PENERAPAN MACHINE LEARNING DALAM MEMPERSONALISASI PROFIL PENGGUNA BERDASARKAN AKTIVITAS PEMBELAJARAN ONLINE

Proposal Lomba ICONIC-IT

Diajukan Untuk Memenuhi Salah Satu Syarat

Penyelesaian Lomba

ICONIC IT 2024

Nama Tim:

Royals Outlaws

Nama Anggota Tim:

Thesion Marta Sianipar (1206210004)

Rendika Nurhartanto Suharto (1206210011)

Rizal Rahman Rizkika (1206210010)

SURABAYA

2024

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR RUMUS

BAB I
LATAR BELAKANG

Penerapan *machine learning* dalam mempersonalisasi profil pengguna berdasarkan aktivitas pembelajaran *online* adalah topik yang semakin relevan di era digital ini. Dengan semakin berkembangnya teknologi dan meningkatnya penggunaan *platform* pembelajaran *online*, personalisasi menjadi penting untuk meningkatkan efektivitas dan efisiensi belajar. *Personalization* membantu menciptakan pengalaman belajar yang lebih sesuai dengan kebutuhan, minat, dan kemampuan individu, yang secara signifikan dapat meningkatkan hasil pembelajaran dan keterlibatan siswa.

Penelitian menunjukkan bahwa sistem rekomendasi yang dipersonalisasi dapat meningkatkan penggunaan sumber daya pendidikan dan mendorong otonomi serta efisiensi belajar siswa. Sebagai contoh, penelitian oleh Xiao et al. (2018) mengembangkan sistem rekomendasi yang menggabungkan berbagai algoritma untuk mempersonalisasi pengalaman belajar di platform *Massive Open Online Courses* (MOOCs). Sistem ini terbukti dapat meningkatkan efisiensi penggunaan sumber daya pendidikan dan membantu siswa belajar secara lebih mandiri[1].

Selain itu, platform pembelajaran *online* yang mempersonalisasi konten pembelajaran sesuai dengan kebutuhan pengguna telah terbukti lebih efektif dalam meningkatkan hasil belajar dibandingkan dengan metode pembelajaran tradisional. Penelitian oleh St-Hilaire et al. (2021) menemukan bahwa *platform* yang mempersonalisasi lingkungan belajar dan menyediakan umpan balik yang sesuai dengan profil individu menunjukkan peningkatan signifikan dalam hasil pembelajaran dibandingkan dengan platform yang menggunakan pendekatan tradisional.[2]

Penelitian lainnya juga mendukung pentingnya personalisasi dalam pembelajaran online. Prihar et al. (2022) mengembangkan *Automatic Personalized Learning Service* (APLS) yang menggunakan algoritma multi-armed bandit untuk merekomendasikan dukungan pembelajaran yang paling efektif bagi siswa. APLS terbukti dapat meningkatkan pembelajaran siswa hingga 10% lebih tinggi dibandingkan algoritma tradisional.[3]

Dalam konteks pendidikan modern, personalisasi pembelajaran tidak hanya memenuhi kebutuhan individual, tetapi juga menjadi solusi efektif untuk mengatasi tantangan dalam pembelajaran online yang bersifat massal. MOOCs, misalnya, menghadapi tantangan besar dalam menyampaikan materi pendidikan yang relevan kepada peserta dari berbagai latar belakang. Teknologi personalisasi memungkinkan platform ini untuk menyesuaikan materi berdasarkan kebutuhan spesifik pengguna, yang pada akhirnya dapat meningkatkan keterlibatan pengguna dan keberhasilan pembelajaran.

Platform seperti eTutor yang dikembangkan oleh Tekin et al. (2014) menunjukkan bahwa personalisasi dalam pembelajaran online memungkinkan adaptasi metodologi

pengajaran berdasarkan umpan balik real-time dari siswa, sehingga dapat memaksimalkan kinerja siswa dalam ujian akhir sambil meminimalkan waktu yang diperlukan untuk belajar.[4]

Komleva dan Vilyavin (2020) juga mengembangkan *platform* digital yang memungkinkan pembuatan kursus online yang adaptif secara personal. *Platform* ini tidak hanya memungkinkan penyesuaian konten kursus sesuai dengan tingkat kompetensi individu siswa, tetapi juga berkontribusi dalam mengurangi beban pengajar dan meningkatkan kualitas kursus yang ditawarkan.[5]

Dalam proyek ini, beberapa model machine learning dipilih untuk dibandingkan dalam mempersonalisasi profil pengguna berdasarkan aktivitas pembelajaran *online*, yaitu *Quadratic Discriminant Analysis* (QDA), *Extreme Gradient Boosting* (XGBoost), dan *K Neighbors Classifier* (KNN).

1. *Quadratic Discriminant Analysis* (QDA) dipilih karena kemampuannya dalam menangani masalah klasifikasi di mana kelompok kelas tidak seimbang dan memiliki variabilitas antar kelas yang berbeda. QDA sangat cocok untuk memodelkan variasi kebutuhan belajar pengguna yang mungkin tidak linier dan tidak homogen. Penelitian oleh Mudrák et al. (2020) menunjukkan bahwa penggunaan model adaptif yang disesuaikan dengan karakteristik individu pengguna dapat meningkatkan motivasi dan hasil belajar.[6]

2. *Extreme Gradient Boosting* (XGBoost) adalah model yang sangat efektif untuk menangani dataset besar dan kompleks dengan akurasi tinggi. XGBoost sering digunakan dalam berbagai aplikasi karena kemampuannya untuk mengatasi overfitting dan memberikan prediksi yang akurat, terutama dalam konteks big data yang sering ditemui dalam platform pembelajaran online. Penelitian oleh Guo et al. (2023) mendukung penggunaan teknologi digital untuk personalisasi pembelajaran yang memerlukan pendekatan komputasi yang canggih seperti XGBoost.[7]

3. *K Neighbors Classifier* (KNN) dipilih karena kesederhanaannya dan efektivitasnya dalam mengelompokkan data berdasarkan kedekatan dalam ruang fitur. KNN sangat cocok untuk memodelkan hubungan lokal antara data pengguna, seperti dalam sistem rekomendasi yang memerlukan pengelompokan pengguna dengan karakteristik belajar yang serupa. Lee dan Ferwerda (2017) dalam penelitiannya menunjukkan bahwa pendekatan berbasis KNN dapat meningkatkan keterlibatan pengguna dengan menyesuaikan konten sejak awal penggunaan platform pembelajaran online.[8]

Dengan memilih model-model ini, proyek ini berusaha untuk memaksimalkan akurasi prediksi dalam memberikan rekomendasi konten yang dipersonalisasi, sehingga dapat membantu pengguna mencapai hasil belajar yang optimal dengan metode pembelajaran yang paling sesuai dengan kebutuhan mereka.

BAB II
TUJUAN DAN MANFAAT DIKEMBANGKANNYA MODEL DAN ANALISIS

II.1 Tujuan

Penelitian ini bertujuan untuk mengembangkan pendekatan personalisasi dalam pembelajaran *online* dengan memanfaatkan teknologi *machine learning*. Tujuan-tujuan khusus dari penelitian ini adalah sebagai berikut:

1. Mengidentifikasi faktor-faktor penting yang mempengaruhi kinerja dan aktivitas belajar pengguna di platform pendidikan *online*. Melakukan analisis mendalam terhadap data aktivitas belajar pengguna di platform pendidikan *online* untuk mengidentifikasi variabel-variabel kunci yang berkontribusi terhadap kinerja akademik. Variabel-variabel ini mungkin mencakup frekuensi akses, durasi belajar, jenis konten yang sering diakses, serta interaksi dengan fitur-fitur lain di platform. Identifikasi faktor ini akan dilakukan dengan menggunakan teknik eksplorasi data dan analisis statistik.

2. Membangun model *machine learning* yang mampu memprediksi profil pengguna berdasarkan data aktivitas belajar mereka. Mengembangkan dan menguji beberapa model machine learning seperti *Quadratic Discriminant Analysis* (QDA), *Extreme Gradient Boosting* (XGBoost), dan *K Neighbors Classifier* (KNN) untuk memprediksi profil pengguna. Model-model ini akan dilatih menggunakan dataset aktivitas pengguna untuk mempersonalisasi konten pembelajaran yang ditawarkan. Kinerja model akan dievaluasi berdasarkan akurasi, kecepatan, dan kemampuannya dalam menangani dataset yang beragam dan dinamis.

3. Menyediakan wawasan berharga bagi pengelola *platform* untuk mengembangkan kurikulum yang lebih efektif dan sesuai dengan kebutuhan pengguna. Menghasilkan rekomendasi praktis bagi pengelola platform pendidikan online untuk mengoptimalkan kurikulum yang ada berdasarkan hasil prediksi profil pengguna. Wawasan yang diperoleh dari model machine learning yang dikembangkan akan membantu dalam menyusun strategi personalisasi yang lebih efektif, yang dapat mencakup penyesuaian materi ajar, metode pengajaran, dan pemberian umpan balik yang lebih relevan dan tepat waktu. Wawasan ini diharapkan dapat membantu meningkatkan hasil belajar dan keterlibatan pengguna secara keseluruhan.

II.2 Manfaat

Penelitian ini diharapkan dapat memberikan berbagai manfaat, baik bagi pengguna *platform* pendidikan *online*, pengelola *platform*, maupun perkembangan teknologi dalam bidang pendidikan secara umum. Berikut adalah beberapa manfaat utama dari penelitian ini:

1. Meningkatkan Efektivitas Pembelajaran bagi Pengguna

   Pengguna *platform* pendidikan online dapat memperoleh pengalaman belajar yang lebih dipersonalisasi dan sesuai dengan kebutuhan serta preferensi mereka. Dengan adanya konten yang disesuaikan, pengguna dapat lebih fokus dan efektif dalam belajar, yang pada akhirnya meningkatkan hasil akademik dan keterlibatan mereka.

2. Memberikan Wawasan bagi Pengelola *Platform* Pendidikan

   Pengelola platform dapat memanfaatkan hasil penelitian ini untuk mengembangkan fitur-fitur dan strategi personalisasi yang lebih canggih dan tepat sasaran. Dengan pemahaman yang lebih baik mengenai faktor-faktor yang mempengaruhi kinerja belajar pengguna, pengelola dapat merancang kurikulum dan metode pengajaran yang lebih adaptif dan responsif.

3. Mengembangkan Teknologi Pembelajaran yang Lebih Maju

   Penelitian ini berkontribusi pada perkembangan teknologi dalam bidang pendidikan dengan mengaplikasikan model machine learning yang inovatif. Model-model ini tidak hanya membantu dalam personalisasi pembelajaran, tetapi juga dapat diadaptasi untuk berbagai konteks lain dalam pendidikan, seperti asesmen otomatis dan rekomendasi sumber daya pendidikan.

4. Mempercepat Inovasi dalam Pendidikan *Online*

   Dengan menunjukkan manfaat dari personalisasi berbasis *machine learning*, penelitian ini dapat mendorong adopsi teknologi serupa di berbagai platform pendidikan lainnya. Hal ini dapat mempercepat inovasi dan peningkatan kualitas pendidikan online secara umum.

5. Meningkatkan Otonomi dan Motivasi Belajar Siswa

   Dengan pembelajaran yang lebih dipersonalisasi, siswa dapat merasa lebih termotivasi dan memiliki kendali lebih besar atas proses belajar mereka. Ini dapat mendorong siswa untuk menjadi lebih mandiri dan bertanggung jawab dalam mencapai tujuan pembelajaran mereka.

II.3 Keluaran yang Diharapkan

Penelitian ini diharapkan menghasilkan beberapa keluaran spesifik yang dapat memberikan dampak positif pada bidang pendidikan online, terutama dalam aspek personalisasi pembelajaran. keluaran yang diharapkan meliputi:

1. Pengembangan Model *Machine Learning* untuk Personalisasi Pembelajaran

   Terbentuknya model machine learning yang mampu memprediksi profil pengguna berdasarkan data aktivitas belajar mereka. Model ini diharapkan dapat memberikan rekomendasi konten pembelajaran yang lebih akurat dan sesuai dengan kebutuhan individu pengguna.

2. Peningkatan Efisiensi dan Kualitas Pembelajaran *Online*

   Implementasi model yang dikembangkan dalam platform pendidikan online diharapkan dapat meningkatkan efisiensi belajar pengguna, termasuk dalam hal waktu yang diperlukan untuk mencapai tujuan pembelajaran serta kualitas hasil belajar yang dicapai.

3. Penyusunan Rekomendasi Strategis untuk Pengelola *Platform*

   Hasil penelitian diharapkan memberikan wawasan berharga bagi pengelola *platform* pendidikan *online* dalam mengembangkan kurikulum dan fitur personalisasi yang lebih efektif. Ini mencakup rekomendasi tentang bagaimana konten dan umpan balik dapat disesuaikan dengan kebutuhan pengguna berdasarkan prediksi model.

4. Peningkatan Keterlibatan dan Motivasi Belajar Pengguna

   Diharapkan bahwa personalisasi yang lebih baik akan meningkatkan keterlibatan pengguna dengan platform pembelajaran, serta meningkatkan motivasi mereka untuk belajar. keluaran ini dapat diukur melalui peningkatan frekuensi penggunaan *platform*, durasi belajar, dan kepuasan pengguna.

5. Validasi dan Pengujian Efektivitas Model

   keluaran penting lainnya adalah validasi empiris dari efektivitas model-model yang digunakan, termasuk *Quadratic Discriminant Analysis* (QDA), *Extreme Gradient Boosting* (XGBoost), dan *K Neighbors Classifier* (KNN). Ini akan mencakup evaluasi akurasi prediksi, kemampuan model untuk menangani dataset yang beragam, dan dampaknya terhadap hasil pembelajaran.

Dengan tercapainya keluaran-keluaran ini, penelitian diharapkan dapat memberikan kontribusi nyata dalam meningkatkan personalisasi dan efektivitas pembelajaran *online*, serta memberikan dasar bagi pengembangan lebih lanjut dalam bidang ini.

