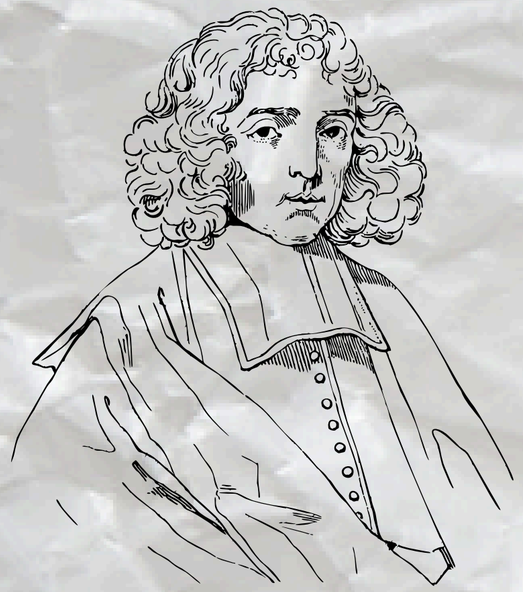
Hierarchical Clustering on Indians Diabetes

Our Members



Vito - Odi - Calvin (VOC)

Knowledge Source

This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. The objective of the dataset is to diagnostically predict **whether or not a patient has diabetes**, based on certain diagnostic measurements included in the dataset. Several constraints were placed on the selection of these instances from a larger database. In particular, all patients here are **females** at least 21 years old of **Pima Indian** heritage where lived along the Gila and Salt Rivers in Arizona, United States.

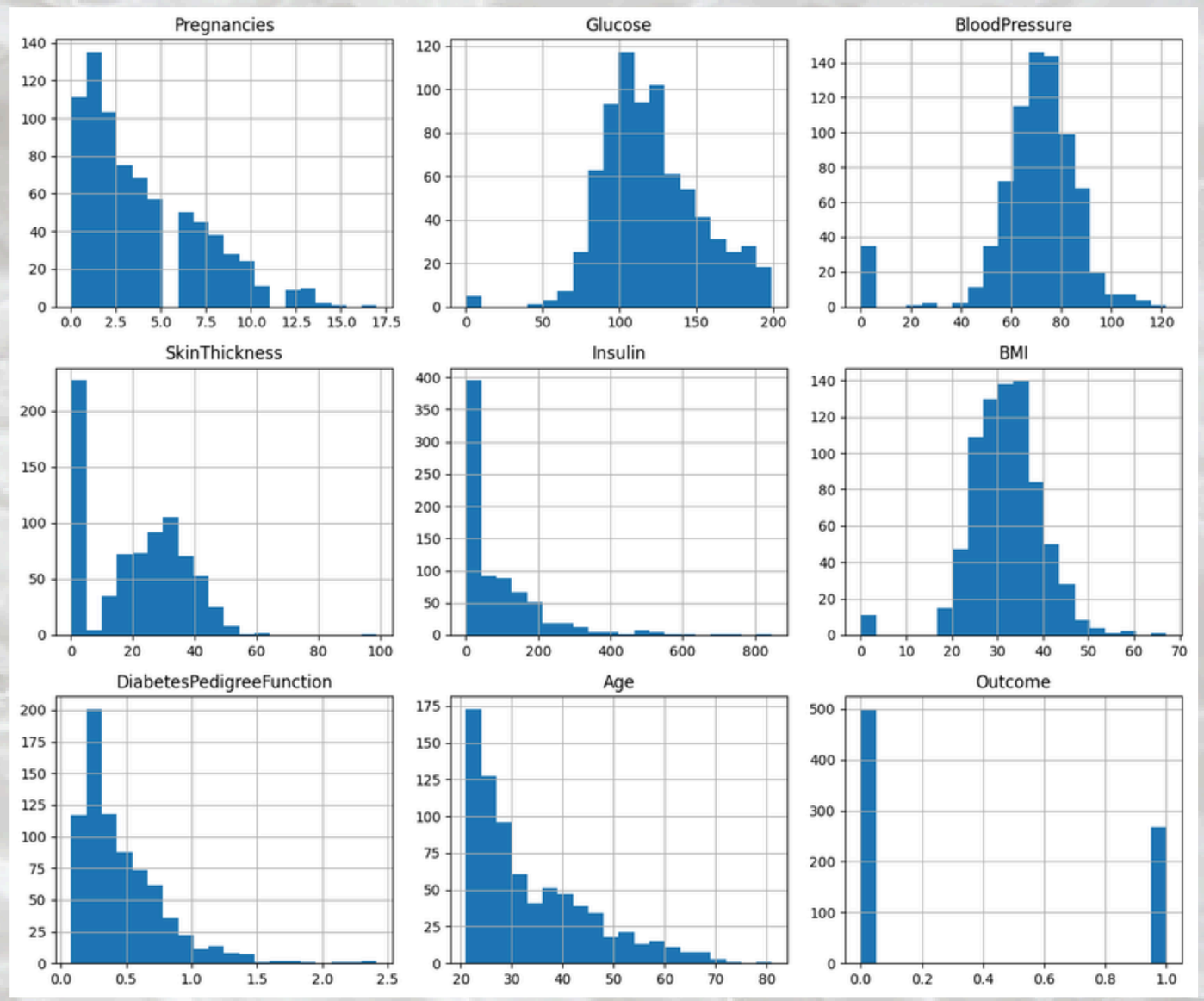
Dataset Source

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
...	...	...	...	...	...	...	...	...	...
763	10	101	76	48	180	32.9	0.171	63	0
764	2	122	70	27	0	36.8	0.340	27	0
765	5	121	72	23	112	26.2	0.245	30	0
766	1	126	60	0	0	30.1	0.349	47	1
767	1	93	70	31	0	30.4	0.315	23	0

