OPEN ACCESS
Indo-JC
ISSN 2460-9056
journals.telkomuniversity.ac.id/indojc

Hybrid Recommendation System for University Major Selection Using Sentence-Transformers, FAISS, and Rule-Based Matching Studicase at SMK Telkom Jakarta.

Muhammad Kahfi Bachtiar #1, Eko Darwiyanto #2, Thofik Nugroho #3
 # School of Computing, Telkom University Bandung, Indonesia
[1] muhammadkahfi@student.telkomuniversity.ac.id
[2] ekodarwiyanto@telkomuniversity.ac.id
[3] thofiknugroho@telkomuniversity.ac.id

Abstract

For vocational high school students, choosing a college major is crucial because it affects career suitability, skill development, and academic motivation. This study develops a hybrid recommendation system for students at SMK Telkom Jakarta who major in Computer and Network Engineering (TKJ) and Software Engineering (RPL). The proposed system combines Sentence-Transformers for semantic representation, FAISS for vector similarity search, and rule-based matching for domain-aware re-ranking. Free-form text represents students' varied interests, preferences, hobbies, and vocational skills, with optional report-card grades. Sentence-Transformers generate embeddings, while FAISS indexes them for similarity search; the candidate majors are then re-ranked using competency-linearity rules, interest keywords, and optional academic requirements. The system was evaluated on 60 labeled profiles using both classification metrics (Precision, Recall, F1-Score) and ranking-aware metrics (MAP, MRR, NDCG), complemented by a usability questionnaire. In the Top-3 scenario, the system achieved Precision = Recall = F1-Score = 0.6944 (0.6818 without report-card input on 44 profiles and 0.7292 with report-card input on 16 profiles), with MAP = 0.657, MRR = 0.972, and NDCG = 0.752, indicating that the most suitable major is almost always ranked first. A Top-2 configuration increased precision to 0.80 at the cost of recall (0.533). Based on an average System Usability Scale (SUS) score of 69.57, the prototype is considered usable and understandable.

Keywords: FAISS, major recommendation, rule-based matching, semantic similarity, Sentence-Transformers, vocational education.

Abstrak

Bagi siswa sekolah menengah kejuruan, memilih jurusan kuliah sangat penting karena memengaruhi kesesuaian karir, pengembangan kompetensi, dan motivasi akademik. Studi ini mengembangkan sistem rekomendasi hibrida untuk siswa SMK Telkom Jakarta yang menempuh jurusan Teknik Komputer dan Jaringan (TKJ) serta Rekayasa Perangkat Lunak (RPL). Sistem ini menggabungkan Sentence-Transformers untuk representasi semantik, FAISS untuk pencarian kesamaan vektor, dan pencocokan berbasis aturan untuk pemeringkatan ulang berbasis domain. Teks bebas digunakan untuk mewakili minat, preferensi, hobi, dan keterampilan kejuruan siswa, dengan nilai rapor sebagai masukan opsional. Sentence-Transformers menghasilkan embedding, FAISS mengindeksnya untuk pencarian kesamaan, lalu jurusan kandidat disusun ulang berdasarkan aturan linearitas kompetensi, kata kunci minat, dan persyaratan akademik opsional. Evaluasi dilakukan pada 60 profil berlabel menggunakan metrik klasifikasi (Presisi, Recall, F1-Score) dan metrik berbasis peringkat (MAP, MRR, NDCG), serta kuesioner kegunaan. Pada skenario Top-3, sistem memperoleh Presisi = Recall = F1-Score = 0,6944 (0,6818 tanpa nilai rapor pada 44 profil dan 0,7292 dengan nilai rapor pada 16 profil), dengan MAP = 0,657, MRR = 0,972, dan NDCG = 0,752. Konfigurasi Top-2 meningkatkan presisi menjadi 0,80 dengan recall yang lebih rendah (0,533). Berdasarkan skor SUS rata-rata 69,57, prototipe ini dinilai mudah digunakan dan dipahami.

Kata Kunci: FAISS, rekomendasi jurusan, rule-based matching, semantic similarity, Sentence-Transformers, pendidikan vokasi.

Received on January, 2026. Accepted on June, 2026

I. INTRODUCTION

The choosing of university major is one of the most important for students who plan to continue their education to the next level of education [1]. The impact of this decision is not limited to academic performance but also touches upon professional orientation and long term career for high school students, the choice is even more specific because they already have technical competencies obtained from their vocational program[2]. For example, students of Software Engineering (RPL) and Computer and Network Engineering (TKJ) typically have learning experiences that are closely related to computing, information systems, computer networks, cybersecurity, and other areas of technology [3].

