

SISTEM REKOMENDASI PEMILIHAN PROGRAM STUDI BERBASIS *HYBRID* MENGGUNAKAN PENDEKATAN *DEEP LEARNING*

Alim Misbullah¹, *Mufid Akbar¹, Nazaruddin¹, Laina Farsiah¹, Husaini¹, Zulfan¹

¹Departemen Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Syiah Kuala, Kopelma Darussalam, Banda Aceh, 23111, Indonesia
E-mail: misbullah@usk.ac.id, mufid.a@mhs.usk.ac.id, anzaro@usk.ac.id, lainafarsiah@usk.ac.id, husaini.muhammad@usk.ac.id, zulfan@usk.ac.id

Abstract

Education plays a critical role in shaping career decisions for the future. However, many students encounter difficulties in selecting suitable academic programs, often stemming from a lack of confidence in their ability to make appropriate decisions. Consequently, students may choose study programs that do not align with their personal characteristics. This study emphasizes the importance of providing comprehensive information about various academic programs offered in higher education and developing tools to assist prospective students in making informed decisions. To address these challenges, a recommendation system using Hybrid Filtering technology has been developed. The system integrates Content-Based Filtering and Collaborative Filtering methods within the TensorFlow Recommenders System (TFRS) framework. The study utilized data from undergraduate students of the Faculty of Mathematics and Natural Sciences (FMIPA) across seven academic programs. By employing 10 features representing students' interests and talents, the recommendation system generated accurate and tailored suggestions for study programs. The model was trained and evaluated using both real and augmented (augmented) datasets with predefined hyperparameters. Results demonstrated that using only the real dataset achieved a Top-1 accuracy of 0.59 and a Top-5 accuracy of 0.97. When incorporating the augmented dataset, the Top-1 accuracy improved to 0.66, while the Top-5 accuracy reached 1.0. The findings reveal that combining real and augmented datasets enhances average accuracy by approximately 10% compared to using the real dataset alone. Additionally, the study program recommendations produced by the model showed significant improvement in quality. A web-based recommendation system utilizing the TFRS model was developed and positively evaluated by FMIPA students. User feedback indicated high satisfaction with the system's recommendations, demonstrating its effectiveness in guiding students toward suitable academic programs.

Keywords: *recommendation system, study program selection, hybrid filtering, tensorflow recommenders, decision support system*

Abstrak

Pendidikan memainkan peran penting dalam menentukan keputusan karier di masa depan. Namun, banyak mahasiswa mengalami kesulitan dalam memilih program studi yang sesuai, yang sering kali disebabkan oleh kurangnya kepercayaan diri dalam membuat keputusan yang tepat. Akibatnya, mahasiswa dapat memilih

SISTEM REKOMENDASI PEMILIHAN PROGRAM STUDI BERBASIS *HYBRID* MENGGUNAKAN PENDEKATAN *DEEP LEARNING*

program studi yang tidak selaras dengan karakteristik pribadi mereka. Penelitian ini menyoroti pentingnya menyediakan informasi yang komprehensif tentang berbagai program studi yang ditawarkan di perguruan tinggi serta pengembangan alat bantu untuk membantu calon mahasiswa dalam membuat keputusan yang lebih terinformasi. Untuk mengatasi tantangan ini, dikembangkan sistem rekomendasi berbasis teknologi Hybrid Filtering. Sistem ini mengintegrasikan metode Content-Based Filtering dan Collaborative Filtering dalam kerangka kerja TensorFlow Recommenders System (TFRS). Penelitian ini menggunakan data dari mahasiswa Sarjana Fakultas Matematika dan Ilmu Pengetahuan Alam (FMIPA) yang terdiri dari tujuh program studi. Dengan memanfaatkan 10 fitur yang mencerminkan minat dan bakat siswa, sistem rekomendasi dapat menghasilkan saran program studi yang akurat dan sesuai. Model dilatih dan dievaluasi menggunakan dataset nyata (real dataset) serta dataset tambahan (augmented dataset) dengan hyperparameter yang telah ditentukan. Hasil menunjukkan bahwa penggunaan dataset nyata saja menghasilkan akurasi Top-1 sebesar 0,59 dan akurasi Top-5 sebesar 0,97. Sementara itu, dengan menambahkan augmented dataset, akurasi Top-1 meningkat menjadi 0,66 dan akurasi Top-5 mencapai 1,0. Temuan ini menunjukkan bahwa penggabungan dataset nyata dengan dataset tambahan meningkatkan rata-rata akurasi sekitar 10% dibandingkan hanya menggunakan dataset nyata. Selain itu, rekomendasi program studi yang dihasilkan oleh model juga menunjukkan peningkatan kualitas yang signifikan. Sistem rekomendasi berbasis web yang dikembangkan menggunakan model TFRS telah diuji efektivitasnya dan mendapatkan respons positif dari pengguna, yaitu mahasiswa FMIPA. Umpan balik pengguna menunjukkan tingkat kepuasan yang tinggi terhadap rekomendasi program studi yang diberikan oleh sistem.

Kata Kunci: *sistem rekomendasi, pemilihan program studi, hybrid filtering, tensorflow recommenders, sistem pendukung keputusan*

1. Pendahuluan

Setiap individu berhak mendapatkan pendidikan, baik melalui pendidikan formal di sekolah maupun melalui pendidikan nonformal di luar sekolah. Pendidikan memberikan bekal bagi peserta didik atau mahasiswa dalam mencari pekerjaan, sesuai dengan program studi yang mereka ambil di sekolah atau universitas [1]. Pendidikan juga menjadi salah satu pendorong untuk melakukan pemilihan karier pada masa depan. Pemilihan karier yang tepat membutuhkan perencanaan yang baik. Hal ini penting karena karier akan berdampak pada kehidupan di masa mendatang. Siswa perlu mengevaluasi kemampuan mereka sebelum memutuskan karier yang akan diambil. Fakta di lapangan menunjukkan bahwa banyak mahasiswa yang membuat kesalahan dalam memilih program studi. Siswa juga menghadapi tantangan dalam mengambil keputusan saat ingin melanjutkan pendidikan ke tingkat yang lebih tinggi [2].

