`. Langkah ini bertujuan untuk mereduksi dimensi data menjadi dua principal component utama. Reduksi dimensi ini dilakukan untuk menyederhanakan data berdimensi tinggi tanpa kehilangan informasi penting secara signifikan.

7. Visualisasi Hasil PCA



Hasil reduksi dimensi menggunakan dua principal component divisualisasikan dalam bentuk scatter plot. Visualisasi ini digunakan untuk melihat pola penyebaran data dan hubungan antar sampel. Selain itu, pemisahan antara kelas kanker jinak (Benign) dan ganas (Malignant) dapat diamati secara visual, sehingga membantu dalam memahami struktur data setelah dilakukan PCA.

8. Kontribusi Fitur terhadap Principal Component

```
[12] 0d
pca_components = pd.DataFrame(
    pca_2.components_,
    columns=X.columns,
    index=["PC1", "PC2"]
)

print("Kontribusi fitur terhadap PC1 dan PC2:")
print(pca_components.T.sort_values(by="PC1", ascending=False).head())

--- Kontribusi fitur terhadap PC1 dan PC2:
          PC1      PC2
concave points_mean  0.260854 -0.034768
concavity_mean      0.258400  0.060165
concave points_worst 0.250886 -0.008257
compactness_mean     0.239285  0.151892
perimeter_worst      0.236640 -0.199878
```

Pada tahap terakhir, ditampilkan kontribusi masing-masing fitur terhadap principal component menggunakan atribut `components_`. Nilai ini menunjukkan seberapa besar pengaruh setiap fitur terhadap PC1 dan PC2. Analisis kontribusi fitur ini penting untuk memahami karakteristik medis mana yang paling berperan dalam membentuk struktur data hasil reduksi dimensi.

LINK GITHUB : https://github.com/Syauqi366/SyauqiRabbani_MachineLearning/tree/main