KERJA PRAKTIK - EF234603

# Prediksi Ketepatan Waktu Kelulusan Mahasiswa Universitas Airlangga Berdasarkan Performa Empat Semester Pertama

Universitas Airlangga - Kampus MERR C
Jl. Dr. Ir. H. Soekarno, Mulyorejo, Kec. Mulyorejo, Surabaya, Jawa Timur 60115
Periode: 21 Maret 2024 - 1 Juni 2024

**Oleh:**

Hemakesha Ramadhani Heriqbaldi 50250201209

**Pembimbing Jurusan**
Dr.Eng. Darlis Herumurti, S.Kom., M.Kom.

**Pembimbing Lapangan**
Yunus Abdul Halim, S.Si. M.Kom

DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2024

**KERJA PRAKTIK – EF234603**

## Prediksi Ketepatan Waktu Kelulusan Mahasiswa Universitas Airlangga Berdasarkan Performa Empat Semester Pertama

Universitas Airlangga - Kampus MERR C
Jl. Dr. Ir. H. Soekarno, Mulyorejo, Kec. Mulyorejo, Surabaya, Jawa Timur 60115
Periode: 21 Maret 2024 - 1 Juni 2024

**Oleh:**

Hemakesha Ramadhani Heriqbaldi 50250201209

**Pembimbing Jurusan**
Dr.Eng. Darlis Herumurti, S.Kom., M.Kom.

**Pembimbing Lapangan**
Yunus Abdul Halim, S.Si. M.Kom

DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2024

# DAFTAR ISI

*[Halaman ini sengaja dikosongkan]*

# DAFTAR GAMBAR

*[Halaman ini sengaja dikosongkan]*

# DAFTAR TABEL

*[Halaman ini sengaja dikosongkan]*

# DAFTAR PSEUDOCODE

**LEMBAR PENGESAHAN**
**KERJA PRAKTIK**

**Prediksi Ketepatan Waktu Kelulusan Mahasiswa Universitas Airlangga Berdasarkan Performa Empat Semester Pertama**

Oleh:

Hemakesha Ramadhani Heriqbaldi 5025201209

Disetujui oleh Pembimbing Kerja Praktik:

1. Dr.Eng. Darlis Herumurti, S.Kom., M.Kom.
   NIP. 197712172003121001

(Pembimbing Departemen)

2. Yunus Abdul Halim, S.Si. M.Kom.
   NIP. 197501232008121002

(Pembimbing Lapangan)

*[Halaman ini sengaja dikosongkan]*

**Prediksi Ketepatan Waktu Kelulusan Mahasiswa Universitas Airlangga Berdasarkan Performa Empat Semester Pertama**


| | |
|---|---|
| Nama Mahasiswa | : Hemakesha Ramadhani Heriqbaldi |
| NRP | : 5025201209 |
| Departemen | : Teknik Informatika FTEIC-ITS |
| Pembimbing Departemen | : Dr.Eng. Darlis Herumurti, S.Kom., M.Kom. |
| Pembimbing Lapangan | : Yunus Abdul Halim, S.Si. M.Kom |


## ABSTRAK


*Kelulusan tepat waktu mahasiswa adalah faktor krusial bagi mahasiswa dan universitas. Ketepatan waktu kelulusan mahasiswa merupakan salah satu tolak ukur terpenting bagi universitas serta menjadi indikator penting dalam mengukur efektivitas program universitas. Tugas yang dikerjakan saat melakukan Kerja Praktik adalah pengembangan model prediksi ketepatan waktu kelulusan mahasiswa. Dimana model dapat memprediksi seorang mahasiswa dapat lulus tepat waktu atau tidak berdasarkan data empat semester pertama.*

*Model prediksi dibuat dengan menggunakan algoritma XGBoost dengan menggunakan beberapa fitur, yaitu terdiri dari data performa akademik mahasiswa selama empat semester pertama. Pengamatan pengaruh fitur terhadap model prediksi juga dilakukan menggunakan nilai SHAP.*



***Kata Kunci : Python, Prediksi, XGBoost, Kelulusan Mahasiswa***

*[Halaman ini sengaja dikosongkan]*

# KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah SWT atas penyertaan dan karunia-Nya sehingga penulis dapat menyelesaikan salah satu kewajiban penulis sebagai mahasiswa Departemen Teknik Informatika ITS yaitu Kerja Praktik yang berjudul: Prediksi Ketepatan Waktu Kelulusan Mahasiswa Universitas Airlangga Berdasarkan Performa Empat Semester Pertama.

Penulis menyadari bahwa masih banyak kekurangan baik dalam melaksanakan kerja praktik maupun penyusunan buku laporan kerja praktik ini. Namun penulis berharap buku laporan ini dapat menambah wawasan pembaca dan dapat menjadi sumber referensi.

Melalui buku laporan ini penulis juga ingin menyampaikan rasa terima kasih kepada orang-orang yang telah membantu menyusun laporan kerja praktik baik secara langsung maupun tidak langsung antara lain:

1. Kedua orang tua penulis.
2. Dr.Eng. Darlis Herumurti, S.Kom., M.Kom. selaku dosen pembimbing kerja praktik.
3. Yunus Abdul Halim, S.Si. M.Kom. selaku pembimbing lapangan selama kerja praktik berlangsung.
4. Teman-teman penulis yang senantiasa memberikan semangat ketika penulis melaksanakan KP.

Surabaya, 8 Juli 2024
Hemakesha Ramadhani Heriqbaldi

*[Halaman ini sengaja dikosongkan]*

# BAB I
# PENDAHULUAN

## 1.1. Latar Belakang

Kelulusan tepat waktu mahasiswa adalah faktor krusial bagi mahasiswa dan universitas. Ketepatan waktu kelulusan mahasiswa merupakan salah satu tolak ukur terpenting bagi universitas serta menjadi indikator penting dalam mengukur efektivitas program universitas. Hal ini mendorong Universitas Airlangga untuk melakukan evaluasi akademik tiga kali selama 8 semester guna membantu mahasiswa lulus tepat waktu. Evaluasi dilakukan pada akhir semester 2, semester 4, serta semester 8. Evaluasi dilakukan pada akhir semester 2, semester 4, dan semester 8. Pada semester 2, mahasiswa dapat melanjutkan studi jika mencapai minimal 20 SKS dan IPK minimal 1.00. Pada semester 4, mahasiswa harus mencapai minimal 40 SKS dan IPK minimal 2.00, serta dilakukan evaluasi terhadap aspek non-akademik. Evaluasi pada semester 8 mengharuskan mahasiswa mencapai minimal 80 SKS dan IPK minimal 2.00.

Perfroma serta ketepatan waktu kelulusan mahasiswa hanya bisa diamati di evaluasi akhir semester delapan. Hal ini menyebabkan kesulitan dalam melakukan intervensi dini untuk membantu mahasiswa secara akademik.

Oleh karena itu, dibutuhkan suatu cara untuk memprediksi ketepatan waktu kelulusan mahasiswa. Terutama dengan adanya program MBKM terbuka bagi mahasiswa semester lima keatas, kegiatan akademik mahasiswa semakin padat dan penuh perhatian. Dengan itu, pihak universitas memerlukan cara untuk membantu menentukan apakah mahasiswa memerlukan aksi intervensi akademik untuk meningkatkan performa mahasiswa atau tidak. Kerja praktik ini memberi kesempatan untuk merancang model prediksi ketepatan waktu kelulusan mahasiswa berdasarkan data empat semester pertama mahasiswa.

## 1.2. Tujuan

Tujuan kerja praktik ini adalah menyelesaikan kewajiban nilai kerja praktik sebesar 4 sks dan membantu Universitas Airlangga untuk menyelesaikan permasalahan evaluasi performa mahasiswa.

## 1.3. Manfaat

Manfaat yang diperoleh dengan adanya model prediksi ketepatan waktu kelulusan mahasiswa berdasarkan data empat semester pertama mahasiswa antara lain adalah membantu pihak universitas dalam menilai dan memprediksi performa mahasiswa. Dengan itu, pihak universitas memiliki patokan dalam menentukan keperluannya aksi intervensi akademik bagi mahasiswa. Jadi dapat mengurangi telatnya aksi intervensi akademik serta mengurangi kemungkinan ketidak tepatan waktu kelulusan mahasiswa.

## 1.4. Rumusan Masalah

Rumusan masalah dari kerja praktik ini adalah sebagai berikut:

1. Bagaimana pengembangan model prediksi ketepatan waktu kelulusan mahasiswa berdasarkan data keempat semester awal.
2. Bagaimana pengaruh fitur yang digunakan terhadap prediksi yang dilakukan dengan model XGBoost.

## 1.5. Lokasi dan Waktu Kerja Praktik

Kerja praktik ini dilaksanakan pada waktu dan tempat sebagai berikut:

Lokasi : Hybrid (Universitas Airlangga C)
Waktu : 18 Maret 2024 – 21 Juni 2024
Hari Kerja : Senin – Jumat
Jam Kerja : Fleksibel (dinilai melalui penyelesaian proyek)

## 1.6. Metodologi Kerja Praktik

Metodologi pembuatan buku kerja praktik meliputi :

### 1.6.1. Perumusan Masalah

Untuk mengetahui target pembuatan model prediksi, rapat serta diskusi dilakukan dengan tim satu data mengenai detail proyek. rapat dan diskusi menjelaskan alasan, kebutuhan, dan tujuan pembuatan model prediksi. Dibahas juga terkait pembagian pengembangan model prediksi.