Permasalahan dalam pengambilan keputusan program studi merupakan hasil dari ketidakpercayaan terhadap kemampuan diri dalam menentukan bidang atau program studi yang tepat. Akibatnya, beberapa orang memilih program studi yang tidak sesuai dengan karakteristik pribadi mereka [3]. Pemilihan program studi berdasarkan bakat, minat, dan kepribadian merupakan bagian dari pelayanan pemilihan dan perencanaan individual yang bertujuan untuk membantu peserta didik dalam memilih program studi yang sesuai dengan bakat, minat, dan kepribadian mereka secara tepat [4].

Berdasarkan informasi dan data yang dikumpulkan, hasil perhitungan dapat menjadi rekomendasi bagi siswa dalam memilih jenjang pendidikan di perguruan tinggi. Namun, masih terdapat keterbatasan fasilitas teknologi yang dapat membantu para lulusan sekolah

dalam memilih program studi yang sesuai dengan minat dan bakat yang mereka inginkan [5]. Untuk mencapai keputusan yang sesuai, diperlukan pendukung keputusan. Proses pengolahan data ini didasarkan pada kriteria atau atribut pendukung, sehingga menghasilkan hasil terbaik dari sejumlah data alternatif. Namun, jika proses pengambilan keputusan tidak melibatkan sistem atau metode yang sesuai, maka kemungkinan hasil keputusan tidak relevan dengan kebutuhan yang ada [5]. Penelitian ini menggunakan teknologi *data mining* yaitu sistem rekomendasi berbasis *hybrid*, untuk membantu memberikan rekomendasi pengambilan keputusan dalam memilih program studi.

Sistem Rekomendasi adalah sebuah sistem yang digunakan untuk menyaring informasi yang berlebihan [6]. Sistem Rekomendasi merupakan jenis sistem penyaringan informasi yang bertujuan untuk memprediksi peringkat atau preferensi yang akan diberikan oleh pengguna terhadap suatu item atau elemen sosial yang sebelumnya belum mereka pertimbangkan. Hal ini dilakukan dengan membangun model dari karakteristik item (*Content-Based Filtering*) dan lingkungan sosial pengguna (*Collaborative Filtering*) [7]. Sistem ini membantu menyaring informasi penting yang diperoleh dari jumlah informasi yang terus berkembang, sesuai dengan preferensi, minat, atau perilaku pengguna terhadap item [8]. Sistem Rekomendasi memberikan manfaat bagi penyedia layanan dan pengguna [9]. Sistem ini mengatasi masalah informasi yang berlebihan yang sering dihadapi pengguna dengan menyediakan rekomendasi konten dan layanan yang disesuaikan dan eksklusif [10].

Oleh karena itu, perlu dipahami dan dikembangkan sebuah fasilitas yang dapat membantu lulusan sekolah dalam memilih program studi yang sesuai dengan menggunakan teknologi data mining yaitu Sistem Rekomendasi berbasis *Hybrid Filtering*. *Hybrid Filtering* adalah penggabungan lebih dari satu metode filtering yang memanfaatkan kelebihan di tiap-tiap filtering. Pada penelitian ini peneliti menggabungkan antara metode *Content-Based Filtering* dan *Collaborative Filtering*. *Content based* memanfaatkan karakter user untuk dapat dicari kesamaan dan *Collaborative Filtering* memanfaatkan Matrix untuk memudahkan dalam pencarian kesamaan antara user.

2. Kajian Pustaka

Sistem pembelajaran mesin, khususnya Supervised Learning, merupakan bentuk machine learning yang sangat umum digunakan, baik dalam *deep learning* maupun aplikasi lainnya. Supervised Learning bekerja dengan menggunakan pasangan data input dan output yang diketahui untuk membangun model yang dapat memprediksi keluaran dari data baru. Dalam konteks sistem rekomendasi, metode ini digunakan untuk mempelajari pola dari data yang telah dilabeli sehingga dapat memberikan rekomendasi yang akurat kepada pengguna berdasarkan data historis [11, 12]. Model Supervised Learning juga sangat efektif dalam meminimalkan perbedaan antara data prediksi dan data sebenarnya melalui optimasi, seperti dalam pengurangan cross-entropy [13].

Sistem rekomendasi (*Recommendation System*) telah menjadi alat penting dalam berbagai aplikasi modern, seperti *e-commerce*, *streaming media*, dan pendidikan. Sistem ini bertujuan untuk memberikan saran terkait item tertentu yang relevan dengan kebutuhan pengguna, seperti produk, artikel, atau program studi. Terdapat tiga entitas utama dalam sistem rekomendasi, yaitu item, pengguna, dan interaksi antara keduanya. Berdasarkan pendekatannya, sistem rekomendasi dapat dikategorikan menjadi tiga jenis: *Content-Based Filtering*, *Collaborative Filtering*, dan *Hybrid Filtering*. *Content-Based Filtering* bekerja dengan menganalisis atribut item yang disukai pengguna sebelumnya,

SISTEM REKOMENDASI PEMILIHAN PROGRAM STUDI BERBASIS *HYBRID* MENGGUNAKAN PENDEKATAN *DEEP LEARNING*

sedangkan *Collaborative Filtering* menganalisis interaksi historis pengguna untuk menemukan pola rekomendasi. *Hybrid Filtering* menggabungkan kedua pendekatan tersebut untuk menghasilkan rekomendasi yang lebih akurat [14, 15].

Pendekatan *Content-Based Filtering* menggunakan deskripsi atribut item untuk membangun profil pengguna berdasarkan item yang telah dinilai atau dibeli. Pendekatan ini memiliki keunggulan, seperti kemampuan merekomendasikan item baru dan kemampuan adaptasi terhadap perubahan preferensi pengguna. Namun, sistem ini sering kali menghadapi keterbatasan dalam hal kurangnya konteks sosial dan data pengguna lainnya [16]. Sementara itu, *Collaborative Filtering* memanfaatkan data interaksi banyak pengguna untuk memberikan rekomendasi. Metode ini sangat populer karena kemampuannya dalam memanfaatkan preferensi kolektif, tetapi sering kali menghadapi masalah *sparsity* data dan *cold start*, terutama ketika terdapat sedikit data untuk pengguna baru [17].