BAB III
METODOLOGI PENGEMBANGAN MODEL YANG DILAKUKAN

III.1 Pendekatan yang Digunakan

III.1.1 CRISPDM - alur pengerjaan

Pendekatan yang digunakan dalam proyek ini adalah **CRISP-DM (*Cross-Industry Standard Process for Data Mining*)**, sebuah metodologi yang banyak diadopsi dalam proyek data mining. Metodologi ini terdiri dari beberapa fase yang saling berkaitan, dirancang untuk memberikan panduan struktural dalam proses pengembangan data mining. Setiap fase dalam CRISP-DM dirancang untuk memastikan bahwa proyek berjalan dengan sistematis dan efisien, mulai dari pemahaman bisnis hingga penerapan model. Penjelasan lebih lanjut mengenai setiap fase akan dijelaskan dalam sub bab berikut ini.

Gambar 1 Proses CRISP-DM

Sumber: https://ruthsitorus.medium.com/mempelajari-modeling-cross-industry-standard-process-for-data-mining-atau-crisp-dm-166735c14159

1. *Business Understanding*

   Tahap pertama dalam pendekatan CRISP-DM (*Cross-Industry Standard Process for Data Mining*) adalah *Business Understanding*, di mana fokus utamanya adalah memahami tujuan bisnis yang ingin dicapai melalui proyek *data mining*. Langkah ini melibatkan identifikasi masalah bisnis, penetapan tujuan proyek, dan perumusan strategi untuk mencapai tujuan tersebut. Dengan memahami konteks bisnis, tim data dapat menentukan pertanyaan-pertanyaan spesifik yang perlu dijawab dan bagaimana data dapat digunakan untuk memberikan solusi yang tepat. Hal ini juga melibatkan komunikasi yang erat

dengan pemangku kepentingan untuk memastikan bahwa proyek ini selaras dengan kebutuhan bisnis.[9]

2. *Data Understanding*

Pada tahap *Data Understanding*, fokusnya adalah mengumpulkan data awal dan memahami karakteristik dasar dari data tersebut. Langkah ini melibatkan eksplorasi data awal untuk mengidentifikasi pola, outlier, dan hubungan antar variabel. Selain itu, pemahaman data juga membantu dalam menentukan kualitas data, mengidentifikasi adanya data yang hilang atau anomali, serta memahami distribusi variabel. Tahap ini sangat penting karena memungkinkan pengambil keputusan untuk melihat apakah data yang ada dapat menjawab pertanyaan bisnis yang telah diidentifikasi sebelumnya.[10]

3. *Data Preparation*

Tahap *Data Preparation* melibatkan proses pembersihan dan pengubahan data mentah menjadi format yang dapat digunakan oleh model. Langkah ini mencakup berbagai aktivitas seperti penanganan data yang hilang, penghapusan *outlier*, transformasi variabel, pengkodean fitur kategorikal, dan normalisasi atau standardisasi data. *Data preparation* sering kali menjadi tahap yang paling memakan waktu dalam proyek *data mining*, tetapi sangat penting untuk memastikan bahwa data yang digunakan adalah berkualitas tinggi dan relevan untuk analisis lebih lanjut.[11]

4. *Modeling*

Dalam tahap *Modeling*, data yang telah dipersiapkan digunakan untuk membangun model prediktif atau deskriptif yang dapat memberikan wawasan dari data. Tahap ini melibatkan pemilihan algoritma yang tepat, melatih model dengan data latih, dan mengoptimalkan parameter model untuk meningkatkan kinerja. Berbagai teknik seperti regresi, pohon keputusan, atau neural networks dapat digunakan tergantung pada jenis data dan tujuan analisis. Pemodelan sering kali bersifat iteratif, dengan model yang dibangun, dievaluasi, dan disempurnakan secara berulang-ulang hingga mencapai kinerja yang diinginkan.[12]

5. *Evaluation*

Tahap *Evaluation* bertujuan untuk menilai seberapa baik model yang telah dibangun dalam memenuhi tujuan bisnis dan memberikan jawaban yang relevan terhadap pertanyaan bisnis. Evaluasi melibatkan penggunaan berbagai metrik seperti akurasi, *precision*, *recall*, dan *F1-score* untuk model klasifikasi, atau MSE dan MAE untuk model regresi. Selain itu, evaluasi juga mencakup validasi terhadap data uji atau melalui metode *cross-validation* untuk memastikan model tidak *overfitting* dan memiliki generalisasi yang baik pada data baru. Jika model tidak memenuhi harapan, perlu dilakukan revisi atau iterasi pada tahap sebelumnya.[13]

6. *Data Presentation*

   Data Presentation adalah tahap akhir dalam CRISP-DM di mana hasil dari analisis dan model disajikan kepada pemangku kepentingan. Presentasi hasil ini bisa dalam bentuk laporan tertulis, presentasi visual, *dashboard* interaktif, atau kombinasi dari semuanya. Tujuan dari data *presentation* adalah untuk memastikan bahwa hasil yang disajikan mudah dipahami dan dapat diinterpretasikan dengan baik oleh pemangku kepentingan yang mungkin tidak memiliki latar belakang teknis. Pada tahap ini, penting untuk menekankan implikasi bisnis dari temuan dan rekomendasi untuk tindakan lebih lanjut.[14]

III.1.2 *Model Selection* (Metode Evaluasi Model, *Cross Validation*)

Selain pendekatan CRISP-DM, proyek ini juga menggunakan pendekatan ***Model Selection*** dalam machine learning. *Model Selection* adalah proses penting dalam *machine learning* yang bertujuan untuk memilih model terbaik dari sejumlah model kandidat berdasarkan kinerja pada data tertentu. Ada dua metode utama yang digunakan dalam pemilihan model: Evaluasi Model dan *Cross-Validation*. Berikut adalah penjelasan lebih lanjut tentang kedua metode tersebut:

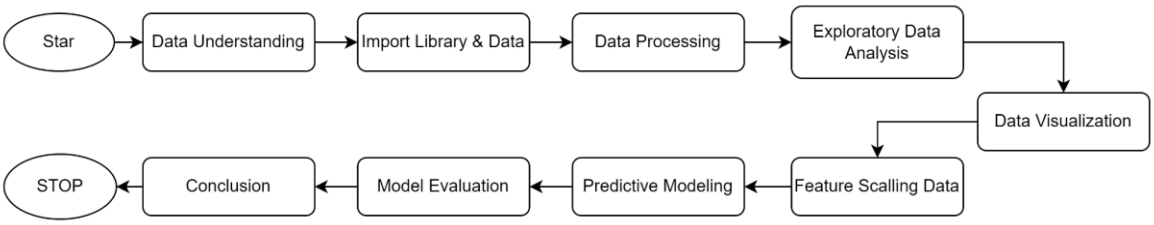
1. Metode Evaluasi Model

   Metode Evaluasi Model berfokus pada pengukuran kinerja model dengan menggunakan berbagai metrik evaluasi seperti akurasi, *precision*, *recall*, *F1-score*, dan *area under the curve* (AUC) untuk klasifikasi, atau *mean squared error* (MSE) dan *mean absolute error* (MAE) untuk regresi. Metrik ini memberikan gambaran tentang seberapa baik model memprediksi data baru, dan digunakan untuk membandingkan beberapa model kandidat. Dalam beberapa kasus, *trade-off* antara metrik tertentu harus diperhatikan, seperti antara precision dan recall. Evaluasi ini sering dilakukan pada data yang terpisah dari data yang digunakan untuk melatih model, yaitu data validasi atau data uji.[15]

2. *Cross-Validation*

   Cross-Validation adalah metode yang lebih canggih untuk mengevaluasi dan memilih model dengan cara yang lebih andal. Metode ini membagi dataset menjadi beberapa subset atau "*folds*". Model kemudian dilatih pada subset tertentu dan diuji pada subset lainnya, dengan proses ini diulang beberapa kali hingga setiap subset menjadi data uji satu kali. Teknik yang paling umum adalah *k-fold cross-validation*, di mana dataset dibagi menjadi k bagian yang sama. Hasil dari setiap iterasi kemudian dirata-rata untuk memberikan estimasi yang lebih stabil dari kinerja model. *Cross-validation* sangat berguna untuk mengurangi overfitting dan memastikan bahwa model yang dipilih memiliki generalisasi yang baik pada data yang tidak terlihat.[16]

III.2 Tahapan Pengembangan



Gambar 2 Tahap-tahap pengembangan

1. *Data Understanding*

   Data Understanding adalah tahap awal yang penting dalam setiap proyek data, di mana tujuan utamanya adalah untuk memahami sifat dan struktur data yang tersedia. Pada tahap ini, kita perlu mengeksplorasi berbagai aspek data, termasuk jenis data (numerik, kategorikal, atau teks), distribusi statistik, dan hubungan antar fitur. Memahami data dengan baik memungkinkan peneliti untuk mengidentifikasi masalah potensial seperti data yang hilang, *outlier*, dan ketidakkonsistenan. Langkah ini juga melibatkan pemahaman konteks bisnis atau domain yang relevan untuk memastikan bahwa analisis selanjutnya akan sesuai dengan tujuan proyek.[17]

2. *Import Library* dan Data

   Import Library dan Data adalah langkah teknis pertama dalam analisis data, di mana pustaka yang diperlukan diimpor dan dataset dimuat ke dalam lingkungan kerja. Pustaka seperti Pandas, NumPy, dan Matplotlib adalah alat dasar yang digunakan untuk manipulasi data, perhitungan matematis, dan visualisasi. Langkah ini memastikan bahwa semua alat dan data yang dibutuhkan tersedia dan siap digunakan untuk analisis lebih lanjut. Setelah pustaka diimpor, data dimuat dari berbagai sumber (misalnya, file CSV, SQL *database*) ke dalam *DataFrame* untuk mempermudah proses analisis.[18]

3. *Data Processing*

   Data Processing adalah proses penting yang mencakup pembersihan, pengubahan, dan penggabungan data. Langkah ini bertujuan untuk memastikan bahwa data siap digunakan dalam model dengan kualitas yang optimal. *Data Processing* mencakup penanganan data yang hilang, penghapusan *outlier*, transformasi data (misalnya, *encoding* variabel kategorikal), dan normalisasi atau standardisasi data. Proses ini memastikan bahwa data yang digunakan dalam model bebas dari masalah kualitas yang dapat menyebabkan kesalahan dalam hasil analisis.[12]

4. *Exploratory Data Analysis* (EDA)

   EDA adalah proses eksplorasi data secara visual dan statistik untuk memahami distribusi, pola, dan hubungan dalam data. EDA digunakan untuk mengidentifikasi fitur penting, mendeteksi *outlier*, dan mengamati hubungan antar variabel. Teknik-teknik visualisasi seperti histogram, *scatter plot*, dan *box plot* sering digunakan dalam tahap ini. EDA juga membantu dalam mengidentifikasi fitur-fitur yang relevan yang dapat digunakan dalam pemodelan prediktif.[19]

5. *Data Visualization*

   Data Visualization adalah proses menampilkan data dalam bentuk visual seperti grafik dan diagram yang membantu dalam pemahaman dan komunikasi hasil analisis. Visualisasi data memudahkan identifikasi pola, tren, dan *outlier* yang mungkin tidak terlihat dalam data mentah. Alat seperti Matplotlib, Seaborn, dan Plotly digunakan untuk membuat berbagai jenis grafik,

yang membantu dalam menggambarkan hasil analisis secara efektif kepada audiens non-teknis.[14]

6. *Feature Scaling*

   Feature Scaling adalah proses penting dalam pemodelan prediktif, di mana data diubah sehingga fitur-fitur memiliki skala yang sama. Hal ini penting untuk algoritma yang mengandalkan pengukuran jarak, seperti *K-Nearest Neighbors* (KNN) dan *Support Vector Machines* (SVM). Teknik yang umum digunakan termasuk normalisasi (mengubah data menjadi rentang 0 hingga 1) dan standardisasi (mengubah data sehingga memiliki mean 0 dan standar deviasi 1). *Feature scaling* membantu memastikan bahwa model tidak memberikan bobot berlebih pada fitur tertentu hanya karena skala mereka lebih besar.[20]

7. *Predictive Modeling*

   Predictive Modeling adalah tahap di mana model dibangun untuk memprediksi hasil dari data baru berdasarkan data yang ada. Model prediktif dapat dibangun menggunakan berbagai algoritma *machine learning* seperti regresi linear, *decision tree*, dan *neural networks*. Proses ini melibatkan pembelajaran dari data latih dan kemudian menerapkan model untuk memprediksi hasil pada data uji. Tujuan utama dari predictive modeling adalah untuk mengembangkan model yang dapat menghasilkan prediksi yang akurat dan handal.[21]

8. *Model Evaluation*

   Model Evaluation adalah proses untuk mengukur kinerja model dan memastikan bahwa model tersebut memenuhi tujuan yang diinginkan. Evaluasi dilakukan dengan menggunakan berbagai metrik seperti akurasi, *precision, recall, F1-score* untuk klasifikasi, atau *mean squared error* (MSE) dan *mean absolute error* (MAE) untuk regresi. Teknik evaluasi yang lebih canggih seperti *cross-validation* juga sering digunakan untuk memastikan bahwa model tidak overfitting dan memiliki generalisasi yang baik terhadap data baru.[13]

9. *Conclusion*

   Kesimpulan adalah tahap terakhir dari proyek di mana hasil dari semua analisis dan pemodelan dirangkum dan disajikan dalam bentuk yang dapat dipahami. Kesimpulan ini mencakup diskusi tentang keberhasilan model dalam memenuhi tujuan awal, serta rekomendasi untuk perbaikan atau penelitian lebih lanjut. Selain itu, bagian ini juga mencakup refleksi atas tantangan yang dihadapi selama proses dan potensi keterbatasan dari pendekatan yang digunakan.[22]

BAB IV

ANALISIS KEBUTUHAN ANALISIS DAN *MODELING*

IV.1 Analisis Kebutuhan

IV.1.1 Kebutuhan Data

Untuk mencapai tujuan dalam memprediksi profil pengguna berdasarkan aktivitas pembelajaran mereka, beberapa jenis data diperlukan:

IV.1.1.1 Data Yang Diperlukan

a) Identitas Pengguna:
   - Nama (NAME): Memberikan konteks dan pengenalan unik pengguna.
   - ID Pengguna (USER_ID): Pengenal unik untuk setiap pengguna yang akan digunakan sebagai indeks.

b) Aktivitas Pembelajaran:
   - Jam Belajar: HOURS_DATASCIENCE, HOURS_BACKEND, HOURS_FRONTEND: Mencatat jumlah jam yang dihabiskan untuk belajar di bidang *data science*, pengembangan *backend*, dan *frontend*

c) Jumlah Kursus:
   - NUM_COURSES_BEGINNER_DATASCIENCE, NUM_COURSES_BEGINNER_BACKEND, NUM_COURSES_BEGINNER_FRONTEND: Jumlah kursus tingkat pemula yang diikuti.
   - NUM_COURSES_ADVANCED_DATASCIENCE, NUM_COURSES_ADVANCED_BACKEND, NUM_COURSES_ADVANCED_FRONTEND: Jumlah kursus tingkat lanjutan yang diikuti.

d) Skor Rata-Rata:
   - AVG_SCORE_DATASCIENCE, AVG_SCORE_BACKEND, AVG_SCORE_FRONTEND: Rata-rata skor yang diperoleh dari kursus-kursus yang diikuti di masing-masing bidang.

e) Target Prediksi:
   - Profil Pengguna (PROFILE): Kategori atau deskripsi profil pengguna berdasarkan aktivitas dan performa belajar mereka.