Decision-support approaches use structured criteria such as report-card grades, interests, and manually weighted attributes. Methods such as AHP and rule-based decision systems can be used when the input data are numerical, but this approach is still less flexible for students who express interests in free text, such as 'I like web design' or 'I like server configuration [4].

Recent developments of natural language processing provide opportunities for students to present their interest and describe university majors in a semantic form. Sentence-Transformers can convert sentences into dense embedding vectors, enabling the system to compare meanings rather than just exact words [5]. Vector search libraries such as FAISS can search for similarities and then can retrieve similar items and semantics efficiently [6]. However, semantic similarity is not enough for education recommendations because semantic relevance does not always represent academic suitability. For example, a student wrote about creative design but the recommendation system still needs to consider the major background and minimum academic requirement.

Therefore, the research proposes a recommendation system that integrates three components, namely sentence-transformers, FAISS and rule-based matching. Sentence-transformers are used to encode student profiles and descriptions into semantic vectors. FAISS is used to efficiently retrieve candidate majors and then rule-based matching as a processing and re-ranking mechanism to ensure recommendations remain aligned with student competencies, interests and report-card grades [7]. The aim is to design and implement a prototype recommendation system that can help students at SMK Telkom Jakarta to choose a university major systematically and accurately.

The main contribution of this study is the integration of semantic search with explicit educational rules for major recommendation[8]. Rather than relying solely on fixed rules or numerical scores, the system allows students to describe themselves in natural language while retaining domain control through competency alignment and academic rules. The approach is implemented as a web-based prototype in which students input their profiles and view final recommendations with explanations[9]. In addition, this paper reports a baseline comparison against single-method approaches, an error analysis of incorrect recommendations, and ranking-aware evaluation metrics (MAP, MRR, NDCG), which were not addressed in earlier prototype-only studies[10].

II. LITERATURE REVIEW

A. Major Recommendation and Educational Decision Support

A recommendation system aims to help students in choosing a major and career paths[11]. Student characteristics may include academic achievement, learning preferences, and career expectations. In the context of connecting student characteristics with program characteristics. program aspects can include curriculum focus, graduate profiles and career prospects. Some of them are displayed in multiple text forms, which may pose challenges to the recommendation process.

Previous research on major recommendation has used several approaches, from knowledge-based recommendation and decision-support systems to hybrid methods [12]. Content-based methods are useful for comparing user profiles with item descriptions, while collaborative methods can capture patterns across users but need enough interaction data and face cold-start problems[13]. Most recommender systems in higher education still rely on structured academic data and rarely use free-text student input. Some studies also combine several criteria with optimization techniques to improve accuracy, but they still need predefined features and weights. Because each single approach has its own weakness, hybrid systems are generally preferred, as they combine the strengths of more than one technique[14].

B. Comparison of Related Studies

Some studies that use decision support systems such as AHP-based major selection provide clear weighting so that the decisions can be interpreted[15]. However, these systems are generally more suitable for numerical or categorical criteria, such as academic scores, and are less flexible in processing free-text input. In contrast, the hybrid system base can increase personalization profile matching the user with the program or career description[16]. However, these systems can be determined before adequate user interaction but always dependent on explicit features.

Recent studies on semantic retrieval show that transformer-based sentence embeddings can improve text matching by capturing meaning instead of only lexical similarity. Transformer-based models have also been used for curriculum recommendation and give better matching than keyword-based methods. FAISS further supports scalability by enabling efficient similarity search in vector databases [17]. However, semantic retrieval alone is not enough to guarantee the appropriateness of recommendations in an educational context: a major that is textually similar to a student's interests may not match their vocational competencies or the academic requirements of the program. Rule-based methods are still valuable because they add explicit and transparent domain constraints, but they are less flexible when used in isolation [18].

Based on this comparison, there are still three gaps in the literature. First, structured decision-support systems need numerical or categorical input, so they cannot handle free-text student expression. Second, purely semantic retrieval methods can find textually similar majors, but they do not check competency linearity or academic eligibility. Third, most existing prototypes are not compared with single-method baselines and are not evaluated using ranking-aware metrics. This study fills these gaps by using Sentence-Transformers and FAISS to obtain semantic candidates and applying explicit educational rules for re-ranking, so the gap between semantic relevance and academic suitability can be reduced. Unlike pure structured decision-support systems, the proposed method can process students' free-text interests while still applying educational rules.