Untuk mengatasi kekurangan masing-masing pendekatan, *Hybrid Filtering* dikembangkan dengan menggabungkan kelebihan *Content-Based Filtering* dan *Collaborative Filtering*. Sistem ini mampu meningkatkan akurasi rekomendasi dengan mengatasi masalah *sparsity* data dan *cold start*. Dalam penelitian terbaru, pendekatan hybrid terbukti lebih efektif dibandingkan metode tunggal, terutama dalam menangkap hubungan kompleks antara pengguna dan item yang didasarkan pada fitur intrinsik maupun pola perilaku pengguna [18, 19].

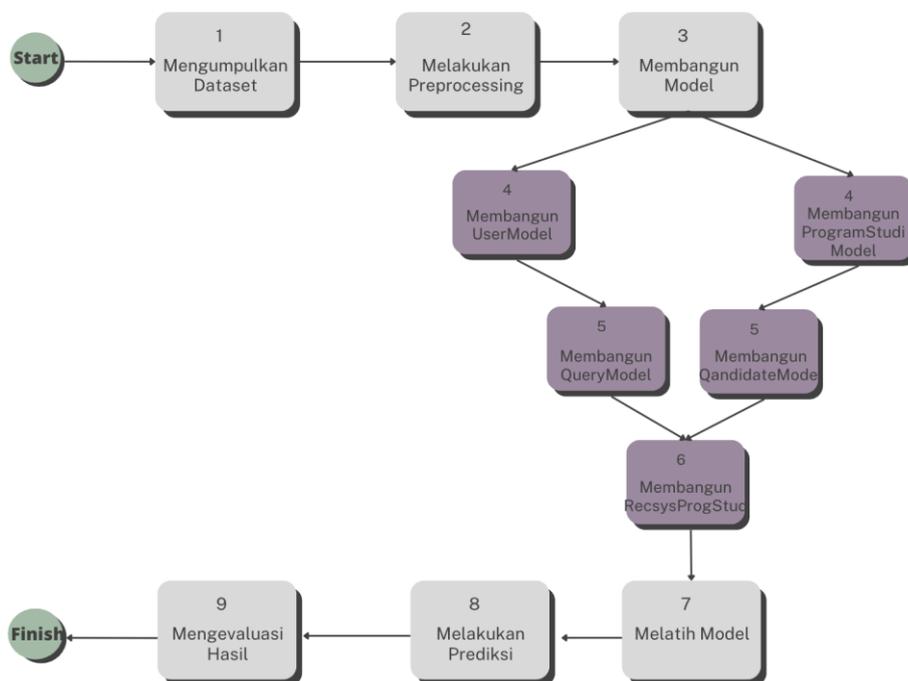
Dalam sistem rekomendasi modern, penerapan *deep learning* semakin umum digunakan untuk meningkatkan performa model rekomendasi. *Deep learning* mampu menangkap hubungan yang kompleks dan non-linear dalam data, serta memanfaatkan berbagai sumber informasi seperti teks, visual, dan konteks pengguna. Dengan kerangka kerja seperti *TensorFlow Recommenders System* (TFRS), sistem rekomendasi dapat dirancang dengan fleksibilitas tinggi untuk menggabungkan informasi pengguna, item, dan konteks. TFRS juga memungkinkan pembuatan model *multi-task* yang dapat mengoptimalkan beberapa tujuan rekomendasi secara bersamaan, sehingga memberikan hasil yang lebih personal dan relevan [20].

Penelitian sebelumnya menunjukkan bahwa penggunaan pendekatan hybrid dan *deep learning* dalam sistem rekomendasi memberikan hasil yang menjanjikan. Contohnya, penelitian [21] yang membuktikan bahwa penggabungan teknik TensorFlow dengan pendekatan kolaboratif dapat menyelesaikan masalah *cold start* dan memberikan rekomendasi buku yang relevan. Penelitian yang dilakukan oleh Tiara [22] menunjukkan bahwa pendekatan *Content-Based Filtering* mampu mencapai akurasi hingga 94% dalam memberikan rekomendasi jurusan kuliah. Selain itu, penelitian yang dilakukan oleh Muthi [23] menunjukkan bahwa pendekatan *hybrid* dengan metode Naive Bayes mampu memberikan akurasi sebesar 85,5% dalam sistem rekomendasi program studi. Hal ini menunjukkan bahwa kombinasi pendekatan berbasis machine learning dan *deep learning* memberikan kontribusi signifikan terhadap pengembangan sistem rekomendasi yang efektif.

3. Metode Penelitian

Penelitian ini menggunakan *toolkit* Kaldi dengan mengikuti *recipe* Voxceleb v2. Diagram alir dari penelitian ini dapat dilihat pada Gambar 1. Dimulai dari pengumpulan *dataset* yang berupa berkas-berkas audio, *pre-processing* data audio, persiapan berkas dalam format Kaldi, ekstraksi fitur, pembangunan model, pengujian model, dan analisis hasil pengujian model. *Dataset* utama yang digunakan adalah Voxceleb1, VoxCeleb2,

dan INF19. Sedangkan *dataset* RIRs Noise dan MUSAN hanya digunakan untuk melakukan augmentasi. Setelah *dataset* utama dikumpulkan, dilakukan *pre-processing* berkas audio agar dapat diterima oleh Kaldi. Lalu, disiapkan berkas-berkas dalam format Kaldi agar dapat diproses. Selanjutnya, digunakan dua konfigurasi MFCC untuk ekstraksi fitur dan dihitung VAD. Khusus *dataset* untuk pembangunan model, dilakukan augmentasi menggunakan *dataset* RIRs Noise dan MUSAN, normalisasi fitur menggunakan CMVN, dan *filtering* atau penyaringan. Setelah fitur-fitur sudah tersedia, dibangun empat model dengan menggunakan kombinasi dari dua konfigurasi MFCC dan dua arsitektur DNN yang memanfaatkan TDNN. Setiap model yang telah dibangun diuji menggunakan data pengujian dengan memanfaatkan x-vectors yang diekstrak menggunakan masing-masing model dan dianalisis hasil pengujian model untuk mencari model terbaik berdasarkan akurasi tertinggi dan durasi ekstraksi x-vectors tersingkat.

