Klasifikasi Kinerja Mahasiswa menggunakan Algoritma *Machine Learning*

A. Latar Belakang

Pendidikan adalah pondasi utama dalam membangun masyarakat yang berkualitas dan berkelanjutan [1]. Di era informasi dan teknologi saat ini, pendidikan tinggi memainkan peran yang sangat penting dalam mempersiapkan individu dengan pengetahuan dan keterampilan yang dibutuhkan untuk menghadapi tuntutan dunia kerja yang semakin beragam [2]. Oleh karena itu, perhatian terhadap peningkatan mutu pendidikan tinggi dan kinerja mahasiswa adalah hal yang harus diperhatikan oleh pemerintah yang bekerja sama dengan institusi perguruan tinggi.

Kualitas pendidikan tidak hanya dinilai dari aspek pengajaran di dalam kelas, tetapi juga dari pemahaman yang mendalam tentang kinerja mahasiswa dan pencapaian lulusan setelah mereka menyelesaikan studi di perguruan tinggi [2]. Perguruan tinggi dan universitas memiliki tanggung jawab besar untuk mengidentifikasi faktor-faktor yang memengaruhi hasil belajar mahasiswa dan memberikan bantuan yang diperlukan untuk memastikan kesuksesan mereka. Terdapat banyak faktor yang berperan dalam menentukan hasil akhir pendidikan mahasiswa, baik faktor yang bersumber dari mahasiswa itu sendiri maupun faktor eksternal [3]. Sayangnya, seringkali mahasiswa mengalami *drop-out* atau tidak melanjutkan perkuliahan mereka karena tidak mampu menyelesaikan studi mereka [1]. Dropout merupakan masalah paling problematis yang harus diatasi oleh institusi pendidikan tinggi untuk meningkatkan keberhasilannya [1]. Namun, pemahaman tentang faktor-faktor ini, terutama pada tahap awal pendidikan, seringkali menjadi tugas yang rumit dan memerlukan pendekatan yang sistematis.

Dalam konteks ini, *Data Science* (DS) telah muncul sebagai bidang ilmu yang sangat berguna dalam menganalisis banyak persoalan pada dunia nyata, salah satunya adalah terkait kinerja mahasiswa [4]. DS menggabungkan matematika, statistik, pemrograman, dan pemahaman domain untuk menggali wawasan berharga dari data. Dengan menerapkan DS, institusi pendidikan tinggi dapat mengumpulkan, menganalisis, dan memanfaatkan data mahasiswa untuk mengidentifikasi pola, tren, dan faktor-faktor yang berkontribusi terhadap kinerja mereka.

Salah satu tujuan utama dari penggunaan DS dalam pendidikan tinggi adalah untuk meningkatkan kualitas pembelajaran dan hasil belajar mahasiswa. Dengan pemahaman tentang faktor-faktor yang memengaruhi kinerja mahasiswa, institusi pendidikan dapat mengambil langkah-langkah proaktif untuk memberikan bantuan dan dukungan yang sesuai. Hal ini mencakup memberikan rekomendasi yang tepat waktu, menyesuaikan kurikulum, dan menciptakan lingkungan pembelajaran yang lebih efektif.

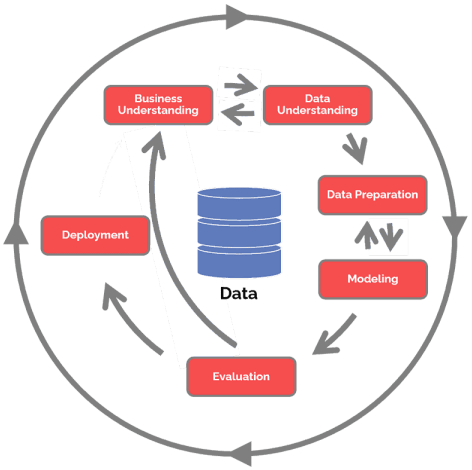
Oleh karena itu, proyek ini bertujuan untuk mengimplementasikan DS dalam menganalisis kinerja mahasiswa pada tahap awal pendidikan mereka di institusi pendidikan tinggi. Proyek ini akan menggali data mahasiswa, menerapkan teknik-teknik seperti pemrosesan data, pemodelan prediktif, dan evaluasi, dan pada akhirnya memberikan wawasan berharga dan rekomendasi yang dapat membantu meningkatkan kinerja dan pengalaman belajar mahasiswa. Dengan memanfaatkan DS, diharapkan pihak berwenang dapat mengoptimalkan pembelajaran, meningkatkan efisiensi, dan menciptakan lingkungan pendidikan yang mendukung pertumbuhan akademik dan perkembangan pribadi mahasiswa.

B. Tujuan Pembelajaran

Setelah menyelesaikan materi ini, peserta diharapkan dapat:

1. Memahami konsep dasar *data science* dan bagaimana penggunaannya dapat diaplikasikan dalam konteks analisis kinerja mahasiswa.
2. Menguasai langkah-langkah utama dalam mengembangkan proyek Ilmu Data, termasuk pengumpulan data, pra-pemrosesan data, pemodelan, evaluasi, dan *deployment* menggunakan metodologi CRISP-DM (*CRoss Industry Standard Process for Data Mining*).
3. Menggunakan alat-alat, pustaka (*library*), dan teknik *data science* seperti Python, Pandas, Matplotlib, dan Scikit-Learn untuk menganalisis dan memodelkan kinerja mahasiswa.
4. Menerapkan hasil analisis untuk memberikan wawasan dan rekomendasi yang berguna dalam meningkatkan kinerja mahasiswa.

C. Konsep CRISP-DM



Gambar 1. Proses pada CRISP-DM [5]

CRISP-DM adalah singkatan dari *Cross Industry Standard Process for Data Mining*. CRISP-DM adalah metodologi yang sangat terstruktur yang digunakan dalam dunia Data Science dan analisis data untuk memandu langkah-langkah yang harus diambil dalam proyek penambangan data atau analisis data. Metodologi ini membantu para profesional data untuk merencanakan, merancang, dan melaksanakan proyek penambangan data dengan lebih efektif dan sistematis. Berikut adalah tahapan yang dapat dilakukan [6]:

1. Pemahaman Bisnis (*Business Understanding*)
   Tahap pertama adalah memahami tujuan bisnis atau masalah yang ingin diselesaikan melalui analisis data. Tahapan ini melibatkan pertanyaan seperti apa yang ingin dicapai dengan proyek ini, dan bagaimana hasil analisis akan digunakan untuk mengambil keputusan.
2. Pemahaman Data (*Data Understanding*)
   Pada tahap ini, data yang diperlukan untuk proyek dikumpulkan, dieksplorasi, dan dianalisis. Proses ini mencakup pemahaman terhadap sumber data, karakteristik data, serta masalah kualitas data.