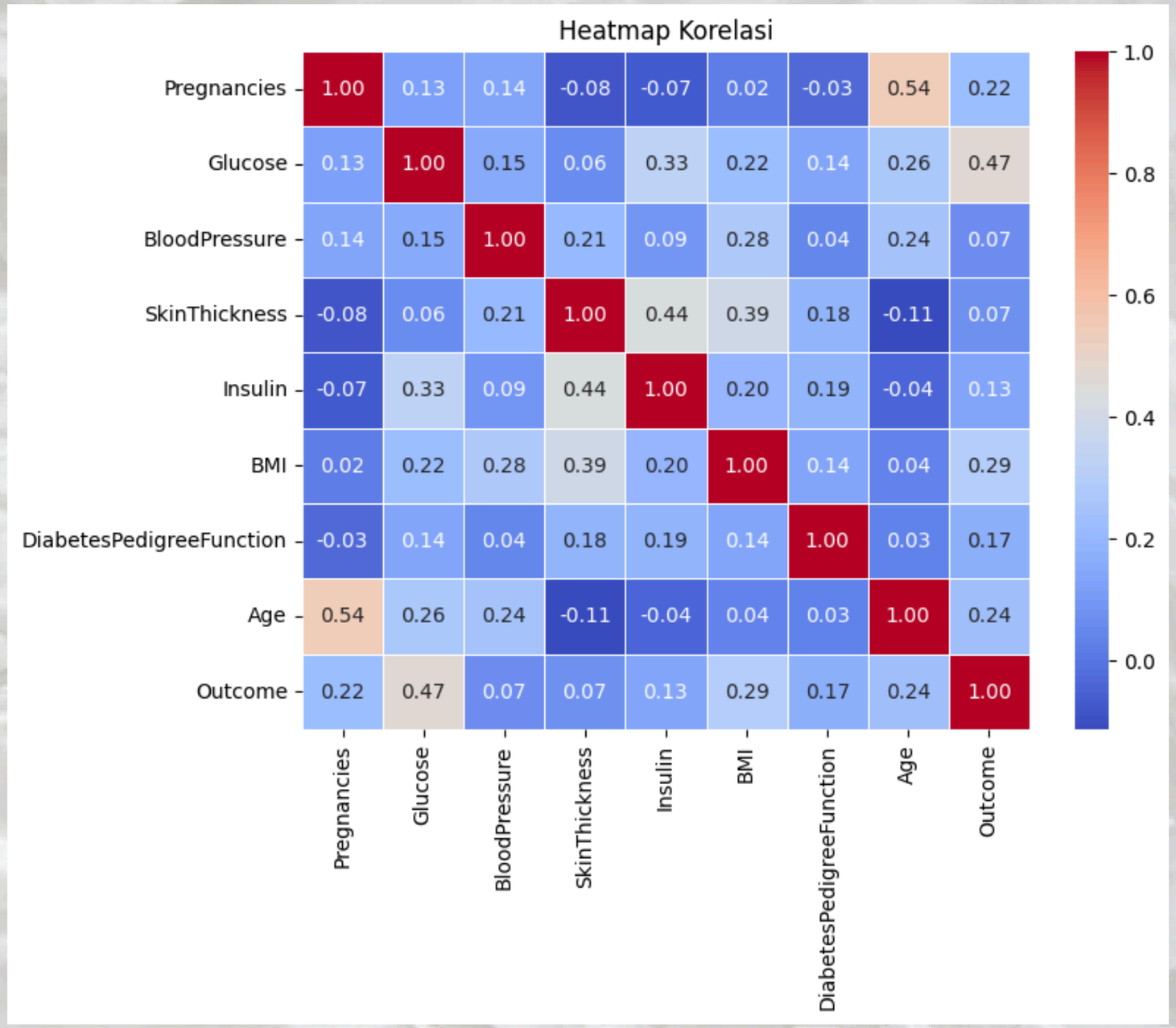
```
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   Pregnancies               768 non-null    int64  
 1   Glucose                   768 non-null    int64  
 2   BloodPressure             768 non-null    int64  
 3   SkinThickness             768 non-null    int64  
 4   Insulin                   768 non-null    int64  
 5   BMI                       768 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64  
 8   Outcome                   768 non-null    int64  
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

- **Pregnancies:** Jumlah total kehamilan yang pernah dialami.
- **Glucose:** Kadar gula dalam darah. Indikator kunci diabetes.
- **BloodPressure:** Tekanan darah diastolik (angka bawah).
- **SkinThickness:** Ketebalan lipatan kulit, digunakan untuk mengukur lemak tubuh.
- **Insulin:** Kadar hormon insulin dalam darah.
- **BMI (Body Mass Index):** Indeks Massa Tubuh, ukuran obesitas berdasarkan berat dan tinggi badan.
- **DiabetesPedigreeFunction:** Skor yang menunjukkan risiko diabetes berdasarkan riwayat keluarga.
- **Age:** Usia pasien dalam tahun.
- **Outcome:** Kolom target; 1 berarti pasien menderita diabetes, 0 berarti tidak.

Distribusi Data Awal



Correlation Heatmap Before Manipulation



Data Manipulation

	Zero_Count	Zero_Percent
Pregnancies	111	14.45
Glucose	5	0.65
BloodPressure	35	4.56
SkinThickness	227	29.56
Insulin	374	48.70
BMI	11	1.43
DiabetesPedigreeFunction	0	0.00
Age	0	0.00
Outcome	500	65.10

```
# Drop the specified columns from the features
df_treated = df_treated.drop(columns=['Insulin', 'SkinThickness'])
print("Columns remaining in features:", df_treated.columns.tolist())
```

```
cols_with_invalid_zeros = ["Glucose", "BloodPressure", "BMI"]

df_treated = df.copy()
df_treated[cols_with_invalid_zeros] = df_treated[cols_with_invalid_zeros].replace(0, np.nan)