IV.1.1.2 Sumber Data

Dataset yang digunakan akan disediakan oleh panitia Innovative *Competition and National Informatics Conference* (ICONIC-IT). Data ini sudah mencakup semua metrik yang relevan untuk analisis dan prediksi.

IV.1.1.3 Bagaimana Data Tersebut Digunakan

Data tersebut akan digunakan untuk melatih model *machine learning* dalam memprediksi profil pengguna berdasarkan berbagai metrik aktivitas pembelajaran. Proses ini menggunakan metode *data life cycle* yang komprehensif, yang mencakup pengelolaan *Master Data Management* (MDM) untuk memastikan konsistensi dan integritas data[23]. Setiap langkah dalam siklus hidup data mulai dari pengumpulan data, pemrosesan, analisis eksplorasi, pelatihan model, dan evaluasi hasil prediksi akan diatur dengan prinsip *Data Governance* untuk menjamin kualitas data yang optimal[23]. Hal ini penting agar model yang dilatih dapat menghasilkan prediksi yang akurat dan andal sesuai dengan profil pengguna.

IV.1.2 Kebutuhan Infrastruktur

Infrastruktur teknologi yang diperlukan untuk proyek ini mencakup perangkat keras dan perangkat lunak berikut:

IV.1.2.1 Perangkat Keras

a) Spesifikasi Perangkat:
   - Prosesor: Intel(R) Core(TM) i5-10300H CPU @ 2.50GHz (8CPUs), ~2.5GHz.
   - RAM: 16GB.
   - GPU: NVIDIA GeForce RTX 2060 with Max-Q Design.

Spesifikasi ini mendukung pengolahan data yang besar dan kompleks, serta memungkinkan pelatihan model *machine learning* dengan efisien.

IV.1.2.2 Perangkat Lunak

Sistem operasi yang digunakan adalah Windows 11 x64 sebagai lingkungan utama untuk semua aktivitas pengembangan prediksi. Bahasa pemrograman utama yang digunakan yaitu python untuk semua proses *data life cycle* hingga modeling machine learning.

a) Library Utama:
   - NumPy: Digunakan untuk komputasi numerik dan operasi array.
   - Pandas: Digunakan untuk manipulasi data dan analisis data struktural.
   - Matplotlib: Digunakan untuk visualisasi data, pembuatan grafik dan plot.
   - Scikit-learn: Digunakan untuk implementasi algoritma machine learning dan preprocessing data.
   - Pingouin: Digunakan untuk analisis statistik yang mudah dan komprehensif.

b) IDE/Editor:

   Jupyter Notebook via Google Colab: Digunakan untuk pengembangan, eksperimen model, dan dokumentasi kode secara interaktif.

IV.1.2.3 Lingkungan Pengembangan

Lingkungan pengembangan adalah bagian penting dari proyek ini, di mana saya menggunakan *virtual environment* bawaan dari Google Colab. Ini memungkinkan isolasi proyek dan manajemen dependensi secara efisien. Selain itu, Google Drive digunakan sebagai tempat penyimpanan dataset dan backup hasil eksperimen, memudahkan akses dan pengelolaan data di berbagai perangkat. Dengan kombinasi ini, proyek dapat dijalankan dengan lancar dari tahap awal hingga akhir.

IV.2 Hasil dan Pembahasan

Sub-bab ini memberikan gambaran mengenai tahap *processing data*, hasil eksperimen dan analisis data terkait prediksi menggunakan model *machine learning Quadratic Discriminant Analysis* (QDA), *Extreme Gradient Boosting* (XGBoost), dan *K-Nearest Neighbors* (KNN) terhadap data profil pengguna berdasarkan aktivitas pembelajarannya. Analisis mendalam juga dilakukan untuk mengungkapkan wawasan yang diperoleh dari data tersebut. Selain itu, evaluasi model akan dilakukan untuk menilai kinerja ketiga algoritma, memberikan pemahaman yang lebih komprehensif mengenai faktor-faktor yang mempengaruhi hasil prediksi, serta menetapkan metode seleksi model untuk mendapatkan model terbaik.

IV.2.1 Data Processing

Pada tahap pemrosesan data, dilakukan pemeriksaan menyeluruh terhadap dataset yang digunakan, termasuk pengecekan pencilan (*outliers*), keseimbangan label, dan nilai yang hilang (*missing value*). Hasil pemeriksaan menunjukkan adanya masalah *missing value* dan *outliers* dalam data. Oleh karena itu, beberapa metode pembersihan data dilakukan, seperti imputasi nilai hilang dan penanganan pencilan.

IV.2.1.1 Periksa dan Imputasi Missing Value

Missing values adalah masalah umum yang terjadi ketika sebagian data tidak lengkap atau hilang, biasanya disebabkan oleh berbagai alasan seperti kerusakan alat, perhitungan yang tidak akurat, data yang tidak tercatat, atau masalah teknis lainnya[24]. Pada tahap imputasi *missing value*, data dinilai cukup baik dan bersih untuk dataset yang terdiri dari 20 ribu baris. Namun, upaya maksimal dilakukan untuk menghilangkan *missing value* sekecil apa pun. Untuk mempertahankan seluruh data yang ada dengan cara yang tepat, dilakukan imputasi dengan memisahkan data berdasarkan labelnya. Karena data latih (*train*) sudah memiliki label, hasil imputasi lebih mampu merepresentasikan nilai dari masing-masing label tersebut. Setelah proses imputasi selesai, *missing value* tidak lagi ditemukan. Berikut adalah hasil pemeriksaan sebelum *handling missing value*.

Tabel 1 Hasil pemeriksaan sebelum handling missing value

VARIABLE	COUNT OF NULL
NAME	0
USER_ID	0
HOURS_DATASCIENCE	14
HOURS_BACKEND	53
HOURS_FRONTEND	16
NUM_COURSES_BEGINNER_DATASCIENCE	26
NUM_COURSES_BEGINNER_BACKEND	18
NUM_COURSES_BEGINNER_FRONTEND	39
NUM_COURSES_ADVANCED_DATASCIENCE	2
NUM_COURSES_ADVANCED_BACKEND	8
NUM_COURSES_ADVANCED_FRONTEND	37
AVG_SCORE_DATASCIENCE	220
AVG_SCORE_BACKEND	84
AVG_SCORE_FRONTEND	168
PROFILE	0

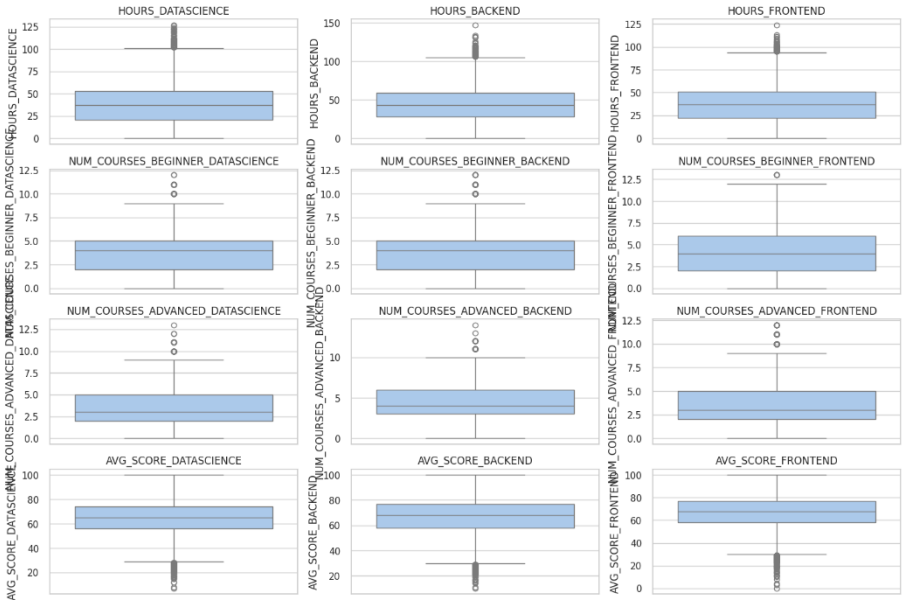
Seperti yang ditunjukkan di atas, jumlah total nilai null mencapai 685 baris. Namun, karena proses imputasi akan dilakukan dengan membagi dataset berdasarkan kelas atau labelnya, diharapkan hasil imputasi akan lebih merepresentasikan karakteristik masing-masing label.

Setelah pengecekan dilakukan, metode imputasi diterapkan. Kolom-kolom dipisahkan berdasarkan tipe data, yaitu numerik dan kategorikal (tanpa *missing value*). Hanya kolom dengan tipe data numerik yang diimputasi menggunakan nilai rata-rata (*mean*). Imputasi dilakukan secara terpisah untuk setiap kelas berdasarkan variabel target (class), yaitu kolom 'PROFILE'. Dengan pendekatan ini, nilai imputasi yang digunakan lebih akurat dalam merepresentasikan data dari masing-masing kelas karena perhitungannya tidak mencampur data dari kelas yang berbeda.

Hasilnya, *missing value* telah dihilangkan dan data sekarang bersih dari nilai yang hilang.

IV.2.1.2 Periksa dan Penanganan *Outlier*

Setelah imputasi *missing value*, dilakukan pemeriksaan pencilan (*outlier*) dalam dataset. *Outlier* adalah data yang sangat berbeda dari pengamatan lainnya dan kehadiran *outlier* dapat memengaruhi hasil kesimpulan atau keputusan dalam sebuah penelitian[25]. Metode penanganan outlier menggunakan IQR (*Interquartile Range*) telah dipilih karena metode ini efektif dalam mengidentifikasi dan menangani pencilan tanpa terpengaruh oleh distribusi data yang mungkin tidak normal. Metode IQR ini menghitung rentang antara kuartil pertama (Q1) dan kuartil ketiga (Q3) untuk menentukan batas pencilan suatu data. Jika nilai observasi lebih kecil dari $Q1-1.5*IQR$ atau nilainya lebih besar dari $Q3+1.5*IQR$ maka nilai tersebut dianggap pencilan[26]. Dengan demikian, metode ini memungkinkan deteksi *outlier* yang lebih akurat berdasarkan distribusi data yang ada dan selanjutnya adalah handling *outlier* tersebut. Dengan bantuan visualiasi *box-plot* deteksi *outlier* ini akan mempermudah untuk dibaca dan diketahui. Berikut adalah hasilnya.



Gambar 3 Box-plot outlier metode IQR untuk numerikal data

Pada Gambar 3 diatas terlihat sangat jelas bahwa *outlier* memang masih cukup banyak terlihat pada data, sehingga lebih baiknya pada kasus ini dilakukan handling dengan melakukan *capping* atau *dropping* otulier. *Capping* sendiri adalah penetapan batas untuk fitur dan

menetapkan nilai semua outlier yang melebihi batas ke nilai batas tersebut. Namun setelah *trial and error* maka diputuskan menggunakan *dropping* atau mebuang data *outlier*. Berikut adalah jumlah *outlier* yang didapatkan pada data berdasakan metode IQR dan akan langsung di buang setelahnya.

Tabel 2 Hasil pemeriksaan sebelum handling outlier

NUMERICAL VARIABLE	COUNT OF OUTLIER
HOURS_DATASCIENCE	71
HOURS_BACKEND	84
HOURS_FRONTEND	49
NUM_COURSES_BEGINNER_DATASCIENCE	71
NUM_COURSES_BEGINNER_BACKEND	117
NUM_COURSES_BEGINNER_FRONTEND	5
NUM_COURSES_ADVANCED_DATASCIENCE	57
NUM_COURSES_ADVANCED_BACKEND	72
NUM_COURSES_ADVANCED_FRONTEND	132
AVG_SCORE_DATASCIENCE	105
AVG_SCORE_BACKEND	114
AVG_SCORE_FRONTEND	129

Pada tahap ini, terdapat total 1006 baris *outlier*, yang meskipun relatif kecil dibandingkan dengan jumlah total dataset sebesar 20000 baris, tetap perlu dibuang karena dapat mengganggu analisis. Setelah *outlier* dihapus, dataset menjadi lebih bersih, dengan total data akhir sebanyak 19035 baris.

IV.2.1.3 Feature Scaling

Feature Scaling adalah metode skala ulang atau normalisasi data untuk mengonversi nilai numerik dalam dataset ke skala yang seragam, tanpa mendistorsi perbedaan dalam rentang nilai tersebut[27]. Alasan menggunakan *Feature Scaling* karena rentang data pada dataset memang berbeda antara variabel numerik satu dengan yang lainnya, dan ingin membantu mempercepat proses pembelajaran pada *machine learning*[28]. Metode yang digunakan adalah *minmax normalization*

yaitu mengubah ukuran data dari rentang asli sehingga semua nilai berada dalam rentang 0 hingga 1. Normalisasi min-max dapat dinyatakan dalam persamaan berikut:

$$W_{norm} = \left(\frac{W_i - W_{min}}{W_{max} - W_{min}}\right) \quad (1)$$

Menuju pada persamaan (1) diatas, di mana W_i adalah nilai asli, W_{norm} adalah nilai yang telah dinormalisasi, W_{min} adalah nilai minimum, W_{max} dan adalah nilai maksimum[29].