3. Persiapan Data (Data Preparation)
   Data yang dikumpulkan akan dipre-proses dan disiapkan untuk analisis lebih lanjut. Hal ini melibatkan pemrosesan data seperti penghapusan data yang hilang, normalisasi, dan transformasi data.
4. Modeling (Pemodelan)
   Tahap ini melibatkan pemilihan dan penerapan model statistik atau algoritma *Machine Learning* untuk menganalisis data. Proses ini mencakup pelatihan, validasi, dan evaluasi model.
5. Evaluasi (*Evaluation*)
   Hasil dari model dan analisis dievaluasi untuk memeriksa sejauh mana model tersebut memenuhi tujuan bisnis yang ditentukan di tahap awal. Evaluasi juga mencakup pengukuran performa model dan identifikasi apakah model tersebut sesuai atau perlu disesuaikan.
6. Implementasi (Deployment)
   Pada tahap akhir, hasil analisis diterapkan dalam konteks bisnis atau organisasi. Tujuan akhirnya adalah mengimplementasikan model dalam produksi atau memberikan rekomendasi kepada pemangku kepentingan.

D. Implementasi Proyek

Link notebook:

https://colab.research.google.com/drive/1kasvMMls-_Btu85G52a05crABDbQSkII?usp=sharing

1. Pemahaman Bisnis (*Business Understanding)*

Dalam konteks penjelasan sebelumnya tentang dataset yang digunakan untuk membangun model klasifikasi mahasiswa yang akan drop out atau berhasil akademik, berikut adalah penjelasan lebih lengkap:

a. Tujuan Bisnis

Proyek ini bertujuan untuk meningkatkan kualitas pendidikan tinggi dengan pemahaman yang lebih baik tentang faktor-faktor yang memengaruhi kinerja mahasiswa dan secara simultan mengurangi tingkat drop-out di antara mahasiswa.

b. Pemahaman Masalah

Dalam pemahaman bisnis ini, kita akan mengidentifikasi faktor-faktor yang mempengaruhi kinerja mahasiswa, seperti riwayat akademik, demografi, dan faktor sosial-ekonomi. Selain itu, kita akan membangun model klasifikasi untuk memprediksi apakah seorang mahasiswa akan terdaftar, *drop out,* atau lulus akademik.

c. *Stakeholder* atau Pemangku Kepentingan

Dalam konteks pemangku kepentingan (*stakeholder*), proyek ini melibatkan perguruan tinggi/universitas, mahasiswa, pemerintah dan regulator pendidikan, serta para peneliti atau analis data yang akan mengembangkan model.

d. Kriteria Keberhasilan

Kriteria keberhasilan proyek ini mencakup tingkat akurasi yang layak dari model klasifikasi dalam memprediksi *drop-out* atau kesuksesan akademik mahasiswa. Selain itu, pencapaian kualitas pendidikan yang lebih baik akan tercermin dalam kemampuan model untuk mengidentifikasi faktor-faktor yang dapat

ditingkatkan atau diintervensi, yang akan mendukung upaya untuk meningkatkan pengalaman belajar mahasiswa.

e. Batasan dan Kendala

Batasan dan kendala dalam proyek ini mencakup keterbatasan data, termasuk data historis mahasiswa yang mungkin tidak lengkap atau terperinci. Selain itu, perlu memperhatikan ketatnya kebijakan privasi yang berlaku saat menggunakan data mahasiswa untuk memastikan perlindungan data pribadi dan kepatuhan terhadap regulasi privasi yang berlaku.

2. Pemahaman Data (*Data Understanding*)

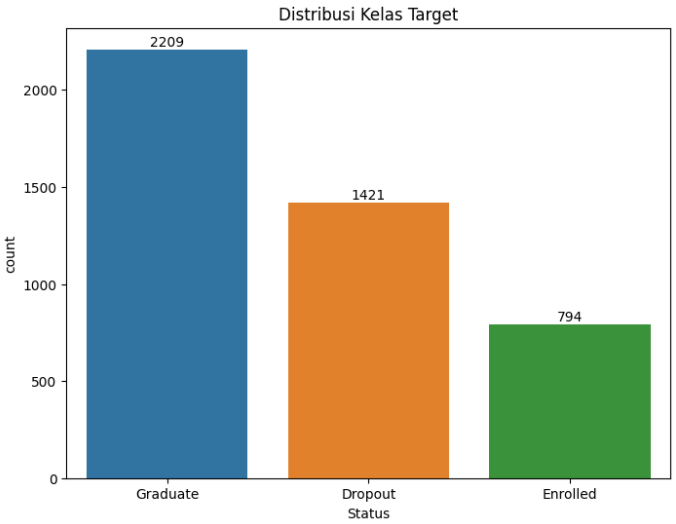
a. Pengenalan Dataset

Dataset yang digunakan pada proyek diperoleh dari UCI Machine Learning Repository dan dikumpulkan oleh Realinho, Valentim, Vieira Martins, Mónica, Machado, Jorge, dan Baptista, Luís pada tahun 2021 [1], [7]. Dataset ini berisi beragam jenis data, termasuk data demografis, sosial-ekonomi, makroekonomi, data saat pendaftaran mahasiswa, dan data pada akhir semester pertama dan kedua. Dataset mencakup catatan mahasiswa dari tahun akademik 2008/2009 hingga 2018/2019, dengan data dari 17 program sarjana berbagai bidang. Dataset ini tersedia dalam format CSV dengan 4424 data, 35 atribut, 1 target dan tanpa nilai *null* maupun duplikat. Tipe data dalam dataset ini sangat beragam, dengan sebagian besar atribut berupa bilangan bulat (int64) dan bilangan pecahan (float64). Namun, ada juga satu atribut yang memiliki tipe data objek (object) yang berisi informasi teks. Berikut adalah atribut yang digunakan pada kasus ini:

Kategori Data	Atribut
Data Demografis	● Status Pernikahan (Marital Status) ● Kebangsaan (Nationality) ● Terdisplasi (Displaced) ● Jenis Kelamin (Gender) ● Usia saat Pendaftaran (Age at Enrollment) ● Status Internasional (International)
Data Sosial-Ekonomi	● Kualifikasi Ibu (Mother’s Qualification) ● Kualifikasi Ayah (Father’s Qualification) ● Pekerjaan Ibu (Mother’s Occupation) ● Pekerjaan Ayah (Father’s Occupation) ● Kebutuhan Pendidikan Khusus (Educational Special Needs) ● Peminjam (Debtor) ● Kelunasan Uang Sekolah (Tuition Fees Up to Date) ● Penerima Beasiswa (Scholarship Holder)
Data Makroekonomi	● Tingkat Pengangguran (Unemployment Rate) ● Tingkat Inflasi (Inflation Rate) ● Produk Domestik Bruto (GDP)
Data Akademik saat Pendaftaran	● Mode Pendaftaran (Application Mode) ● Urutan Pendaftaran (Application Order) ● Program Studi (Course) ● Kehadiran Siang/Malam (Daytime/Evening

	Attendance) ● Kualifikasi Sebelumnya (Previous Qualification)
Data Akademik pada Akhir Semester Pertama	● Mata Kuliah Kurikuler Semester Pertama (Credited) ● Mata Kuliah Terdaftar Semester Pertama (Enrolled) ● Mata Kuliah dengan Evaluasi Semester Pertama (Evaluations) ● Mata Kuliah yang Lulus Semester Pertama (Approved) ● Nilai Mata Kuliah Semester Pertama (Grade) ● Mata Kuliah Semester Pertama tanpa Evaluasi (Without Evaluations)
Data Akademik pada Akhir Semester Kedua	● Mata Kuliah Kurikuler Semester Kedua (Credited) ● Mata Kuliah Terdaftar Semester Kedua (Enrolled) ● Mata Kuliah dengan Evaluasi Semester Kedua (Evaluations) ● Mata Kuliah yang Lulus Semester Kedua (Approved) ● Nilai Mata Kuliah Semester Kedua (Grade) ● Mata Kuliah Semester Kedua tanpa Evaluasi (Without Evaluations)
Target	● Status (Status)

Pada kolom ‘Status’, terdapat 3 kelas atau keluaran yang akan dihasilkan oleh algoritma yang dipilih, yaitu "Graduate" (Lulus) sebanyak 2209 data, "Dropout" (Putus Sekolah) sebanyak 1421 data, dan "Enrolled" (Terdaftar) sebanyak 794 data.



b. Pemilihan Atribut

Selanjutnya, karena dataset memiliki banyak atribut, kita akan melakukan seleksi atribut untuk mengoptimalkan sumber daya dan menghindari penggunaan atribut yang tidak memberikan kontribusi signifikan terhadap hasil akhir model. Dalam kasus ini, kita akan

menggunakan metode Korelasi. Korelasi adalah ukuran statistik yang mengukur sejauh mana dua variabel bergerak bersama-sama dalam rentang -1 hingga 1. Kita akan menghitung nilai korelasi dengan metode *Pearson Correlation* antara setiap atribut dan kolom ‘Status’, dan hanya memilih 10 atribut dengan korelasi tertinggi untuk digunakan dalam pembangunan model lebih lanjut. Berikut adalah hasil nilai korelasi masing-masing atribut terhadap kolom ‘Status’:

```
Urutan Korelasi Tertinggi dengan Kolom 'Status':
Status                                          1.000000
Curricular_units_2nd_sem_approved               0.624157
Curricular_units_2nd_sem_grade                  0.566827
Curricular_units_1st_sem_approved               0.529123
Curricular_units_1st_sem_grade                  0.485207
Tuition_fees_up_to_date                         0.409827
Scholarship_holder                              0.297595
Curricular_units_2nd_sem_enrolled               0.175847
Curricular_units_1st_sem_enrolled               0.155974
Admission_grade                                 0.120889
Displaced                                       0.113986
Previous_qualification_grade                    0.103764
Curricular_units_2nd_sem_evaluations            0.092721
Application_order                               0.089791
Daytime_evening_attendance                      0.075107
Curricular_units_2nd_sem_credited               0.054004
Curricular_units_1st_sem_credited               0.048150
Curricular_units_1st_sem_evaluations            0.044362
GDP                                             0.044135
Course                                          0.034219
Unemployment_rate                               0.008627
International                                   0.003934
Fathers_qualification                          -0.001393
Fathers_occupation                             -0.001899
Mothers_occupation                             -0.005629
Educational_special_needs                      -0.007353
Nacionality                                    -0.014801
Inflation_rate                                 -0.026874
Mothers_qualification                          -0.043178
Previous_qualification                         -0.056039
Curricular_units_1st_sem_without_evaluations   -0.068702
Marital_status                                 -0.089804
Curricular_units_2nd_sem_without_evaluations   -0.094028
Application_mode                               -0.221747
Gender                                         -0.229270
Debtor                                         -0.240999
Age_at_enrollment                              -0.243438
```

Terdapat 10 atribut dalam dataset yang memiliki nilai korelasi tertinggi terhadap kolom 'Status,' yang merupakan atribut target dalam penelitian ini. Atribut-atribut ini memberikan kontribusi signifikan terhadap prediksi status mahasiswa, baik itu lulus, putus sekolah, atau terdaftar. Dalam urutan dari yang memiliki korelasi tertinggi hingga terendah, atribut pertama adalah "Curricular_units_2nd_sem_approved" dengan korelasi sebesar 0.624157, yang diikuti oleh "Curricular_units_2nd_sem_grade" dengan korelasi 0.566827. atribut-atribut berikutnya yang juga berpengaruh kuat termasuk "Curricular_units_1st_sem_approved" (korelasi: 0.529123) dan "Curricular_units_1st_sem_grade" (korelasi: 0.485207).

