



**Klasifikasi
Karakteristik Mahasiswa
Berpotensi Drop-Out Menggunakan
Algoritma K-Nearest Neighbors
yang Ditingkatkan dengan
Principal Component Analysis**

By Adisty Regina Pramnesti

Identifikasi Research Gap

Keterbatasan Metode Sebelumnya

Penelitian sebelumnya menunjukkan bahwa meskipun algoritma klasifikasi telah digunakan, masih ada tantangan dalam mengidentifikasi mahasiswa dengan potensi drop-out secara akurat, terutama dalam kondisi data yang kompleks.

Penggunaan PCA dan KNN

Meskipun PCA dan KNN telah digunakan secara terpisah dalam penelitian sebelumnya, disini saya mencoba menggabungkan kedua metode ini untuk meningkatkan akurasi prediksi potensi drop-out mahasiswa.

Variabel yang di perhitungkan

Penelitian sebelumnya mungkin tidak mempertimbangkan semua variabel yang relevan (misalnya, status ekonomi, kehadiran, dan faktor demografis lainnya) dalam model prediksi mereka karena menggunakan PCA



Rumusan Permasalahan

Bagaimana melakukan klasifikasi performa mahasiswa yang berpotensi drop out menggunakan kombinasi Principal Component Analysis (PCA) untuk reduksi dimensi dan optimasi algoritma K-Nearest Neighbors (KNN) guna meningkatkan akurasi klasifikasi?



Tujuan Penelitian

01

Mengimplementasikan algoritma K-Nearest Neighbor (KNN) yang ditingkatkan dengan PCA (Principal Component Analysis) untuk mengklasifikasikan data karakteristik mahasiswa yang berpotensi drop-out di salah satu kampus di Jawa Timur.

02

Merancang model algoritma K-Nearest Neighbor (KNN) yang ditingkatkan dengan PCA (Principal Component Analysis) untuk mengklasifikasikan data karakteristik mahasiswa yang berpotensi drop-out di salah satu kampus di Jawa Timur.

03

Mengetahui Tingkat akurasi model algoritma K-Nearest Neighbor (KNN) yang ditingkatkan dengan PCA (Principal Component Analysis) untuk mengklasifikasikan data karakteristik mahasiswa yang berpotensi drop-out di salah satu kampus di Jawa Timur.

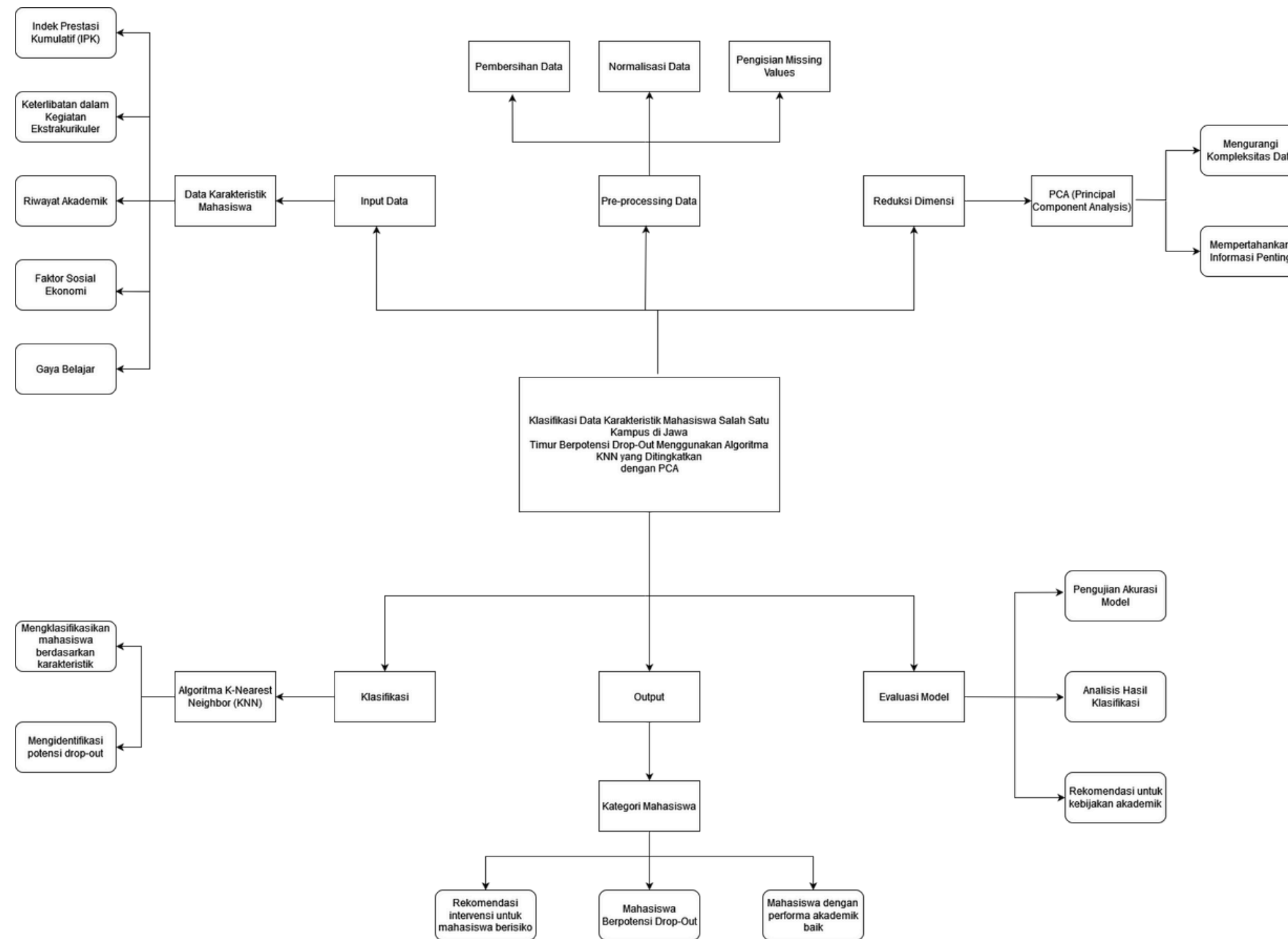
04

Memahami hasil klasifikasi yang dihasilkan oleh model algoritma K-Nearest Neighbor (KNN) yang ditingkatkan dengan PCA (Principal Component Analysis) untuk mengklasifikasikan data karakteristik mahasiswa yang berpotensi drop-out di salah satu kampus di Jawa Timur.