### 1.6.2. Studi Literatur

Setelah mendapat gambaran bagaimana model prediksi yang akan dikembangkan, ditentukan tinjuan apa saja yang akan digunakan dalam pengembangan model prediksi. Tinjauan yang dipakai meliputi Python, matplotlib, sklearn, SHAP, dan *library* lainnya.

### 1.6.3. Implementasi Sistem

Implementasi merupakan realisasi dari metode yang telah didefinisikan pada studi literasi, ditentukan urutan metode yang akan digunakan, sehingga dapat memunculkan hasil performa prediksi model yang kuat. Pada tahap ini, dijelaskan tahap tahap yang dilakukan untuk mencapai prediksi model terhadap data empat semester pertama mahasiswa.

### 1.6.4. Pengujian dan Evaluasi

Setelah model telah dilalatih, perlu adanya evaluasi untuk menguji performa model. Tahap ini juga digunakan untuk menganalisis pengaruh setiap fitur yang digunakan dalam pelatihan model terhadap performa prediksi model.

### 1.6.5. Kesimpulan dan Saran

Pada tahap ini, dipaparkan kesimpulan yang dapat diambil dari tahap implementasi sistem serta tahap pengujian dan evaluasi. Saran juga dipaparkan agar pekerjaan dapat dikembangkan lagi dengan hasil yang lebih bagus.

## 1.7. Sistematika Laporan

### 1.7.1. Bab I Pendahuluan

Bab ini berisi latar belakang, tujuan, manfaat, rumusan masalah, lokasi dan waktu kerja praktik, metodologi, dan sistematika laporan.

### 1.7.2. Bab II Profil Perusahaan

Bab ini berisi gambaran umum Universitas Airlangga mulai dari profil dan lokasi perusahaan.

### 1.7.3. Bab III Tinjauan Pustaka

Bab ini berisi dasar teori dari teknologi yang digunakan dalam menyelesaikan proyek kerja praktik.

### 1.7.4. Bab IV Implementasi Sistem

Bab ini berisi uraian tahap - tahap yang dilakukan untuk proses pengembangan model.

### 1.7.5. Bab V Pengujian dan Evaluasi

Bab ini berisi hasil uji coba dan evaluasi dari model prediksi yang telah dikembangkan selama pelaksanaan kerja praktik.

### 1.7.6. Bab VI Kesimpulan dan Saran

Bab ini berisi kesimpulan dan saran yang didapat dari proses pelaksanaan kerja praktik.

# BAB II
# PROFIL PERUSAHAAN

## 2.1. Profil Universitas Airlangga Surabaya

Universitas Airlangga merupakan instansi pendidikan yang bertempat di kota Surabaya. Universitas Airlangga bertujuan untuk menjadi institusi yang mandiri, inovatif, dan terkemuka di tingkat nasional dan internasional, memelopori pengembangan ilmu pengetahuan, teknologi, humaniora, dan seni yang dilandasi oleh moralitas agama.

## 2.2. Lokasi

Jl. Dr. Ir. H. Soekarno, Mulyorejo, Kec. Mulyorejo, Surabaya, Jawa Timur 60115

*[Halaman ini sengaja dikosongkan]*

# BAB III
# TINJAUAN PUSTAKA

## 3.1. Python

Python adalah bahasa pemrograman tingkat tinggi yang dikenal dengan kesederhanaan dan keserbagunaannya. Diciptakan oleh Guido van Rossum dan pertama kali dirilis pada tahun 1991, Python memiliki kelebihan pada kemudahan penggunaannya, sehingga dapat digunakan oleh pemula dan juga merupakan bahasa pemrograman yang cukup kuat bagi *developer* berpengalaman.

Python memiliki *support* untuk berbagai paradigma pemrograman, termasuk pemrograman prosedural, berorientasi objek, dan fungsional. *Library* standar Python yang luas dan ketersediaan berbagai paket pihak ketiga telah menjadikannya pilihan populer untuk berbagai aplikasi, mulai dari pengembangan web dan analisis data hingga kecerdasan buatan dan komputasi ilmiah.

## 3.2. Google Colab

Google Colab, atau Google Colaboratory, adalah platform berbasis cloud gratis yang dikembangkan oleh Google Research yang memungkinkan pengguna untuk menulis dan menjalankan kode Python melalui browser mereka. Platform ini sangat cocok untuk pembelajaran mesin, analisis data, dan pendidikan. Google Colab menyediakan lingkungan notebook Jupyter tanpa memerlukan pengaturan atau konfigurasi apa pun, sehingga pengguna dapat mengakses sumber daya komputasi yang kuat, termasuk GPU dan TPU, tanpa biaya. Hal ini menjadikannya alat yang mudah diakses dan nyaman bagi para ilmuwan data, peneliti, dan pelajar.

Salah satu fitur utama Google Colab adalah integrasinya dengan Google Drive, yang memungkinkan pengguna untuk dengan mudah menyimpan dan membagikan buku catatan mereka.

Selain itu, Colab mendukung pemasangan berbagai *library* Python, membuatnya serbaguna untuk berbagai jenis proyek. Platform ini juga memfasilitasi kolaborasi dengan memungkinkan beberapa pengguna untuk bekerja pada notebook yang sama secara bersamaan. Dengan kombinasi kemudahan penggunaan, kemampuan komputasi yang kuat, dan fitur-fitur kolaboratif, Google Colab telah menjadi pilihan populer bagi mereka yang terlibat dalam ilmu data dan pembelajaran mesin.

## 3.3. SHAP

SHAP (SHapley Additive exPlanations) adalah *framework* terpadu untuk menginterpretasikan prediksi dari model pembelajaran mesin. Dikembangkan oleh Lundberg dan Lee di tahun 2017, SHAP memanfaatkan prinsip-prinsip dari teori permainan kooperatif, khususnya konsep nilai Shapley, untuk mengaitkan kontribusi setiap fitur ke prediksi akhir. Metode ini memastikan bahwa kontribusi masing-masing fitur didistribusikan secara adil berdasarkan kontribusi marjinal mereka di semua kombinasi fitur yang memungkinkan, memberikan pendekatan yang konsisten secara teoritis untuk atribusi fitur.

Salah satu kelebihan yang signifikan dari SHAP adalah kemampuannya untuk menyediakan interpretabilitas lokal dan global. Interpretabilitas lokal mengacu pada penjelasan prediksi individu, di mana nilai SHAP menunjukkan dampak dari setiap fitur pada prediksi tertentu. Interpretabilitas global, di sisi lain, menggabungkan penjelasan individu ini untuk memberikan wawasan tentang perilaku keseluruhan model. SHAP kompatibel dengan berbagai jenis model *machine learning*, menjadikannya alat serbaguna untuk memahami dan men-debug model yang kompleks, meningkatkan transparansi, dan memastikan kepercayaan pada sistem *AI*.

## 3.4. Scikit-learn

Scikit-learn, sering disingkat sklearn, adalah *library machine learning* gratis dan *open source* untuk bahasa pemrograman Python. Sklearn dilengkapi dengan berbagai alat untuk data *mining* serta analisis data, menjadikannya sumber daya utama untuk mengembangkan model *machine learning*. Dibangun di atas NumPy, SciPy, dan matplotlib, sklearn terintegrasi secara mulus dengan *library-library* tersebut untuk menyediakan algoritma yang efisien untuk klasifikasi, regresi, pengelompokan, reduksi dimensi, pemilihan model, dan *preprocessing*.

## 3.5. NumPy

NumPy, yang merupakan singkatan dari Numerical Python, adalah pustaka dasar untuk komputasi ilmiah dalam Python. NumPy menyediakan dukungan untuk *array* dan matriks multi-dimensi yang besar, serta kumpulan fungsi matematika untuk mengoperasikan *array* secara efisien. NumPy banyak digunakan untuk komputasi numerik dan menjadi dasar bagi banyak pustaka komputasi ilmiah lainnya di Python, seperti SciPy, pandas, dan scikit-learn.

## 3.6. Pandas

Pandas adalah *library* analisis dan manipulasi data bersifat *open source* untuk bahasa pemrograman Python. Pandas menyediakan struktur data dan fungsi yang diperlukan untuk bekerja pada data terstruktur dengan lancar dan intuitif. Diperkenalkan oleh Wes McKinney pada tahun 2008, Pandas dibangun di atas NumPy dan menawarkan alat manipulasi data tingkat tinggi, terutama melalui dua struktur data intinya: Series (satu dimensi) dan DataFrame (dua dimensi).