Selanjutnya, "Tuition_fees_up_to_date" (korelasi: 0.409827) dan "Scholarship_holder" (korelasi: 0.297595) juga memiliki korelasi yang cukup signifikan. Terdapat pula beberapa atribut lain yang memiliki pengaruh, seperti "Curricular_units_2nd_sem_enrolled" (korelasi: 0.175847) dan "Curricular_units_1st_sem_enrolled" (korelasi: 0.155974). "Admission_grade" (korelasi: 0.120889) dan "Displaced" (korelasi: 0.113986) juga menjadi faktor penting dalam prediksi status mahasiswa. Informasi ini menjadi landasan penting dalam memahami faktor-faktor yang memengaruhi kinerja mahasiswa dalam konteks penelitian ini.

c. Analisis Statistik Deskriptif dari Atribut Terpilih

Dalam bagian ini, menjelaskan statistika deskriptif membantu untuk memberikan gambaran yang lebih baik tentang dataset yang digunakan dalam proyek analisis kinerja mahasiswa. Statistik deskriptif adalah metode yang digunakan untuk merangkum karakteristik dasar dari data, seperti pemusatan data (mean, median), sebaran data (rentang, simpangan baku), dan distribusi data (histogram, diagram batang). Dalam konteks ini, telah diidentifikasi sepuluh fitur dengan nilai korelasi tertinggi terhadap kolom 'Status,' yang mengindikasikan hubungan antara fitur-fitur ini dengan status mahasiswa (lulus, putus sekolah, atau terdaftar). Statistik deskriptif membantu dalam memberikan pemahaman yang lebih dalam tentang sejauh mana fitur-fitur ini mempengaruhi hasil akhir mahasiswa dalam dataset, serta memberikan pandangan yang lebih lengkap tentang karakteristik dataset tersebut.

Atribut	Mean	Standar Deviasi	Min	Q1	Median atau Q2	Q3	Max
Curricular_units_2nd_sem_approved	4.435805	3.014764	0.000000	2.000000	5.000000	6.000000	20.000000
Curricular_units_2nd_sem_grade	10.230206	5.210808	0.000000	10.750000	12.200000	13.333333	6.000000
Curricular_units_1st_sem_approved	4.706600	3.094238	0.000000	3.000000	5.000000	18.571429	26.000000
Curricular_units_1st_sem_grade	10.640822	4.843663	0.000000	11.000000	12.285714	13.400000	18.875000
Tuition_fees_up_to_date	0.880651	0.324235	0.000000	1.000000	1.000000	1.000000	1.000000
Scholarship_holder	0.248418	0.432144	0.000000	0.000000	0.000000	0.000000	1.000000

Curricular_units_2nd_sem_enrolled	6.232143	2.195951	0.000000	5.000000	6.000000	7.000000	23.000000
Curricular_units_1st_sem_enrolled	6.270570	2.480178	0.000000	5.000000	6.000000	7.000000	26.000000
Admission_grade	126.978119	14.482001	95.000000	117.900000	126.100000	134.800000	190.000000
Displaced	0.548373	0.497711	0.000000	0.000000	1.000000	1.000000	1.000000

3. Persiapan Data (*Data Preparation*)

a. Penghapusan nilai *null*

Pada dataset yang digunakan, tidak terdapat nilai null maupun duplikat, sehingga tidak perlu dilakukan tahapan penghapusan atau penggantian nilai. Oleh karena itu, dataset yang digunakan dapat dianggap bersih dan siap untuk melanjutkan ke tahapan persiapan data selanjutnya.

b. Transformasi data

Pada tahap transformasi data, dilakukan konversi dari representasi data yang sebelumnya bersifat kategorikal atau berbentuk huruf (tipe objek) menjadi representasi data yang bersifat numerik. Hal ini menjadi penting karena algoritma analisis data yang akan digunakan hanya dapat mengolah data dalam bentuk numerik. Dalam dataset yang telah dipilih, terdapat tiga kolom yaitu 'Tuition_fees_up_to_date', 'Scholarship_holder', dan 'Displaced', yang masih memiliki tipe data objek. Masing-masing kolom ini memiliki nilai-nilai unik 'Yes' dan 'No'. Oleh karena itu, agar data dapat digunakan untuk analisis lebih lanjut, maka akan dilakukan pengubahan representasi ini, di mana 'Yes' akan diubah menjadi '1' dan 'No' menjadi '0' menggunakan *library* LabelEncoder.

c. Pemisahan Data (*Data Splitting*)

Dalam tahap pemisahan data, dilakukan pembagian data menjadi dua set, yaitu set pelatihan (train set) dan set pengujian (test set). Proses ini memiliki peran penting dalam pengembangan model machine learning untuk mengevaluasi kinerja model secara obyektif. Menggunakan fungsi `train_test_split` dari pustaka Scikit-Learn, data dibagi menjadi dua bagian: data pelatihan (train set) dan data pengujian (test set). Set pelatihan (train set) terdiri dari 3539 sampel dan 10 fitur, sedangkan set pengujian (test set) terdiri dari 885 sampel dengan jumlah fitur yang sama.

Pemisahan data ini penting karena selanjutnya model machine learning akan dilatih menggunakan data pelatihan dan diuji pada data pengujian untuk mengukur kemampuan model dalam menggeneralisasi hasilnya ke data yang belum pernah dilihat sebelumnya. Ini merupakan langkah kunci dalam pengembangan model yang dapat diandalkan untuk melakukan prediksi dengan baik.

d. Normalisasi

Pada tahap Normalisasi, dilakukan penskalaan fitur dengan menggunakan library yang disebut `StandardScaler`. Tujuannya adalah untuk memastikan bahwa semua fitur dalam dataset memiliki skala yang seragam. Pertama, dilakukan perhitungan statistik penting seperti rata-rata dan deviasi standar dari setiap fitur pada data pelatihan untuk memahami sebaran data fitur-fitur tersebut. Informasi statistik yang telah dihitung ini kemudian digunakan untuk mengubah skala data pelatihan dan data pengujian. Hasilnya adalah data yang telah disesuaikan sehingga memiliki rata-rata 0 dan deviasi standar 1.

Proses normalisasi ini penting karena beberapa algoritma machine learning dapat menghasilkan prediksi yang tidak akurat jika fitur-fitur memiliki skala yang berbeda-beda. Dengan penskalaan fitur ini, memastikan bahwa semua fitur memiliki pengaruh yang setara dalam pembentukan model, sehingga meningkatkan kemampuan model untuk membuat prediksi yang lebih baik.