TABLE I

COMPARISON OF METHODS RELATED TO THIS STUDY

Method	Strength	Limitation	Role in This Research
Structured decision support / AHP	Clear and interpretable weighted criteria	Less flexible for free-text student input	Motivates the need for a more flexible approach
Content-based recommendation	Matches user profile and item description	Often depends on explicit features or keyword overlap	Used as conceptual basis for profile-major matching
Sentence-Transformers	Captures semantic meaning of free text	Requires model resources and input quality control	Encodes student profiles and major descriptions
FAISS	Efficient nearest-neighbor vector retrieval	Does not apply educational constraints by itself	Retrieves Top-N candidate majors
Rule-based matching	Transparent, controllable, and domain-aware	Depends on completeness of rules and keywords	Re-ranks and filters candidates

III. Research Method

A. Research Design

A research and development (R&D) approach is used in this study because the main output is a working prototype and its evaluation. The research process starts with problem identification, literature review, data collection, system design, implementation, testing and analysis. This prototype is specially designed for students of SMK Telkom Jakarta majoring in RPL and TKJ. The developed system was evaluated with recommendation metrics and usability evaluations.

The research problem is divided into two. The first question is, how can the design of a hybrid recommendation architecture be done by integrating sentence transformers, FAISS, and rule-based matching to visualize vocational competencies and students' interests in relevant college majors? Second, how to evaluate the accuracy, relevance, and usability of the developed system? These questions underlie the strategies of system design and evaluation.

B. Data Requirements and Preparation

Four types of data are required by the system: student profile data, university major data, rule data, and evaluation data. The student data includes free-text interests, preferences, hobbies, vocational competencies, and optional report-card scores. The major data includes the study program name, description, relevant competencies, interest keywords, career opportunities, and example universities. The rule data encodes three main rules: competency relevance, keyword adjustment, and academic-score thresholds. The evaluation data serves as the reference for assessment because it contains labeled student profiles paired with their relevant majors as ground truth.

The preprocessing stage was kept simple because Sentence-Transformers already handles tokenization internally. The preprocessing steps included converting text into lowercase, removing excessive whitespace, and merging student profile fields into a single query text[19]. The combined fields produced a richer representation because they included not only the intended major, but also competencies, interests, preferences, and career-related information.

Two scenarios with labeled test data are used in the evaluation, comprising 60 labeled profiles in total. The first scenario uses 44 profiles without report-card scores; the ground truth for each profile is the set of relevant majors derived from its competencies and interests. The second scenario uses 16 profiles additionally labeled with report-card grades in mathematics, English, and physics. For every profile the ground-truth set contains the relevant majors, and the system output is evaluated for both Top-2 and Top-3 recommendations. Precision, recall, F1-score, and ranking-aware metrics are then computed by comparing the system's Top-K output with the ground-truth labels.

The scope of the dataset reflects a deliberate design choice in this study. Rather than covering all vocational programs, the research intentionally focuses on the two programs most relevant to the computing-related majors targeted by the system, namely Software Engineering (RPL) and Computer and Network Engineering (TKJ). This scoping keeps the data collection focused and time-efficient while ensuring that the collected profiles are closely aligned with the system's domain. Within this scope, 60 labeled profiles were obtained, and because not all students completed the questionnaire, this set represents the realistically obtainable sample rather than an arbitrary cut-off.

TABLE II
DATA ATTRIBUTES USED IN THE SYSTEM

Data Type	Attributes	Function in the System
Student profile	Competency, interest, preference, hobby, optional report-card scores	Represents the user query and personalization context
Major database	Major name, description, relevant competency, keywords, career prospects, university examples	Represents recommendation items and explanation content
Rule data	Competency linearity, keyword bonus, optional report-card requirements	Controls re-ranking, filtering, and explanation
Evaluation data	Labeled student profiles and ground-truth majors	Used to calculate Precision, Recall, and F1-Score

C. System Architecture

In the system architecture there are five main modules. the first module is the input interface to enter vocational competencies and free-text profiles. The second module is embedding, which converts student profiles and major descriptions into vector representations using sentence transformers. The third module is the FAISS similarity search module, which retrieves candidate majors based on vector similarity. The fourth module is rule-based matching, which carries out filtering and re-ranking based on educational domain rules. The fifth module is the output and evaluation module, which displays recommendations and support the evaluation process.

The system process begins when students fill in their profiles. It contains the optional data such as competencies, interests, preferences, hobbies and report card score. The text-based profile is then combined into a single query and converted into a vector representation through the embedding process. The query is then matched against the vector representations of the indexed study programs to generate a Top-N candidate list using FAISS. The candidates found are not directly presented to the user, but are processed by a rule-based module to calibrate scores based on suitability of vocational competencies, matching interest keywords and optional academic criteria. The final results are presented as Top-K ranked recommendations, with scores and supporting explanations.