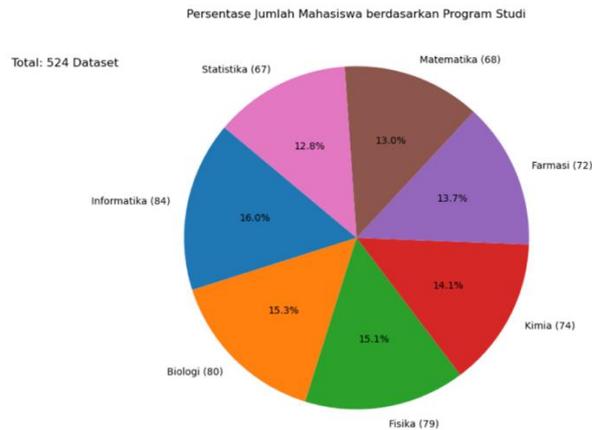


Gambar 1. Alur tahapan penelitian

A. Mengumpulkan Dataset

Tahapan pertama dalam metodologi penelitian ini adalah pengumpulan dataset. Informasi yang relevan tentang pengguna, seperti atribut, perilaku, atau konten yang diakses, dikumpulkan untuk membangun profil pengguna sebagai dasar rekomendasi. Data ini diperoleh dari mahasiswa Universitas Syiah Kuala melalui *Google Form* yang dirancang dengan lebih dari 10 pertanyaan. Pertanyaan yang terdapat pada *Google Form* akan menjadi fitur dalam pembuatan *dataset*. Fitur tersebut akan mendeskripsikan karakteristik *user* yang nantinya akan dijadikan sebagai pedoman rekomendasi. Fitur tersebut terdiri dari jenis kelamin, jurusan SMA, nilai matematika, nilai bahasa Inggris, nilai bahasa Indonesia, preferensi pekerjaan, mata pelajaran yang diminati, kriteria diri, topik dan bidang yang diminati, kecerdasan yang dimiliki dan keahlian yang dimiliki. Rincian jumlah dan penyebaran data di tiap program studi tersebut dapat ditemukan pada Gambar 2.

SISTEM REKOMENDASI PEMILIHAN PROGRAM STUDI BERBASIS *HYBRID* MENGGUNAKAN PENDEKATAN *DEEP LEARNING*



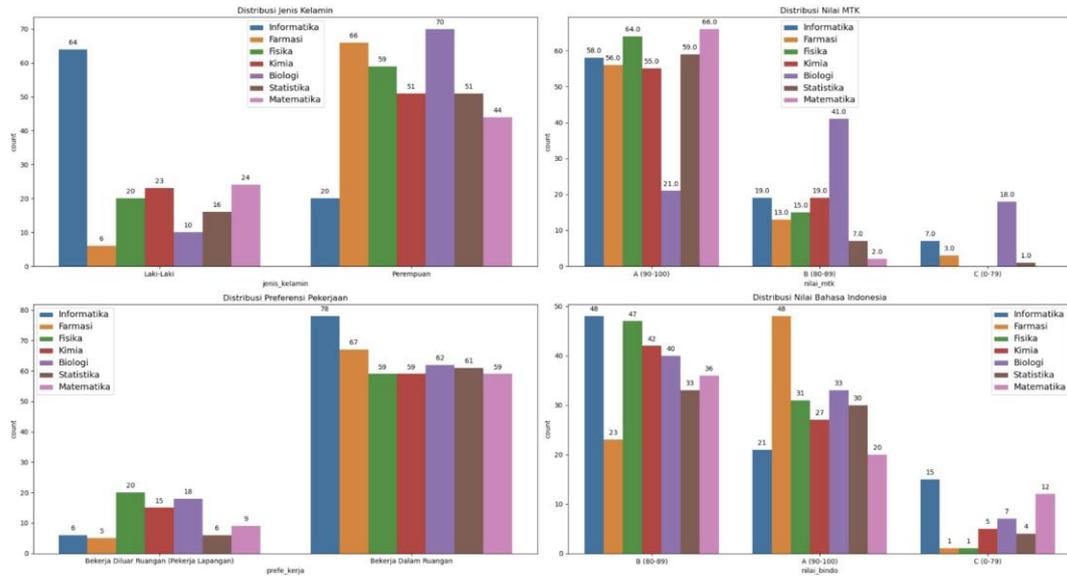
Gambar 2. Visualisasi persentase data pada setiap program studi

Selanjutnya, pada *dataset* tersebut dilakukan proses augmentasi untuk mengamati apakah performa model dapat ditingkatkan dengan penggunaan *dataset* augmentasi. *Dataset* augmentasi berisi *augmented* data sebanyak 50 data tiap program studi. Karakteristik mahasiswa dalam *augmented* data ini diambil dari beberapa penelitian terdahulu yang mempelajari karakteristik mahasiswa dalam memilih program studi tertentu. Tujuan dari *dataset* augmentasi adalah untuk mendukung *dataset* asli dalam menyortir karakteristik mahasiswa yang sebenarnya memilih program studi tertentu dan tidak mengurangi variasi *dataset* aslinya.

B. *Pre-processing Data*

Dataset yang telah dikumpulkan tidak dapat langsung digunakan karena mengandung noise seperti tanda baca, spasi ekstra, atau format yang tidak standar. Proses *pre-processing* dilakukan untuk membersihkan data dan menyiapkannya agar kompatibel dengan TensorFlow Recommenders System (TFRS). Tujuan utama *pre-processing* adalah meningkatkan kualitas data dan memastikan data siap digunakan dalam pengembangan model rekomendasi. Beberapa tahap pemrosesan tersebut antara lain:

- 1) Pengolahan *dataset* ini akan dilakukan dengan memanfaatkan *library* Pandas (*Python Data Analysis Library*). Proses analisis *dataset* ini bertujuan untuk mendapatkan informasi yang tersembunyi, memahami karakteristik data, dan memahami konteks data. Pada Gambar 3, diagram tersebut menampilkan distribusi mahasiswa berdasarkan jenis kelamin, nilai Matematika, Bahasa Indonesia dan preferensi pekerjaan. Dari distribusi jenis kelamin, terlihat bahwa jumlah perempuan lebih tinggi dibandingkan laki-laki di beberapa jurusan, meskipun terdapat variasi antar bidang studi. Pada distribusi nilai Matematika dan Bahasa Indonesia, mayoritas mahasiswa memperoleh nilai dalam kategori A (tinggi) dan B (sedang), sementara kategori C (rendah) memiliki jumlah yang lebih sedikit. Sementara itu, dalam hal preferensi pekerjaan, sebagian besar mahasiswa lebih memilih bekerja dalam ruangan dibandingkan di luar ruangan.