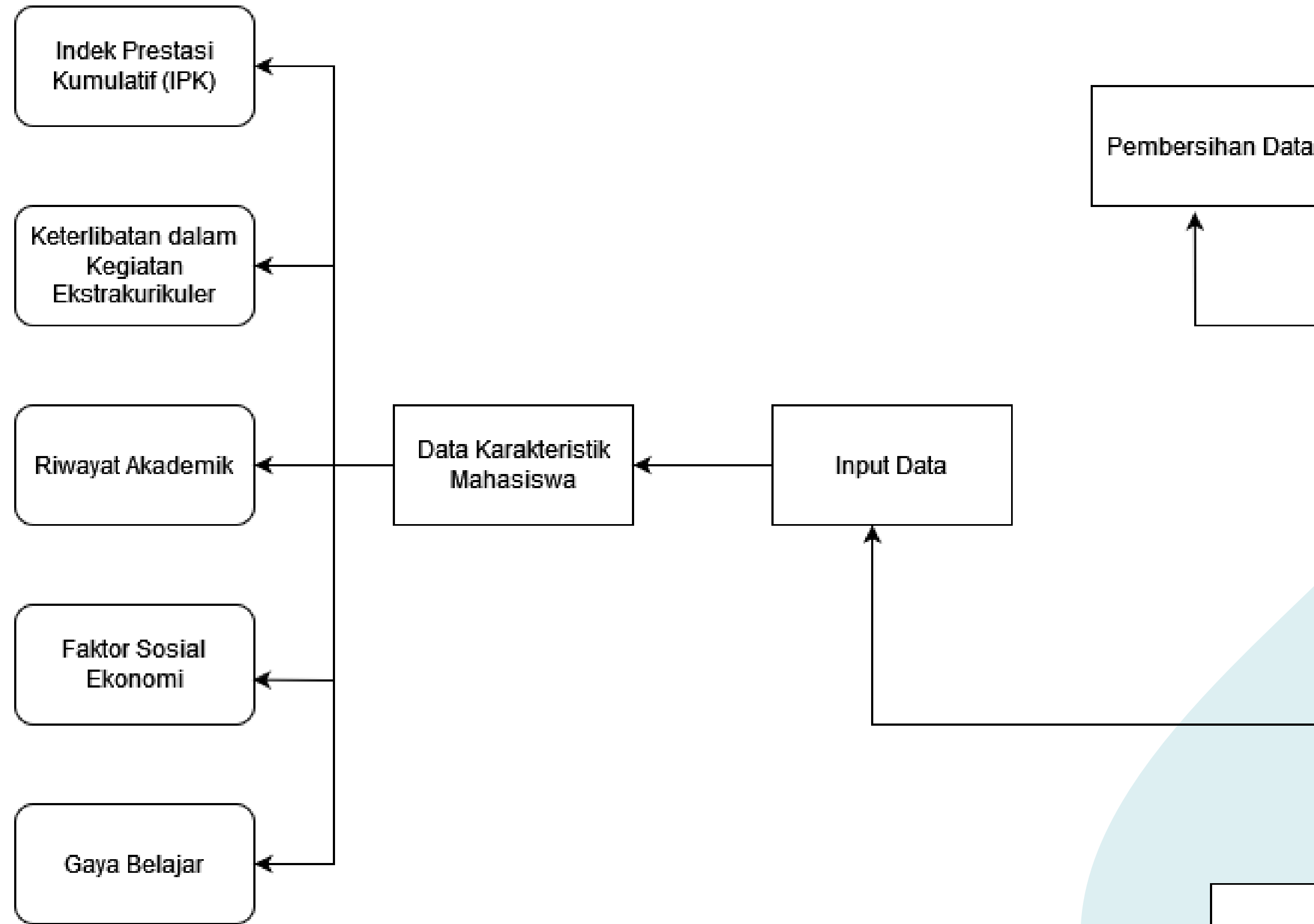
05

Merancang model algoritma prediksi yang diimplementasikan dalam bentuk API (Application Programming Interface) berbasis Python, sehingga dapat diakses dan digunakan oleh pengguna untuk memprediksi potensi drop-out mahasiswa secara praktis.