# mean impute for Glucose, BMI, BloodPressure, SkinThickness
mean_cols = ["Glucose", "BMI", "BloodPressure"]
mean_imp = SimpleImputer(strategy='mean')
df_treated[mean_cols] = mean_imp.fit_transform(df_treated[mean_cols])
```

Data Manipulation

From This:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
...	...	...	...	...	...	...	...	...	...
763	10	101	76	48	180	32.9	0.171	63	0
764	2	122	70	27	0	36.8	0.340	27	0
765	5	121	72	23	112	26.2	0.245	30	0
766	1	126	60	0	0	30.1	0.349	47	1
767	1	93	70	31	0	30.4	0.315	23	0

To This:

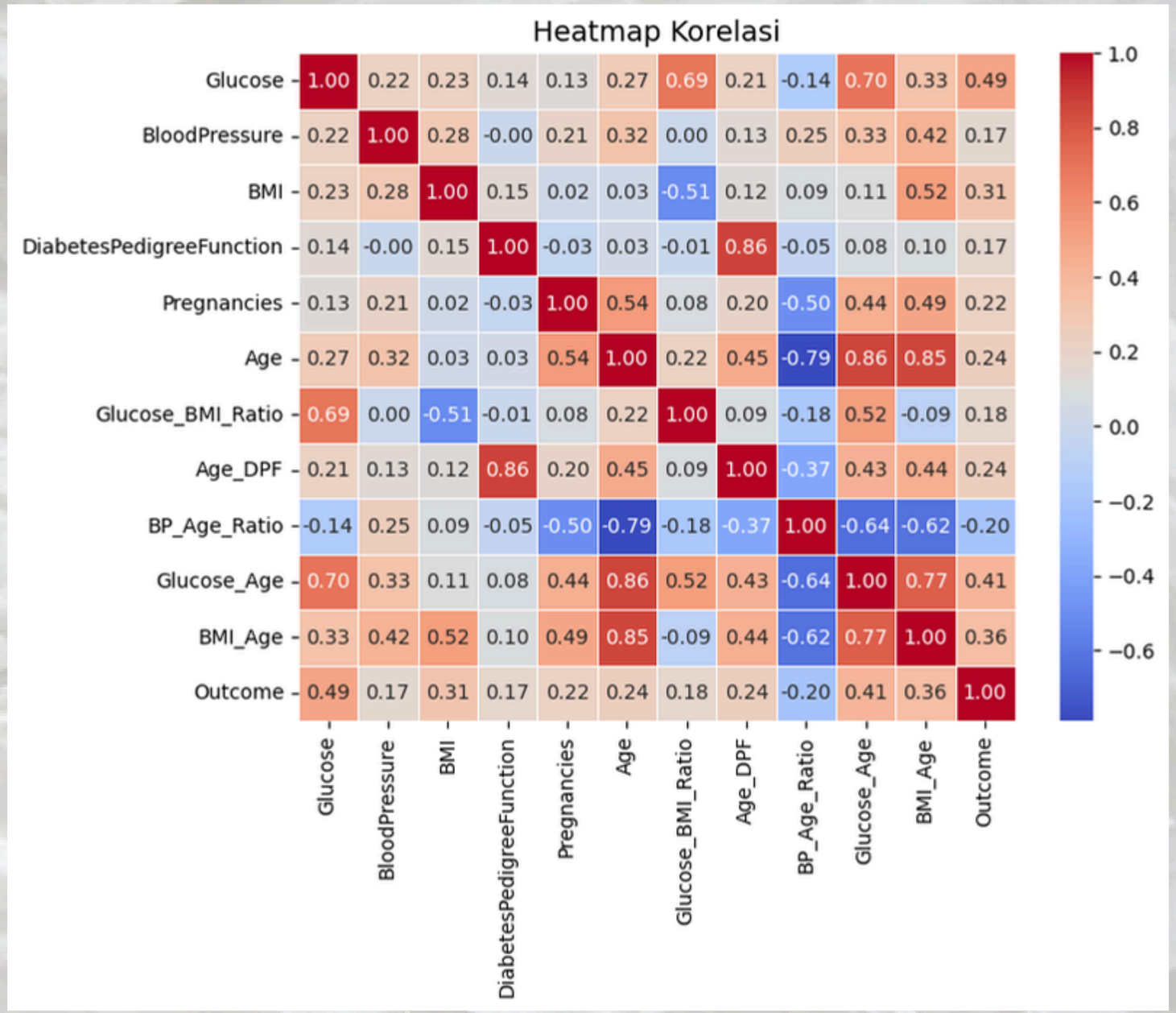
	Pregnancies	Glucose	BloodPressure	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148.0	72.0	33.6	0.627	50	1
1	1	85.0	66.0	26.6	0.351	31	0
2	8	183.0	64.0	23.3	0.672	32	1
3	1	89.0	66.0	28.1	0.167	21	0
4	0	137.0	40.0	43.1	2.288	33	1
...	...	...	...	...	...	...	...
763	10	101.0	76.0	32.9	0.171	63	0
764	2	122.0	70.0	36.8	0.340	27	0
765	5	121.0	72.0	26.2	0.245	30	0
766	1	126.0	60.0	30.1	0.349	47	1
767	1	93.0	70.0	30.4	0.315	23	0

Feature Engineering

	Pregnancies	Glucose	BloodPressure	BMI	DiabetesPedigreeFunction	Age	Outcome	Glucose_BMI_Ratio	Age_DPF	BP_Age_Ratio	Glucose_Age	BMI_Age
0	6	148.0	72.0	33.6	0.627	50	1	4.404762	31.350	1.411765	7400.0	1680.0
1	1	85.0	66.0	26.6	0.351	31	0	3.195489	10.881	2.062500	2635.0	824.6
2	8	183.0	64.0	23.3	0.672	32	1	7.854077	21.504	1.939394	5856.0	745.6
3	1	89.0	66.0	28.1	0.167	21	0	3.167260	3.507	3.000000	1869.0	590.1
4	0	137.0	40.0	43.1	2.288	33	1	3.178654	75.504	1.176471	4521.0	1422.3
...	...	...	...	...	...	...	...	...	...	...	...	...
763	10	101.0	76.0	32.9	0.171	63	0	3.069909	10.773	1.187500	6363.0	2072.7
764	2	122.0	70.0	36.8	0.340	27	0	3.315217	9.180	2.500000	3294.0	993.6
765	5	121.0	72.0	26.2	0.245	30	0	4.618320	7.350	2.322581	3630.0	786.0
766	1	126.0	60.0	30.1	0.349	47	1	4.186046	16.403	1.250000	5922.0	1414.7
767	1	93.0	70.0	30.4	0.315	23	0	3.059210	7.245	2.916667	2139.0	699.2