Gambar 3. Visualisasi dataset menggunakan diagram batang

- 2) Nama kolom yang panjang dalam *dataset* yang dikumpulkan dari pertanyaan dalam *Google Forms* sebelumnya diubah lebih singkat untuk dijadikan nama fitur. Hal ini bertujuan untuk mengurangi potensi kesalahan penulisan dan mempermudah proses pemrosesan selanjutnya.



Gambar 4. Nama kolom yang digunakan sebagai fitur

- 3) *Dataset* Pandas yang sebelumnya digunakan untuk mempermudah proses EDA sekarang akan digantikan dengan *Tensor Dataset*. Perubahan ini diperlukan karena penelitian ini menggunakan *framework Tensorflow*. Framework ini hanya dapat menerima input *dataset* dari *Tensor Dataset* agar dapat digunakan dalam proses pembelajaran model.
- 4) Total fitur dalam *dataset* mencapai 16 kolom. Namun, tidak semua kolom tersebut dapat digunakan karena beberapa diantaranya tidak memiliki relevansi sebagai ciri-ciri mahasiswa dalam pemilihan Program Studi. Beberapa fitur yang tidak digunakan termasuk 'Timestamp', 'Nama Lengkap', 'Jurusan SMA', dan dua verifikasi *dataset*

SISTEM REKOMENDASI PEMILIHAN PROGRAM STUDI BERBASIS *HYBRID* MENGGUNAKAN PENDEKATAN *DEEP LEARNING*

lainnya. Jadi, jumlah fitur yang benar-benar digunakan untuk pembelajaran model adalah 10, yang antara lain mencakup 'Jenis Kelamin', 'Program Studi', 'Nilai Matematika', 'Nilai Bahasa Indonesia', 'Nilai Bahasa Inggris', 'Mata Pelajaran', 'Preferensi Pekerjaan', 'Kriteria Diri', 'Topik dan Bidang', 'Kecerdasan', 'Bakat dan Keahlian'.

- 5) Setiap fitur dijadikan unik dengan tujuan untuk mempermudah pembuatan kosa kata (*vocabulary*) dari fitur tersebut. Nantinya, fungsi `StringLookup` akan membutuhkan inputan *vocabulary* yang unik dari setiap fitur. Langkah membuat nilai unik dari setiap fitur dapat dilakukan dengan memanfaatkan fungsi `np.unique` dari modul *Numpy*.

C. Membangun Model

Dalam fase pembangunan model, sistem rekomendasi menggunakan *TensorFlow Recommenders System* (TFRS). TFRS menawarkan proses kerja yang lengkap dan beratur, mulai dari pemrosesan *dataset* hingga pembangunan model dan menghasilkan rekomendasi. Model ini terdiri dari lima bagian, yaitu *UserModel*, *QueryModel*, *ProgramStudiModel*, dan *CandidateModel*, serta *RecsysProgramStudi*. Setiap model memiliki tujuan khususnya dalam rangka memperoleh hasil yang diinginkan.

1) Membangun *UserModel*

Dalam proses pembangunan model, *dataset* yang digunakan pada *UserModel* adalah *dataset* karakteristik mahasiswa yang telah dijelaskan pada proses pengumpulan *dataset*. Model ini bertujuan untuk menangkap fitur atau karakteristik *user* yang nantinya membentuk kerangka kerja yang berurutan, dan mencakup *layer* untuk mengubah dari *string* ke *embedding*. Proses ini melibatkan dua *layer*, yaitu *StringLookup* dan *Embedding*. Sebagai contoh, fitur seperti jenis kelamin yang memiliki dua nilai, Laki-Laki dan Perempuan, akan dikodekan menjadi 0 dan 1, kemudian 0 dan 1 akan diubah menjadi vektor menggunakan. Setelah semua fitur menjadi vektor, proses selanjutnya adalah membangun *QueryModel*.

2) Membangun *QueryModel*

Proses membangun *QueryModel* dimulai dengan mengonversi fitur *user* menjadi pola yang dapat dimengerti oleh neural network. Pola yang diekstrak tersebut menghasilkan informasi yang berguna untuk rekomendasi program studi. *QueryModel* akan menggunakan *UserModel* untuk membuat representasi *embedding* dari data pengguna. Model ini menerima input dari *user* dan menggunakan *UserModel* untuk menghasilkan representasi *embedding* dari input tersebut. *Embedding* ini kemudian melewati *dense layer* pada arsitektur neural network. Hasil akhir dari proses ini adalah representasi numerik dari fitur *user*, yang dapat digunakan untuk prediksi atau rekomendasi.

3) Membangun *ProgramStudiModel*

Seperti halnya *UserModel*, *ProgramStudiModel* dibangun untuk *encoding* data dari program studi. *Dataset* yang digunakan berasal dari *dataset* program studi yang telah dijelaskan pada proses pengumpulan *dataset*. *Dataset* ini memuat informasi unik mengenai Program Studi di FMIPA. Model ini mengikuti langkah yang serupa dengan *UserModel*, yaitu mengalami proses transformasi dari *string* menjadi *integer* menggunakan `tf.keras.layers.StringLookup`, dan dari *integer* ke vektor. Selain itu, *TextVectorization* digunakan untuk mengonversi teks menjadi urutan token, diikuti oleh lapisan *Embedding* dan *GlobalAveragePooling1D* untuk menghasilkan representasi global dari teks tersebut.