Pandas memiliki keunggukan dalam menangani dan memproses data terstruktur, seperti file CSV, basis data SQL, dan spreadsheet Excel. Pandas menyediakan kemampuan untuk

pembersihan, transformasi, dan visualisasi data, sehingga menjadikannya sebagai *library* utama bagi para ilmuwan dan analis data. Dengan kemampuan manipulasi data yang kuat dan fleksibel, Pandas memungkinkan pengguna untuk melakukan operasi seperti penyaringan, pengelompokan, dan penggabungan set data dengan mudah. Penggunaannya yang luas dan *support* komunitas yang kuat telah menjadikan Pandas sebagai alat penting dalam ekosistem analisis data.

## 3.7. Matplotlib

Matplotlib adalah sebuah *library* komprehensif untuk membuat visualisasi statis, animasi, dan interaktif dalam Python. Awalnya dikembangkan oleh John D. Hunter pada tahun 2003, Matplotlib memungkinkan pengguna untuk membuat plot, histogram, diagram batang, plot sebar, dan banyak lagi dengan tingkat pengaturan yang tinggi. Matplotlib menyediakan API berorientasi objek untuk menyematkan plot ke dalam aplikasi menggunakan toolkit GUI tujuan umum seperti Tkinter, wxPython, Qt, atau GTK.

Salah satu kekuatan Matplotlib adalah kemampuannya untuk menghasilkan visualisasi dengan kualitas publikasi dalam berbagai format dan lingkungan interaktif di berbagai *platform*. Modul pyplot *library* Matplotlib, yang dirancang untuk menyerupai kemampuan plotting MATLAB, menyederhanakan proses pembuatan dan pengaturan visualisasi. Hal ini membuat Matplotlib menjadi pilihan populer bagi para ilmuwan, insinyur, dan analis data yang perlu menganalisis dan menyajikan data secara visual. Dokumentasi yang luas dan dukungan komunitas yang aktif berkontribusi pada ketahanan dan keserbagunaannya.

## 3.8. XGBoost

XGBoost, singkatan dari eXtreme Gradient Boosting, adalah *library machine learning* yang terukur dan efisien untuk *gradient boosting*, sebuah metode *ensemble learning*. Dikembangkan oleh Tianqi Chen dan dirilis sebagai proyek sumber terbuka pada tahun 2014, XGBoost telah menjadi sangat populer karena performa dan kecepatannya dalam menangani dataset skala besar dan model yang kompleks. XGBoost banyak digunakan dalam kompetisi ilmu data dan aplikasi dunia nyata karena kemampuannya untuk meningkatkan akurasi prediksi secara signifikan.

XGBoost meningkatkan *gradient boosting* tradisional dengan mengimplementasikan beberapa pengoptimalan algoritmik dan peningkatan sistem, seperti *tree pruning* pemrosesan paralel, dan *cache awareness*. Peningkatan ini membuat XGBoost menjadi lebih cepat dan lebih efisien dalam hal sumber daya komputasi. Selain itu, XGBoost menyediakan parameter regularisasi untuk mencegah *overfitting* dan mendukung berbagai fungsi objektif, sehingga fleksibel untuk berbagai jenis tugas pemodelan prediktif, termasuk regresi, klasifikasi, dan pemeringkatan.

## 3.9. Random Forest

Random Forest adalah metode *ensemble learning* yang digunakan untuk klasifikasi, regresi, dan fungsi-fungsi lain yang beroperasi dengan membangun banyak *decision trees* selama waktu pelatihan. Diperkenalkan oleh Leo Breiman pada tahun 2001, metode ini menggabungkan prediksi dari beberapa *decision trees* untuk meningkatkan akurasi dan mengontrol *overfitting*. "Forest" dalam Random Forest mengacu pada kumpulan *decision trees*, masing-masing dilatih pada subset acak dari data dan subset acak dari fitur.

Salah satu kekuatan utama dari Random Forest adalah kemampuannya untuk menangani dataset yang besar dengan dimensi yang lebih tinggi dan ketahanannya terhadap *noise* dan

*overfitting*. Dengan merata-ratakan hasil dari setiap *tree*, Random Forest mengurangi varians model, sehingga menghasilkan prediksi yang lebih akurat dan stabil. Selain itu, metode ini memberikan skor kepentingan fitur, yang membantu dalam memahami kontribusi setiap fitur terhadap prediksi, menjadikannya alat yang kuat untuk pemilihan dan interpretasi fitur.

# BAB IV
# IMPLEMENTASI SISTEM

Pada bab ini akan menjelaskan tahap implementasi yang dilakukan pada data yang diberikan oleh Universitas Airlangga selama kerja praktik.

## 4.1. Menyambungkan Google Drive ke Google Colab

Pertama, Google Drive disambungkan dengan *notebook* Google Colab agar dapat memuat dataset yang diperlukan. Proses ini dipaparkan pada kode semu 4.1.

```
1. Drive.mount('/content/drive')
```

Pseudocode 4.1 Penyambungan Google Drive

## 4.2. Import Library

Setelah Google Drive telah disambungkan ke Google Colab, langkah selanjutnya adalah meng-*import library* yang dibutuhkan untuk segala operasi yang akan dilakukan. *Library* digunakan dalam tahap pemrosesan data serta untuk memvisualisasi data, pelatihan model, evaluasi model, pemilihan dan *tuning* model, serta serialisasi *file*.

## 4.3. Import Dataset

Setelah *library* telah di-*import*, langkah selanjutnya adalah memasukkan *dataset* yang akan digunakan kedalam Google Colab. Hal ini dicapai menggunakan Google Drive yang sudah disambungkan. Proses dipaparkan pada kode semu 4.2.

```
sks_data_types ← {
    'NIM_MHS': str,
    'ID_MHS': str,
    'NO_UJIAN': str
}
abs_data_types ← {
    'ID_MHS': str,
    'ID_SEMESTER': str,
    'NIM_MHS': str
}
aln_data_types ← {
    'ID_MHS': str,
    'NIM_MHS': str,
    'ID_SEMESTER': str,
    'NIM_MHS': str,
    'ID_FAKULTAS': str,
    'ID_PROGRAM_STUDI': str
}
df_sks ← pd.read_excel(r'/content/.../file.xlsx', dtype=sks_data_types)
df_sks.info()
df_abs ← pd.read_excel(r'/content/.../file.xlsx', dtype=abs_data_types)
df_abs.info()
df_aln ← pd.read_excel('/content/.../file.xlsx', dtype=aln_data_types)
df_aln.info()
```

Pseudocode 4.2 *Import* dataset ke dalam Google Colab

Tabel 4.1, 4.2, dan 4.3 memaparkan bentuk dari ketiga dataset yang digunakan, dengan tabel 4.1 memaparkan dataset sks mahasiswa, tabel 4.2 memaparkan dataset absensi mahasiswa, dan tabel 4.3 memaparkan dataset alumni mahasiswa:

a. df_sks

Tabel 4.1 Dataset df_sks

| NIM_MHS | 121611233001 | 121611233001 | 121611233001 |
|---|---|---|---|
| THN_ANGKATAN_MHS | 2016 | 2016 | 2016 |

| | | | |
|---|---|---|---|
| TAHUN_AJARAN | 2016/2017 | 2016/2017 | 2017/2018 |
| NM_SEMESTER | Ganjil | Genap | Ganjil |
| NM_PROGRAM_STUDI | Bahasa dan Sastra Inggris | Bahasa dan Sastra Inggris | Bahasa dan Sastra Inggris |
| NO_UJIAN | 4160000028 | 4160000028 | 4160000028 |
| NISN | 9980257941 | 9980257941 | 9980257941 |
| ID_MHS | 135452 | 135452 | 135452 |
| SKSTOTAL | 19.0 | 24.0 | 23.0 |

b. df_abs

Tabel 4.2 Dataset df_abs

| | | | |
|---|---|---|---|
| ID_SEMESTER | 226 | 243 | 263 |
| ID_MHS | 155382 | 152284 | 180590 |
| NIM_MHS | 151611913038 | 0317111332 07 | 0420114331 53 |
| TAHUN_AJARAN | 2017/2018 | 2019/2020 | 2022/2023 |
| NM_SEMESTER | Ganjil | Genap | Ganjil |
| PROSENTASE_KEHADIRAN | 0.00 | 88.93 | 0.00 |

c. df_aln

Tabel 4.3 Dataset df_aln

| | | | |
|---|---|---|---|
| ID_MHS | 142060 | 142092 | 142085 |
| NIM_MHS | 151610383037 | 151610383069 | 151610383062 |
| ID_SEMESTER | 197 | 197 | 197 |
| TAHUN_AJARAN | 2014/2015 | 2014/2015 | 2014/2015 |
| GROUP_SEMESTER | Ganjil | Ganjil | Ganjil |
| IPS_MHS | 2.79 | 2.93 | 3.19 |
| THN_ANGKATAN_MHS | 2016 | 2016 | 2016 |
| ID_FAKULTAS | 15 | 15 | 15 |
| ID_JENJANG | 4 | 4 | 4 |
| ID_PROGRAM_STUDI | 244 | 244 | 244 |
| PENGHASILAN_ORTU_MHS | 5500000.0 | 2812000.0 | 4500000.0 |
| KELAMIN_PENGGUNA | 1.0 | 2.0 | 2.0 |
| SKOR_SKP | 131.0 | 119.0 | 214.0 |
| NM_STATUS_PENGGUNA | Lulus | Lulus | Lulus |
| LAMA_STUDI | 13.0 | 14.0 | 7.0 |
| TAHUN_LULUS | 2022.0 | 2023.0 | 2019.0 |