4. Pemodelan (*Modelling*)

Pada pembangunan klasifikasi kinerja mahasiswa, digunakan beberapa algoritma Machine Learning (ML), yaitu Decision Tree, Random Forest, dan KNN. Pemilihan algoritma bergantung pada karakteristik dataset dan tujuan analisis. Decision Tree cocok digunakan untuk pemahaman lebih dalam tentang hubungan antar variabel, Random Forest untuk meningkatkan akurasi, dan KNN untuk masalah yang kompleks dan berubah-ubah. Selain itu, akan dilakukan Hyperparameter Tuning menggunakan metode Gridsearch CV.

Gridsearch CV adalah sebuah teknik yang digunakan untuk mencari kombinasi terbaik dari parameter-parameter yang ada dalam algoritma ML. Teknik ini akan secara sistematis mencoba berbagai kombinasi parameter yang telah ditentukan sebelumnya, seperti kedalaman maksimum pohon dalam Decision Tree, jumlah pohon dalam Random Forest, atau jumlah tetangga terdekat dalam KNN.

Tujuan dari hyperparameter tuning menggunakan Gridsearch CV adalah untuk meningkatkan kinerja model dengan mencari parameter-parameter optimal yang menghasilkan hasil terbaik, seperti akurasi atau presisi. Dengan demikian, proses hyperparameter tuning dengan Gridsearch CV akan membantu memaksimalkan potensi algoritma yang digunakan dalam menganalisis kinerja mahasiswa. Parameter yang digunakan dan rentangnya masing-masing dapat ditampilkan pada Tabel I, II, dan III

Tabel I Parameter Hyperparameter Tuning Decision Tree

Decision Tree		
Parameter	Fungsi	Nilai
criterion	Menentukan metrik yang digunakan untuk mengukur kualitas pemisahan simpul.	['gini', 'entropy']
max_depth	Mengontrol kedalaman maksimum dari pohon keputusan.	[None, 10, 20]
min_samples_split	Menentukan jumlah sampel minimum untuk membagi simpul.	[2, 5]
min_samples_leaf	Mengatur jumlah sampel minimum untuk simpul daun.	[1, 2]
Parameter terbaik Decision Tree: {'criterion': 'gini', 'max_depth': 10, 'min_samples_leaf': 1, 'min_samples_split': 5}		

Tabel II Parameter Hyperparameter Tuning Random Forest

Random Forest		
Parameter	Keterangan	Nilai
n_estimators	Jumlah pohon keputusan dalam ensemble.	[100, 200, 300]
max_depth	Kedalaman maksimum setiap pohon keputusan.	[None, 10, 20, 30]
min_samples_split	Jumlah sampel minimum untuk membagi simpul.	[2, 5, 10]

min_samples_leaf	Jumlah sampel minimum untuk simpul daun.	[1, 2, 4]
Parameter terbaik untuk Random Forest: {'max_depth': 10, 'min_samples_leaf': 4, 'min_samples_split': 10, 'n_estimators': 100}		

Tabel III Parameter Hyperparameter Tuning KNN

K-Nearest Neighbors (KNN)		
Parameter	Keterangan	Nilai
n_neighbors	Menentukan jumlah tetangga terdekat yang akan digunakan untuk membuat prediksi.	[3, 5, 7]
weights	Menentukan metode bobot yang digunakan dalam perhitungan jarak antara titik data.	['uniform', 'distance']
P	Parameter ini mengontrol jenis jarak yang digunakan dalam perhitungan.	[1, 2]
Parameter terbaik untuk KNN: Hyperparameter terbaik: {'n_neighbors': 7, 'p': 1, 'weights': 'uniform'}		

5. Evaluasi (*Evaluation*)

Setelah melatih model, model tersebut digunakan untuk membuat prediksi pada data pengujian, dan kemudian dilakukan perhitungan matrik evaluasi seperti akurasi, presisi, recall, dan F1-Score. Hasil matrik evaluasi ini memberikan gambaran tentang sejauh mana model KNN mampu memprediksi kinerja mahasiswa dengan baik. Evaluasi adalah tahap untuk membandingkan hasil yang diperoleh dari implementasi dengan kriteria dan standar yang telah ada untuk mendapatkan angka keberhasilan dari implementasi [8].

Pada kasus ini, digunakan confusion matrix sebagai metode validasi model yang telah dibangun. Confusion matrix yaitu sebuah tabel yang terdiri atas banyaknya baris data uji yang diprediksi benar dan tidak benar oleh model klasifikasi, tabel ini diperlukan untuk menentukan kinerja suatu model klasifikasi [9]. Adapun parameter performansi yang digunakan adalah Akurasi, Recall, Precision, dan F1-Score. Confusion matrix untuk kelas prediksi dan kelas actual dapat dilihat pada Tabel 3, dimana TP, TN, FP, FN berturut-turut adalah *True Positive, True Negative, False Positive, dan False Negative* yang ditampilkan pada Tabel IV.

Tabel IV *Confusion Matrix*

Kategori	**Kelas Prediksi (*Predicted Class*)**		
Kelas Aktual (*Actual Class*)		Positif (+)	Negatif (-)
	Positif (+)	TP	FN
	Negatif (-)	FP	TN

Akurasi (Q) merupakan tingkat kedekatan antara nilai prediksi yang telah dilakukan dengan nilai yang sebenarnya. Akurasi digunakan untuk melakukan evaluasi agar mengetahui banyaknya jumlah label prediksi yang tepat sesuai dengan label aktual. Akurasi memiliki konsep yaitu semakin besar nilai Akurasi nya, maka performansi dari klasifikasi yang telah dilakukan akan semakin baik. Recall atau Sensitivity (SE) adalah tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi. Precision (PR) adalah tingkat ketepatan antara informasi yang diharapkan oleh pengguna atau user dengan jawaban yang dihasilkan oleh sistem yang telah dibangun. F1-Score adalah perhitungan untuk mengukur performansi dari gabungan nilai Precision dan Recall.

$$Akurasi = \frac{TP + TN}{TP + FP + TN + FN}$$

$$Recall = \frac{TP}{TP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$F1 - Score = 2\,x\,\frac{Precision\,x\;Recall}{Precision + Recall}$$

Berikut hasil validasi model setelah Hyperparameter Tuning terdapat pada Tabel 5:

Tabel V Hasil Evaluasi

Algoritma	Akurasi	Precision	Recall	F1-Score
Decision Tree	73.22%	67.13%	65.94%	66.41%
Random Forest	**76.27%**	70.59%	67.32%	68.26%
KNN	72.09%	65.14%	62.52%	63.17%

Hasil analisis menunjukkan bahwa Random Forest memiliki performa terbaik dengan akurasi sekitar 76.27%, presisi sekitar 70.59%, recall sekitar 67.32%, dan F1-Score sekitar 68.26%. Decision Tree memiliki performa sedang dengan akurasi sekitar 73.22%, presisi sekitar 67.13%, recall sekitar 65.94%, dan F1-Score sekitar 66.41%. Sedangkan KNN memiliki performa yang lebih rendah dibandingkan kedua algoritma lainnya dengan akurasi sekitar 72.09%, presisi sekitar 65.14%, recall sekitar 62.52%, dan F1-Score sekitar 63.17%. Dengan demikian, Random Forest menjadi pilihan terbaik untuk memprediksi kinerja mahasiswa berdasarkan hasil evaluasi performa algoritma.

6. Implementasi (*Deployment*)

Pada tahap akhir, hasil analisis diterapkan dalam konteks bisnis atau organisasi. Tujuannya adalah mengimplementasikan model dalam produksi atau memberikan rekomendasi kepada pemangku kepentingan. Sebelumnya, telah diidentifikasi bahwa algoritma Random Forest memiliki akurasi tertinggi, yaitu sekitar 76.27%, sehingga model tersebut akan digunakan untuk memprediksi data baru. Dalam tahap deployment ini, kita akan menggunakan library Streamlit untuk mengembangkan antarmuka pengguna yang

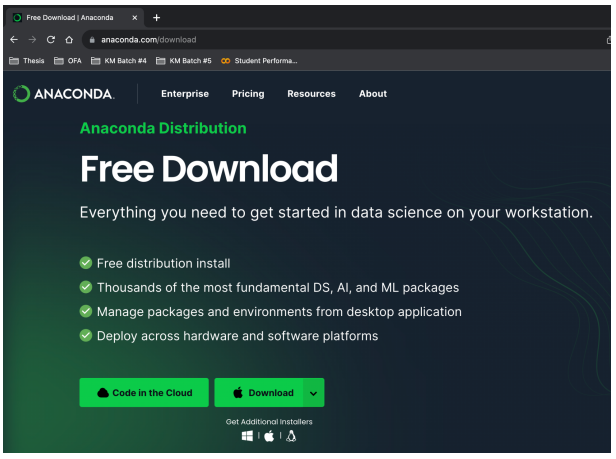
interaktif. Antarmuka pengguna ini akan memungkinkan pengguna (misalnya, staf akademik atau *administrator*) untuk memasukkan data mahasiswa yang baru dan mendapatkan prediksi tentang kinerja mahasiswa tersebut. Hal ini akan membantu dalam mengambil tindakan yang tepat dan memberikan dukungan yang diperlukan kepada mahasiswa berdasarkan prediksi model. Dengan menggunakan Streamlit, kita dapat dengan mudah mengintegrasikan model Random Forest ke dalam antarmuka pengguna yang ramah pengguna tanpa perlu pengetahuan pemrograman yang mendalam.

Dalam panduan ini, akan dijelaskan cara menggunakan Anaconda untuk membuat dan menjalankan aplikasi Streamlit dalam lingkungan virtual yang terisolasi. Dengan menggunakan virtual environment, Anda dapat mengelola dependensi proyek dengan lebih baik, menghindari konflik dengan paket sistem, dan memastikan kestabilan aplikasi. Anaconda adalah platform open-source yang digunakan untuk pengelolaan dan distribusi lingkungan pemrograman, paket, dan perangkat lunak data science. Anaconda menyediakan cara yang sangat efisien untuk mengatur lingkungan pengembangan data science, termasuk pengelolaan dependensi dan paket, serta menyediakan banyak paket yang umum digunakan dalam analisis data, ilmu data, dan pengembangan aplikasi.

Anaconda juga dilengkapi dengan manajer paket bernama Conda, yang memungkinkan pengguna untuk membuat dan mengelola lingkungan virtual yang berisi versi-versi spesifik dari paket-paket Python dan alat-alat lain yang diperlukan untuk proyek tertentu. Ini memungkinkan pengembang data science untuk menjalankan proyek mereka dalam lingkungan yang terisolasi, sehingga menghindari konflik dependensi antar-paket.

Berikut adalah langkah-langkah lengkap dan terstruktur untuk menjalankan aplikasi Streamlit dengan bantuan Anaconda dalam sebuah virtual environment:

1. Unduh dan instal Anaconda dari situs web resmi: https://www.anaconda.com/products/distribution

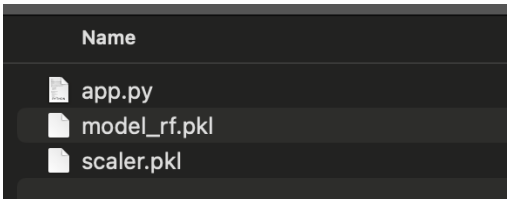


   Pilih versi Anaconda yang sesuai dengan sistem operasi Anda (Windows, macOS, atau Linux) dan unduh installer-nya. Lalu, ikuti panduan instalasi yang tersedia di situs web Anaconda untuk menginstal Anaconda di komputer Anda. Pastikan untuk memilih opsi "Add Anaconda to my PATH environment variable" selama proses instalasi.