Berikut adalah contoh beberapa variabel sebelum dan sesudah diterapkannya normalisasi *min-max*.

Tabel 3 Contoh hasil dari sebelum dan sesudah normalisasi min-max

Sebelum		Sesudah	
HOURS_DATASCIENCE	NUM_COURSES_ADVANCED_BACKEND	HOURS_DATASCIENCE	NUM_COURSES_ADVANCED_BACKEND
37	1	0,366337	0,1
16	6	0,158416	0,6
…	…	…	…
38	5	0,376238	0,5
30	5	0,29703	0,5

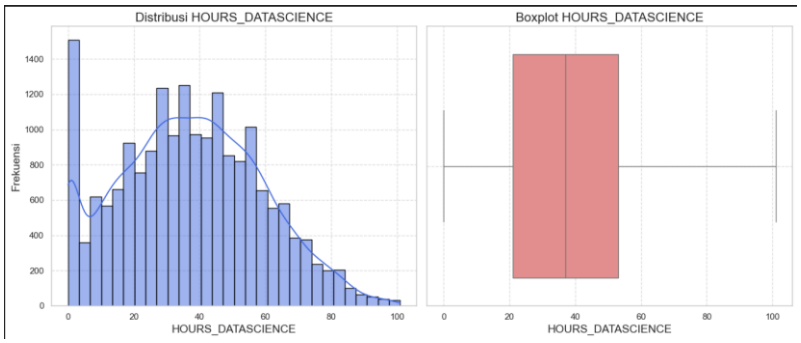
IV.2.2 Exploratory Data Analysis

Exploratory Data Analysis (EDA) adalah sebuah proses penting dalam *data science* yang bertujuan untuk mengenal dan memahami dataset secara mendalam. EDA menggunakan teknik pencarian heuristik untuk menemukan relasi signifikan antara variabel dalam dataset yang besar, memungkinkan kita untuk mendapatkan pemahaman yang lebih baik tentang kondisi data yang kita miliki[30], [31]. Proses ini melibatkan metode eksplorasi data dengan menerapkan teknik aritmatika sederhana dan visualisasi grafis untuk meringkas data observasi, mengidentifikasi pola, menguji hipotesis, serta memeriksa asumsi terkait data[25].

IV.2.2.1 *Univariate Analysis*

Univariate Analysis berfokus pada analisis satu variabel untuk memahami distribusi dan karakteristik data. Pada Gambar 1, analisis variabel `HOURS_DATASCIENCE` menggunakan histogram dan boxplot menunjukkan adanya *skewness* positif. Histogram mengindikasikan bahwa sebagian besar data terkonsentrasi di sisi kiri, dengan beberapa nilai yang jarang dan lebih tinggi di sisi kanan. Ini

menunjukkan bahwa banyak pengguna menghabiskan waktu relatif sedikit untuk *Data Science*, tetapi ada beberapa pengguna yang menghabiskan waktu secara signifikan lebih banyak.

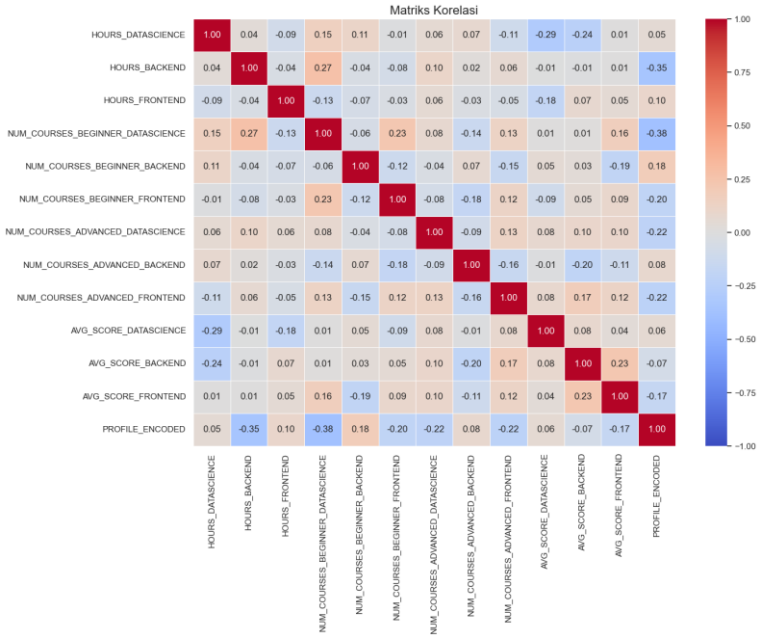


Gambar 4 Histogram dan Boxplo dari HOURS_DATASCIENCE

Pada Gambar 4 *boxplot* mengkonfirmasi adanya *skewness* positif dengan *whisker* bagian kanan yang lebih panjang dibandingkan *whisker* bagian kiri, menunjukkan bahwa sebagian besar data terpusat di bawah median dan terdapat outlier di sisi kanan distribusi. Hal ini menunjukkan bahwa `HOURS_DATASCIENCE` tidak terdistribusi simetris dan memiliki beberapa nilai ekstrem yang tinggi, yang dapat mempengaruhi rata-rata dan memberikan wawasan tambahan mengenai pola distribusi waktu pengguna dalam Data Science.

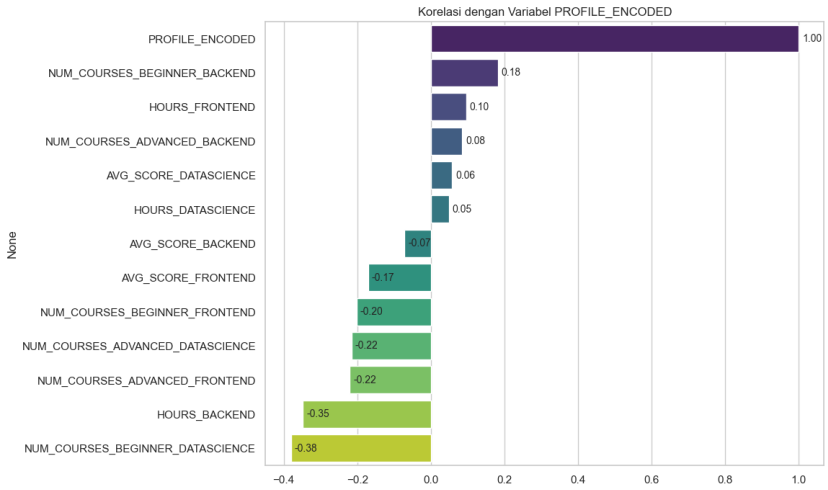
IV.2.2.2 *Bivariate Analysis*

Bivariate Analysis mengeksplorasi hubungan antara dua variabel dengan menggunakan teknik seperti *scatter plot*, *heatmap*, dan matriks korelasi. Heatmap matriks korelasi pada Gambar 5 memberikan wawasan tentang kekuatan dan arah hubungan antara pasangan variabel.



Gambar 5 Korelasi antar fitur menggunakan Heatmap

Analisis ini mengungkapkan beberapa hubungan signifikan. Misalnya, terdapat korelasi negatif sedang antara **HOURS_DATASCIENCE** dan **AVG_SCORE_DATASCIENCE** (-0.29), yang menunjukkan bahwa semakin banyak waktu yang dihabiskan untuk *Data Science*, semakin rendah skor rata-rata yang diperoleh, meskipun hubungan ini tidak sangat kuat. Korelasi negatif yang sama juga ditemukan antara **HOURS_DATASCIENCE** dan **AVG_SCORE_BACKEND** (-0.29), menunjukkan bahwa waktu yang dihabiskan untuk *Data Science* mungkin berdampak negatif atau tidak signifikan terhadap skor di bidang *Backend*. Sebaliknya, **HOURS_BACKEND** memiliki korelasi positif sedang dengan **NUM_COURSES_BEGINNER_DATASCIENCE** (0.27), menunjukkan bahwa individu yang menghabiskan lebih banyak waktu pada *Backend* cenderung mengambil lebih banyak kursus pemula di *Data Science*. Selain itu, **NUM_COURSES_BEGINNER_DATASCIENCE** dan **NUM_COURSES_BEGINNER_FRONTEND** juga menunjukkan korelasi positif sedang (0.27), yang menunjukkan bahwa individu yang mengikuti lebih banyak kursus pemula di *Data Science* juga cenderung mengambil kursus pemula di *Frontend.*

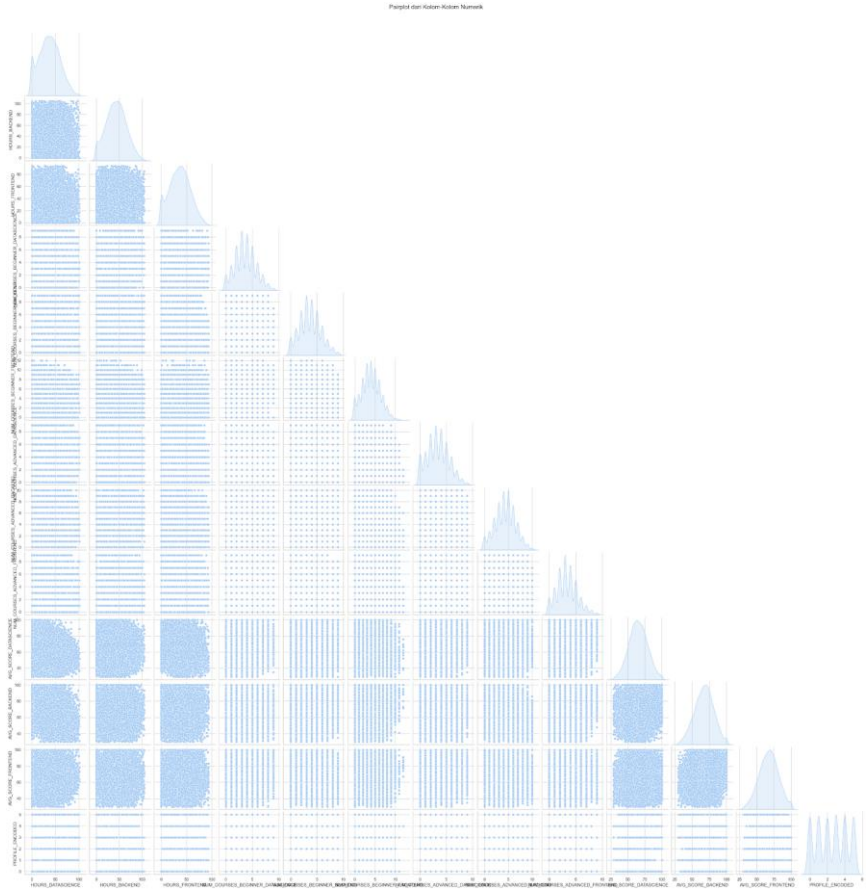


Gambar 6 Korelasi dengan Variabel PROFILE_ENCODED

Pada Gambar 6, analisis korelasi dengan **PROFILE_ENCODED** menunjukkan hubungan negatif sedang antara **NUM_COURSES_BEGINNER_DATASCIENCE** dan **PROFILE_ENCODED** (-0.38), mengindikasikan bahwa semakin banyak kursus pemula yang diambil, profil pengguna cenderung berada pada kategori yang lebih rendah. Hal ini mungkin berarti bahwa pengguna dengan profil lebih tinggi tidak lagi mengikuti kursus pemula. Korelasi negatif juga ditemukan antara **HOURS_BACKEND** dan **PROFILE_ENCODED** (-0.35), yang menunjukkan bahwa semakin banyak waktu yang dihabiskan untuk belajar *Backend*, semakin rendah profil pengguna dalam *encoding*, kemungkinan karena pengguna baru menghabiskan lebih banyak waktu untuk belajar *Backend* sementara pengguna lebih berpengalaman menghabiskan waktu lebih sedikit.

IV.2.2.3 *Multivariate Analysis*

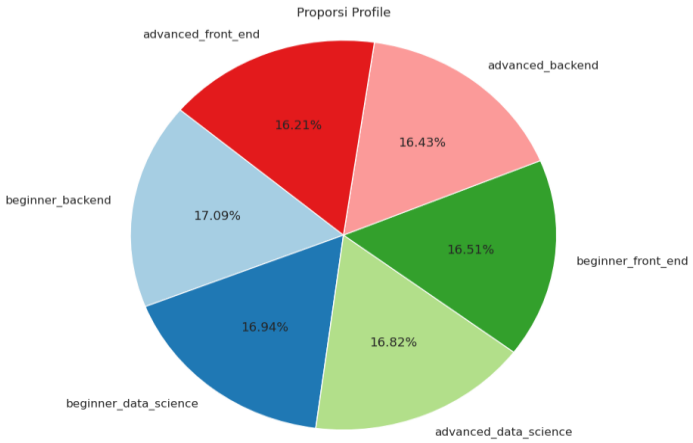
Multivariate Analysis memeriksa hubungan antara lebih dari dua variabel secara simultan. Pairplot adalah alat visualisasi yang efektif untuk analisis multivariat, yang digunakan untuk mengeksplorasi hubungan antara semua pasangan variabel dalam dataset. Dengan menghasilkan *scatter plot* untuk setiap pasangan variabel dan menampilkan distribusi masing-masing variabel di sepanjang diagonal, pairplot memberikan gambaran menyeluruh tentang pola dan hubungan yang ada dalam data. Seperti yang ditunjukkan pada Gambar 7, pairplot memungkinkan kita untuk melihat secara jelas bagaimana variabel-variabel saling berinteraksi dan mengidentifikasi pola atau hubungan yang mungkin tidak terlihat pada analisis univariat atau bivariate.