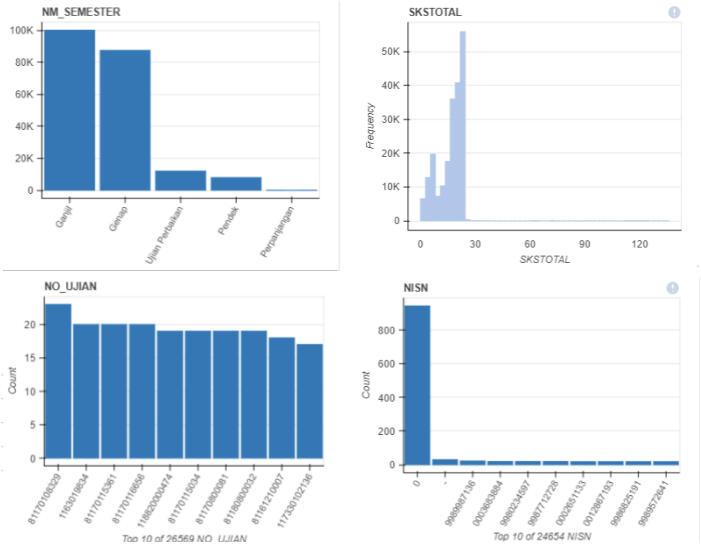
## 4.4. Exploratory Data Analysis

Setelah *dataset* telah di-*import*, langkah selanjutnya adalah melakukan *exploratory data analysis* (EDA). EDA dilakukan untuk mengetahui ciri-ciri dataset dan sebagai basis perencanaan pemrosesan dataset seperti seleksi data, pembersihan data, dan yang lainnya.

EDA dilakukan menggunakan *library* dataprep, yang dapat menampilkan distribusi data setiap kolom dataset, dan ciri ciri dataset secara detail. Proses ini ditampilkan pada kode semu 4.3.

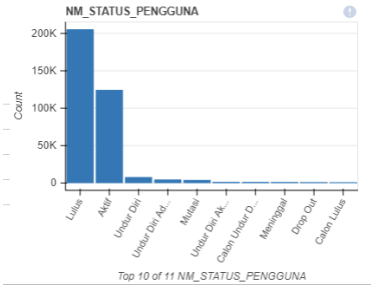
```
1. create_report(df_sks)
2. create_report(df_abs)
3. create_report(df_aln)
```

Pseudocode 4.3 *EDA* menggunakan *library* dataprep



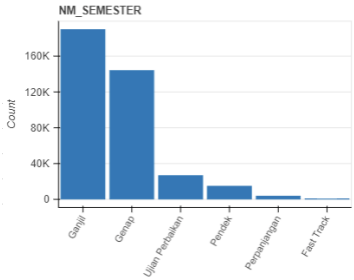

Gambar 4.1 Hasil EDA untuk df_sks

Berdasarkan hasil EDA pada gambar 4.1, ditemukan bahwa ada tiga nilai dalam kolom ‘NM_SEMESTER’ pada dataset ‘df_sks’ yang perlu dihapuskan, yaitu 'Ujian Perbaikan', 'Pendek', dan 'Perpanjangan'. Hal ini dilakukan karena jenis semester yang digunakan hanyalah ‘Ganjil’ dan ‘Genap’. Kedua, kolom ‘SKSTOTAL’ memiliki nominal yang mustahil, dengan nilai sebesar 136 SKS. Dimana seharusnya mahasiswa hanya bisa mengambil 24 sks. Karena itu, data ‘SKSTOTAL’ yang lebih dari 24 sks harus dihapuskan. Penemuan terakhir pada ‘ds_sks’ adalah keberadaan dua kolom yang tidak digunakan yaitu 'NO_UJIAN' dan 'NISN'.




Gambar 4.2 Hasil EDA untuk df_aln

Selanjutnya, gambar 4.2 menampilkan penemuan pada dataset ‘df_aln’, yaitu perlunya penghapusan data pada kolom ‘NM_STATUS_PENGGUNA’ yang memiliki nilai selain ‘Lulus’. Hal ini dilakukan karena data yang diproses hanyalah mahasiswa yang telah lulus.




Gambar 4.3 Hasil EDA untuk df_abs

Terakhir, gambar 4.3 menampilkan penemuan pada dataset ‘df_abs’, dimana penghapusan perlu dilakukan untuk baris yang tidak digunakan dalam ‘NM_SEMESTER’. Baris yang dimaksud adalah baris dengan nilai 'Ujian Perbaikan', 'Pendek', 'Perpanjangan', dan 'Fast Track'. Hal ini dilakukan karena hanya semester ‘Ganjil’ dan ‘Genap’ akan diproses. Terakhir, kolom ‘ID_SEMESTER’ perlu dihapuskan karena tidak digunakan.

## 4.5. Seleksi Data

Setelah pengamatan Dataset dilakukan, hal selanjutnya adalah seleksi dan memproses data sesuai dengan penemuan dataset dalam proses EDA.

a. df_sks

Pertama, penghapusan data dengan nilai yang tidak digunakan dalam kolom ‘NM_SEMESTER’, yaitu 'Ujian Perbaikan', 'Pendek', dan 'Perpanjangan'. Proses ini dipaparkan dalam kode semu 4.4.

```
1. values_to_drop ← ['Ujian Perbaikan', 'Pendek', 'Perpanjangan']
2. df_sks2 ← df_sks[~df_sks['NM_SEMESTER'].isin(values_to_drop)]
```

Pseudocode 4.4 Penghapusan nilai yang tidak digunakan

proses selanjutnya dipaparkan dalam kode semu 4.5, dimana kolom ‘TAHUN_AJARAN’ dan ‘NM_SEMESTER’ digabungkan untuk menjadi kolom baru yaitu ‘combined_columns’ yang digunakan sebagai pengurutan data setiap mahasiswa. Setelah penggabungan dilakukan, data diurutkan berdasarkan tiga kolom yaitu 'THN_ANGKATAN_MHS', 'NIM_MHS', dan 'combined_columns'.

```
df_sksc ←
df_sks2.assign(combined_columns=df_sks2['TAHUN_AJARAN'] + ' '
+ df_sks2['NM_SEMESTER'])
df_skscs ← df_sksc.sort_values(by=['THN_ANGKATAN_MHS',
'NIM_MHS', 'combined_columns'], ascending=[True, True, True])
```

Pseudocode 4.5 Pembuatan kolom baru

Selanjutnya, dilakukan penghapusan data dalam ‘SKSTOTAL’ dengan nilai yang lebih dari 24 serta penghapusan dua kolom yang tidak digunakan yaitu 'NO_UJIAN' dan 'NISN'. Penghapusan data duplikat juga dilakukan pada tahap ini. Proses ini dipaparkan dalam kode semu 4.6.

```
df_sks3 ← df_skscs[df_skscs['SKSTOTAL'] ≤ 24]
df_sks5 ← df_sks4.drop(columns=['index', 'NO_UJIAN', 'NISN'])
df_sks6 ← df_sks5.drop_duplicates()
```

Pseudocode 4.6 Penghapusan baris dan kolom

Terakhir, dilakukan pengambilan data empat semester pertama untuk semua mahasiswa. Hal ini dilakukan karena data yang dipakai untuk proses prediksi adalah data empat semester pertama mahasiswa. Proses ini dipaparkan dalam kode semu 4.7.

```
Procedure sem_n_mahasiswa(df, n)
selected_rows ← []
current_student ← None
count ← 0
for each index, row in df.iterrows() do
    if row['NIM_MHS'] ≠ current_student then
        current_student ← row['NIM_MHS']
        count ← 0
    end if
    if count < n then
        selected_rows.append(row)
        count += 1
    end if
end for
return pd.DataFrame(selected_rows)
```

Pseudocode 4.7 Pengambilan data empat semester pertama

b. df_aln

Pertama, dilakukan proses penhapusan data pada kolom 'NM_STATUS_PENGGUNA' dengan nilai selain 'Lulus' serta dilakukan pengecekan data bernilai *Null* pada seluruh kolom. Proses ini dipaparkan dalam kode semu 4.8.

```
df_aln2 ← df_aln[df_aln['NM_STATUS_PENGGUNA'] == 'Lulus']
print(df_aln3.isna().sum())
df_aln4 ← df_aln3.dropna()
```

Pseudocode 4.8 penghapusan baris yang tidak digunakan

Tabel 4.4 Jumlah nilai null dalam fitur dataset df_aln

| Kolom | Jumlah nilai Null |
|---|---|
| IPS_MHS | 3318 |
| PENGHASILAN_ORTU_MHS | 1500 |
| KELAMIN_PENGGUNA | 976 |

| | |
|---|---|
| SKOR_SKP | 2013 |
| LAMA_STUDI | 1229 |
| TAHUN_LULUS | 1229 |

Berdasarkan hasil pengamatan pada tabel 4.4, ditemukan ada berberapa data *Null* yang ada di dalam kolom ‘IPS_MHS’, ‘PENGHASILAN_ORTU_MHS’, ‘KELAMIN_PENGGUNA’, ‘SKOR_SKP’, ‘LAMA_STUDI’, dan ‘TAHUN_LULUS’. Seluruh data bernilai *Null* dihapuskan.