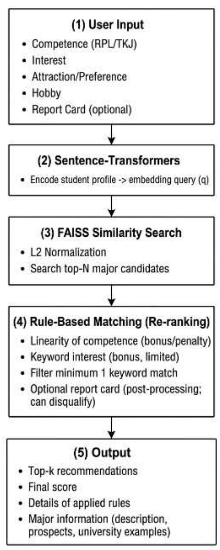

Fig. 1. System Architecture of the Proposed Recommendation System

TABLE III
SYSTEM MODULES AND THEIR FUNCTIONS

Module	Input	Main Process	Output
Input interface	Competency, interest, preference, hobby, optional scores	Collects and validates profile fields	Student profile object
Embedding module	Student profile and major descriptions	Encodes text into dense vectors	Profile and major embeddings
FAISS module	Query embedding and major index	Retrieves nearest candidate majors	Top-N raw candidates
Rule-based module	Candidates, competency, keywords, optional scores	Applies bonus, penalty, filtering, and re-ranking	Final Top-K recommendations
Evaluation module	Recommendations and ground truth	Computes recommendation metrics and usability summary	Evaluation results

D. Embedding, Similarity Search, and Scoring Procedure

The embedding module uses the multilingual Sentence-Transformers model paraphrase-multilingual-MiniLM-L12-v2. This model was chosen because the system input is written in Indonesian and might have a mix of technical terms[20]. The major descriptions are embedded once at load time, the student profiles are embedded at recommendation time. The profile query is built by using information from interest, preference, hobby and competency.

The similarity search module uses FAISS with IndexFlatIP. Before being added to the index and before query search, vectors are normalized using L2 normalization. With this normalization, inner product search behaves similarly to cosine similarity. FAISS returns more candidates than the final Top-K so that the rule-based module has enough alternatives for re-ranking.

The final recommendation score is calculated by fusing the semantic similarity score and rule-based adjustments[21]. The scoring process is simplified to: Final Score = Semantic Similarity Score + Competency Linearity Adjustment + Interest Keyword Bonus + Optional Report-Card Adjustment. Competency linearity provides a bonus if the major is related to the student's competency and a penalty if it is less related. Interest keywords give a small bonus to prevent repeated keyword matches from dominating the score. Candidates may be disqualified or adjusted by optional report-card rules when academic requirements are not met.

In the implemented prototype, the semantic similarity score has a base weight of 1.0 (the normalized cosine score returned by FAISS); competency linearity adds +0.25 when the major is aligned with the student's RPL/TKJ competency and -0.15 when it is not; each matched interest keyword adds +0.05 up to a maximum of +0.15; and the optional report-card rules apply variable adjustments (for example +0.08, -0.05, or +0.15) or disqualify a major when minimum requirements are not met. FAISS initially retrieves k + 10 candidates (13 candidates for K = 3, expandable up to k + 25 and capped at the 17 majors available in the dataset) before rule-based re-ranking. The concrete weight of each rule-based adjustment is summarized in Table IV.

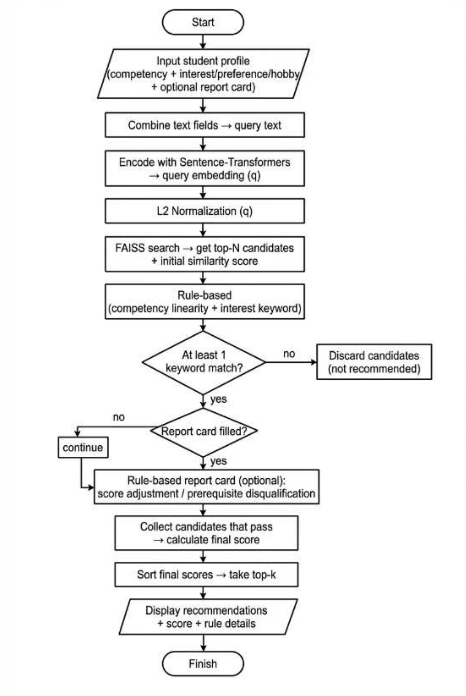

Fig. 2. Recommendation Process Flowchart

TABLE IV
SUMMARY OF RULE-BASED ADJUSTMENTS

Rule Category	Condition	Adjustment Purpose	Weight / Value
Competency linearity	RPL or TKJ matches the relevant competency of the major	Prioritizes majors aligned with vocational background	+0.25 if aligned, -0.15 if not
Interest keyword	Student free-text input contains major-related keywords	Increases relevance to personal interests	+0.05 per keyword (max +0.15)
Strict keyword filter	No important keyword appears in the query text	Reduces overly broad recommendations	Filter only (no score change)
Report-card requirement	Scores do not satisfy major-specific minimum values	Supports academic feasibility filtering	Varies, e.g. +0.08, -0.05, +0.15; may disqualify
Fallback relaxation	Strict filtering produces fewer than K results	Ensures the system still returns enough recommendations	Expands pool k+10 up to k+25 (max 17 majors)