2. Buat file app.py dengan kode sebagai berikut:

```python
import streamlit as st
import pickle
import numpy as np

# Judul aplikasi
st.title("Aplikasi Prediksi Status Performa Mahasiswa")

import streamlit as st

# Input fitur-fitur
Curricular_units_1st_sem_enrolled = st.slider("Jumlah SKS yang Didaftarkan Mahasiswa pada Semester 1",
min_value=0, max_value=26, value=0)
Curricular_units_1st_sem_approved = st.slider("Jumlah SKS yang Lulus Mahasiswa pada Semester 1", min_value=0,
max_value=26, value=0)
Curricular_units_1st_sem_grade = st.number_input("Nilai Semester 1", min_value=0.0, max_value=19.0, value=0.0)

Curricular_units_2nd_sem_enrolled = st.slider("Jumlah SKS yang Didaftarkan Mahasiswa pada Semester 2",
min_value=0, max_value=23, value=0)
Curricular_units_2nd_sem_approved = st.slider("Jumlah SKS yang Lulus Mahasiswa pada Semester 2", min_value=0,
max_value=20, value=0)
Curricular_units_2nd_sem_grade = st.number_input("Nilai Semester 2", min_value=0.0, max_value=19.0, value=0.0)

# 1 Yes 0 No
Tuition_fees_up_to_date = st.radio("Pelunasan Uang Pendidikan (Iya (1); Tidak (0))", ("1", "0"))
# 1 Yes 0 No
Scholarship_holder = st.radio("Penerima Beasiswa (Iya (1); Tidak (0))", ("1", "0"))
Admission_grade = st.number_input("Nilai Penerimaan", min_value=0.0, max_value=200.0, value=0.0)
Displaced = st.radio("Apakah Mahasiswa Orang Terlantar? (Iya (1); Tidak (0))", ("1", "0"))


# Data dalam bentuk list
data = [
    [
        Curricular_units_2nd_sem_approved,
        Curricular_units_2nd_sem_grade,
        Curricular_units_1st_sem_approved,
        Curricular_units_1st_sem_grade,
        Tuition_fees_up_to_date,
        Scholarship_holder,
        Curricular_units_2nd_sem_enrolled,
        Curricular_units_1st_sem_enrolled,
        Admission_grade,
        Displaced
    ]
]

# Load model dan skaler yang telah disimpan sebelumnya
scaler = pickle.load(open('scaler.pkl', 'rb'))
best_model = pickle.load(open('model_rf.pkl', 'rb'))

# Ketika tombol "Prediksi" ditekan
if st.button("Prediksi"):
    # Standardisasi data
    data_scaled = scaler.transform(data)

    # Prediksi hasil Status
    hasil_prediksi = best_model.predict(data_scaled)
    hasil_prediksi = int(hasil_prediksi)

    # Mapping hasil prediksi ke label yang sesuai
    if hasil_prediksi == 0:
        status = "Dropout"
    elif hasil_prediksi == 1:
        status = "Enrolled"
    else:
        status = "Graduate"

    # Menampilkan hasil prediksi
    st.write(f"Hasil Prediksi Status: {status}")
```

3. Simpan file app.py dalam satu folder bersama dengan model pickle yang telah diperoleh selama proses pembelajaran di Google Colaboratory.



4. Buka terminal atau command prompt (CMD) pada folder tempat Anda menyimpan 3 file di atas. Pada contoh ini, 3 file tersebut disimpan dalam folder ‘Streamlit’.

```
Last login: Mon Oct  9 13:15:07 on console
angelmetanosaafinda@Angels-MacBook-Pro Streamlit %
```

5. Buat sebuah virtual environment baru untuk proyek Streamlit menggunakan perintah berikut:
   conda create --name kinerja_mahasiswa

```
angelmetanosaafinda@Angels-MacBook-Pro Streamlit % conda create --name kinerja_mahasiswa
Collecting package metadata (current_repodata.json): done
Solving environment: done


==> WARNING: A newer version of conda exists. <==
  current version: 22.9.0
  latest version: 23.9.0

Please update conda by running

    $ conda update -n base -c defaults conda



## Package Plan ##

  environment location: /Users/angelmetanosaafinda/opt/anaconda3/envs/kinerja_mahasiswa



Proceed ([y]/n)? y

Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate kinerja_mahasiswa
#
# To deactivate an active environment, use
#
#     $ conda deactivate

Retrieving notices: ...working... done
```

6. Aktifkan virtual environment yang telah Anda buat dengan menjalankan perintah berikut:
   conda activate kinerja_mahasiswa

```
angelmetanosaafinda@Angels-MacBook-Pro Streamlit % conda activate kinerja_mahasiswa
(kinerja_mahasiswa) angelmetanosaafinda@Angels-MacBook-Pro Streamlit %
```

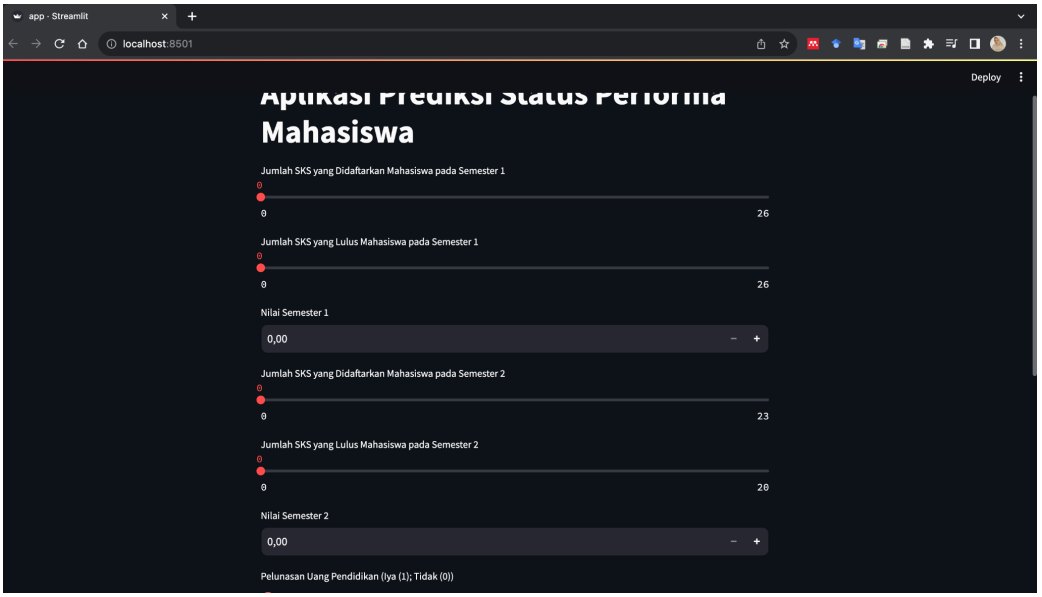
7. Setelah virtual environment diaktifkan, Anda dapat menginstal Streamlit ke dalamnya dengan perintah:
   pip install streamlit

```
(kinerja_mahasiswa) angelmetanosaafinda@Angels-MacBook-Pro Streamlit % pip install streamlit
Requirement already satisfied: streamlit in /Library/Frameworks/Python.framework/Versions/3.9/lib/python3.9/site-packages (1.27.2)
Requirement already satisfied: altair<6,>=4.0 in /Library/Frameworks/Python.framework/Versions/3.9/lib/python3.9/site-packages (from streamlit) (5.1.2)
Requirement already satisfied: blinker<2,>=1.0.0 in /Library/Frameworks/Python.framework/Versions/3.9/lib/python3.9/site-packages (from streamlit) (1.6.2)
Requirement already satisfied: cachetools<6,>=4.0 in /Library/Frameworks/Python.framework/Versions/3.9/lib/python3.9/site-packages (from streamlit) (5.3.0)
Requirement already satisfied: click<9,>=7.0 in /Library/Frameworks/Python.framework/Versions/3.9/lib/python3.9/site-packages (from streamlit) (8.1.3)
Requirement already satisfied: importlib-metadata<7,>=1.4 in /Library/Frameworks/Python.framework/Versions/3.9/lib/python3.9/site-packages (from streamlit) (6
.6.0)
```