Selanjutnya, kolom ‘TAHUN_AJARAN’ dan ‘NM_SEMESTER’ digabungkan untuk menjadi kolom baru yaitu ‘combined_columns’ yang digunakan sebagai pengurutan data setiap mahasiswa. Setelah penggabungan dilakukan, data diurutkan berdasarkan tiga kolom yaitu 'THN_ANGKATAN_MHS', 'NIM_MHS', dan 'combined_columns'. Setelah itu, penghapusan kolom yang tidak digunakan yaitu ‘ID_SEMESTER’. Dan terakhir, dilakukan pengambilan data empat semester pertama untuk semua mahasiswa. Seluruh proses ini dipaparkan dalam kode semu 4.9.

```
df_aln5 ←
df_aln4.assign(combined_columns=df_aln4['TAHUN_AJARAN']
+ ' ' + df_aln4['GROUP_SEMESTER'])
df_aln6 ← df_aln5.sort_values(by=['THN_ANGKATAN_MHS',
'NIM_MHS', 'combined_columns'], ascending=[True, True, True])
df_aln8 ← df_aln7.drop('ID_SEMESTER', axis=1)
df_aln11 ← df_aln10.groupby('ID_MHS').head(4)
```

Pseudocode 4.9 Pengurutan df_aln menggunakan kolom baru

c. df_abs

Pertama, dilakukan pengecekan data bernilai *Null* pada seluruh kolom. Proses pengecekan ini dipaparkan dalam kode semu 4.10.

```
1. print("Jumlah data Null pada tiap kolom:")
2. print(df_abs.isna().sum())
3. df_abs2 ← df_abs.dropna()
```

Pseudocode 4.10 pengecekan dan penghapusan data null

Tabel 4.5 Jumlah nilai null dalam fitur dataset df_abs

| Kolom | Jumlah nilai Null |
|---|---|
| ID_SEMESTER | 867 |
| TAHUN_AJARAN | 867 |
| NM_SEMESTER | 867 |
| PROSENTASE_KEHADIRAN | 867 |

Berdasarkan hasil pengamatan pada tabel 4.5, ditemukan ada berberapa data *Null* yang ada di dalam kolom ‘PROSENTASE_KEHADIRAN’, ‘TAHUN_AJARAN’, ‘NM_SEMESTER’, dan ‘ID_SEMESTER’. Seluruh data bernilai *Null* dihapuskan.

Kedua, penghapusan data di dalam kolom ‘NM_SEMESTER’ dengan nilai 'Ujian Perbaikan', 'Pendek', 'Perpanjangan', dan 'Fast Track'. Hal ini dilakukan karena hanya semester ‘Ganjil’ dan ‘Genap’ akan di proses. Proses penghapusan data yang tidak digunakan dipaparkan dalam kode semu 4.11.

```
didrop ← ['Ujian Perbaikan', 'Pendek', 'Perpanjangan',
'Fast Track']
df_abs3 ← df_abs2[~df_abs2['NM_SEMESTER'].isin(didrop)]
```

Pseudocode 4.11 Penghapusan nilai yang tidak digunakan

Selanjutnya, kolom ‘TAHUN_AJARAN’ dan ‘NM_SEMESTER’ digabungkan untuk menjadi kolom baru yaitu ‘combined_columns’ yang digunakan sebagai pengurutan data setiap mahasiswa. Setelah penggabungan dilakukan, data diurutkan berdasarkan dua kolom yaitu 'NIM_MHS' dan 'combined_columns'. Setelah itu, penghapusan kolom yang tidak digunakan yaitu ‘ID_SEMESTER’. Dan terakhir, dilakukan pengambilan data empat semester pertama untuk semua mahasiswa. Seluruh proses ini dipaparkan dalam kode semu 4.12.

```
df_abs4 ←
df_abs3.assign(combined_columns=df_abs3['TAHUN_AJARAN'] +
' ' + df_abs3['NM_SEMESTER'])
df_abs5 ← df_abs4.sort_values(by=['NIM_MHS',
'combined_columns'], ascending=[True, True])
df_abs6 ← df_abs5.drop('ID_SEMESTER', axis=1)
df_abs9 ← df_abs8.groupby('ID_MHS').head(4)
```

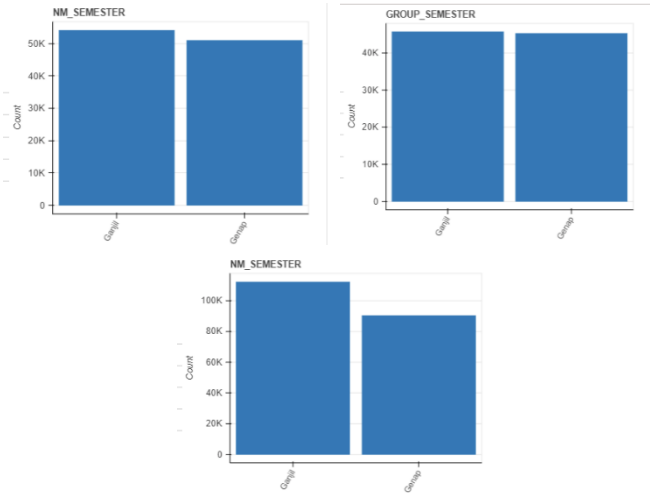
Pseudocode 4.12 Pengurutan df_abs menggunakan kolom baru

## 4.6. Exploratory Data Analysis Setelah Seleksi Data

EDA dilakukan setelah seleksi data untuk mengetahui ciri-ciri data yang telah diseleksi serta mengetahui jika pembersihan data masih perlu dilakukan atau tidak. Proses EDA dipaparkan dalam kode semu 4.13.

```
1. create_report(df_sks8)
2. create_report(df_aln11)
3. create_report(df_abs9)
```

Pseudocode 4.13 *EDA* menggunakan *library* dataprep



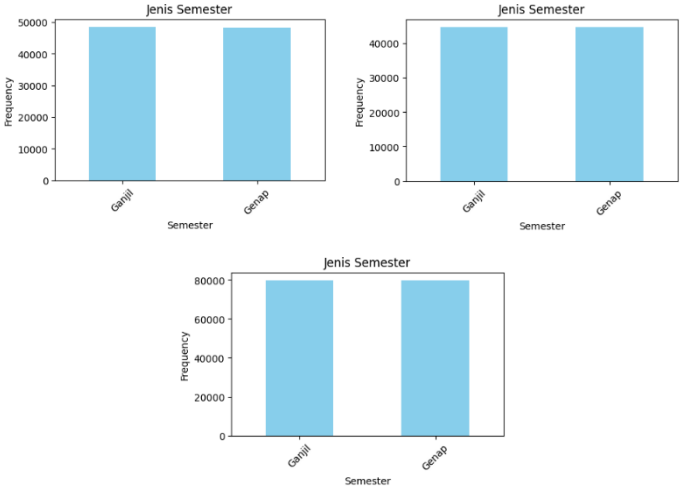

Gambar 4.4 Distribusi nilai semester ketiga dataset

Berdasarkan hasil EDA pada gambar 4.4, ditemukan bahwa ketiga *dataset* masih memiliki ketidakseimbangan nilai data ‘Ganjil’ dan ‘Genap’ dalam kolom semester. perlu dilakukannya pembersihan lebih jauh. Dengan data yang telah terurut berdasarkan semester dan tahun ajaran melalui kolom‘combined_columns’, dapat dilakukan pemilihan data

mahasiswa yang memiliki empat semester, tidak kurang tidak lebih. Proses pemilihan data dipaparkan dalam kode semu 4.14 dengan hasil yang didapatkan dipaparkan dalam gambar 4.5.

```
id_counts ← df_sks8.groupby('NIM_MHS').size()
id_filter ← id_counts[id_counts == 4].index
df_sks9 ← df_sks8[df_sks8['NIM_MHS'].isin(id_filter)]
id_counts ← df_aln11.groupby('NIM_MHS').size()
id_filter ← id_counts[id_counts == 4].index
df_aln12 ← df_aln11[df_aln11['NIM_MHS'].isin(id_filter)]
id_counts ← df_abs9.groupby('NIM_MHS').size()
id_filter ← id_counts[id_counts == 4].index
df_abs10 ← df_abs9[df_abs9['NIM_MHS'].isin(id_filter)]
```

