

## *PCA (Principal Component Analysis)*

### Cara Kerja:

1. Standarisasi dataset sehingga memiliki mean 0 dan variansi 1.
2. Hitung matriks kovarian dari dataset.
3. Lakukan dekomposisi Eigen pada matriks kovarian untuk mendapatkan *eigenvalues* dan *eigenvectors*.
4. Urutkan *eigenvectors* berdasarkan *eigenvalues* tertinggi.
5. Pilih jumlah komponen utama (*principal components*) yang diinginkan dan bentuk matriks proyeksi.
6. Proyeksikan data asli ke ruang baru dengan komponen utama.

### Perbandingan Hasil:

#### Dimensionality Reduction with PCA

```
[15]: X = df.drop('NObeyesdad', axis = 1)
```

#### Scratch

```
[16]: pca_scratch = PCAScratch(n_components = 2)
```

```
[17]: X_pca_scratch = pca_scratch.fit_transform(X)
```

```
[18]: pca_scratch.explained_variance_
```

```
[18]: array([6.41815203e+09, 8.71226477e+01])
```

```
[19]: pca_scratch.explained_variance_ratio_
```

```
[19]: array([9.99999980e-01, 1.35744129e-08])
```

#### Library

```
[20]: pca = PCA(n_components = 2)
```

```
[21]: X_pca = pca.fit_transform(X)
```

```
[22]: pca.explained_variance_
```

```
[22]: array([6.41815203e+09, 8.71226477e+01])
```

```
[23]: pca.explained_variance_ratio_
```

```
[23]: array([9.99999980e-01, 1.35744129e-08])
```

Dengan melihat nilai explained variance dan explained variance ratio yang ada, algoritma PCA antara *scratch* dengan *library* tidak memiliki perbedaan.