8. Di dalam virtual environment student_performance, Anda dapat menjalankan aplikasi Streamlit dengan perintah berikut:
   streamlit run nama_app.py

```
(kinerja_mahasiswa) angelmetanosaafinda@Angels-MacBook-Pro Streamlit % streamlit run app.py

  You can now view your Streamlit app in your browser.

  Local URL: http://localhost:8501
  Network URL: http://192.168.1.17:8501

  For better performance, install the Watchdog module:

  $ xcode-select --install
  $ pip install watchdog
```

9. Setelah aplikasi dijalankan, Anda akan melihat URL lokal tempat aplikasi Streamlit berjalan. Buka URL tersebut di browser Anda untuk melihat aplikasi Streamlit yang Anda buat. Selanjutnya Anda bisa menginputkan nilai untuk masing-masing atribut untuk mendapat hasil klasifikasi yang sesuai.



10. Untuk menutup aplikasi Streamlit yang sedang berjalan, kembali ke terminal atau command prompt, lalu tekan ‘Ctrl + C’.

11. Untuk menonaktifkan virtual environment, Anda dapat menjalankan perintah:
    conda deactivate

E. KESIMPULAN

Dalam proyek ini, dilakukan analisis terhadap kinerja mahasiswa berdasarkan berbagai atribut demografis, ekonomi, dan akademik. Berbagai algoritma machine learning, seperti Decision Tree, Random Forest, dan K-Nearest Neighbors (KNN) diuji untuk memprediksi kinerja mahasiswa. Hasil analisis menunjukkan bahwa algoritma Random Forest memberikan akurasi tertinggi, mencapai tingkat akurasi sebesar 76.27%. Pentingnya penyetelan hyperparameter juga ditekankan dalam proyek ini, dengan menggunakan Grid Search untuk menemukan parameter terbaik untuk setiap algoritma. Selanjutnya, model Random Forest yang telah dioptimalkan dapat diimplementasikan dalam produksi dengan menggunakan library Streamlit, memungkinkan pengguna untuk melakukan prediksi kinerja mahasiswa secara interaktif. Kesimpulannya, proyek ini tidak hanya memberikan wawasan mengenai faktor-faktor yang memengaruhi kinerja mahasiswa, tetapi juga menyediakan alat yang bermanfaat untuk pengambilan keputusan dalam konteks pendidikan.

DAFTAR PUSTAKA

[1] V. Realinho, J. Machado, L. Baptista, and M. V. Martins, “Predicting Student Dropout and Academic Success,” *Data*, vol. 7, no. 11, Art. no. 11, Nov. 2022, doi: 10.3390/data7110146.

[2] B. Sekeroglu, K. Dimililer, and K. Tuncal, “Student Performance Prediction and Classification Using Machine Learning Algorithms,” in *Proceedings of the 2019 8th International Conference on Educational and Information Technology*, in ICEIT 2019. New York, NY, USA: Association for Computing Machinery, Mar. 2019, pp. 7–11. doi: 10.1145/3318396.3318419.

[3] H. Al-Shehri *et al.*, “Student performance prediction using Support Vector Machine and K-Nearest Neighbor,” in *2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*, Apr. 2017, pp. 1–4. doi: 10.1109/CCECE.2017.7946847.

[4] B. Albreiki, N. Zaki, and H. Alashwal, “A Systematic Literature Review of Student’ Performance Prediction Using Machine Learning Techniques,” *Education Sciences*, vol. 11, no. 9, Art. no. 9, Sep. 2021, doi: 10.3390/educsci11090552.

[5] N. Hotz, “What is CRISP DM?,” Data Science Process Alliance. Accessed: Oct. 08, 2023. [Online]. Available: https://www.datascience-pm.com/crisp-dm-2/

[6] “Cross-Industry Standard Process for Data Mining (CRISP-DM),” MMSI BINUS University. Accessed: Oct. 08, 2023. [Online]. Available: https://mmsi.binus.ac.id/2020/09/18/cross-industry-standard-process-for-data-mining-crisp-dm/

[7] “UCI Machine Learning Repository.” Accessed: Oct. 08, 2023. [Online]. Available: https://archive.ics.uci.edu/dataset/697/predict+students+dropout+and+academic+success

[8] M. Mudasir, I. Tahir, and I. Putri, “QUANTITATIVE STRUCTURE AND ACTIVITY RELATIONSHIP ANALYSIS OF 1,2,4-THIADIAZOLINE FUNGICIDES BASED ON MOLECULAR STRUCTURE CALCULATED BY AM1 METHOD,” *Indonesian Journal of Chemistry*, vol. 3, pp. 39–47, Mar. 2003, doi: 10.22146/ijc.21904.

[9] “ANALISIS SENTIMEN TWITTER UNTUK MENGETAHUI KESAN MASYARAKAT TENTANG PELAKSANAAN POMPROV JAWA TIMUR TAHUN 2022 DENGAN PERBANDINGAN METODE NAÏVE BAYES CLASSIFIER DAN DECISION TREE BERBASIS SMOTE | Jurnal Informatika Dan Tekonologi Komputer (JITEK).” Accessed: Oct. 09, 2023. [Online]. Available: https://journal.amikveteran.ac.id/index.php/jitek/article/view/551