- 4) Membangun *CandidateModel*
 Model ini memanfaatkan *ProgramStudiModel* untuk menciptakan representasi *embedding* dari data program studi. Setelah itu, beberapa *dense layer (layer neural network)* dibangun untuk menghasilkan representasi numerik dari data program studi yang dapat digunakan untuk merekomendasikan program studi.
- 5) Membangun *RecsysProgStudi*
 Langkah terakhir dalam proses pembangunan model adalah menggabungkan kedua model yaitu *QueryModel* dan *QandidateModel* serta mencari kesamaan antara pengguna untuk memberikan Rekomendasi Program Studi. Pada fase *Retrieval* ini, kita hanya perlu menggunakan output *QueryModel*. Kueri tersebut akan diubah menjadi vektor dengan menggunakan *Neural Network* yang telah dipelajari sebelumnya. Vektor tersebut kemudian dicocokkan dengan output *QandidateModel* (yang juga sudah melalui proses *encoding* dengan *Neural Network* yang telah dilatih, sehingga memudahkan dalam menghitung kesamaan). Kesamaannya dapat dihitung menggunakan metode seperti *cosine similarity* (yang merupakan perkalian vektor), dan ini akan menghasilkan *similarity score* yang dapat diurutkan. Proses inilah yang dikenal sebagai *Retrieval* secara "*brute force*".

4. Hasil dan Pembahasan

Pada penelitian ini menggunakan dua *dataset*, yaitu *Real Dataset* dan *Augmented Dataset*. Dalam proses pelatihan model, beberapa variasi *learning rate* digunakan dengan nilai sebesar 0.1, 0.05, dan 0.001. Jumlah layer divariasikan, termasuk 1 (satu) *layer* dengan 32 *neuron*, 2 (dua) *layer* dengan 64 dan 32 *neuron*, serta 3 (tiga) *layer* dengan 128, 64, dan 32 *neuron*. *Batch size* diatur dengan ukuran 16, 32, dan 64, dan eksperimen dijalankan dengan jumlah *epoch* sebanyak 2000.

Tabel 1. Hasil eksperimen dengan *hyperparameter* berbeda dengan *real dataset*

No.	Dataset	Learning Rate	Layer	Batch size	Epoch	Akurasi Top-1	Akurasi Top-5			
1	Real Dataset (524)	0.1	1 (32)	16	2000	0.51	0.92			
2				32		0.46	0.92			
3				64		0.43	0.93			
4			2 (64, 32)	16		2000	0.42	0.91		
5				32			0.39	0.94		
6				64			0.42	0.95		
7				3 (128, 64, 32)			16	2000	0.44	0.93
8							32		0.38	0.89
9							64		0.35	0.89
10	Real Dataset (524)	0.05	1 (32)	16	2000	0.59	0.97			
11				32		0.59	0.97			
12				64		0.49	0.96			
13			2 (64, 32)	16		2000	0.52	0.97		
14				32			0.53	0.94		
15				64			0.39	0.97		
16				3 (128, 64, 32)			16	2000	0.46	0.98
17							32		0.57	0.93
18							64		0.51	0.96
19	Real Dataset (524)	0.001	1 (32)	16	2000	0.53	0.99			
20				32		0.53	0.98			
21			64	0.48		0.97				
22			2 (64, 32)	16		2000	0.47	0.98		
23				32			0.49	0.96		

**SISTEM REKOMENDASI PEMILIHAN PROGRAM STUDI
BERBASIS *HYBRID* MENGGUNAKAN PENDEKATAN *DEEP LEARNING***

24		64		0.51	0.97
25		16		0.36	0.95
26	3 (128, 64, 32)	32	2000	0.33	0.92
27		64		0.46	0.95

Proses melatih model dengan berbagai kombinasi parameter seperti yang tertera pada Tabel 1 memerlukan waktu sekitar 7 hingga 20 menit untuk setiap eksekusi kombinasi. Dengan menggunakan *dataset* asli dan melakukan beberapa penyesuaian pada *hyperparameter*, hasil evaluasi performa model yang diukur dengan Akurasi Top-1. Akurasi top-1 mengukur sejauh mana model berhasil memprediksi item atau entitas yang benar-benar relevan di peringkat teratas. Akurasi yang didapat berkisar antara 0.40 hingga 0.56, sementara Akurasi Top-5 berkisar antara 0.86 hingga 0.99.

Dapat diamati bahwa dalam melatih model, penggunaan *learning rate* yang tinggi, yaitu 0.1, menghasilkan akurasi yang cenderung tidak stabil karena langkah-langkah yang terlalu besar dapat menyebabkan model melampaui titik minimum yang optimal dalam fungsi *loss*, dan sebagai akibatnya, model mungkin kesulitan mencapai nilai yang diinginkan. Sebaliknya, penggunaan *learning rate* yang lebih rendah, seperti 0.05 dan 0.001, memberikan akurasi yang lebih stabil, seperti yang terlihat pada Tabel 1.

Penggunaan 1 Layer menghasilkan hasil yang memadai dengan akurasi yang tetap di atas *Baseline*. Menggunakan 2 layer menghasilkan performa yang lebih baik dibandingkan dengan 1 layer yang dapat dilihat pada nilai akurasi yang dihasilkan lebih tinggi. Penggunaan 3 layer tidak memberikan hasil yang lebih baik dibandingkan dengan penggunaan 1 dan 2 layer. Dalam penggunaan konteks ini, peningkatan jumlah layer pada model memang dapat meningkatkan variabilitas dan kompleksitas, namun perlu diimbangi dengan ketersediaan informasi yang memadai untuk membangun kompleksitas tersebut.

Selain itu, dapat dilihat bahwa menggunakan *batch size* 16 menghasilkan akurasi yang secara keseluruhan lebih baik karena *dataset* yang dimiliki masih tergolong sedikit sehingga, model yang belajar dengan menggunakan *batch size* yang kecil akan lebih cepat menyesuaikan dengan karakteristik data daripada menggunakan *batch size* 32 dan 64.