```
Missing per column after treatment:
Pregnancies                 0
Glucose                     0
BloodPressure               0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
Glucose_BMI_Ratio           0
Age_DPF                     0
BP_Age_Ratio                0
Glucose_Age                 0
BMI_Age                     0
dtype: int64
```

Correlation Heatmap After Manipulation



Metodologi

1. Gradient Boosting untuk Feature Selection
2. Standard Scaler dan Dendogram
3. Hierarchical Clustering dengan Agglomerative Approach

Metodologi

1 Gradient Boosting untuk Feature Selection

```python
# Feature Selection with Gradient Boosting
from sklearn.ensemble import GradientBoostingClassifier
import pandas as pd

# Features to consider
features_all = [
    "Glucose", "BloodPressure", "BMI",
    "DiabetesPedigreeFunction", "Age",
    "Glucose_BMI_Ratio", "Age_DPF", "BP_Age_Ratio",
    "Glucose_Age", "BMI_Age", 'Pregnancies'
]

X_fs = df_treated[features_all]
y_fs = df_treated['Outcome']

# Fit Gradient Boosting model
gb_model = GradientBoostingClassifier(
    n_estimators=300,
    learning_rate=0.05,
    max_depth=3,
    random_state=42
)
gb_model.fit(X_fs, y_fs)

# Ambil feature importance
importances = pd.Series(gb_model.feature_importances_, index=features_all).sort_values(ascending=False)
print("=== Gradient Boosting Feature Importances ===")
print(importances)