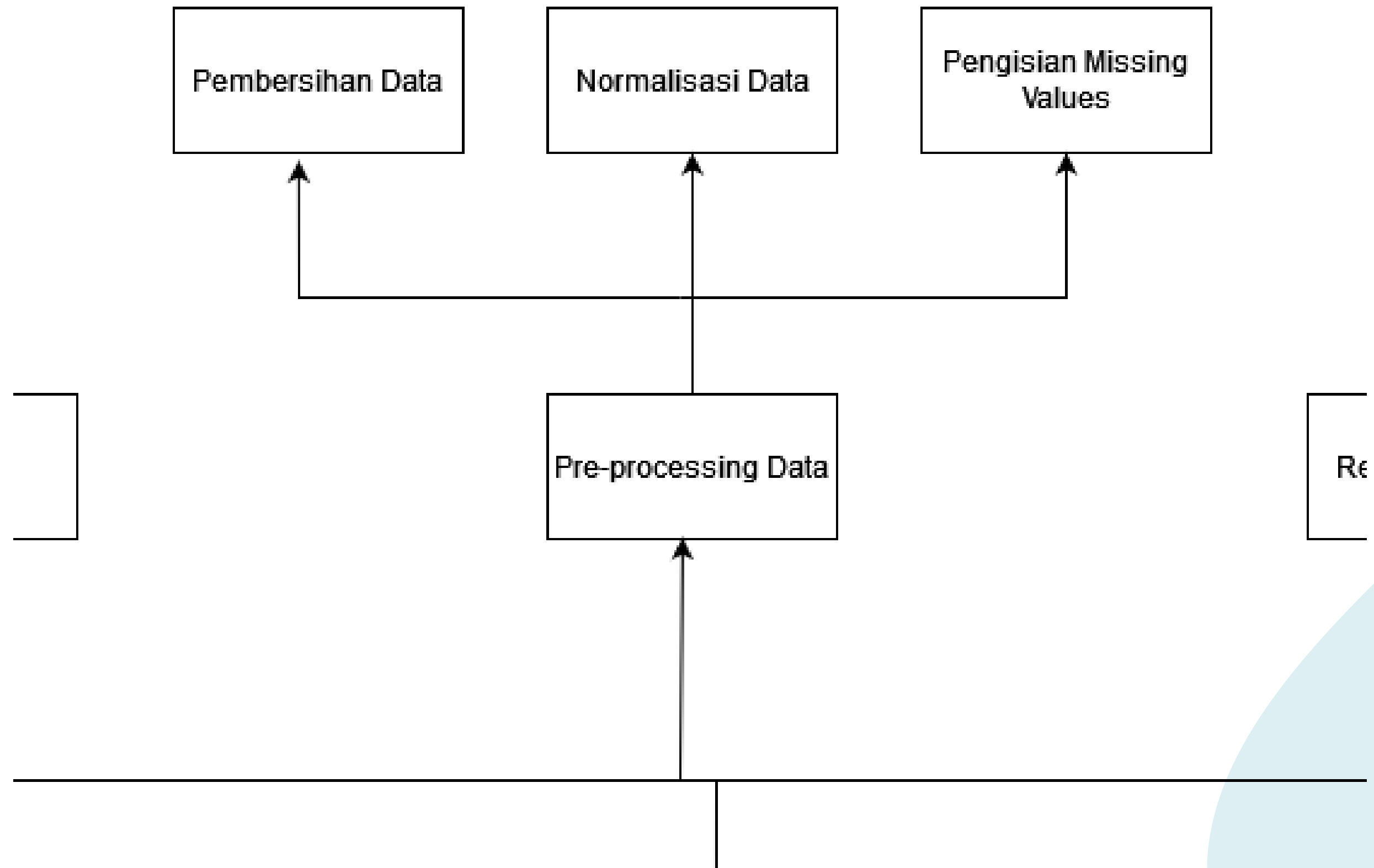
Mind Mapping



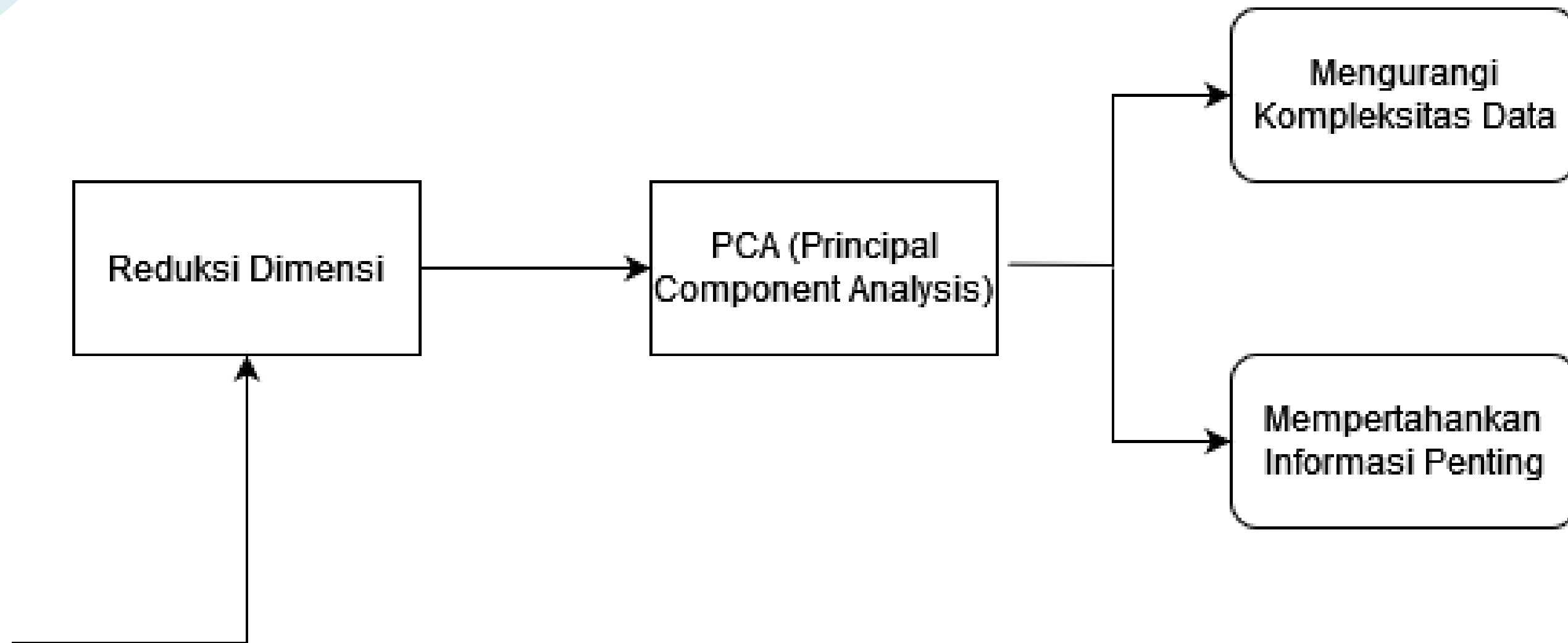
Mind Mapping



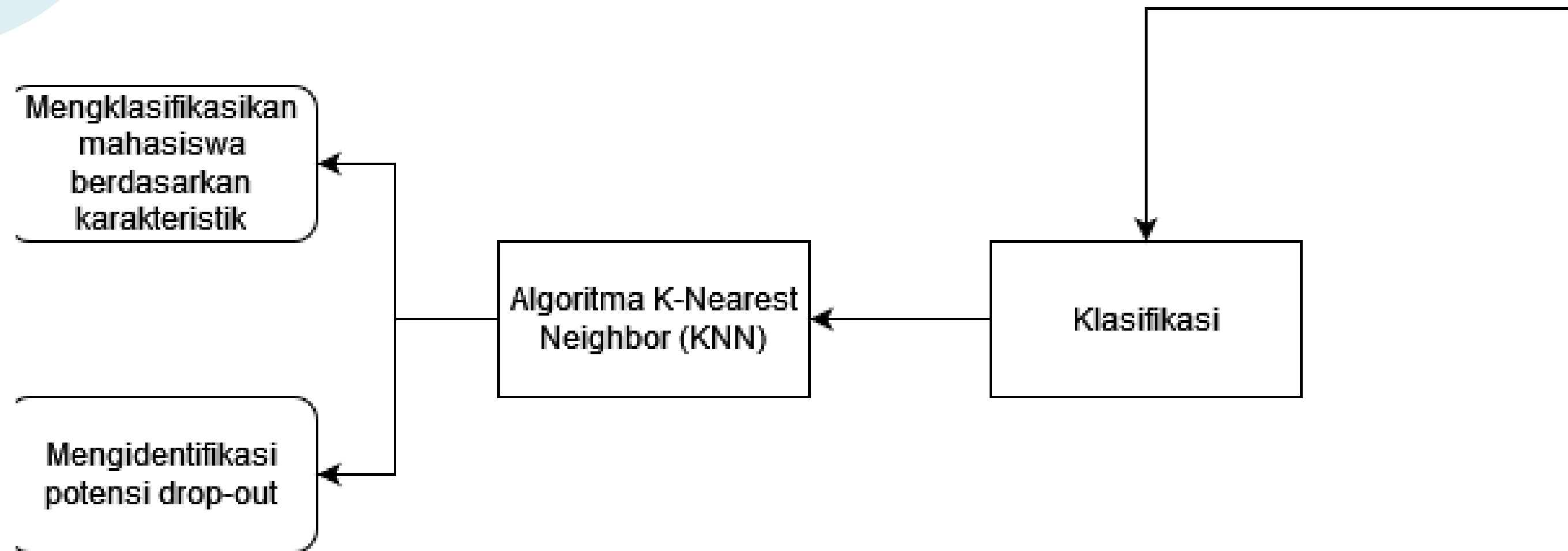
Mind Mapping



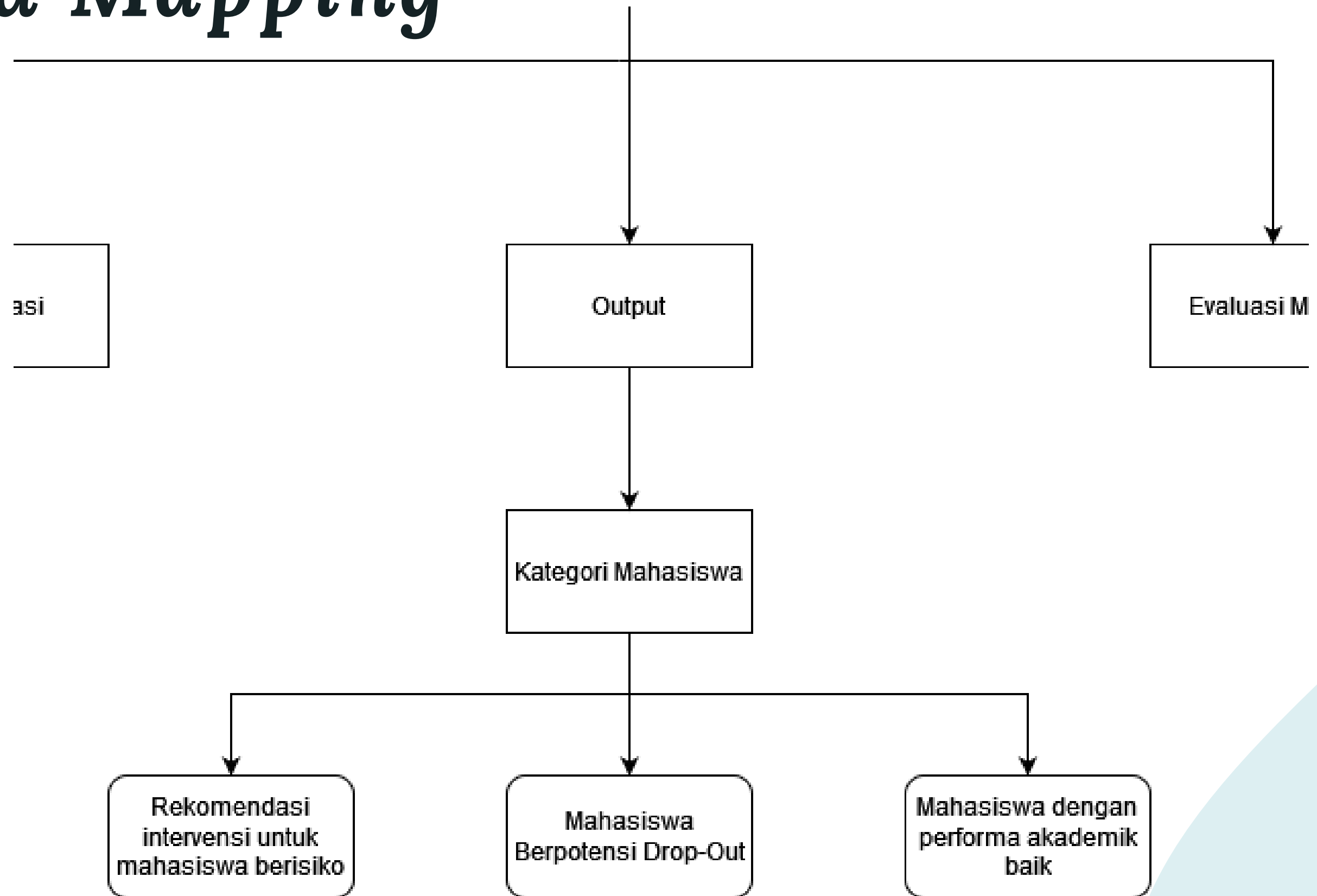
Mind Mapping



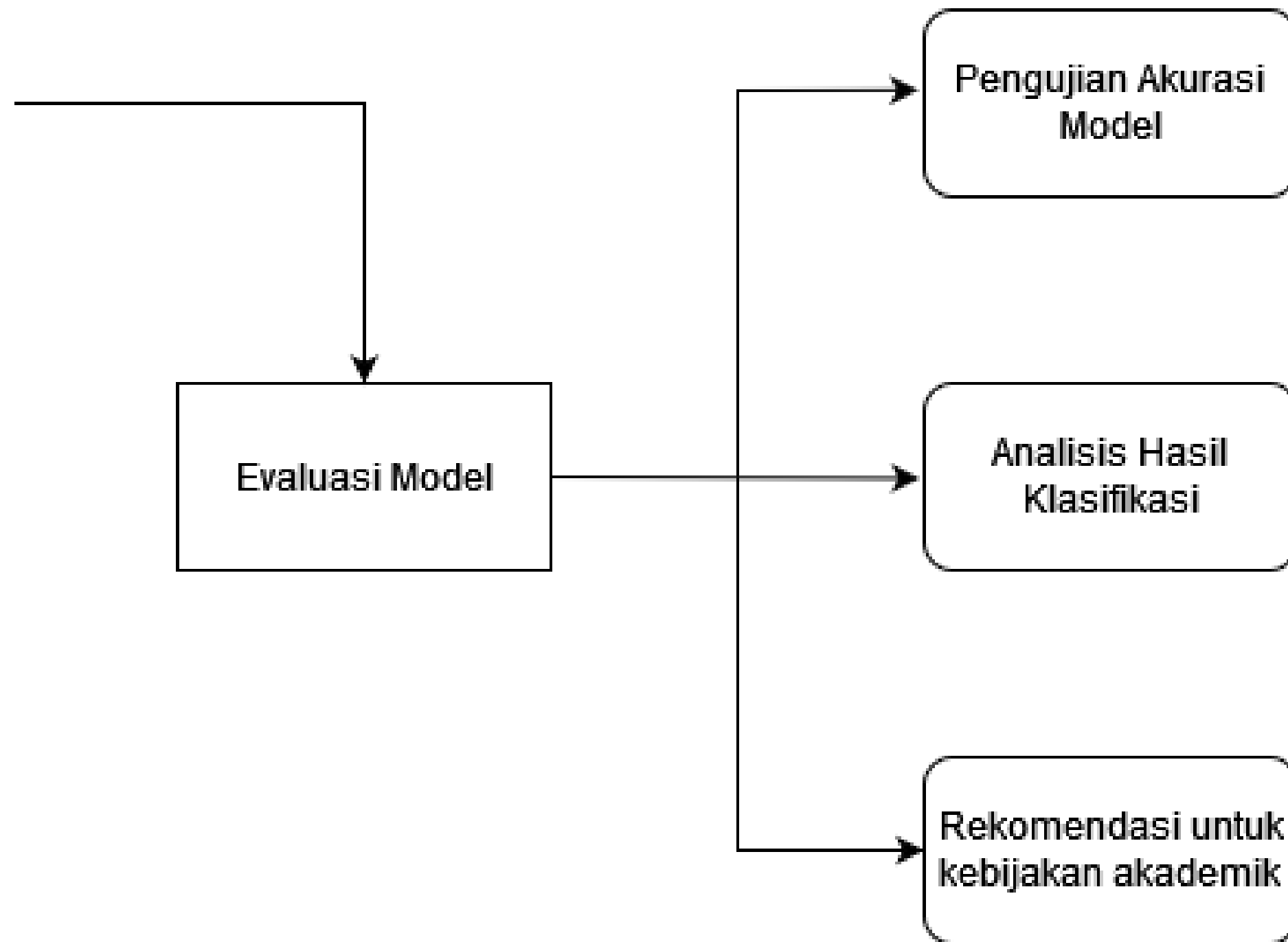
Mind Mapping



Mind Mapping



Mind Mapping



Metode yang di usulkan

Penelitian ini mengusulkan penggunaan kombinasi Principal Component Analysis (PCA) dan K-Nearest Neighbors (KNN) untuk mengklasifikasikan karakteristik mahasiswa yang berpotensi drop-out. Metode ini dirancang untuk meningkatkan akurasi prediksi dengan cara berikut:



REDUKSI DIMENSI

PCA digunakan untuk mengurangi dimensi dataset yang memiliki banyak variabel berkorelasi, sehingga dapat mengekstrak fitur utama tanpa kehilangan informasi penting.