Pseudocode 4.14 Pengambilan data empat semester




Gambar 4.5 Distribusi nilai semester setelah pemilihan data

## 4.7. Penggabungan Dataset

Penggabungan ketiga *dataset* dilakukan menggunakan dua tahap. Pertama, penggabungan *dataset* ‘df_sks’ dengan ‘df_aln’ yang dinamakan ‘merged_df’. Penggabungan dilakukan menggunakan metode ‘outer’ untuk memastikan tidak ada data yang tertinggal. Setelah itu langsung dilanjutkan dengan penggabungan ‘merged_df’ dengan ‘df_abs’. Setelah penggabungan seluruh dataset terselesaikan, hal yang selanjutnya dilakukan adalah memeriksa keberadaannya data dengan nilai *Null*. Proses penggabungan ketiga dataset dipaparkan dalam kode semu 4.15.

```
merged_df ← merge(df_sks9, df_aln12, on=['NIM_MHS',
'combined_columns'], how='outer')
merged_df_2 ← merge(merged_df, df_abs10, on=['NIM_MHS',
'combined_columns'], how='outer')
print(merged_df_2.isna().sum())
merged_df_clean ← merged_df_2.dropna()
```

Pseudocode 4.15 Pengggabungan ketiga dataset

Berdasarkan hasil pengecekan, banyak kolom di dalam *dataset* yang memiliki data bernilai *null*. Seluruh data *Null* dihapuskan. Terakhir, penghapusan kolom yang tidak digunakan dilakukan. Kolom yang dihapuskan merupakan kolom dengan nama duplikat yaitu 'THN_ANGKATAN_MHS_x', 'TAHUN_AJARAN_x', 'TAHUN_AJARAN_y', 'NM_SEMESTER_x', 'ID_MHS_x', dan 'ID_MHS_y'. Proses penghapusan dipaparkan dalam kode semu 4.16.

```
columns_to_drop ← ['THN_ANGKATAN_MHS_x', 'TAHUN_AJARAN_x',
'TAHUN_AJARAN_y', 'NM_SEMESTER_x', 'ID_MHS_x', 'ID_MHS_y']
merged_df_c2 ← merged_df_clean.copy()
merged_df_c2.drop(columns=columns_to_drop, axis=1,
inplace=True)
```

Pseudocode 4.16 Penghapusan kolom yang tidak digunakan

## 4.8. Pivot Dataset

Pivot *dataset* dilakukan agar setiap mahasiswa memiliki satu baris untuk seluruh datanya. Hal ini mempermudah model dalam proses prediksi. Langkah pertama dalam pivot *dataset* adalah membuat penanda pivot berdasarkan semester. Hal ini digunakan untuk memastikan bahwa pivot yang dilakukan merupakan data empat semester mahasiswa itu sendiri, tidak melebihi batas itu. Penanda pivot berbentuk sebuah kolom dengan nama 'semester_number' dan dibuat dengan menghitung jumlah baris dengan 'NIM_MHS' yang sama. Setelah kolom 'semester_number' dibuat, langkah selanjutnya adalah menghapus kolom yang tidak digunakan, yaitu 'NM_PROGRAM_STUDI', 'combined_columns', 'GROUP_SEMESTER', 'KELAMIN_PENGGUNA', 'NM_STATUS_PENGGUNA', 'ID_MHS', dan 'TAHUN_AJARAN', 'NM_SEMESTER_y'. Proses ini dipaparkan dalam kode semu 4.17.

```
1. merged_df_p ← merged_df_c2.copy()
2. merged_df_p['semester_number'] ←
3. merged_df_c2.groupby('NIM_MHS').cumcount() + 1
4. unused_cols ← ['NM_PROGRAM_STUDI', 'combined_columns',
5. 'GROUP_SEMESTER', 'KELAMIN_PENGGUNA', 'NM_STATUS_PENGGUNA',
6. 'ID_MHS', 'TAHUN_AJARAN', 'NM_SEMESTER_y']
7. merged_df_p2 ← merged_df_p.drop(columns=unused_cols)
```

Pseudocode 4.17 Pembuatan kolom baru

Langkah berikutnya dipaparkan dalam kode semu 4.18, yaitu mem-pivot *dataset*, dengan kolom yang ter-pivot adalah 'IPS_MHS', 'SKSTOTAL', 'PROSENTASE_KEHADIRAN', dan 'SKOR_SKP' dimana keempatnya memiliki data dari semester satu sampai empat. Sedangkan, kolom yang tidak ter-pivot adalah 'NIM_MHS', 'THN_ANGKATAN_MHS_y', 'ID_FAKULTAS', 'ID_JENJANG', 'ID_PROGRAM_STUDI',

'PENGHASILAN_ORTU_MHS', 'LAMA_STUDI', dan 'TAHUN_LULUS' dimana kolom-kolom tersebut hanya memiliki satu nilai, karena itu tidak perlu untuk mem-pivot ketujuh kolom tersebut.

```
columns_to_keep ← ['NIM_MHS', 'THN_ANGKATAN_MHS_y',
'ID_FAKULTAS', 'ID_JENJANG', 'ID_PROGRAM_STUDI',
'PENGHASILAN_ORTU_MHS', 'LAMA_STUDI', 'TAHUN_LULUS']
columns_to_pivot ← ['IPS_MHS', 'SKSTOTAL',
'PROSENTASE_KEHADIRAN', 'SKOR_SKP']
pivot_df ← merged_df_p2.pivot_table(index=columns_to_keep,
columns='semester_number',
values=columns_to_pivot).reset_index()
pivot_df.reset_index(drop=True)
```

Pseudocode 4.18 Fungsi pivot dataset

Langkah terakhir adalah merubah nama kolom. Hal ini dilakukan karena setelah proses pivot *dataset*, kolom menjadi multiindex, yaitu kolom yang memiliki banyak tingkat. Kolom multiindex dapat menyusahkan proses permodelan karena inkompatibilitasnya. Perubahan nama dilakukan kepada seluru kolom untuk menghilangkan multiindex. Langkah terakhir dipaparkan dalam kode semu 4.19.

```
new_column_names ← {
 ('IPS_MHS', 1): 'IPS_MHS_sem1',
 ('IPS_MHS', 2): 'IPS_MHS_sem2',
 ...
 ('LAMA_STUDI', ''): 'LAMA_STUDI',
 ('TAHUN_LULUS', ''): 'TAHUN_LULUS'
 }
pivot_df.columns ← [new_column_names.get(col, col) for col
in pivot_df.columns]
```

Pseudocode 4.19 Perubahan nama kolom menjadi *single index*

## 4.9. Formasi Dataset

Tahap terakhir yang dilakukan sebelum proses *modelling* adalah *formatting dataset*. Hal ini dilakukan untuk mempermudah proses *modelling* dengan cara mengubah dan mengatur *format dataset* sesuai dengan format yang dibutuhkan. Namun, sebelum *formatting* dilakukan, ada berberapa proses pengolahan data yang belum dilkakukan. Pertama adalah membuat kolom 'CLASS' sebagai kolom target prediksi. Kolom 'CLASS' dibuat berdasarkan kolom 'lama_studi' dengan data yang memiliki lama studi kurang dari atau sama dengan '8.0' dinyatakan lulus tepat waktu atau ditandakan dengan '0'. Sedangkan, data yang tidak memenuhi persyaratan tersebut dinyatakan tidak lulus tepat waktu atau ditandakan dengan '1'. Proses pembuatan kolom target dipaparkan dalam kode semu 4.20.

```
1. pivot_df['CLASS'] ← None
2. pivot_df9 ← pivot_df.copy()
3. Procedure classify_lama_studi(lama_studi)
4.     if lama_studi ≤ 8.0 then
5.         return '0'
6.     else
7.         return '1'
8.     end if
9. end Procedure
10. pivot_df['CLASS'] ←
11. pivot_df['LAMA_STUDI'].apply(classify_lama_studi)
```

Pseudocode 4.20 Pembuatan kolom target model prediksi

Kedua, yaitu menghapus data dengan nilai *Null* dalam seluruh dataset dan menghapus data dengan nilai '5' dan '4' pada kolom 'ID_JENJANG'. Hal ini dilakukan karena jenjang yang digunakan hanyalah '1' yaitu jenjang S1. Proses penghapusan data yang tidak digunakan dipaparkan dalam kode semu 4.21.

```
1. pivot_df1 ← pivot_df.dropna()
2. values_to_drop ← [5, 4]
3. pivot_df1_1 ←
4. pivot_df1[~pivot_df1['ID_JENJANG'].isin(values_to_drop)]
```

Pseudocode 4.22 Penghapusan data yang tidak digunakan

Hal terakhir yang dilakukan adalah menghapus seluruh kolom yang tidak digunakan, yaitu 'NIM_MHS', 'THN_ANGKATAN_MHS', 'PENGHASILAN_ORTU_MHS', 'LAMA_STUDI', 'TAHUN_LULUS', 'ID_JENJANG', 'ID_PROGRAM_STUDI', 'ID_FAKULTAS', 'SKOR_SKP_sem1', 'SKOR_SKP_sem2', 'SKOR_SKP_sem3', 'SKOR_SKP_sem4', 'SKOR_SKP_sem1', 'SKOR_SKP_sem2', 'SKOR_SKP_sem3', dan 'SKOR_SKP_sem4'. Proses penghapusan dipaparkan dalam kode semu 4.22.

```
1. unused_cols_final ← ['NIM_MHS', ... , 'SKOR_SKP_sem4']
2. pivot_df_no_skp ←
3. pivot_df1_1.drop(columns=unused_cols_final)
```

Pseudocode 4.21 Penghapusan kolom yang tidak digunakan

Hasil dari penghapusan adalah kolom yang digunakan untuk proses prediksi, yaitu :

- 'IPS_MHS_sem1',
- 'IPS_MHS_sem2',
- 'IPS_MHS_sem3',
- 'IPS_MHS_sem4',
- 'PROSENTASE_KEHADIRAN_sem1',
- 'PROSENTASE_KEHADIRAN_sem2',
- 'PROSENTASE_KEHADIRAN_sem3',
- 'PROSENTASE_KEHADIRAN_sem4',
- 'SKSTOTAL_sem1',
- 'SKSTOTAL_sem2',

- 'SKSTOTAL_sem3',
- 'SKSTOTAL_sem4',
- 'CLASS'.