# Pilih top-N fitur
top_features = importances.head(6).index.tolist()
print("\nTop Features for Clustering (Gradient Boosting):", top_features)
```

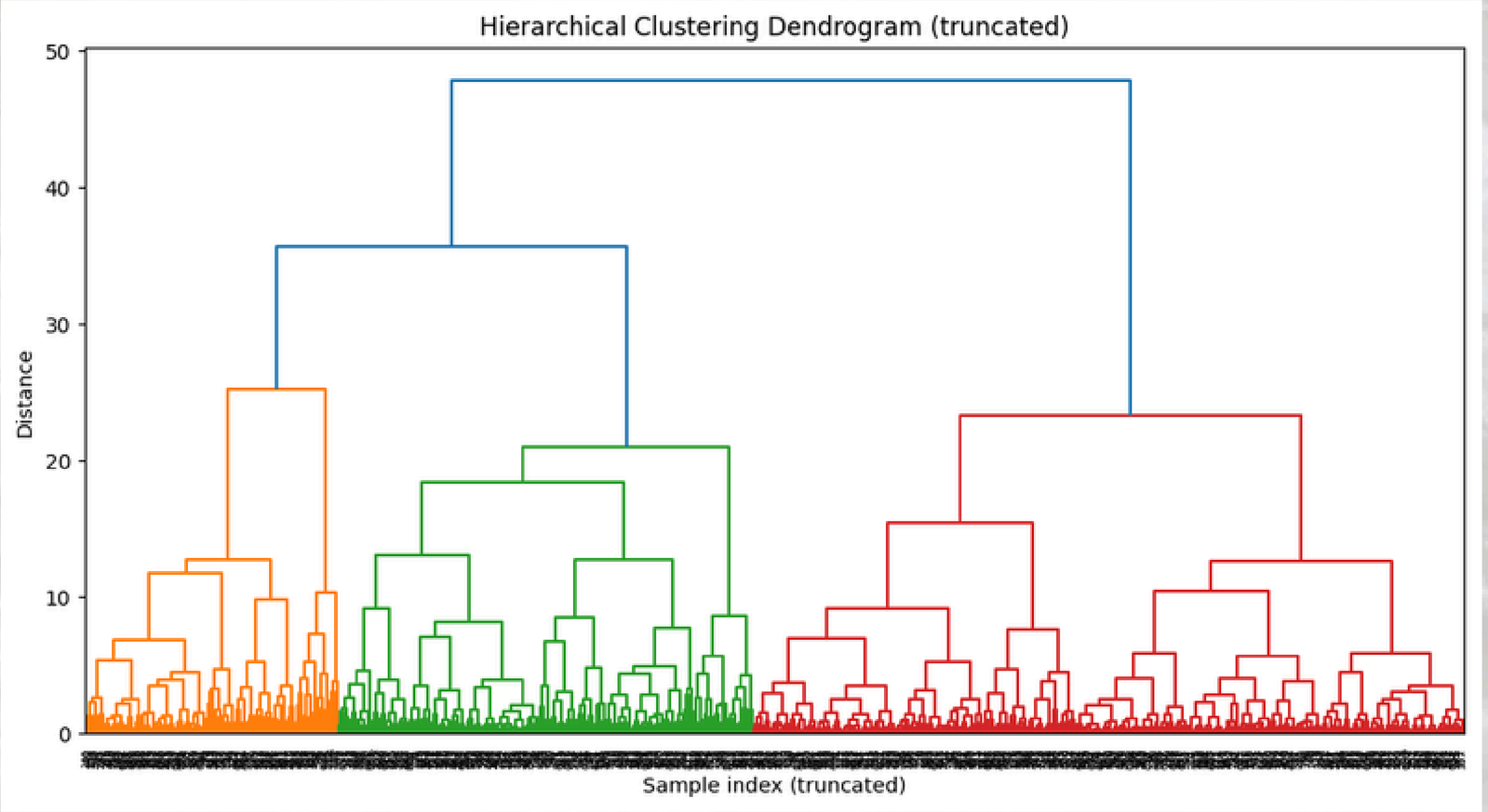
```
=== Gradient Boosting Feature Importances ===
Glucose_Age                 0.321528
Glucose                     0.150833
BMI                         0.142342
BMI_Age                     0.098601
Age_DPF                     0.074437
DiabetesPedigreeFunction    0.056232
Glucose_BMI_Ratio           0.041532
Pregnancies                 0.041082
Age                         0.032774
BP_Age_Ratio                0.030313
BloodPressure               0.010326
dtype: float64
```

Top Feature :
['Glucose_Age', 'Glucose', 'BMI', 'BMI_Age', 'Age_DPF', 'DiabetesPedigreeFunction']

Metodologi

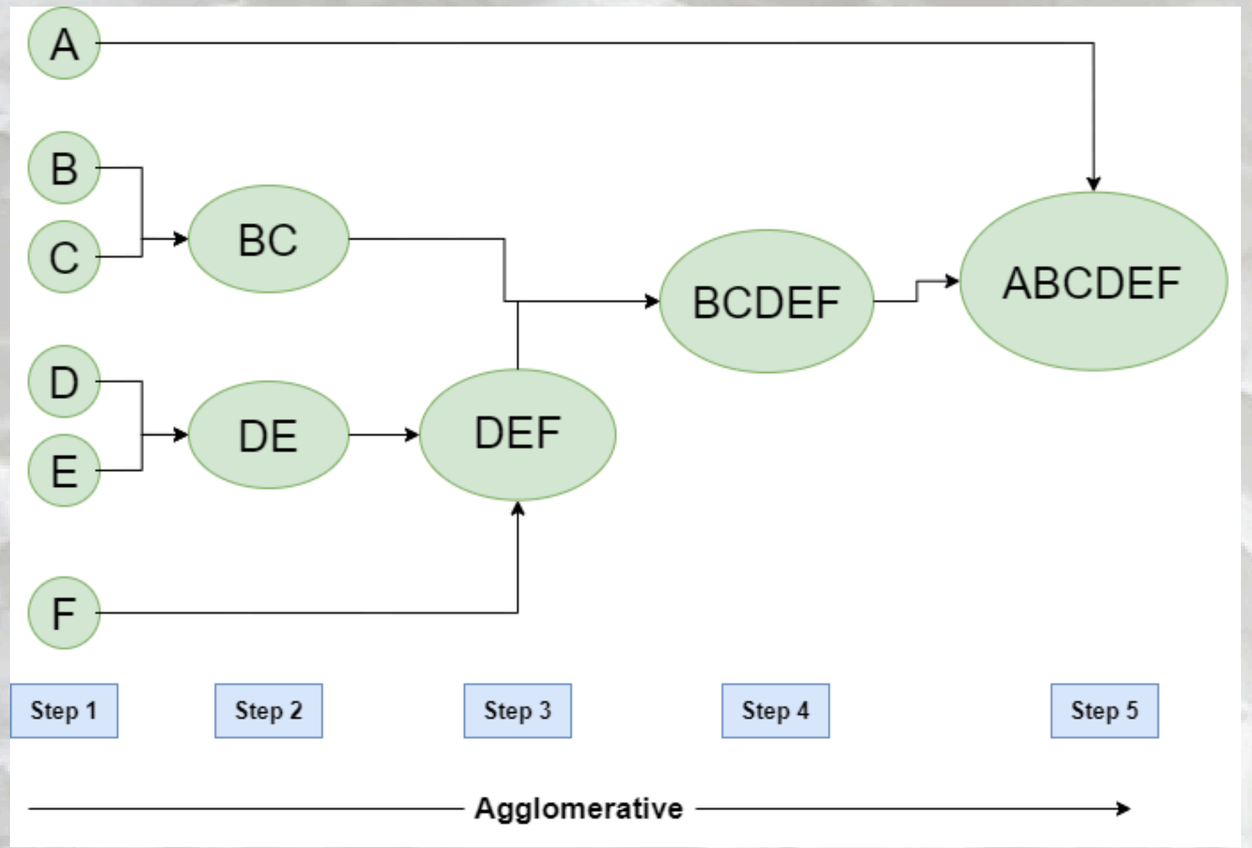
2 Standard Scaler dan Dendogram

```
features_selection =
top_features # From Gradient
Boosting

X_complete =
df_treated[features_selectio
n].copy()
X = X_complete

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



Metodologi

3 Hierarchical Clustering dengan Agglomerative Approach



it is also known as the bottom-up approach or hierarchical agglomerative clustering (HAC). Bottom-up algorithms treat each data as a singleton cluster at the outset and then successively agglomerate pairs of clusters until all clusters have been merged into a single cluster that contains all data.