Ini membantu dalam mengurangi kompleksitas data dan meningkatkan efisiensi algoritma KNN.

KLASIFIKASI dengan KNN

Setelah reduksi dimensi, algoritma KNN diterapkan untuk mengklasifikasikan mahasiswa berdasarkan kedekatan karakteristik data. KNN efektif dalam menangani data dengan struktur non-linear dan mampu mengelompokkan mahasiswa ke dalam kategori risiko drop-out yang berbeda.



Metrik Pengujian

01

AKURASI

Mengukur seberapa banyak prediksi model yang benar dibandingkan dengan total prediksi.

02

PRECISION

Mengukur proporsi prediksi positif yang benar dari semua prediksi positif yang dibuat oleh model.

03

Recall (Sensitivitas)

Mengukur proporsi prediksi positif yang benar dari semua kasus positif sebenarnya.

04

F1 Score

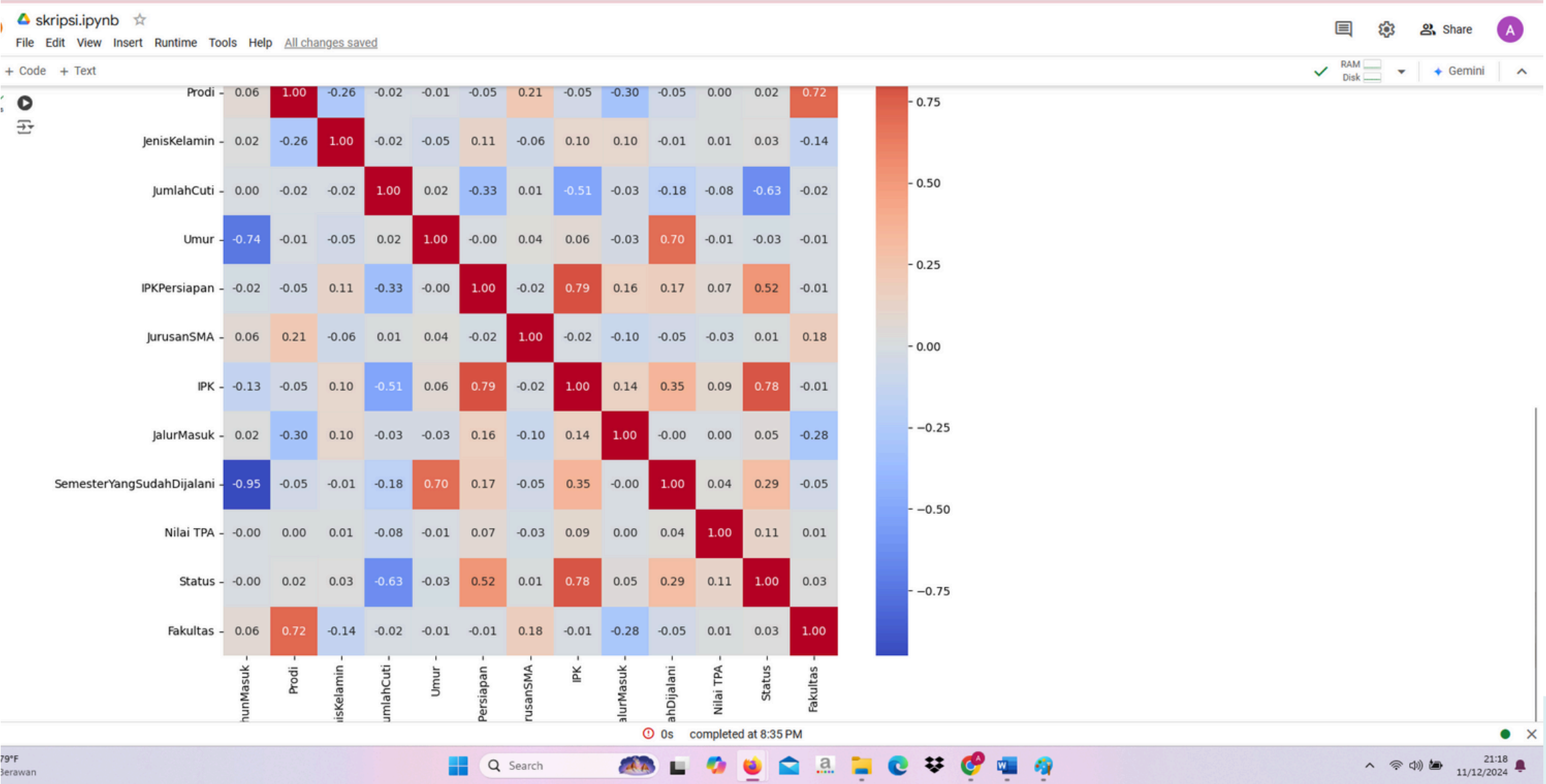
Merupakan rata-rata harmonis dari precision dan recall, memberikan gambaran keseluruhan tentang kinerja model dalam hal keseimbangan antara keduanya.

05

Classification Report

Menyediakan ringkasan lengkap tentang precision, recall, F1 score, dan support untuk setiap kelas dalam model klasifikasi.

Progress Riset



Progress Riset

Dalam penelitian ini, kami mengkaji karakteristik mahasiswa yang berpotensi drop-out dengan menggunakan algoritma K-Nearest Neighbors (KNN) yang ditingkatkan dengan Principal Component Analysis (PCA). Proses riset dimulai dengan pengumpulan data dari sistem manajemen kampus, yang mencakup berbagai atribut akademik, sosial, dan demografis mahasiswa. Data yang diperoleh mencakup informasi penting seperti nilai akademik, jumlah kehadiran, status ekonomi, dan informasi demografis lainnya.