Gambar 7 Multivariate Analysis menggunakan Pairplot & Scatter Plot

IV.2.3 Hasil Wawasan Berharga

1. Proporsi Profile

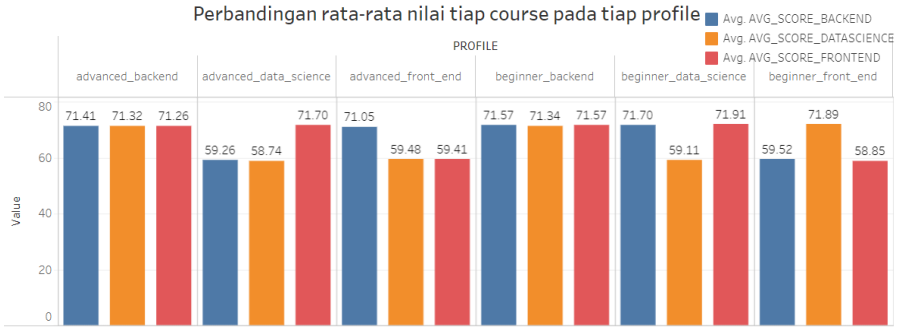


Gambar 8 Grafik proporsi profile

Pada Gambar 8 proporsi Profil Siswa diagram lingkaran yang menunjukkan proporsi dari masing-masing profil siswa yang mengikuti kursus. Profil-profil tersebut adalah:

Beginner Front-End: 16.51%, *Beginner Back-End*: 17.09%, *Beginner Data Science*: 16.94%, *Advanced Front-End*: 16.21%, *Advanced Back-End*: 16.43%, *Advanced Data Science*: 16.82% Proporsi ini menunjukkan distribusi yang relatif merata di antara berbagai profil siswa, dengan sedikit perbedaan di antaranya.

2. Perbandingan Rata-Rata Nilai

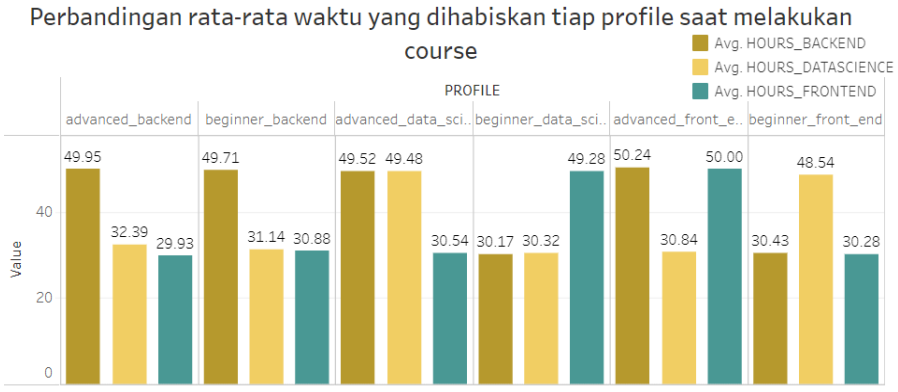


Gambar 9 Grafik perbandingan rata-rata nilai tiap course pada tiap profile

Gambar 9 Menunjukkan perbandingan Rata-Rata Nilai Tiap *Course* pada Tiap Profil Grafik batang di bagian ini membandingkan rata-rata nilai yang diperoleh oleh setiap profil siswa dalam kursus yang berbeda. Data yang ditampilkan adalah rata-rata nilai untuk kursus *Backend*, *Data Science*, dan *Front-End*:

Advanced Back-End: Nilai rata-rata berkisar di sekitar 71.26-71.41. *Beginner Back-End*: Nilai rata-rata berkisar di sekitar 71.34-71.57. *Advanced Data Science*: Nilai rata-rata berkisar di sekitar 58.74-59.26. *Beginner Data Science*: Nilai rata-rata berkisar di sekitar 59.11-71.91. *Advanced Front-End*: Nilai rata-rata berkisar di sekitar 59.48-71.05. *Beginner Front-End*: Nilai rata-rata berkisar di sekitar 58.85-71.89. Perbandingan ini mengindikasikan bahwa siswa pada profil *backend*, baik *beginner* maupun *advanced*, cenderung mendapatkan nilai yang lebih tinggi dibandingkan dengan profil lainnya.

3. Perbandingan Rata-Rata *Spend Time*

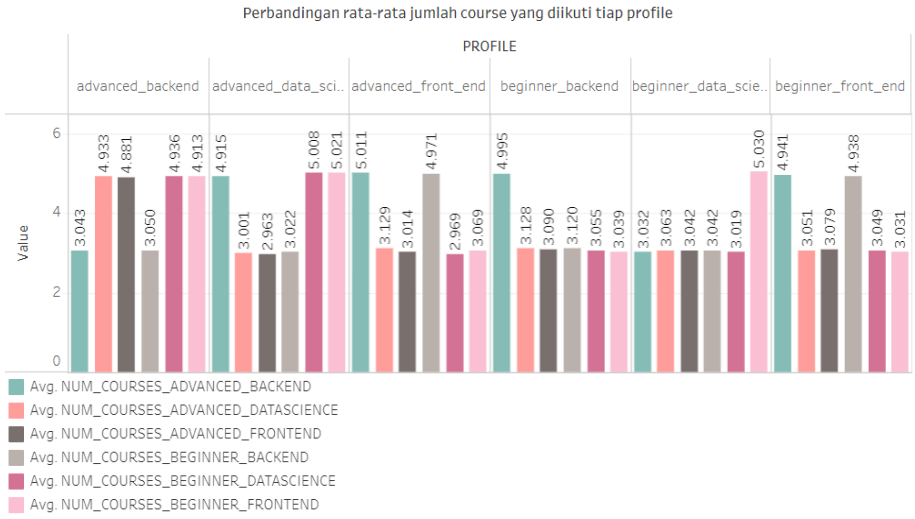


Gambar 10 Grafik perbandingan rata-rata waktu yang dihabiskan tiap profile saat melakukan course

Pada Gambar 10 perbandingan Rata-Rata Waktu yang Dihabiskan Tiap Profil Saat Melakukan Kursus Grafik batang ini menunjukkan perbandingan rata-rata waktu (dalam jam) yang dihabiskan oleh setiap profil siswa dalam mengikuti kursus:

Advanced Back-End: Menghabiskan sekitar 49.95 jam untuk *backend*, 32.39 jam untuk *data science*, dan 29.93 jam untuk *front-end*. *Beginner Back-End*: Menghabiskan sekitar 49.71 jam untuk *backend*, 31.14 jam untuk *data science*, dan 30.88 jam untuk *front-end*. *Advanced Data Science*: Menghabiskan sekitar 49.52 jam untuk *backend*, 30.54 jam untuk *data science*, dan 30.17 jam untuk *front-end*. *Beginner Data Science*: Menghabiskan sekitar 49.48 jam untuk backend, 30.32 jam untuk data science, dan 30.32 jam untuk *front-end*. *Advanced Front-End:* Menghabiskan sekitar 49.28 jam untuk *backend*, 50.24 jam untuk data science, dan 30.84 jam untuk *front-end*. *Beginner Front-End*: Menghabiskan sekitar 48.54 jam untuk *backend*, 30.43 jam untuk *data science*, dan 30.28 jam untuk *front-end*. Data ini menunjukkan bahwa siswa dengan profil *advanced* cenderung menghabiskan waktu lebih banyak dalam kursus mereka dibandingkan dengan siswa dengan profil *beginner*, terutama pada kursus *backend*.

4. Perbandingan Rata-Rata Jumlah *Course*

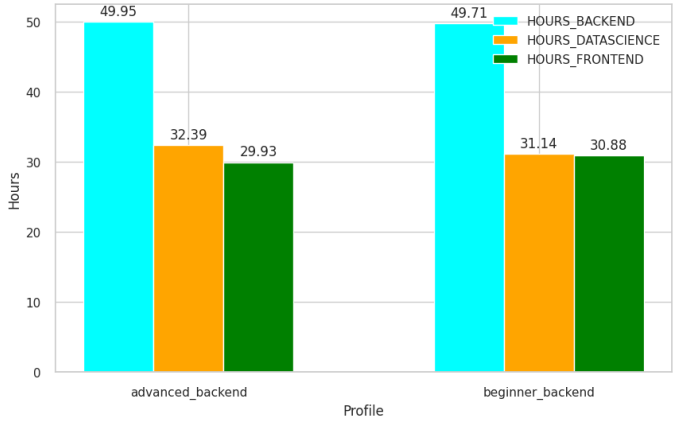


Gambar 11 Grafik perbandingan rata-rata jumlah course yang diikuti tiap profile

Pada Gambar 11 perbandingan Rata-Rata Jumlah Kursus yang Diikuti Tiap Profil Bagian terakhir dari *dashboard* adalah grafik batang yang menunjukkan rata-rata jumlah kursus yang diikuti oleh setiap profil siswa:

Advanced Back-End: Rata-rata mengikuti sekitar 4.933 kursus. *Advanced Data Science*: Rata-rata mengikuti sekitar 4.936 kursus. *Advanced Front-End*: Rata-rata mengikuti sekitar 4.971 kursus. *Beginner Back-End*: Rata-rata mengikuti sekitar 3.128 kursus. *Beginner Data Science*: Rata-rata mengikuti sekitar 3.093 kursus. *Beginner Front-End*: Rata-rata mengikuti sekitar 3.079 kursus. Rata-rata ini menunjukkan bahwa siswa dengan profil *advanced* cenderung mengikuti lebih banyak kursus dibandingkan dengan siswa *beginner*.

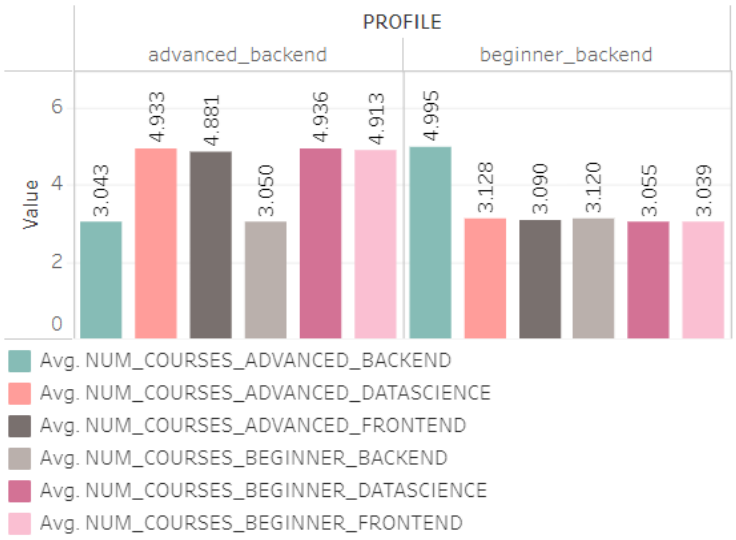
5. Profil *Backend*
   - Perbandingan Rata-Rata waktu belajar dari *Advance* dan *Beginner*



Gambar 12 Perbandingan Rata-Rata waktu belajar dari Advance dan Beginner pada profil backend

Gambar 12 menunjukkan waktu belajar untuk profil *backend advance* jauh lebih tinggi (49.95 jam) dibandingkan dengan profil *beginner* (31.14 jam). Ini menunjukkan bahwa kursus tingkat *advance* dalam pengembangan *backend* mungkin memerlukan lebih banyak waktu belajar karena materi yang lebih kompleks dan mendalam.

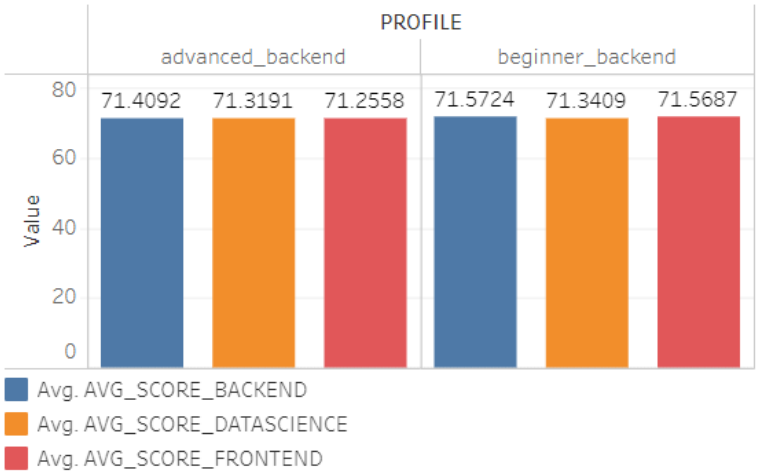
- Perbandingan Rata Jumlah Kursus dari *Advance* dan *Beginner*



Gambar 13 Perbandingan Rata Jumlah Kursus dari Advance dan Beginner pada profil backend

Gambar 13 menunjukkan rata-rata jumlah kursus yang diambil oleh profil *backend advance* lebih tinggi (4.933) dibandingkan dengan profil *backend beginner* (3.128). Hal ini menunjukkan bahwa pembelajar yang sudah berada di level *advance* cenderung mengambil lebih banyak kursus *backend*, mungkin karena materi yang lebih spesifik dan mendalam diperlukan di tingkat ini.

- Perbandingan Rata Nilai Dari *Advance* dan *Beginner*

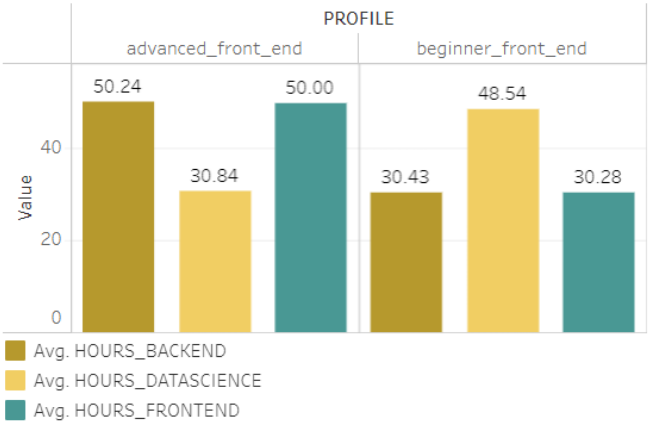


Gambar 14 Perbandingan Rata Nilai Dari Advance dan Beginner pada profil backend

Gambar 14 menunjukkan nilai yang dicapai oleh pembelajar backend cukup konsisten antara profil *beginner* (71.3409) dan profil *advance* (71.4092). Ini bisa menunjukkan bahwa sistem penilaian tidak terlalu bervariasi dengan tingkat kesulitan kursus, atau baik pembelajar *beginner* maupun *advance* mencapai tingkat kompetensi yang serupa dalam kursus *backend*.