Dengan *dataset* yang bersih, *formatting dataset* dapat dilakukan. Hal yang pertama dilakukan adalah membelah *dataset* menjadi *dataset train* dan *test*. Proses pembelahan ini dipaparkan pada kode semu 4.23. Tabel 4.6 Menunjukkan bentuk dataset yang akan dibelah menjadi dataset *train* dan *test*.

Tabel 4.6 Dataset pivot_df_no_skp

| | | | |
|---|---|---|---|
| IPS_MHS_sem1 | 3.20 | 3.59 | 3.18 |
| IPS_MHS_sem2 | 3.02 | 3.48 | 3.67 |
| IPS_MHS_sem3 | 3.52 | 3.67 | 3.63 |
| IPS_MHS_sem4 | 3.68 | 3.68 | 3.09 |
| PROSENTASE_KEHADIRAN_sem1 | 85.40 | 88.89 | 88.89 |
| PROSENTASE_KEHADIRAN_sem2 | 95.72 | 97.87 | 96.91 |
| PROSENTASE_KEHADIRAN_sem3 | 97.54 | 97.54 | 97.54 |
| PROSENTASE_KEHADIRAN_sem4 | 100.00 | 100.00 | 100.00 |
| SKSTOTAL_sem1 | 17 | 22 | 17 |
| SKSTOTAL_sem2 | 22 | 22 | 11 |
| SKSTOTAL_sem3 | 24 | 24 | 12 |
| SKSTOTAL_sem4 | 10 | 10 | 4 |
| CLASS | 0 | 0 | 0 |

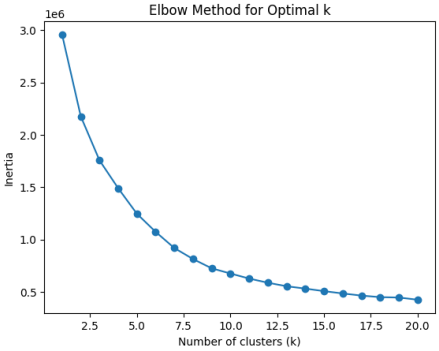
```
1. X ← pivot_df_no_skp.drop(columns=['CLASS'])
2. y ← pivot_df_no_skp['CLASS']
3. (X_train, X_test, y_train, y_test) ← train_test_split(X, y,
4. test_size=0.2, random_state=42)
```

Pseudocode 4.23 Pembelahan dataset menjadi *train* dan *test*

Setelah pembelahan terselesaikan, langkah selanjutnya yang dilakukan adalah menggunakan Kmeans undersampling untuk meratakan nilai ‘0’ dan ‘1’ pada kolom target ‘CLASS’. Jumlah kluster yang digunakan untuk KMeans undersampling adalah 3, berdasarkan plot ‘Elbow Method’ pada gambar 4.6. Proses penerapan KMeans undersampling dipaparkan dalam kode semu 4.24.

```
1. X_majority ← X_train[y_train == 0]
2. y_majority ← y_train[y_train == 0]
3. from sklearn.cluster import KMeans
4. from imblearn.under_sampling import ClusterCentroids
5.  inertia_values ← empty list
6.  for k in range(1, 21) do
7.      kmeans ← KMeans(n_clusters=k, random_state=42)
8.      kmeans.fit(X_majority)
9.      inertia_values.append(kmeans.inertia_)
10. end for
11. Plot inertia_values against range(1, 21) with marker 'o'
12. Set xlabel to 'Number of clusters (k)'
13. Set ylabel to 'Inertia'
14. Set title to 'Elbow Method for Optimal k'
15. Show the plot
16. kmeans ← KMeans(n_clusters=7)
17. kmeans.fit(X_majority)
18. representative_samples_indices ← empty list
19. for each cluster_center in kmeans.cluster_centers_ do
20.     distances ← np.linalg.norm(X_majority - cluster_center,
21.     axis=1)
22.     representative_samples_indices.append(np.argmin
23.     (distances))
27. end for
28. undersampler ← ClusterCentroids(sampling_strategy='auto',
29. random_state=42)
30. (X_undersampled, y_undersampled) ←
31. undersampler.fit_resample(X_train, y_train)
```

Pseudocode 4.24 Fungsi penerapan KMeans undersampling




Gambar 4.6 Plot Elbow Method

Terakhir, dibuat variabel ‘X_real’ yang berisi kolom fitur dan ‘y_real’ yang berisi kolom target untuk seluruh dataset. Kedua variabel ini akand igunakan dalam evaluasi menggunakan *cross validation*. Proses pembelahan dipaparkan dalam kode semu 4.25.

```
1. X_real ← pivot_df_no_skp.drop(columns=['CLASS'])
2. y_real ← pivot_df_no_skp['CLASS']
```

Pseudocode 4.25 Pembelahan dataset untuk *cross validation*

## 4.10. Modelling

*Modelling* dilakukan dengan pertama menginisialisasi model serta fungsi *grid search*. *Grid search* dilakukan untuk menemukan nilai hyperparameter terbaik bagi model XGBoost. Proses penggunaan *grid search* dipaparkan dalam kode semu 4.26.

```
1. from sklearn.model_selection import GridSearchCV
2. param_grid ← {
3.  'max_depth': [1, ..., 20],
4.  'learning_rate': [0.1, 0.01, 0.001],
5.  'subsample': [0.1, ..., 1]
6.  }
7. model_bin ← xgb.XGBClassifier(objective="binary:logistic")
8. grid_search ← GridSearchCV(model_bin, param_grid, cv=5,
9. scoring='roc_auc')
10.grid_search.fit(X_undersampled, y_undersampled)
```

Pseudocode 4.26 Fungsi penggunaan *grid search*

Setelah itu, *grid search* dijalankan, dan paramter terbaik yang ditemukan menggunakan grid search digunakan dalam model XGBoost. Proses implementasi dipaparkan pada kode semu 4.27.

```
1. best_xgb_model ← XGBClassifier(
2.  max_depth=grid_search.best_params_['max_depth'],
3.  learning_rate=grid_search.best_params_['learning_rate'],
4.  subsample=grid_search.best_params_['subsample'],
5.  objective="binary:logistic")
```

Pseudocode 4.27 Implementasi parameter terbaik

Langkah terakhir adalah untuk melakukan *fitment* model XGBoost terhadap data *train* dan menguji model tersebut dengan memprediksi data *test*. Proses *fitment* dipaparkan dalam kode semu 4.28.

```
1. best_xgb_model.fit(X_undersampled, y_undersampled)
2. xgb_predictions ← best_xgb_model.predict(X_test)
```

Pseudocode 4.28 *Fitment* XGBoost serta prediksi data *test*

## 4.11. Evaluasi

Evaluasi model dilakukan menggunakan tiga metode, yaitu menggunakan data *test* dengan metrik ukur skor Accuracy, ROC-AUC, dan F1. Proses pengujian dipaparkan dalam kode semu 4.29.

```
1. accuracy_xgb ← accuracy_score(y_test, xgb_predictions)
2. roc_auc_xgb ← roc_auc_score(y_test, xgb_predictions)
3. f1_xgb ← f1_score(y_test, xgb_predictions)
```

Pseudocode 4.29 Pengujian XGBoost dengan dataset *test*

kedua, menggunakan seluruh dataset dengan *cross validation*. Hal ini dilakukan untuk memastikan bahwa model tidak mengalami *overfitting* atau *underfitting*. metrik ukur skor yang digunakan adalah Accuracy, ROC-AUC, dan F1. Proses dipaparkan dalam kode semu 4.30.

```
kf ← KFold(n_splits=5, shuffle=True, random_state=42)
acc_score_as ← cross_val_score(best_xgb_model, X_real,
y_real, cv=kf, scoring='accuracy')
ra_score_as ← cross_val_score(best_xgb_model, X_real,
y_real, cv=kf, scoring='roc_auc')
fone_score_as ← cross_val_score(best_xgb_model, X_real,
y_real, cv=kf, scoring='f1')
```

Pseudocode 4.30 Pengujian XGBoost dengan *cross validation*

Ketiga, menggunakan Plot BeeSwarm SHAP untuk mengetahui pengaruh setiap fitur terhadap prediksi model XGBoost. Proses ini dipaparkan dalam kode semu 4.31.

```
!pip install shap
import shap as shp
explainer ← shp.Explainer(best_xgb_model)
shap_values_bin ← explainer(X_test)
shp.plots.beeswarm(shap_values_bin, max_display=16)
```

Pseudocode 4.31 Fungsi penggunaan SHAP

*[Halaman ini sengaja dikosongkan]*

# BAB V
# PENGUJIAN DAN EVALUASI

Bab ini menjelaskan tahap uji coba terhadap model XGBoost untuk memprediksi kelulusan mahasiswa berdasarkan performa empat semester pertama. Pengujian dilakukan untuk memastikan fungsionalitas dan model prediksi.