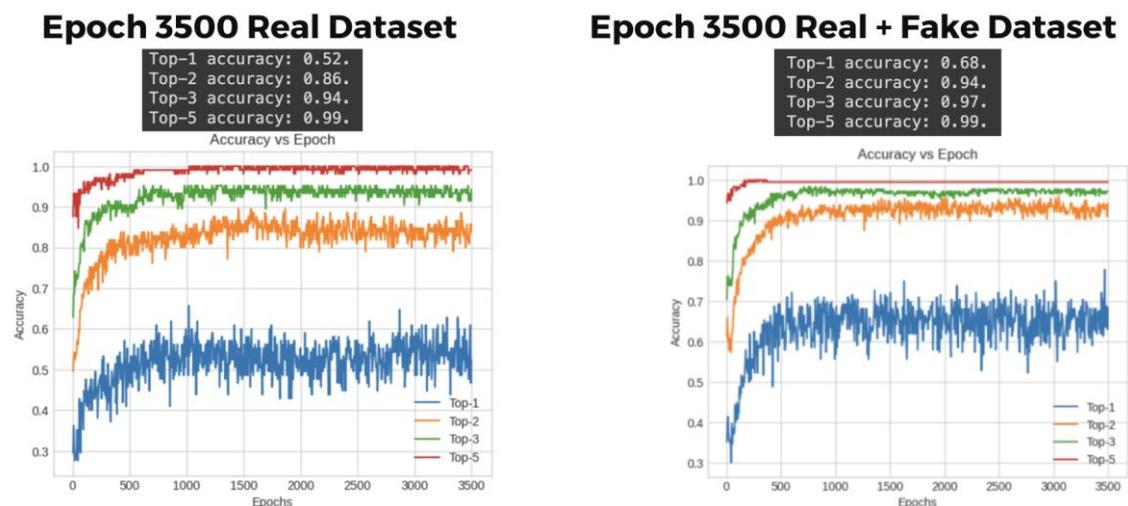
Tabel 2. Hasil eksperimen hyperparameter dengan *real* dan *augmented dataset*

No	Dataset	Learning Rate	Layer	Batch size	Epoch	Akurasi Top-1	Akurasi Top-5
1				16		0.59	0.98
2	<i>Real Dataset +</i>	0.1	1 (32)	32	2000	0.55	0.98
3				64		0.58	0.98
4				16		0.59	0.97
5	<i>Augmented Dataset (350)</i>		2 (64, 32)	32	2000	0.56	1.00
6				64		0.53	0.98
7				16		0.57	0.97
8			3 (128, 64, 32)	32	2000	0.55	0.95
9				64		0.64	0.98
10				16		0.59	0.98
11	<i>Real Dataset +</i>	0.05	1 (32)	32	2000	0.53	0.97
12				64		0.60	0.98
13				16		0.61	0.99
14	<i>Augmented Dataset (350)</i>		2 (64, 32)	32	2000	0.58	0.98
15				64		0.59	0.98
16				16		0.53	0.99
17			3 (128, 64, 32)	32	2000	0.51	0.99
18				64		0.57	0.97

19				16		0.65	1.00
20			1 (32)	32	2000	0.64	0.99
21	<i>Real</i>			64		0.66	1.00
22	<i>Dataset +</i>			16		0.63	0.99
23	<i>Augmented</i>	0.001	2 (64, 32)	32	2000	0.59	0.99
24	<i>Dataset</i>			64		0.65	0.97
25	(350)			16		0.64	1.00
26			3 (128, 64, 32)	32	2000	0.59	0.98
27				64		0.56	0.98

Terlihat pada Tabel 2 bahwa pelatihan model memerlukan waktu sekitar 10 hingga 30 menit untuk satu kali eksekusi kombinasi seperti yang tertera pada tabel di atas. Dengan menggunakan *augmented dataset* dan melakukan beberapa penyesuaian pada *hyperparameter*, hasil evaluasi performa model yang diukur dengan akurasi top-1 dan top-5 menunjukkan peningkatan performa dibandingkan dengan model yang hanya menggunakan *real dataset* tanpa tambahan *Augmented dataset*. Hal ini dapat dijelaskan karena *augmented dataset* berisi 50 *augmented dataset* yang mendukung karakteristik mahasiswa dalam memilih program studi tertentu, tanpa menghilangkan variasi dari *real dataset* tersebut *epoch* 3500.

Setelah dijalankan dengan dua kondisi, yaitu *menggunakan real dataset* ditambah dengan *augmented dataset* dan dengan *hyperparameter* yang telah ditentukan, hasilnya menunjukkan bahwa menggabungkan *real dataset* dengan *augmented dataset* meningkatkan akurasi rata-rata sekitar 10%, dibandingkan dengan pembelajaran model hanya dengan menggunakan *real dataset*. Selain itu, rekomendasi program studi yang dihasilkan oleh model juga menjadi lebih baik. Pada *epoch* ke-3500 dengan dua skenario, yakni dengan *real dataset* dan tambahan *augmented dataset*, tingkat akurasi pada Top-1 berkisar antara 0.5 hingga 0.59, sedangkan pada Top-2 berkisar antara 0.8 hingga 0.86. Berbeda ketika *augmented dataset* ditambahkan, yaitu Top-1 berkisar antara 0.6 hingga 0.75, dan Top-2 berkisar antara 0.9 hingga 0.96. Dengan melibatkan 3500 *epoch*, akurasi yang dicapai tetap stabil dan berada pada rentang angka tersebut yang dapat dilihat pada Gambar 5.



Gambar 5. Hasil training model TensorFlow Recommenders System (TFRS)

5. Kesimpulan

Berdasarkan penelitian, jumlah dataset yang terbatas menjadi tantangan dalam implementasi model berbasis *Deep learning*, yang dapat menghambat kinerja optimal sistem. Meskipun fitur yang digunakan sudah menunjukkan hasil yang baik, akurasi sistem rekomendasi dapat ditingkatkan dengan menambahkan fitur tambahan, seperti hobi, gaya belajar, kemampuan *problem-solving*, dan penghasilan orang tua. Model TFRS mencapai performa terbaik pada sistem rekomendasi program studi dengan akurasi Top-1 sebesar 0,59 dan Top-5 sebesar 0,97 ketika menggunakan dataset nyata, *hyperparameter learning rate* 0,05, satu lapisan dengan 32 neuron, *batch size* 16, dan 2000 *epoch*. Sebaliknya, performa terburuk untuk *dataset* ini menunjukkan akurasi Top-1 sebesar 0,33 dan Top-5 sebesar 0,92 dengan konfigurasi *hyperparameter learning rate* 0,001, tiga lapisan (128, 64, 32 neuron), *batch size* 32, dan 2000 *epoch*. Ketika *dataset* nyata dikombinasikan dengan *dataset* sintetis, akurasi model meningkat menjadi Top-1 sebesar 0,66 dan Top-5 sebesar 1,0 dengan konfigurasi *hyperparameter learning rate* 0,001, satu lapisan dengan 32 neuron, *batch size* 64, dan 2000 *epoch*. Namun, performa terburuk untuk kombinasi *dataset* ini menghasilkan akurasi Top-1 sebesar 0,51 dan Top-5 sebesar 0,99 dengan konfigurasi *hyperparameter learning rate* 0,05, tiga lapisan (128, 64, 32 neuron), *batch size* 32, dan 2000 *epoch*. Temuan ini menunjukkan bahwa kombinasi *dataset* dan pemilihan *hyperparameter* yang tepat dapat meningkatkan akurasi sistem rekomendasi secara signifikan.