E. Evaluation Procedure

Each profile is processed by the system to produce its Top-K major recommendations, and the recommendation quality is evaluated in a Top-K setting. The ground truth for each profile was defined prior to evaluation: for every labeled profile, the set of relevant majors was determined from the student's stated competencies and interests following competency-major linearity guidelines, and this relevant set was recorded in a spreadsheet. A recommended major is counted as a True Positive (TP) when it appears in the profile's relevant set and as a False Positive (FP) when it does not; a relevant major that the system fails to return within the Top-K is counted as a False Negative (FN). The TP, FP, and FN values were tabulated and verified manually in a spreadsheet for all 60 profiles, and micro-average aggregation was used to compute precision, recall, and F1-score.

TABLE V
EVALUATION METRIC FORMULA

Metric	Formula	Description
Precision	$\frac{TP}{TP+FP}$	Measures the proportion of recommended majors that are relevant.
Recall	$\frac{TP}{TP+FN}$	Measures the proportion of relevant majors that are successfully retrieved by the system.
F1-Score	$2x\frac{(Precision\ x\ Recall)}{(Precision + Recall)}$	Measures the balance between Precision and Recall.
MAP	$MAP = \left(\frac{1}{N}\right)\sum_{i=1}^{N} AP_i$	Mean of the average precision across query profiles; rewards placing relevant majors higher in the ranking.
MRR	$MRR = \left(\frac{1}{N}\right)\sum_{i=1}^{N} \left(\frac{1}{rank_i}\right)$	Mean reciprocal rank of the first relevant major; emphasizes the position of the first correct recommendation.
NDCG	$NDCG = \frac{DCG}{IDCG}$	Normalized discounted cumulative gain; discounts relevant majors that appear lower in the list.

Besides the classification metrics above, we also report three ranking-aware metrics that are common in recommender systems: Mean Average Precision (MAP), Mean Reciprocal Rank (MRR), and Normalized Discounted Cumulative Gain (NDCG). These metrics matter because, in real use, it is not enough for a relevant major to appear somewhere in the Top-K. What really helps a student is seeing the most suitable option near the top of the list. In other words, these metrics show how well the system orders its recommendations, not just whether the correct major is present. The formal definitions are summarized in Table V. We also measure usability with the System Usability Scale (SUS). After trying the prototype, the users filled in a short questionnaire, and the resulting score reflects how easy the interface was to understand and how comfortable the overall recommendation flow felt. This step is important because a system can be technically accurate but still fail in practice if students find it confusing to use.

IV. RESULTS AND DISCUSSION

A. Implementation Result

Streamlit was used in the Python implementation of the prototype. Sentence-transformers for creating embeddings, FAISS-CPU for similarity matching, NumPy and pandas for data processing and streamlit for the user interface are the main libraries used. The department database, department recommendations and system evaluation are the three main tabs that make up the user interface. Users can enter their skills and profile text, choose how many recommendations they want to provide, and optionally enter report-card grades on the recommendations tab. A summary of evaluation metrics is available on the evaluation tab. Users can view the stored data stored in the system by selecting the database tab.

Cards sorted by rank are used to display the recommended results. The name of the program, grades, a brief description, examples of universities, career opportunities, and an explanation of the matching criteria are all listed on each card. Because users can view the recommended study programs and understand the reasoning behind them, this design promotes transparency. For example, recommendations can be based on alignment with RPL competencies, programming-related keywords, or relevant academic requirements.

B. *User Interface Screenshots*

A web-based user interface was also included in the implemented prototype to facilitate communication between students and the recommendation system. To demonstrate that the proposed approach has been implemented as an interactive prototype and not merely as an algorithmic experiment, screenshots are included.

Recommendation metrics and usability evaluations remain the primary sources of validation, with screenshots serving as supporting evidence for the implementation.