Metodologi

3 Hierarchical Clustering dengan Agglomerative Approach

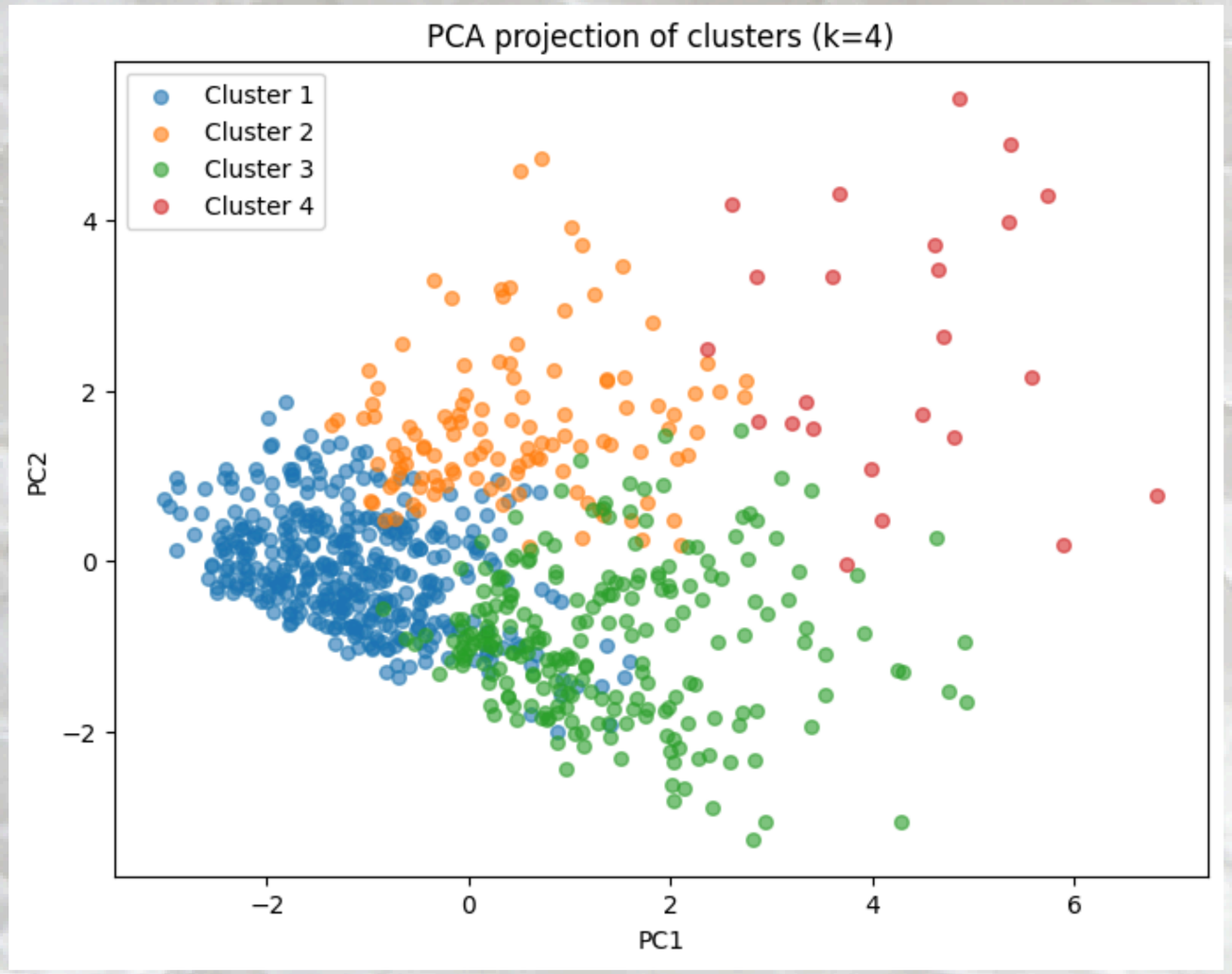
```python
def try_agglomerative(X_scaled, y_true, ks=[2,3,4,5]):
    results = {}
    for k in ks:
        model = AgglomerativeClustering(n_clusters=k, linkage='ward')
        labels = model.fit_predict(X_scaled) + 1

        # silhouette
        sil = silhouette_score(X_scaled, labels)

        # accuracy (purity, pakai majority label mapping)
        cluster_to_label = {}
        for cl in np.unique(labels):
            mask = labels == cl
            maj_label = np.bincount(y_true[mask]).argmax()
            cluster_to_label[cl] = maj_label
        y_pred = np.array([cluster_to_label[cl] for cl in labels])
        accuracy = accuracy_score(y_true, y_pred)

        results[k] = {
            'model': model,
            'labels': labels,
            'silhouette': sil,
            'accuracy': accuracy,
            'cluster_counts': np.bincount(labels)
        }

        print(f'k={k:2d}  silhouette={sil:.4f}  accuracy={accuracy:.4f}  cluster_counts={np.bincount(labels)}')
    return results
```