6. Profil Frontend

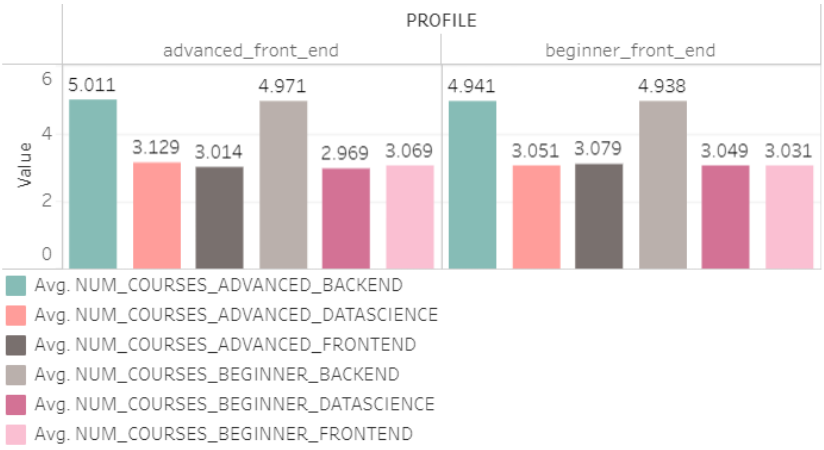
● Perbandingan Rata-Rata waktu belajar dari *Advance* dan *Beginner*



Gambar 15 Perbandingan Rata-Rata waktu belajar dari Advance dan Beginner pada profil frontend

Gambar 15 menunjukkan profil *frontend advance* menghabiskan waktu belajar rata-rata 50.24 jam, yang sebanding dengan profil *backend advance*. Sebaliknya, pembelajar *beginner* menghabiskan waktu yang jauh lebih sedikit (30.43 jam). Perbedaan ini bisa terjadi karena meningkatnya kompleksitas materi di domain *frontend* saat pembelajar beralih ke level *advance*, yang umumnya membutuhkan lebih banyak waktu untuk dipelajari dan dikuasai.

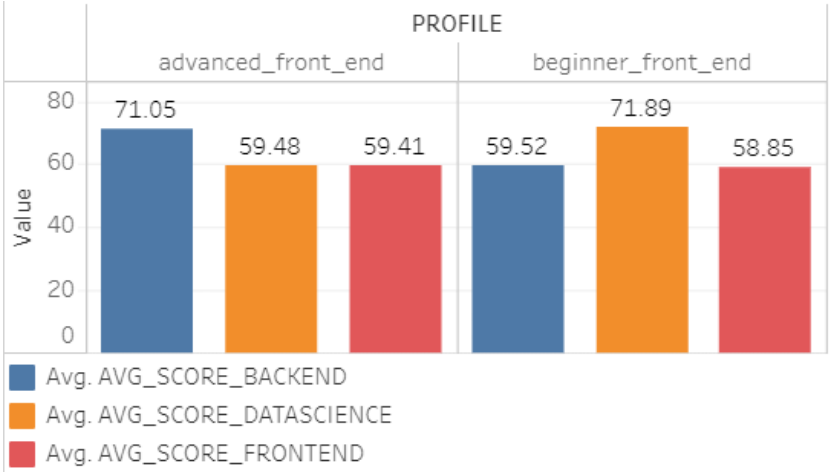
● Perbandingan Rata Jumlah Kursus dari *Advance* dan *Beginner*



Gambar 16 Perbandingan Rata Jumlah Kursus dari Advance dan Beginner pada profil frontend

Gambar 16 menunjukkan pembelajar *frontend advance* juga mengambil kursus sedikit lebih banyak (5.011) dibandingkan dengan pembelajar *frontend beginner* (4.938). Namun, perbedaannya lebih kecil dibandingkan dengan profil *backend*. Ini bisa menunjukkan bahwa pembelajar *frontend*, baik di tingkat *beginner* maupun *advance*, memiliki konsistensi yang lebih dalam mengambil kursus dibandingkan dengan pembelajar *backend*.

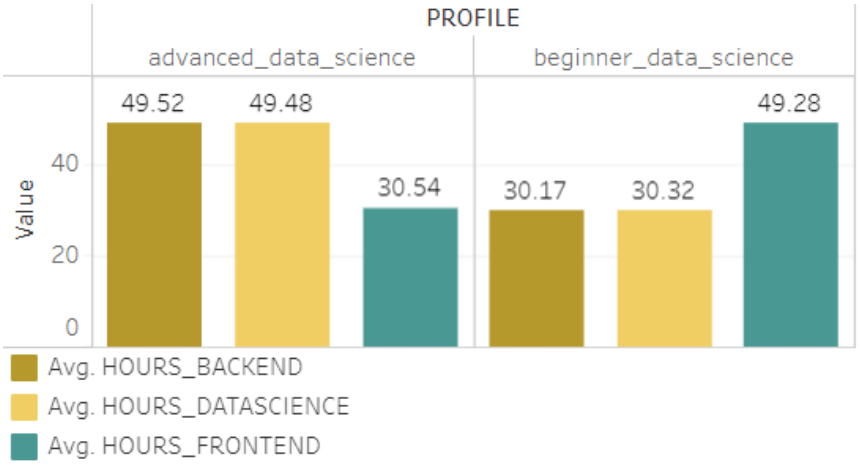
- Perbandingan Rata Nilai Dari *Advance* dan *Beginner*



Gambar 17 Perbandingan Rata Nilai Dari Advance dan Beginner pada profil frontend

Gambar 17 menunjukkan pembelajar *frontend advance* memiliki rata-rata nilai yang sedikit lebih rendah (71.05) dibandingkan dengan pembelajar *beginner* (71.89). Ini bisa mengindikasikan bahwa kursus *frontend advance* lebih menantang, yang mengakibatkan sedikit penurunan nilai meskipun tingkat keahlian pembelajar meningkat.

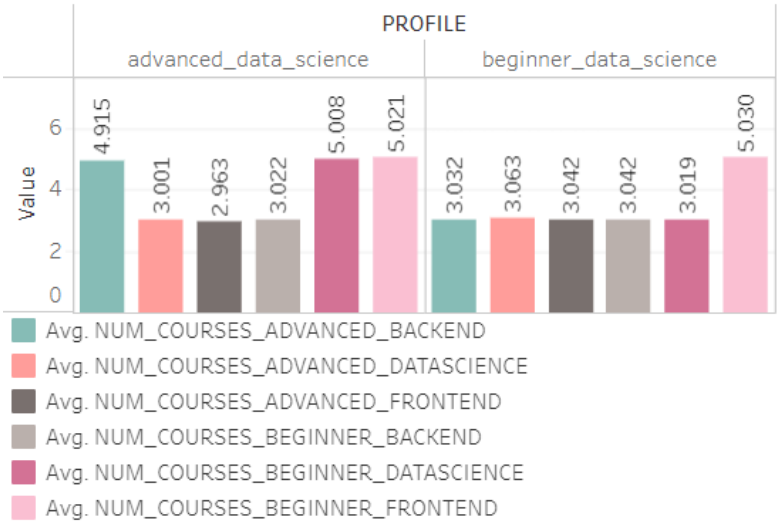
7. Profil *Data Science*
   - Perbandingan Rata-Rata waktu belajar dari *Advance* dan *Beginner*



Gambar 18 Perbandingan Rata-Rata waktu belajar dari Advance dan Beginner pada profil data science

Gambar 18 menunjukkan waktu belajar untuk *beginner data science* mengikuti pola yang sama, dengan profil *advance* menghabiskan 49.52 jam dan profil *beginner* menghabiskan sekitar 30.54 jam. Seperti *backend* dan *frontend*, *data science* menjadi lebih memakan waktu saat pembelajar beralih dari *beginner* ke *advance*, karena topik yang lebih rumit seperti *machine learning* dan analitik *advance* memerlukan waktu yang lebih lama untuk dikuasai.

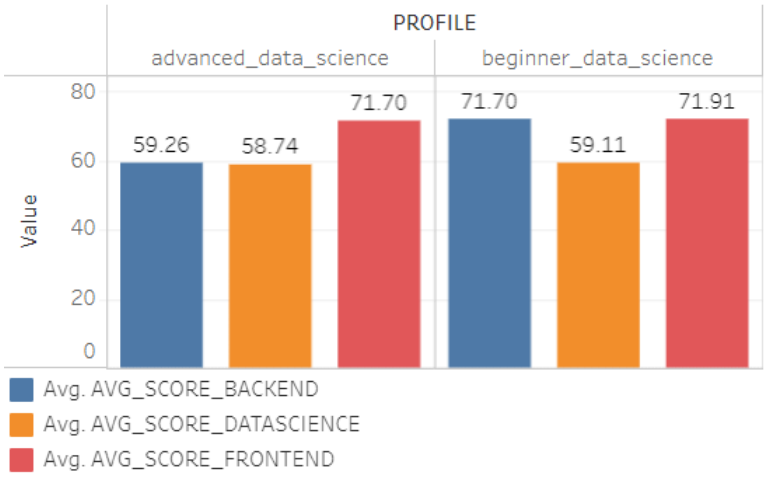
● Perbandingan Rata Jumlah Kursus dari *Advance* dan *Beginner*



Gambar 19 Perbandingan Rata Jumlah Kursus dari Advance dan Beginner pada profil data science

Gambar 19 menunjukkan pembelajar *data science* di level *advance* cenderung mengambil lebih banyak kursus (4.915) dibandingkan dengan pembelajar *beginner* (3.042). Seperti halnya pembelajar *backend*, pembelajar *data science* di level *advance* mungkin membutuhkan lebih banyak kursus untuk memperdalam pemahaman mereka terhadap materi yang lebih kompleks.

● Perbandingan Rata Nilai Dari *Advance* dan *Beginner*



Gambar 20 Perbandingan Rata Nilai Dari Advance dan Beginner pada profil data science

Gambar 20 menunjukkan pembelajar *data science* di level *beginner* cenderung memiliki nilai yang lebih tinggi (71.91) dibandingkan dengan pembelajar *advance* (59.26), serupa dengan tren di *frontend*. Penurunan nilai untuk pembelajar *data science advance* mungkin disebabkan oleh tingkat kesulitan yang lebih tinggi dalam topik *advance* seperti algoritma *machine learning* yang memerlukan pemahaman teknis yang lebih dalam.

Secara keseluruhan, visualisasi ini menunjukkan bahwa seiring dengan kemajuan pembelajar dari beginner ke *advance* di bidang *backend*, *frontend*, dan *data science*, mereka cenderung mengambil lebih banyak kursus dan menghabiskan lebih banyak waktu untuk belajar. Namun, nilai yang dicapai tidak selalu meningkat dengan pengalaman, terutama di bidang *frontend* dan *data science*, di mana pembelajar *advance* justru memiliki nilai sedikit lebih rendah dibandingkan *beginner*. Hal ini menunjukkan bahwa kursus *advance* lebih menantang, sehingga meskipun pembelajar mendapatkan lebih banyak pengetahuan dan keterampilan, kompleksitas materi yang diajarkan dapat mengakibatkan penilaian yang lebih sulit dan menghasilkan nilai rata-rata yang lebih rendah.

IV.2.4 Pemilihan dan Pelatihan Model

IV.2.4.1 Tahap *Pre-Modeling*

Pada tahap *pre-modeling*, metode *splitting* data merupakan langkah krusial yang harus dilakukan sebelum melatih model. Proses ini memastikan bahwa model yang dikembangkan dievaluasi dengan baik dan dapat diandalkan pada data yang belum terlihat sebelumnya. Proporsi data yang digunakan adalah 80% untuk pelatihan (*train*) dan 20% untuk pengujian (*test*). Dengan memanfaatkan fungsi *train_test_split* dari scikit-learn, tahapan splitting dataset menjadi lebih mudah. Setelah proses *splitting*, total data yang diperoleh adalah 15.228

baris untuk data pelatihan dan 3.807 baris untuk data pengujian. *Stratified splitting* juga diterapkan untuk memastikan keseimbangan label dengan mempertimbangkan beberapa bobot untuk setiap class yang diperhitungan oleh fungsi *train_test_split*.

IV.2.4.2 Model Selection

Pada tahap ini, pemilihan model melibatkan tiga model utama, yaitu QDA, XGBoost, dan KNN. Ketiga model ini dipilih setelah melalui berbagai *trial and error* serta evaluasi kinerja terhadap model-model lain, seperti *Random Forest* (RF), *Decision Tree* (DT), *Logistic Regression* (LR), *Support Vector Machine* (SVM), LightGBM, dan lainnya. Keputusan untuk menggunakan ketiga model ini didasarkan pada kinerja superior mereka dalam menyelesaikan masalah yang dihadapi dibandingkan dengan model-model lainnya. Berikut penjelasan singkat mengenai ketiga model tersebut.

Tabel 4 Penjelasan singkat mengenai model

Model	Penjelasan
QDA	Model klasifikasi yang mengasumsikan setiap kelas memiliki distribusi Gaussian dengan matriks kovarians berbeda, cocok untuk data dengan varians antar kelas yang berbeda dan distribusi mendekati normal
XGBoost	Algoritma *boosting* yang kuat, ideal untuk data dengan struktur kompleks dan banyak fitur, serta efektif dalam menangani *outlier* dan data dengan ketidakseimbangan kelas
KNN	Algoritma yang bekerja dengan mencari tetangga terdekat untuk klasifikasi, cocok untuk data dengan jumlah fitur moderat dan distribusi kelas yang jelas, tetapi kurang efisien untuk data yang sangat besar

Dibawah ini adalah parameter-parameter yang digunakan untuk masing-masing model:

Tabel 5 Parameter yang digunakan oleh model

Model	Paramater

QDA	{'priors': None, 'reg_param': 0.0, 'store_covariance': False, 'tol': 0.0001}
XGBoost	{'objective': 'multi:softprob', 'eval_metric': 'mlogloss', 'use_label_encoder': False, 'missing': nan}
KNN	{'algorithm': 'auto', 'leaf_size': 30, 'metric': 'minkowski', 'n_neighbors': 5, 'p': 2, 'weights': 'uniform'}

1) QDA: Parameter priors mengatur proporsi kelas (jika tidak ditentukan secara eksplisit), reg_param digunakan untuk regularisasi, store_covariance menentukan apakah matriks kovarians disimpan, dan tol mengatur toleransi konvergensi.
2) XGBoost: Parameter *objective* menentukan jenis tugas, eval_metric mengukur log loss untuk multi-kelas, use_label_encoder menghindari label encoder bawaan, dan missing mengatur penanganan nilai yang hilang.
3) KNN: Parameter *algorithm* mengatur metode pencarian, leaf_size mengontrol ukuran daun pada tree, metric menghitung jarak dengan p mengatur norm Minkowski, n_neighbors menetapkan jumlah tetangga, dan *weights* menunjukkan semua tetangga memiliki bobot yang sama.