Proses Cleaning Data

Setelah pengumpulan data, langkah selanjutnya adalah melakukan preprocessing atau pembersihan data. Proses ini meliputi beberapa tahap penting:

1. **Pembersihan Data:** Mengidentifikasi dan menghapus data yang hilang atau tidak konsisten untuk memastikan kualitas data yang digunakan dalam analisis. Ini termasuk penghapusan entri duplikat dan perbaikan nilai yang tidak valid.
2. **Normalisasi Data:** Melakukan normalisasi pada fitur numerik agar semua atribut berada dalam skala yang sama. Ini penting untuk menghindari dominasi fitur tertentu dalam algoritma KNN yang berbasis jarak.
3. **Penghapusan Variabel Tidak Relevan:** Mengidentifikasi dan menghapus variabel yang tidak memberikan kontribusi signifikan terhadap model prediksi potensi drop-out mahasiswa.

Dengan proses cleaning ini, dilakukan untuk memastikan bahwa dataset siap untuk digunakan dalam analisis lebih lanjut.

Proses Cleaning Data

penghapusan data.ipynb

☆

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

Reconnect Gemini

[] # 7. Menampilkan 5 data teratas setelah pembersihan dalam bentuk DataFrame
print("\nData setelah pembersihan:")
cleaned_data.head()

Data setelah pembersihan:

	ID	TahunMasuk	Prodi	JenisKelamin	JumlahCut	Umur	Pendapatan	IPK	Persiapan	JurusanSMA	IPK	JalurMasuk	SemesterYangSudahDijalani	Nilai TPA	Status	Fakultas
0	ID00001	2021.0	S-1 TEKNIK ELEKTRO	L	0.0	21.0	Rp. 2.500.001 - Rp. 3.000.000	3.68	SMA/MA IPA	3.55	SBMPTN		4.0	452.0	Tidak DO	Fakultas Teknik Elektro dan Komputer
1	ID00002	2021.0	S.Tr. TEKNOLOGI REKAYASA OTOMASI	L	0.0	22.0	Rp. 1.500.001 - Rp. 2.000.000	1.42	SMA/MA IPA	1.42	PKM Mandiri		4.0	550.0	Tidak DO	Fakultas Vokasi
3	ID00004	2020.0	S-1 BIOLOGI	P	0.0	23.0	Rp. 1.500.001 - Rp. 2.000.000	3.25	SMA/MA IPA	3.11	SNBT KIP-K		6.0	406.0	Tidak DO	Fakultas Seni
4	ID00005	2020.0	S-1 SISTEM INFORMASI	L	0.0	21.0	Lebih dari Rp. 15.000.000	3.56	SMA/MA IPA	3.67	PKM Kemitraan		6.0	612.0	Tidak DO	Fakultas Teknik Elektro dan Komputer
5	ID00006	2019.0	S-1 TEKNIK FISIKA	P	0.0	23.0	Rp. 7.500.001 - Rp. 10.000.000	3.50	SMA/MA IPA	3.41	PKM Mandiri		8.0	582.0	Tidak DO	Fakultas Teknik Industri

[] # 8. (Opsional) Menyimpan dataset yang sudah dibersihkan
cleaned_data.to_csv('cleaned_data.csv', index=False)
print("\nDataset yang sudah dibersihkan disimpan sebagai 'cleaned_data.csv'.")

Dataset yang sudah dibersihkan disimpan sebagai 'cleaned_data.csv'.

[] # 9. Mengunduh dataset yang sudah dibersihkan
files.download('cleaned_data.csv')

35m 1s completed at 7:20 PM

79°F Berawan

Search

21:23 11/12/2024

Memulai PCA

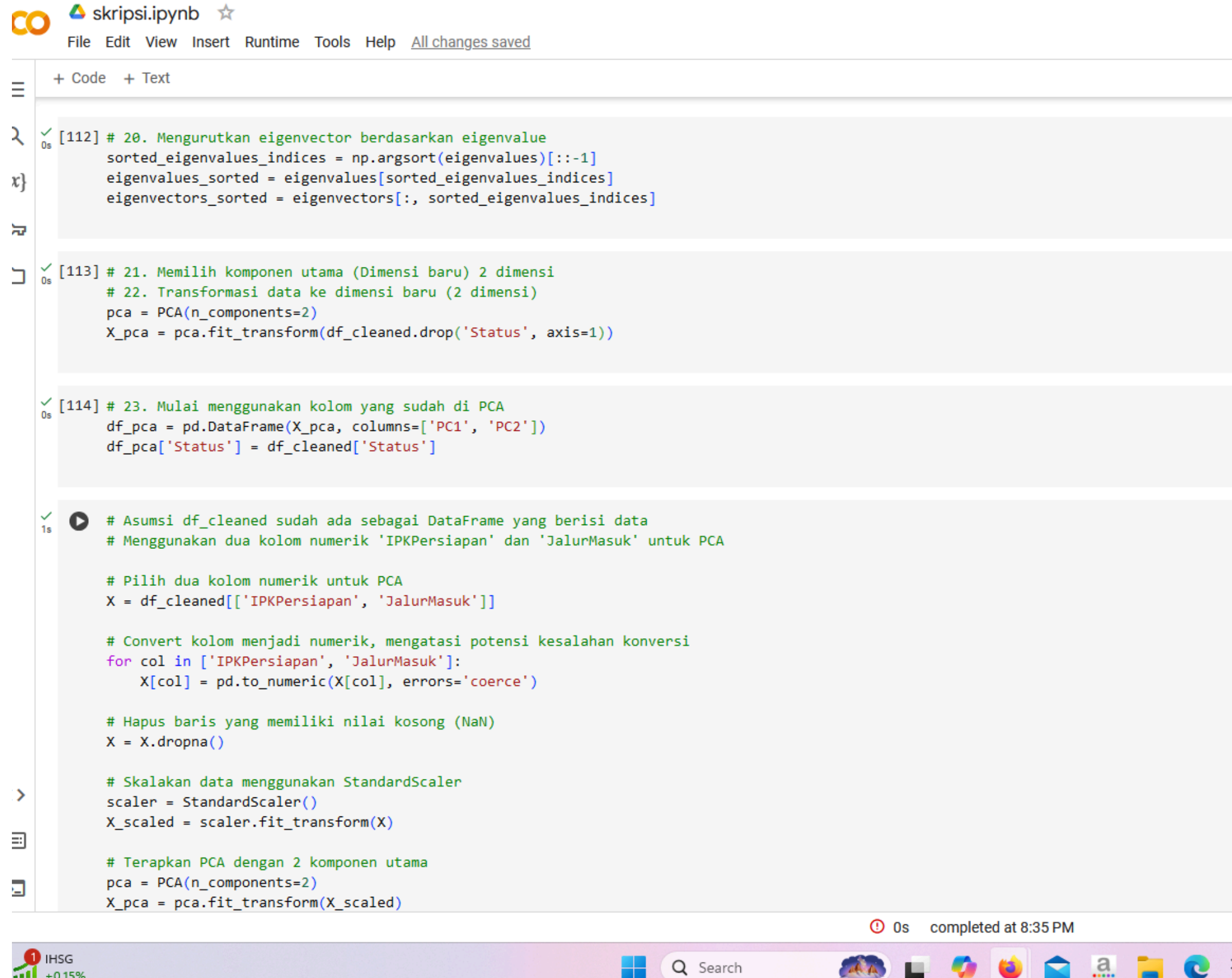
Setelah tahap pembersihan data selesai, dilanjutkan ke tahap berikutnya yaitu penerapan Principal Component Analysis (PCA). PCA digunakan untuk mereduksi dimensi dataset yang memiliki banyak variabel berkorelasi, sehingga dapat mengekstrak fitur utama tanpa kehilangan informasi penting. Dengan menggunakan Google Colab, dapat dengan mudah menerapkan PCA pada dataset yang telah dibersihkan.