Daftar Pustaka

- [1] U. H. S. Maritsa, M. Wafiq, P. R. Anindya, dan M. A. Ma'shum, "Pengaruh Teknologi Dalam Dunia Pendidikan," *Al-Mutharahah: Jurnal Penelitian dan Kajian Sosial Keagamaan*, vol. 18, no. 2, pp. 91-100, 2021.
- [2] H. Sucipto, "Sistem pendukung keputusan penentuan jurusan Sekolah Menengah Atas dengan metode saw," *Sisfotenika*, vol. 6, no. 2, pp. 146-157, 2016, doi: 10.30700/jst.v6i2.113.
- [3] R. Febriantomo dan A. Suharnan, "Pengaruh self-efficacy terhadap pengambilan keputusan karier siswa SMA," *Jurnal Psikologi Pendidikan dan Konseling*, vol. 1, no. 2, pp. 56-63, 2015.
- [4] M. L. Linda dan H. Muridan, "Pemilihan Jurusan Kuliah Berdasarkan Bakat, Minat dan Kepribadian," *Jurnal Bimbingan Konseling dan Psikologi Pendidikan*, vol. 1, no. 1, pp. 1-12, 2020.
- [5] R. Rizka, R. E. Putri, Y. Yusman, dan M. Fajar, "Sistem Rekomendasi Jurusan Kuliah dalam Pengambilan Keputusan Menggunakan Metode MOORA," *Jurnal Penerapan Sistem Informasi*, vol. 4, no. 2, pp. 364-373, 2023.
- [6] J. A. Konstan dan J. Riedl, "Recommender systems: from algorithms to user experience," *User Modeling and User-Adapted Interaction*, vol. 22, pp. 101-123, 2012, doi: 10.1007/s11257-011-9112-x.
- [7] L. Sharma dan A. Gera, "A Survey of Recommendation System: Research Challenges," *International Journal of Engineering Trends and Technology (IJETT)*, vol. 4, no. 5, pp. 1989-1991, 2013.
- [8] W. Pan dan W. Li, "Research paper recommendation with topic analysis," in *Proc. 2010 International Conference on Computer Design and Applications*, vol. 4, pp. V4-264, 2010, doi: 10.1109/ICCD.2010.5541170.

- [9] P. Pu, L. Chen, dan R. Hu, "A user-centric evaluation framework for recommender systems," in Proc. Fifth ACM Conf. on Recommender Systems, pp. 157-164, 2011.
- [10] M. Jalali, N. Mustapha, M. N. Sulaiman, dan A. Mamat, "WebPUM: A Web-based Recommendation System to predict user future movements," *Expert Systems with Applications*, vol. 37, no. 9, pp. 6201-6212, 2010, doi: 10.1016/j.eswa.2010.02.105.
- [11] Y. LeCun, Y. Bengio, dan G. Hinton, "*Deep learning*," *Nature*, vol. 521, no. 7553, pp. 436-444, 2015, doi: 10.1038/nature14539.
- [12] G. James, D. Witten, T. Hastie, dan R. Tibshirani, *An Introduction to Statistical Learning with Applications in R*, New York: Springer, 2014.
- [13] X. Zhao, X. Du, dan Z. Wang, "Cross-entropy loss and its applications in *deep learning*," *Pattern Recognition Letters*, vol. 144, pp. 27-34, 2021, doi: 10.1016/j.patrec.2021.01.015.
- [14] I. M. Spartalis, "A survey on recommendation systems: Classification and evaluation," *International Journal of Artificial Intelligence Applications*, vol. 8, no. 2, pp. 1-14, 2017.
- [15] P. Melville dan V. Sindhvani, "Recommender systems," in *Encyclopedia of Machine Learning*, D. Sammut and G. Webb, Eds., Boston: Springer, 2010, pp. 829-838.
- [16] C. C. Aggarwal, *Recommender Systems: The Textbook*, Cham: Springer, 2016.
- [17] P. B. Thorat, R. Goudar, dan S. Barve, "Survey on collaborative filtering, content-based filtering and hybrid recommendation system," *International Journal of Computer Applications*, vol. 110, no. 4, pp. 31-36, 2015.
- [18] M. Badole, "A review on hybrid recommendation systems," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, vol. 8, no. 1, pp. 45-49, 2022.
- [19] G. Getha et al., "Hybrid recommendation systems: A comprehensive survey," *Journal of Intelligent Information Systems*, vol. 67, no. 3, pp. 105-122, 2023.
- [20] M. Kula dan J. Chen, "TensorFlow recommenders: Scalable and flexible recommendation systems," in Proc. ACM Conf. on Recommender Systems, pp. 342-343, 2020.
- [21] R. Anandaraj, K. Selvaraj, dan S. Saranya, "Book recommendation system using TensorFlow and collaborative filtering," *International Journal of Advanced Science and Technology*, vol. 143, pp. 120-127, 2021.
- [22] Tiara, A. Putri, dan S. Sukardi, "Content-based filtering for college major recommendation," *Journal of Educational Data Mining*, vol. 15, no. 2, pp. 34-47, 2023.
- [23] A. Muthi, R. Wijaya, dan H. Darmawan, "Hybrid recommendation system using naive Bayes for study program selection," *Journal of Information Systems Research and Innovation*, vol. 11, no. 2, pp. 85-91, 2020.