IV.2.5 Evaluasi Kinerja dan Pembuktian Model

Model evaluasi dalam *machine learning* adalah proses menilai kinerja model yang telah dilatih dengan menggunakan metrik tertentu[32]. Tahap ini sangat krusial karena bertujuan untuk menentukan kinerja dan ketahanan model terhadap data baru yang belum pernah dilihat sebelumnya. Penelitian ini juga bertujuan untuk memahami mengapa salah satu model terbaik dapat unggul dalam deteksi, apakah ada kecocokan khusus antara model dan data yang digunakan, atau faktor lainnya. Evaluasi pada tahap ini memungkinkan model untuk menjadi lebih sempurna dengan menguji ketahanannya terhadap variasi data input yang mungkin mencakup aspek-aspek yang tidak ada dalam data pelatihan[32]. Untuk evaluasi ini, digunakan data *testing* yang sebelumnya telah dipisahkan sebanyak 20% dari seluruh data, yaitu 3.807 baris.

Karena data yang digunakan memang cocok untuk klasifikasi, bukan regresi, maka beberapa metrik evaluasi yang relevan telah dipilih, seperti *Accuracy*, *Precision*, *Recall*, *F1-score*, dan *Receiver Operating Characteristic - Area Under the Curve* (*ROC AUC) Score*.

Tabel 6 Rumus matematika untuk penentuan niali metrik

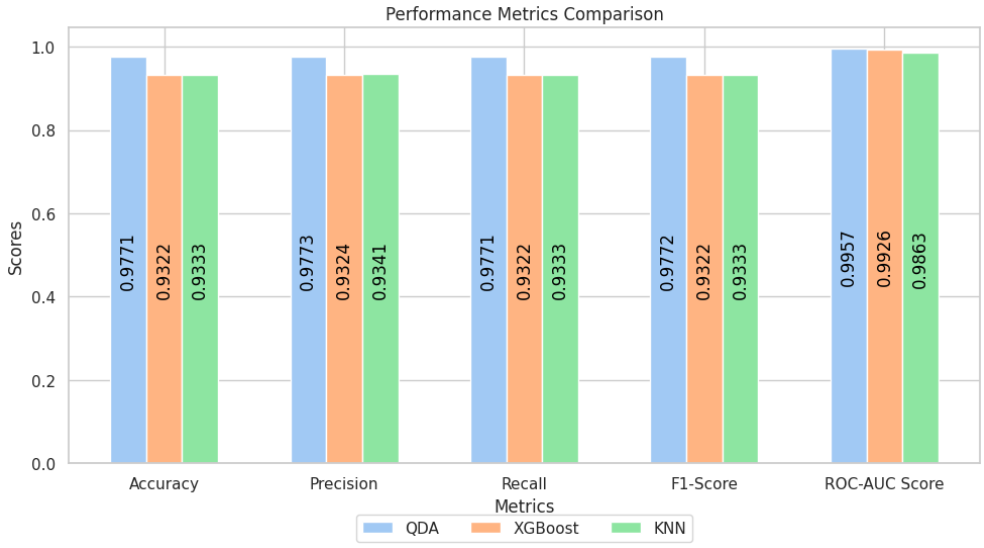
$$Accuracy = \left(\frac{TP + TN}{TP + FP + FN + TN}\right) \quad (2)$$

$$Precision = \left(\frac{TP}{TP + FP}\right) \quad (3)$$

$$Recall = \left(\frac{TN}{TP + TN}\right) \quad (4)$$

$$F1 - score = \left(2\,x\,\frac{Recall\ x\ Precision}{Recall + Precision}\right) \quad (5)$$

Masing-masing metrik ini memiliki fungsi yang berbeda-beda dalam menilai kinerja model. Berikut adalah hasil evaluasi kinerja dari ketiga model yang digunakan.

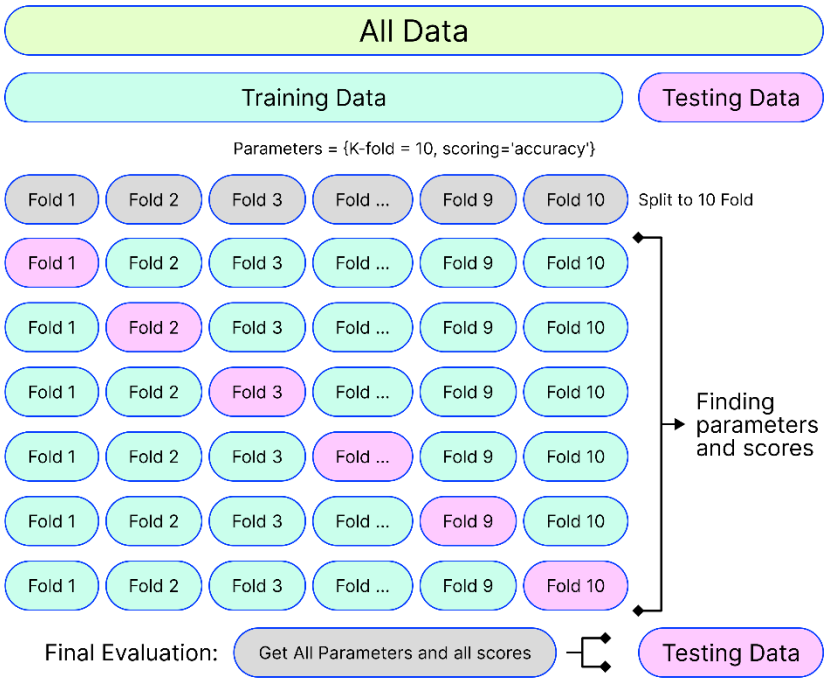


Gambar 21 Barplot untuk metrik evaluasi setiap model

Berdasarkan grafik yang dihasilkan pada Gambar 21, terlihat bahwa model QDA menunjukkan kinerja terbaik dibandingkan dengan XGBoost dan KNN dalam memprediksi profil pengguna berdasarkan aktivitas pembelajaran. QDA memiliki nilai tertinggi di semua metrik evaluasi: *Accuracy* (0.9771), *Precision* (0.9773), *Recall* (0.9771), *F1-Score* (0.9772), dan *ROC-AUC Score* (0.9957). Pada nilai tersebut, QDA menunjukkan keakuratan dan konsistensi tertinggi dalam memprediksi profil pengguna dan membedakan antara kelas-kelas profil.

Precision tinggi berarti prediksi profil oleh QDA lebih akurat. *Recall* yang tinggi menunjukkan kemampuan QDA untuk menangkap hampir semua profil pengguna yang benar. *F1-Score* QDA menunjukkan keseimbangan optimal antara akurasi dan deteksi profil yang benar. Selain itu, nilai *ROC-AUC* yang sangat baik menunjukkan kemampuan QDA dalam membedakan berbagai kelas profil pengguna. Dengan demikian, QDA adalah model yang paling andal untuk studi kasus ini, terutama jika keakuratan dan deteksi profil yang benar adalah prioritas utama. Selanjutnya adalah langkah untuk validasi

IV.2.5.1 Metode Cross Validation Dengan Rerata Skor



Gambar 22 Alur kerja K-fold cross-validation

Cross-validation dilakukan dengan membagi data yang tersedia menjadi set pelatihan dan uji, di mana model dilatih pada set pelatihan dan dievaluasi berdasarkan prediksi pada set uji[33]. Dengan mengulangi proses ini untuk berbagai pembagian data, dapat diestimasi kinerja prediktif rata-rata dari satu atau lebih model. Kinerja prediktif yang divalidasi silang sering dimanfaatkan untuk memperkirakan atau menyetel *hyperparameter* dari sebuah model, atau untuk membandingkan beberapa model diskrit dengan tujuan memilih yang terbaik[34]. Di bawah ini adalah hasil metode *cross-validation* yang digunakan untuk ketiga model yang diuji.

Tabel 7 Hasil cross-validation

Fold	Accuracy_KNN	Accuracy_QDA	Accuracy_XGBoost
1	0.928571	0.971639	0.934349
2	0.939601	0.978992	0.933298
3	0.922269	0.980042	0.938025
4	0.918067	0.972164	0.926471
5	0.935924	0.981618	0.944853
6	0.936416	0.978455	0.936416
7	0.924855	0.976353	0.931161
8	0.925906	0.973726	0.937993
9	0.939569	0.981083	0.93484
10	0.928534	0.97793	0.93011
Mean ± Std	0.9300 ± 0.0071	0.9772 ± 0.0034	0.9348 ± 0.0048

Dari hasil *cross-validation,* QDA adalah model dengan performa terbaik, dengan akurasi rata-rata tertinggi (97.72%) dan konsistensi yang tinggi (standar deviasi 0.0034). Hal ini menandakan bahwa model ini tidak hanya akurat tetapi juga sangat stabil di seluruh fold. XGBoost juga menunjukkan performa yang baik, sedikit lebih baik dari KNN, namun masih di bawah QDA. KNN memiliki performa yang cukup baik tetapi berada di urutan ketiga dari segi akurasi rata-rata. Oleh karena itu, QDA adalah model yang paling direkomendasikan berdasarkan hasil *cross-validation* ini.

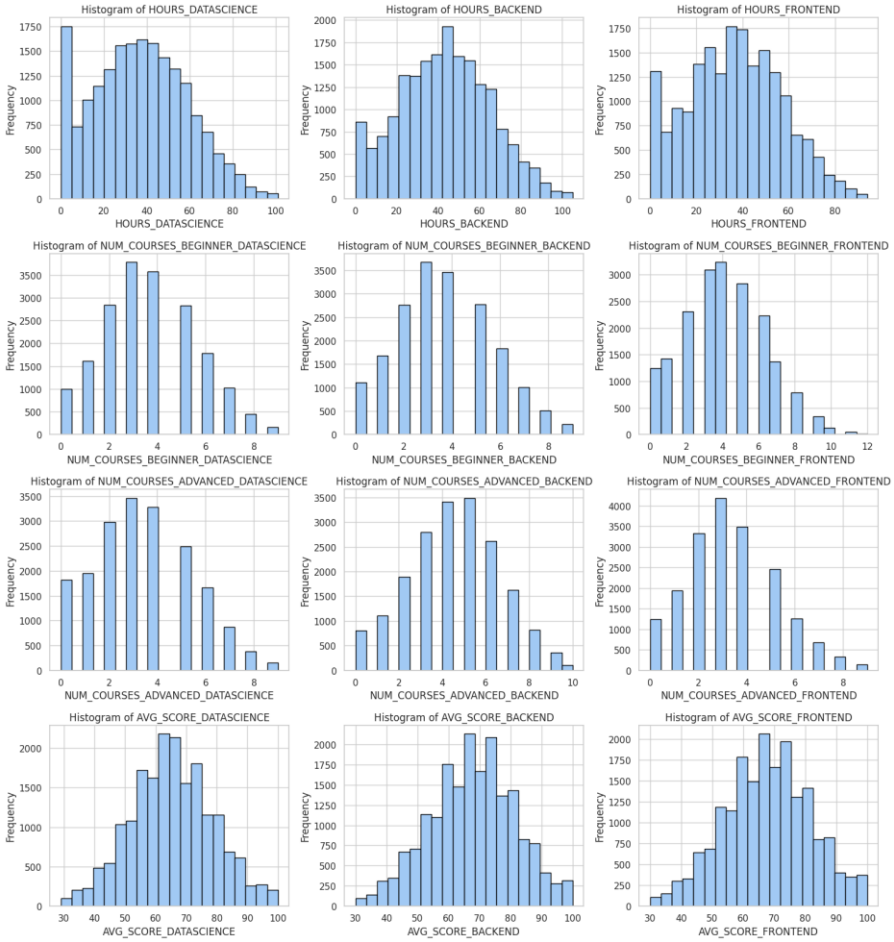
IV.2.5.2 Kesesuaian Model QDA untuk Dataset

Dalam menerapkan model *Quadratic Discriminant Analysis* (QDA) pada dataset ini, dilakukan beberapa pengujian untuk memastikan bahwa asumsi-asumsi dasar yang mendukung efektifitas model QDA telah terpenuhi. Asumsi tersebut antara lain adalah sebagai berikut:

1. Setiap fitur (variabel) dalam dataset Anda mengikuti distribusi normal untuk setiap kelas. Disini menggunakan uji normalitas dengan Histogram dan melihat bagaimana bentuknya pada setiap feature yang digunakan pada modeling. Jika bentuk data pada plot histogram mengarah pada normal atau berbentuk lonceng maka asumsi ini terpenuhi.
2. Setiap kelas memiliki matriks kovarians yang berbeda. Maka akan dilakukan penghitungan matriks kovarians untuk masing-

masing kelas dan membandingkan. Jika matriks kovarians berbeda secara signifikan, asumsi ini terpenuhi.

Pembuktian asumsi pertama dilakukan dengan hasil seperti Gambar 23.



Gambar 23 Hasil pemeriksaan normalitas dengan menggunakan histogram

Pada asumsi pertama terkait normalitas fitur, terlihat bahwa fitur `HOURS_` dan `AVG_SCORE_` dalam dataset menunjukkan distribusi yang mendekati normal, yang sesuai dengan asumsi normalitas yang diperlukan untuk penerapan model *Quadratic Discriminant Analysis* (QDA). Namun, beberapa fitur, seperti `NUM_COURSES_`, terutama pada tingkat pemula, menunjukkan beberapa penyimpangan dari normalitas, seperti adanya skewness atau distribusi bimodal. Meskipun demikian, penyimpangan ini tidak secara *skewness* signifikan menghambat kinerja model QDA. Berdasarkan hasil uji, model QDA tetap menunjukkan performa yang baik, sehingga dapat disimpulkan bahwa, secara keseluruhan, data ini cukup memenuhi asumsi normalitas yang diperlukan untuk penggunaan model QDA.