Memulai PCA

Proses ini dilakukan dengan langkah-langkah sebagai berikut:

1. *Impor Library*: Mengimpor pustaka yang diperlukan seperti `numpy`, `pandas`, dan `sklearn` untuk melakukan PCA.
2. *Menerapkan PCA*: Menggunakan fungsi PCA dari pustaka `sklearn.decomposition` untuk mengurangi dimensi dataset dan mengekstrak komponen utama.
3. *Visualisasi Hasil PCA*: Menggunakan visualisasi untuk memahami bagaimana data terdistribusi setelah reduksi dimensi, serta untuk memeriksa proporsi *varians* yang dijelaskan oleh setiap komponen utama.

Memulai PCA



The screenshot shows a Jupyter Notebook interface with a light blue header bar. The notebook is titled "skripsi.ipynb" and has a star icon. The menu bar includes "File", "Edit", "View", "Insert", "Runtime", "Tools", and "Help", with a link to "All changes saved". The left sidebar contains icons for "Code" and "Text". The main area displays four code cells, each with a green checkmark and a status indicator (0s or 1s). The code cells contain Python code for PCA analysis. The first cell sorts eigenvalues and eigenvectors. The second cell performs PCA transformation. The third cell creates a DataFrame for the PCA results. The fourth cell performs data cleaning and scaling before PCA. The bottom status bar shows "0s completed at 8:35 PM". The Windows taskbar at the bottom includes the Start button, a search bar, and several application icons.

```
[112] # 20. Mengurutkan eigenvector berdasarkan eigenvalue
sorted_eigenvalues_indices = np.argsort(eigenvalues)[::-1]
eigenvalues_sorted = eigenvalues[sorted_eigenvalues_indices]
eigenvectors_sorted = eigenvectors[:, sorted_eigenvalues_indices]

[113] # 21. Memilih komponen utama (Dimensi baru) 2 dimensi
# 22. Transformasi data ke dimensi baru (2 dimensi)
pca = PCA(n_components=2)
X_pca = pca.fit_transform(df_cleaned.drop('Status', axis=1))

[114] # 23. Mulai menggunakan kolom yang sudah di PCA
df_pca = pd.DataFrame(X_pca, columns=['PC1', 'PC2'])
df_pca['Status'] = df_cleaned['Status']

# Asumsi df_cleaned sudah ada sebagai DataFrame yang berisi data
# Menggunakan dua kolom numerik 'IPKPersiapan' dan 'JalurMasuk' untuk PCA

# Pilih dua kolom numerik untuk PCA
X = df_cleaned[['IPKPersiapan', 'JalurMasuk']]

# Convert kolom menjadi numerik, mengatasi potensi kesalahan konversi
for col in ['IPKPersiapan', 'JalurMasuk']:
    X[col] = pd.to_numeric(X[col], errors='coerce')

# Hapus baris yang memiliki nilai kosong (NaN)
X = X.dropna()

# Skalikan data menggunakan StandardScaler
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Terapkan PCA dengan 2 komponen utama
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X_scaled)
```

0s completed at 8:35 PM

Memulai PCA

Proses ini dilakukan dengan langkah-langkah sebagai berikut:

1. *Impor Library*: Mengimpor pustaka yang diperlukan seperti `numpy`, `pandas`, dan `sklearn` untuk melakukan PCA.
2. *Menerapkan PCA*: Menggunakan fungsi PCA dari pustaka `sklearn.decomposition` untuk mengurangi dimensi dataset dan mengekstrak komponen utama.
3. *Visualisasi Hasil PCA*: Menggunakan visualisasi untuk memahami bagaimana data terdistribusi setelah reduksi dimensi, serta untuk memeriksa proporsi *varians* yang dijelaskan oleh setiap komponen utama.