## 5.1. Tujuan Pengujian

Pengujian dilakukan terhadap model prediksi XGBoost menggunakan dataset *test* serta *cross validation* menggunakan seluruh dataset. Random forest diujikan dengan metode yang sama guna membandingkan performa kedua model. Pengaruh Fitur terhadap model prediksi diujikan menggunakan nilai SHAP.

## 5.2. Kriteria Pengujian

Penilaian atas pencapaian tujuan pengujian didapatkan dengan memperhatikan beberapa hasil yang diharapkan berikut :

a. Kemampuan Akurasi model prediksi XGBoost dalam memprediksi data.
b. Kemampuan model prediksi XGBoost dalam membedakan nilai positif dan negatif dalam memprediksi data.
c. Kemampuan model prediksi XGBoost dalam ketepatan dan pengingatan kembali dalam memprediksi data.
d. Pengaruh fitur terhadap prediksi model XGBoost.

## 5.3. Implementasi Database Utama

Skenario pengujian dilakukan dengan menggunakan berbagai metrik sesuai dengan kriteria pengujian. Metrik pengujian yang digunakan adalah :

1. Pengujian skor accuracy menggunakan dataset test
2. Pengujian skor accuracy menggunakan cross validation seluruh dataset
3. Pengujian skor ROC-AUC menggunakan dataset test.
4. Pengujian skor ROC-AUC menggunakan cross validation seluruh dataset.
5. Pengujian skor F1 menggunakan dataset test.
6. Pengujian skor F1 menggunakan cross validation seluruh dataset.
7. Visualisasi Plot BeeSwarm nilai SHAP bagi seluruh fitur model.

## 5.4. Evaluasi Pengujian

Pengujian pertama dilakukan terhadap performa prediksi model XGBoost menggunakan dataset *test* dan juga seluruh dataset menggunakan *cross validation.* Tabel 5.1 menjelaskan hasil uji coba model prediksi XGBoost. Sementara, tabel 5.2 menjelaskan hasil uji coba model Random Forest (RF). Model XGBoost diuji bersamaan dengan model Random Forest (RF) untuk menentukan performa terbaik dari kedua model. Namun, perlu dicatat bahwa model Random Forest menggunakan metode grid search dengan parameter yang lebih banyak, yang dapat membantu keunggulan Random Forest dibandingkan dengan XGBoost.

Tabel 5.1 Hasil pengujian performa model XGBoost

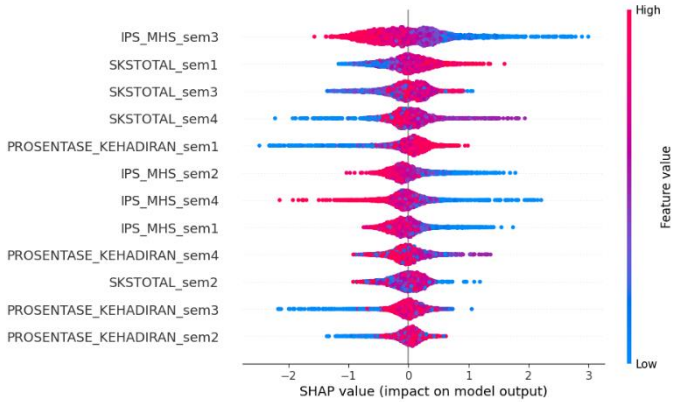
| **Metode Pengujian (XGB)** | **Dataset test** | **CV seluruh dataset** |
|---|---|---|
| Skor Accuracy | 74,32% | 75,45% |
| Skor ROC-AUC | 74,55% | 83,65% |

| | | |
|---|---|---|
| Skor F1 | 72,58% | 72,86% |

Tabel 5.2 Hasil pengujian performa model Random Forest

| **Metode Pengujian (RF)** | **Dataset test** | **CV seluruh dataset** |
|---|---|---|
| Skor Accuracy | 74,41% | 75,49% |
| Skor ROC-AUC | 74,62% | 83,87% |
| Skor F1 | 72,61% | 72,69% |

Pengujian kedua dilakukan terhadap seluruh fitur dataset untuk mengetahui bagaimana setiap fitur mempengaruhi prediksi model XGBoot. Gambar 5.1 menunjukkan hasil uji coba terhadap seluruh fitur menggunakan plot BeeSwarm nilai SHAP.




Gambar 5.1 Plot BeeSwarm nilai SHAP

*[Halaman ini sengaja dikosongkan]*

# BAB VI
# KESIMPULAN DAN SARAN

## 6.1. Kesimpulan

Kesimpulan yang didapat setelah melakukan pengembangan serta evaluasi model prediksi ketepatan waktu kelulusan mahasiswa:

a. Model prediksi XGBoost memiliki kemampuan yang cukup baik dalam memprediksi ketepatan waktu kelulusan mahasiswa dengan skor Accuracy sebesar 75,45%, skor ROC-AUC sebesar 83,65%, dan skor F1 sebesar 72,86% menggunakan pengujian cross validation dengan seluruh dataset. Model Random Forest memiliki peningkatan performa yang tidak signifikan terhadap model XGBoost. Peningkatan terbesar yang dimiliki oleh model Random Forest dalam pengujian cross validation adalah sebesar 0.25% dalam skor ROC-AUC. Namun, perlu dicatat bahwa metode grid search yang dilakukan untuk model Random Forest menghabiskan waktu yang jauh lebih lama, dikarenakan spesifikasi parameter yang lebih banyak dibandingkan oleh model XGBoost.
b. Pengaruh fitur terhadap model prediksi XGBoost dapat dilihat melalui plot BeeSwarm nilai SHAP. Berdasarkan plot, nilai IPS mahasiswa pada semester tiga memiliki penyebaran data dan gradien warna paling rata, yang menandakan hubungan linear pada data nilai IPS mahasiswa semester tiga dengan prediksi ketepatan waktu kelulusan mahasiswa. Sedangkan, data fitur lainnya memiliki kecenderungan nilai SHAP untuk berkumpul di garis tengah grafik, yang menandakan pengaruh yang tidak terlalu besar terhadap prediksi model XGBoost.

## 6.2. Saran

Saran untuk pengembangan model prediksi ketepatan waktu kelulusan mahasiswa :

a. Pemrosesan data yang lebih dalam untuk membantu performa XGBoost.
b. Penggunaan fitur yang lebih beragam untuk mengetahui faktor-faktor lain yang dapat mempengaruhi prediksi ketepatan waktu kelulusan mahasiswa.

# DAFTAR PUSTAKA


[1] Van Rossum, G. (1991). Python Programming Language. [ONLINE] Available at: https://www.python.org/doc/essays/blurb/. [Diakses Juli 2024]

[2] Bisong, E., 2019. *Building machine learning and deep learning models on Google cloud platform* (pp. 59-64). Berkeley, CA: Apress.

[3] Lundberg, S.M. and Lee, S.I., 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, *30*.

[4] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. and Vanderplas, J., 2011. Scikit-learn: Machine learning in python journal of machine learning research. *Journal of machine learning research*, *12*, pp.2825-2830.

[5] Oliphant, T.E., 2006. *Guide to numpy* (Vol. 1, p. 85). USA: Trelgol Publishing.

[6] McKinney, W., 2010, June. Data structures for statistical computing in Python. In *SciPy* (Vol. 445, No. 1, pp. 51-56).

[7] Hunter, J.D., 2007. Matplotlib: A 2D Graphics Environment. Com-464 puting in Science & Engineering, 9 (3), 90–95.

[8] Chen, T. and Guestrin, C., 2016, August. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785-794).

[9] Breiman, L., 2001. Random forests. *Machine learning*, *45*, pp.5-32.

*[Halaman ini sengaja dikosongkan]*

## BIODATA PENULIS


| | |
|---|---|
| Nama | : Hemakesha Ramadhani Heriqbadi |
| Tempat, Tanggal Lahir | : Melbourne, 16 November 2002 |
| Jenis Kelamin | : Laki-Laki |
| Telepon | : +6287851723774 |
| Email | : hemakesha.heriqbaldi@gmail.com |

AKADEMIS

| | |
|---|---|
| Kuliah | : Departemen Teknik Informatika – FTEIC , ITS |
| Angkatan | : 2020 |
| Semester | : 8 (Delapan) |