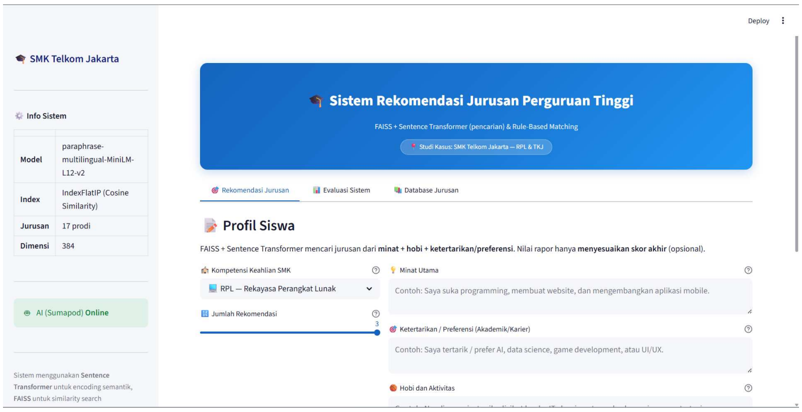

Fig. 3. Main Dashboard and Recommendation Page

Figure 3 shows the system's main dashboard. System details, including the embedding model, FAISS index type, and embedding dimensions, are displayed in the sidebar. Recommendations, system evaluations, and access to the major database are all available via the navigation tabs in the main area.

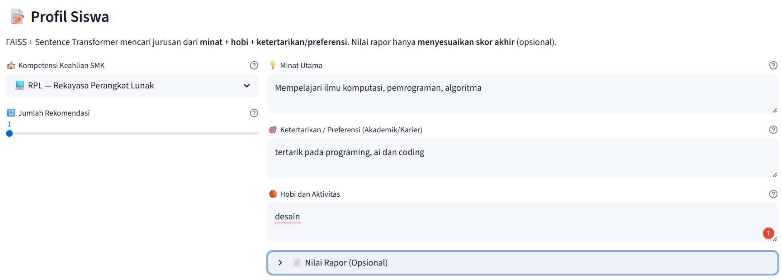

Fig. 4. Student Profile Input Form

The input form for the student profile is shown in Figure 4. Users can select vocational competencies, specify the number of recommendations they want, and enter a free-form description of their interests, academic and career preferences, and hobbies. After that, a query representation for semantic matching is generated from the input.

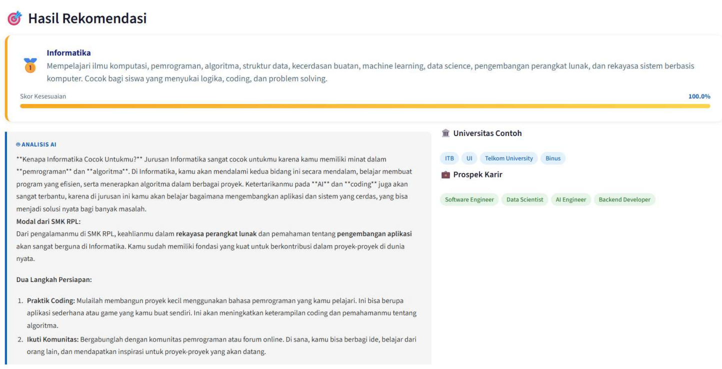

Fig. 5. Recommendation Result Card

Figure 5 shows the recommendation card generated by the system. The results include major recommendations, compatibility scores, brief summaries, examples of universities, career opportunities, and justifications for the recommended majors.

Fig. 6. Optional Report-Card Input

Figure 6 illustrates the optional report card grade entry feature. When this feature is enabled, the system will filter or modify recommendations based on academic eligibility using additional data from Math, English, and Physics grades during the post-processing stage.