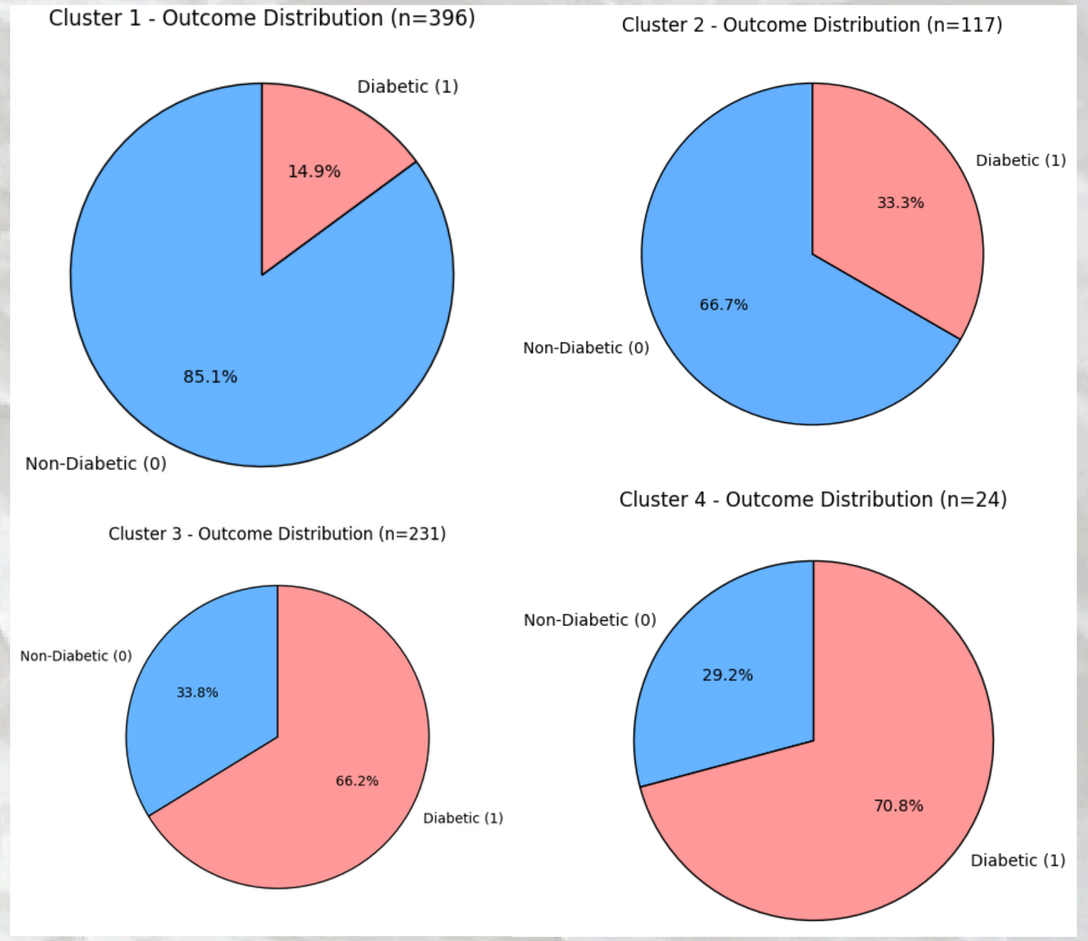
Result

```
k= 2  silhouette=0.2342  accuracy=0.7109  cluster_counts=[  0 372 396]
k= 3  silhouette=0.2577  accuracy=0.7487  cluster_counts=[  0 141 396 231]
k= 4  silhouette=0.2398  accuracy=0.7617  cluster_counts=[  0 396 117 231  24]
k= 5  silhouette=0.1605  accuracy=0.7617  cluster_counts=[  0 231 117 179  24 217]

Best k by Accuracy: 4
accuracy for k=4: 0.7617
```

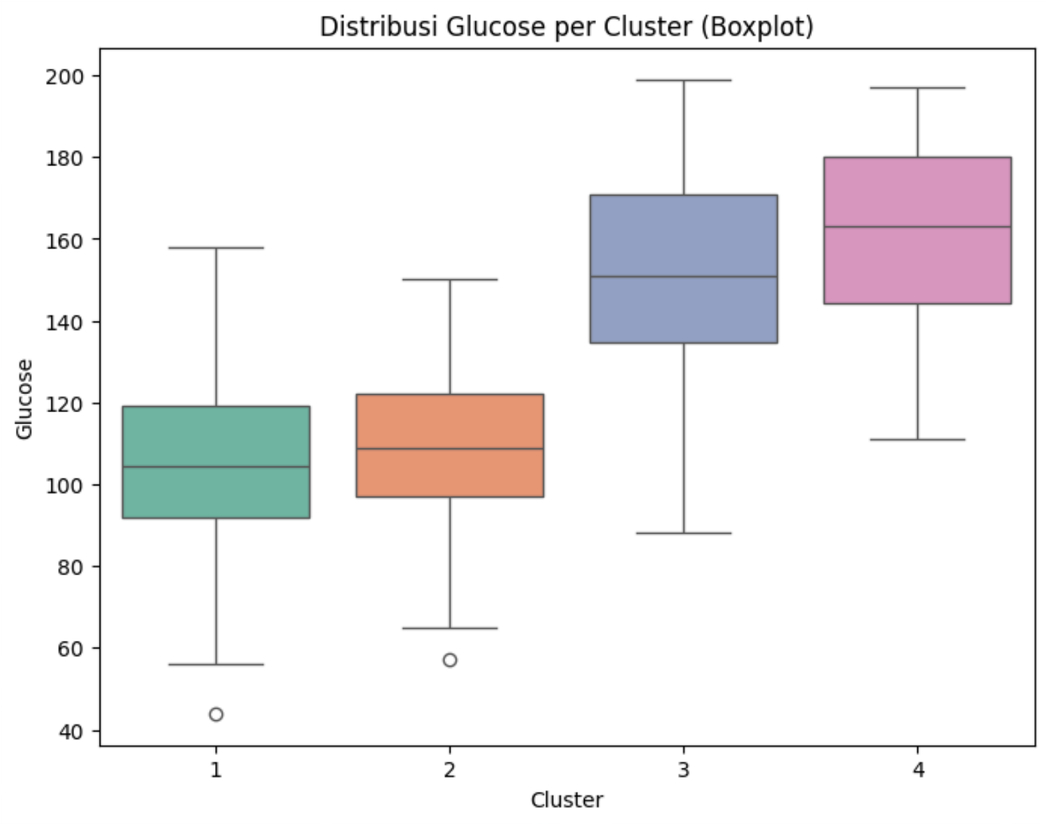
```
Cluster means (original feature scale):
         Glucose_Age     Glucose        BMI      BMI_Age    Age_DPF  \
cluster
1        2965.852625  105.171869  30.635397   876.186057   8.888967
2        3633.555556  109.658120  31.242371  1026.372707  28.450427
3        6126.244881  152.179106  35.736179  1407.691623  16.762359
4        6872.833333  159.333333  36.887500  1581.545833  59.416500

         DiabetesPedigreeFunction   Outcome
cluster
1                        0.321179  0.148990
2                        0.875017  0.333333
3                        0.425900  0.662338
4                        1.435583  0.708333
```