Pembuktian asumsi kedua dilakukan melalui perhitungan determinan dari matriks kovarians dan diperkuat dengan uji Box's M menggunakan library Pingouin.

Tabel 8 Hasil Determinan dari Kovarians seitap class

Class/Label	Determinan Matriks Kovarians
advanced_backend	116112196910806.84
advanced_data_science	104141135216291.61
advanced_front_end	21081722041064.703
beginner_data_science	171475279077700.12
beginner_front_end	75519506539239.06

Tabel 9 Hasil Uji Box's M untuk Memeriksa Kesamaan Kovarians Antar Kelas

Chi2	df	pval	equal_cov
inf	390.0	0.0	FALSE

Seperti yang ditampilkan dalam Table 8. Hasil Determinan dari Kovarians seitap class, yang menunjukkan variasi signifikan dalam nilai determinan antar kelas. Misalnya, determinan untuk kelas `advanced_backend` adalah 116112196910806.84, sedangkan untuk kelas `advanced_front_end` hanya 21081722041064.703, menandakan bahwa kovarians antar kelas tidak seragam. Hasil ini diperkuat oleh uji Box's M yang disajikan dalam Table 9. Hasil Uji Box's M untuk Memeriksa Kesamaan Kovarians Antar Kelas, yang menunjukkan nilai Chi-square (Chi2) yang tak terhingga (inf), derajat kebebasan (df) sebesar 390, dan nilai p (pval) sebesar 0.0, dengan kolom `equal_cov` menunjukkan `False`. Hasil ini menunjukkan bahwa kovarians antar kelas berbeda secara signifikan, sehingga asumsi bahwa setiap kelas memiliki matriks kovarians yang berbeda, yang merupakan syarat penting untuk penerapan model *Quadratic Discriminant Analysis* (QDA), telah terpenuhi dengan baik. Jadi kedua asumsi telah terpenuhi dengan cukup baik, maka tidak heran QDA adalah model terbaik dan cocok untuk dataset.

BAB V

KESIMPULAN

V.1 Ringkasan Temuan

Penelitian ini telah berhasil mengembangkan model personalisasi profil pengguna berdasarkan aktivitas pembelajaran *online* menggunakan tiga model utama: ***Quadratic Discriminant Analysis* (QDA)**, ***Extreme Gradient Boosting* (XGBoost)**, dan ***K-Nearest Neighbors* (KNN)**. Dari hasil evaluasi, model QDA terbukti menjadi yang paling efektif dengan akurasi dan konsistensi tertinggi dibandingkan dua model lainnya. QDA mampu menangkap variasi yang ada dalam data pengguna dan menghasilkan prediksi yang akurat untuk setiap profil pengguna.

Temuan utama lainnya mencakup:

- **Distribusi profil pengguna** yang relatif merata antara level *beginner* dan *advanced*, menunjukkan bahwa metode pembelajaran yang digunakan efektif untuk berbagai tingkat keahlian.
- **Perbedaan dalam waktu belajar dan jumlah kursus** yang diambil antara pengguna *beginner* dan *advanced*, di mana pengguna dengan profil *advanced* cenderung menghabiskan lebih banyak waktu dan mengikuti lebih banyak kursus.
- **Kinerja pengguna** yang tidak selalu lebih tinggi pada level *advanced*, khususnya dalam bidang seperti *data science* dan frontend, mengindikasikan bahwa materi yang lebih kompleks dapat mengakibatkan tantangan yang lebih besar dalam mencapai nilai tinggi.

V.2 Implikasi dan Rekomendasi dari Hasil

Hasil penelitian ini memiliki beberapa implikasi penting bagi pengembangan platform pembelajaran *online*:

1. **Peningkatan Personalisasi**
   Dengan menggunakan model QDA, platform pembelajaran dapat meningkatkan personalisasi konten yang diberikan kepada pengguna, sehingga lebih sesuai dengan kebutuhan dan kemampuan individu.
2. **Pengembangan Kurikulum yang Lebih Adaptif**
   Temuan bahwa pengguna dengan profil *advanced* sering kali menghadapi tantangan lebih besar dalam mencapai nilai tinggi menunjukkan perlunya kurikulum yang lebih adaptif dan dukungan yang lebih kuat bagi pengguna di level ini.
3. **Optimalisasi Pengalaman Pengguna**
   Menyediakan rekomendasi yang lebih tepat waktu dan relevan berdasarkan prediksi profil pengguna dapat meningkatkan keterlibatan dan motivasi pengguna dalam proses belajar.

Rekomendasi:

- Implementasikan model QDA sebagai algoritma utama untuk personalisasi konten di platform pembelajaran *online*.
- Lakukan penyesuaian pada materi kursus yang lebih menantang untuk pengguna di level *advanced*, dengan menyediakan sumber daya tambahan atau metode pengajaran alternatif.
- Terus pantau dan analisis data pengguna untuk memperbarui dan meningkatkan model prediksi secara berkala.

V.3 Keterbatasan dan Arah Penelitian Selanjutnya

Penelitian ini memiliki beberapa keterbatasan yang perlu diperhatikan:

- **Keterbatasan Data**

  Dataset yang digunakan mungkin tidak mencakup semua variabel yang mempengaruhi kinerja pembelajaran pengguna. Selain itu, data yang digunakan terbatas pada *platform* tertentu, sehingga hasilnya mungkin tidak sepenuhnya *generalizable*.

- **Asumsi Model**

  Model QDA mengasumsikan distribusi normal dari data, yang mungkin tidak sepenuhnya tercermin dalam dataset aktual. Meskipun demikian, performa model tetap cukup baik dalam konteks penelitian ini.

Untuk penelitian selanjutnya, beberapa langkah dapat diambil:

- **Pengujian Model di *Platform* Lain**

  Menerapkan dan menguji model di *platform* pembelajaran *online* lainnya untuk melihat apakah hasil yang serupa dapat diperoleh.

- **Penggunaan Dataset yang Lebih Luas dan Beragam**

  Melibatkan dataset dari berbagai sumber dan jenis aktivitas pembelajaran yang lebih beragam untuk meningkatkan generalisasi model.

- **Eksplorasi Model Lain**

  Menguji model *machine learning* lainnya yang mungkin lebih cocok untuk jenis data yang berbeda atau untuk menyelesaikan masalah lain dalam konteks pembelajaran *online*.

DAFTAR PUSTAKA

[1] J. Xiao, M. Wang, B. Jiang, dan J. Li, "A personalized recommendation system with combinational algorithm for online learning," J Ambient Intell Humaniz Comput, vol. 9, no. 3, hlm. 667–677, 2018, doi: 10.1007/s12652-017-0466-8.

[2] F. St-Hilaire dkk., "A Comparative Study of Learning Outcomes for Online Learning Platforms," dalam Artificial Intelligence in Education, I. Roll, D. McNamara, S. Sosnovsky, R. Luckin, dan V. Dimitrova, Ed., Cham: Springer International Publishing, 2021, hlm. 331–337.

[3] E. Prihar, A. Haim, A. Sales, dan N. Heffernan, "Automatic Interpretable Personalized Learning," dalam Proceedings of the Ninth ACM Conference on Learning @ Scale, dalam L@S '22. New York, NY, USA: Association for Computing Machinery, 2022, hlm. 1–11. doi: 10.1145/3491140.3528267.

[4] C. Tekin, J. Braun, dan M. van der Schaar, "eTutor: Online learning for personalized education," dalam 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2015, hlm. 5545–5549. doi: 10.1109/ICASSP.2015.7179032.

[5] Komleva N;Vilyavin D, "Digital platform for creating personalized adaptive online courses," Open Education, vol. 24, no. 2, hlm. 65–72, Apr 2020.

[6] M. Mudrák, M. Turčáni, dan J. Reichel, "Impact of Using Personalized E-Course in Computer Science Education," Journal on Efficiency and Responsibility in Education and Science, vol. 13, no. 4, hlm. 174–188, Des 2020, doi: 10.7160/eriesj.2020.130402.

[7] Y. Guo, Z. Nie, dan M. Wang, "Educational Digitalization Enables Learners to Achieve Personalized Learning Path: Driven by Four Dimensions," dalam 2023 5th International Conference on Computer Science and Technologies in Education (CSTE), 2023, hlm. 146–151. doi: 10.1109/CSTE59648.2023.00032.

[8] M. J. Lee dan B. Ferwerda, "Personalizing Online Educational Tools," dalam Proceedings of the 2017 ACM Workshop on Theory-Informed User Modeling for Tailoring and Personalizing Interfaces, dalam HUMANIZE '17. New York, NY, USA: Association for Computing Machinery, 2017, hlm. 27–30. doi: 10.1145/3039677.3039680.

[9] G. Mariscal, O. Marbán, dan C. Fernández, "A survey of data mining and knowledge discovery process models and methodologies," Knowledge Eng. Review, vol. 25, hlm. 137–166, Jun 2010, doi: 10.1017/S0269888910000032.

[10] Kelleher D John;Tierney Brendan, Data Science. The MIT Press, 2018.

[11] Witten H Ian;Frank Eibe;Hall A Mark;Pal J Christopher, Data Mining: Practical Machine Learning Tools and Techniques, 4th Edition. Elsevier Inc., 2016.

[12] Géron Aurélien, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition. O'Reilly Media, Inc., 2019.

[13] Raschka Sebastian dan Mirjalili Vahid, Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2, 3rd Edition. packt, 2019.

[14] Cairo Alberto, The Truthful Art: Data, Charts, and Maps for Communication. New Riders, 2016.

[15] Peter A. Rogerson, Spatial Statistical Methods for Geography. SAGE Publications Ltd, 2021.

[16] S. Arlot dan A. Celisse, "A survey of cross-validation procedures for model selection," Stat Surv, vol. 4, no. none, hlm. 40–79, Jan 2010, doi: 10.1214/09-SS054.

[17] Z. Xiaojin dan G. Andrew B, Introduction to Semi-Supervised Learning. Springer Cham, 2022.

[18] J. VanderPlas, Python data science handbook: Essential tools for working with data. O'Reilly Media, Inc., 2016.

[19] J. D. Kelleher, B. Mac Namee, dan A. D'Arcy, Fundamentals of Machine Learning for Predictive Data Analytics. The MIT Press, 2015.

[20] M. Kuhn dan K. Johnson, Feature Engineering and Selection: A Practical Approach for Predictive Models. Chapman and Hall/CRC, 2019.

[21] I. Goodfellow, Y. Bengio, dan A. Courvilleng, Deep Learning. The MIT Press, 2016.

[22] R. K. Yin, Case Study Research and Applications Design and Methods, Sixth Edition. COSMOS Corporation, 2018.

[23] M. Ofner, K. Straub, B. Otto, dan H. Oesterle, "Management of the Master Data Lifecycle: A Framework for Analysis," Journal of Enterprise Information Management, vol. 26, hlm. 472–491, Mei 2013, doi: 10.1108/JEIM-05-2013-0026.

[24] A. Prasetya, A. Priyatno, dan N. Nurhaeni, "Penanganan Imputasi Missing Values pada Data Time Series dengan Menggunakan Metode Data Mining," vol. 5, hlm. 56–62, Jun 2023, doi: 10.37034/jidt.v5i1.324.

[25] J. B. Angela, Islamiyah, dan Akhmad Irsyad, "Implementasi Visualisasi Data Berbasis Web Pada Exploratory Data Analysis Profil Kesehatan Kota Samarinda," Kreatif Teknologi dan Sistem Informasi (KRETISI), vol. 1, no. 1, hlm. 9–16, Jul 2023, doi: 10.30872/kretisi.v1i1.447.

[26] Abdurrachman dan F. F. Rachman, "Ketimpangan Indeks Pembangunan Manusia dan Komponennya antar Kabupaten/Kota di Provinsi Kalimantan Selatan 2010-2020," BESTARI: Buletin Statisitikan dan Aplikasi Terkini, vol. 1, no. Vol. 1 No. 01 (2021): Bestari, hlm. 37–45, Jun 2021.

[27] A. Ambarwari, Q. J. Adrian, dan Y. Herdiyeni, "Analisis Pengaruh Data Scaling Terhadap Performa Algoritme Machine Learning untuk Identifikasi Tanaman," Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi), vol. 4, hlm. 117–122, Feb 2020, doi: 10.29207/resti.v4i1.1517.

[28] W. Li dan L. Zhenyu, "A method of SVM with Normalization in Intrusion Detection," Procedia Environ Sci, vol. 11, hlm. 256–262, Agu 2011, doi: 10.1016/j.proenv.2011.12.040.

[29] W. Arifin, I. Ariawan, A. Rosalia, L. Lukman, dan N. Tufailah, "Data scaling performance on various machine learning algorithms to identify abalone sex," Jurnal Teknologi dan Sistem Komputer, vol. 10, hlm. 26–31, Jan 2022, doi: 10.14710/jtsiskom.2021.14105.

[30] F. V. P. Samosir, L. P. Mustamu, E. D. Anggara, A. I. Wiyogo, dan A. Widjaja, "Exploratory Data Analysis terhadap Kepadatan Penumpang Kereta Rel Listrik," Jurnal Teknik Informatika dan Sistem Informasi, vol. 7, no. 2, 2021, doi: 10.28932/jutisi.v7i2.3700.

[31] S. P. Sari dan R. A. Putri, "Analisis Dan visualisasi data penjualan menggunakan Exploratory Data Analysis dan K-Means clustering," Jurnal Sistem Komputer dan Informatika (JSON), vol. 5, no. 2, hlm. 423, 2023, doi: 10.30865/json.v5i2.7180.

[32] G. Varoquaux dan O. Colliot, "Evaluating Machine Learning Models and Their Diagnostic Value," dalam Machine Learning for Brain Disorders, O. Colliot, Ed., New York, NY: Springer US, 2023, hlm. 601–630. doi: 10.1007/978-1-0716-3195-9_20.

[33] L. A. Yates, Z. Aandahl, S. A. Richards, dan B. W. Brook, “Cross validation for model selection: A review with examples from ecology,” Ecol Monogr, vol. 93, no. 1, hlm. e1557, 2023, doi: https://doi.org/10.1002/ecm.1557.

[34] S. Arlot, “V-fold cross-validation improved: V-fold penalization,” 2008. [Daring]. Tersedia pada: https://arxiv.org/abs/0802.0566