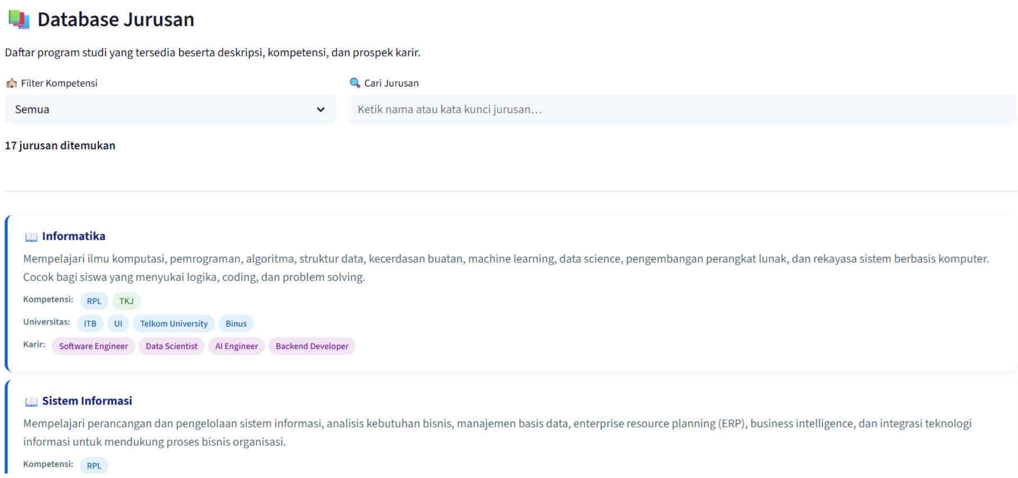

Fig. 7. Major Database Page

Fig. 7 shows the major database page. This page allows users to browse and search available majors, view descriptions, inspect relevant competencies, and check career prospects. This feature supports transparency because users can verify the data used by the recommendation system.

C. Evaluation Without Report-Card Input

In the first scenario, 44 labeled profiles without report-card scores were processed by the system, and each profile received Top-3 major recommendations. The system's output was compared with the ground-truth relevant majors, yielding 90 true positives, 42 false positives, and 42 false negatives, which correspond to a precision, recall, and F1-score of 0.6818. Because each profile has three ground-truth majors and receives three recommendations, precision and recall are numerically equal in this Top-3 setting.

These results indicate that the system retrieves a relevant major in most recommendation slots, although a substantial share of slots are still occupied by majors outside the ground-truth set. The balance between false positives and false negatives shows that the system both recommends some unsuitable majors and overlooks some suitable ones, which is the central challenge in major recommendation. Notably, the ranking-aware metrics are considerably higher than the flat F1-score: with an MRR of 0.962, the first recommendation is almost always relevant, and an NDCG of 0.736 confirms that relevant majors tend to appear near the top of the list.

These findings validate the two research objectives. Semantic embedding, FAISS search, and rule-based re-ranking were successfully integrated into a working architecture, and the system processes students' free-text interests rather than relying solely on fixed numerical criteria. The rule-based layer adds academic and vocational control on top of semantic similarity, which is reflected in the high MRR and NDCG values even when the flat F1-score is moderate.

TABLE VI

RECOMMENDATION QUALITY WITHOUT REPORT-CARD INPUT

Metric	Value	Interpretation
True Positive (TP)	90	Relevant majors successfully recommended
False Positive (FP)	42	Recommended majors that were not in the ground truth
False Negative (FN)	42	Ground-truth majors that were not recommended
Precision	0.6818	Share of recommended majors that were relevant
Recall	0.6818	Share of relevant majors that were retrieved

D. Evaluation With Report-Card Input

In the second scenario, a subset of 16 profiles with report-card grades was evaluated. Report-card grades were applied as an additional post-processing check rather than as the primary basis for recommendation; semantic similarity and rule-based matching still drive the main process. On this subset the system achieved a precision, recall, and F1-score of 0.7292, with an MRR of 1.000 and an NDCG of 0.794.

On this subset the scores are higher than the 0.6944 obtained on the full Top-3 set. This does not necessarily mean that report-card rules improve performance in general, because the 16 report-card profiles form a different and smaller subset whose interests are often closely aligned with technical majors. The report-card layer mainly acts as a feasibility filter: it can adjust or disqualify majors whose academic requirements are not met, which is most visible for majors with creative or non-technical descriptions. The effect of this layer should therefore be interpreted with the small subset size in mind.

Overall, the results suggest that semantic relevance and academic suitability must be balanced rather than treated as identical. Students may be strongly interested in a particular major, yet the system should still flag cases where academic requirements are not met. Report-card grades are therefore treated as supplementary evidence rather than a complete substitute for competency- and interest-based recommendation.

TABLE VII

RECOMMENDATION QUALITY WITH REPORT-CARD INPUT

Metric	Value	Interpretation
Precision	0.7292	Precision on the 16-profile report-card subset
Recall	0.7292	Recall on the 16-profile report-card subset
F1-Score	0.7292	Balanced quality on this subset
Main observation	Higher than 0.6944	Report-card rules act as a feasibility filter on a smaller, more technical subset