Result

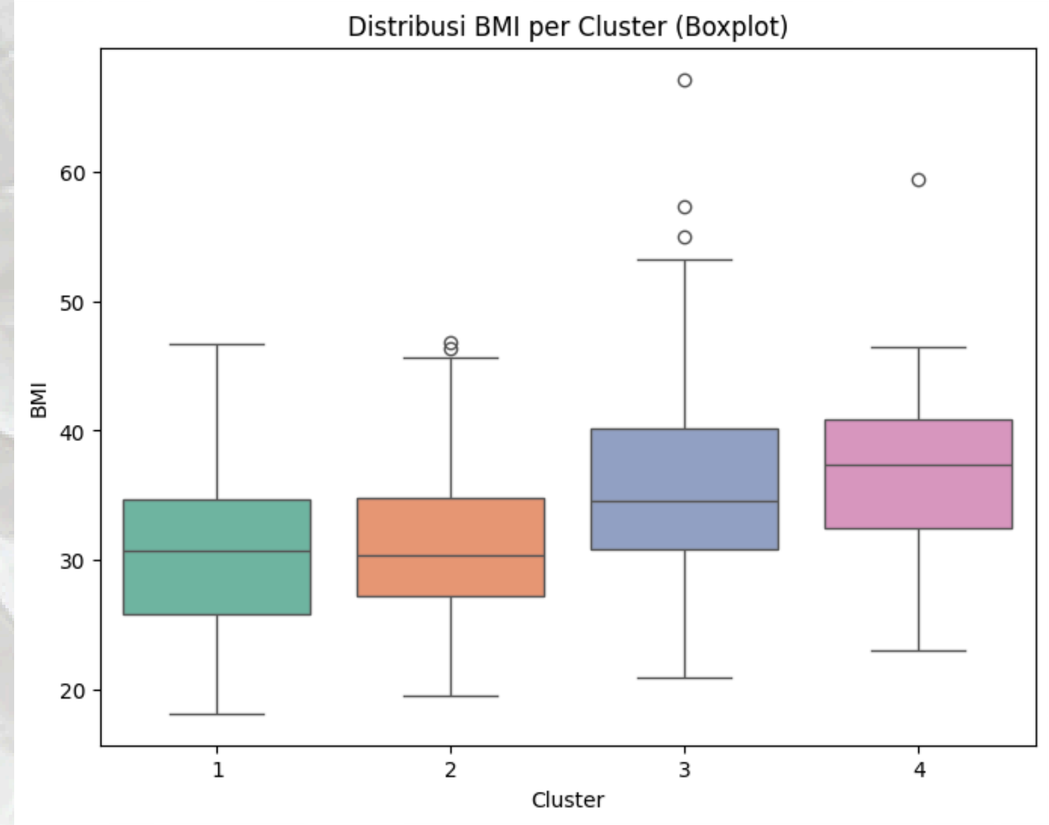

PCA projection of clusters (k=4)
Cluster 1
Cluster 2
Cluster 3
Cluster 4
PC1
PC2
−2
0
2
4
6
−2
0
2
4


Cluster 1 - Outcome Distribution (n=396)
Diabetic (1)
14.9%
85.1%
Non-Diabetic (0)
Cluster 2 - Outcome Distribution (n=117)
Diabetic (1)
33.3%
66.7%
Non-Diabetic (0)
Cluster 3 - Outcome Distribution (n=231)
Non-Diabetic (0)
33.8%
66.2%
Diabetic (1)
Cluster 4 - Outcome Distribution (n=24)
Non-Diabetic (0)
29.2%
70.8%
Diabetic (1)

Boxplot- Glucose & BMI per Cluster



- Cluster 1 → Median Glucose ~100, sebaran rendah, mayoritas normal
- Cluster 2 → Median Glucose ~110, sedikit lebih tinggi dari Cluster 1
- Cluster 3 → Median Glucose ~150, sebaran lebar, mayoritas di atas ambang diabetes
- Cluster 4 → Median Glucose ~160, distribusi terkonsentrasi di level sangat tinggi



- Cluster 1 → Median BMI sekitar 30.6, mayoritas normal–overweight.
- Cluster 2 → Median BMI sekitar 29.8, paling rendah dibanding cluster lain.
- Cluster 3 → Median BMI sekitar 34.5, dengan variasi besar dan beberapa outlier obesitas.
- Cluster 4 → Median BMI sekitar 37.9, paling tinggi dan dominan overweight–obesitas.

Summary

📊 Ringkasan per Cluster

Cluster	n	Prevalensi Diabetes	Risk Level	Ciri Utama	Rekomendasi
1	396	14.9%	Low	Mayoritas pasien di cluster ini masih muda, dengan kadar gula relatif normal dan BMI tidak terlalu tinggi.	pencegahan: edukasi gaya hidup sehat & tetap menjaga berat badan.
2	117	33.33%	Medium	Di sini faktor yang menonjol adalah riwayat keluarga / faktor genetik	screening keluarga dan pemantauan rutin.
3	231	66.23%	High	Di kelompok ini, kadar gula dan BMI sudah jelas tinggi, ditambah usia lebih tua.	Monitoring medis & manajemen glukosa lebih agresif; intervensi gaya hidup & penurunan berat badan
4	24	70.83%	Very High/Critical	kadar gula sangat tinggi, BMI sangat tinggi, dan faktor keluarga juga kuat.	Monitoring medis & manajemen glukosa lebih agresif; perhatikan riwayat keluarga; screening keluarga disarankan; intervensi gaya hidup & penurunan berat badan

Thank You

"Like hierarchical clustering, the VOC tried to split us into branches they could control.
Yet, just as branches lead back to one trunk, Indonesia found its root: independence." – VOC