E. Baseline Comparison

To check whether the hybrid design is better than its single components, the proposed system was compared with two single-method baselines. Both baselines use the same major database and the same 60 profiles with the same ground truth. The first baseline is TF-IDF + Cosine Similarity, which represents every major description and student profile as TF-IDF vectors and ranks majors by cosine similarity without rule-based re-ranking[22]. The second baseline is Rule-Based Matching, which ranks majors using only competency linearity, interest keywords, and report-card requirements, without semantic embeddings. Table VIII shows the Top-3 results on all 60 profiles. The proposed hybrid gives the best ranking quality, with the highest MAP (0.657), MRR (0.972), and NDCG (0.752), so the most suitable major is placed at the top of the list more often than in the two baselines. For the flat F1-score, the hybrid (0.6944) is close to the rule-based baseline (0.7000, a difference of only one correct recommendation out of 180) and clearly higher than the TF-IDF + Cosine baseline (0.6222). This shows that the rule-based part is the main reason for the raw retrieval accuracy, while the semantic part (Sentence-Transformers and FAISS) mostly improves the order of the recommendations and handles free-text interests that strict keyword rules cannot capture, as seen in the higher MRR and NDCG. The TF-IDF baseline gives the lowest result because it only matches words, not meaning. Therefore, the hybrid combines the accuracy of explicit rules with the semantic understanding and better ranking of embedding-based search. The two baselines were re-implemented inside the same environment as the proposed system rather than taken from previously published studies, so this comparison is an internal ablation that isolates the contribution of each component, not an external benchmark against other works.

TABLE VIII
BASELINE COMPARISON ON THE TOP-3 SCENARIO (N=60)

Method	Precision	Recall	F1-Score	MAP	MRR	NDCG
TF-IDF + Cosine Similarity	0.6222	0.6222	0.6222	0.5880	0.9194	0.6833
Rule-Based Matching only	0.7000	0.7000	0.7000	0.6537	0.9056	0.7364
Proposed Hybrid (this study)	0.6944	0.6944	0.6944	0.6574	0.9722	0.7517

F. *Error Analysis*

Even though the Top-3 configuration reached an F1-score of 0.6944, the evaluation still produced 55 false positives and 55 false negatives, which were analyzed to understand the system's errors. Table IX summarizes the main error types. The most frequent false positives were Manajemen Informatika, Rekayasa Perangkat Lunak, Sains Data, and Teknik Komputer. These are broad and popular majors whose descriptions are semantically close to many profiles, so the system recommends them too often[23]. The most frequent false negatives were Informatika, Teknologi Informasi, and Sistem Informasi. These relevant majors were pushed below the Top-3 because broader and semantically similar majors were ranked higher. This shows that the keyword and competency-linearity rules tend to favor generic majors, while the strict keyword filter and the report-card check sometimes remove valid candidates. These parts are the main targets for future improvement.

TABLE IX
ERROR ANALYSIS OF MISCLASSIFIED RECOMMENDATIONS (TOP-3)

Error Type	Most Frequent Majors (count)	Likely Cause
False Positive	Manajemen Informatika (13), Rekayasa Perangkat Lunak (8), Sains Data (7), Teknik Komputer (7)	Broad, popular majors that are semantically close to many profiles are over-recommended
False Negative	Informatika (14), Teknologi Informasi (10), Sistem Informasi (8)	Relevant majors outranked by broader, semantically similar majors or removed by strict keyword / report-card filters

G. Ranking-Aware Evaluation Metrics

Because the classification metrics treat the Top-K set as unordered, ranking-aware metrics were also calculated to measure the quality of the recommendation order. Table X shows the MAP, MRR, and NDCG for the Top-

3 scenario. The high MRR (0.972 overall and 1.000 on the report-card subset) shows that the first recommendation is almost always relevant, while the NDCG of 0.752 shows that relevant majors are placed near the top of the list. These metrics give more information than the flat F1-score, because in practice students usually look at the first one or two recommendations.

TABLE X
RANKING-AWARE EVALUATION RESULTS (TOP-3)

Scenario	MAP	MRR	NDCG
Without report card (N=44)	0.641	0.962	0.736
With report card (N=16)	0.701	1.000	0.794
Overall (N=60)	0.657	0.972	0.752

H. Recommendation Trade-off and Top-K Behavior

The study also examined the effect of the number of recommendations K. Because the system surfaces the most relevant majors first, a smaller K typically yields higher precision but lower recall. The comparison in Table XI confirms this trade-off on the full set of 60 profiles: the Top-2 configuration reaches a precision of 0.80 but a recall of only 0.533 (F1 = 0.64), whereas the Top-3 configuration is balanced, with precision = recall = F1 = 0.6944. The ranking metrics remain high in both settings (MRR around 0.97), indicating that the first recommendation is almost always relevant regardless of K.

The Top-3 approach is considered appropriate for this use case, as students usually need options. There is no standard way of choosing a major at the present time. Students may choose from a variety of related programs, including software engineering and computer science. Students will have a limited number of choices presented to them, i.e. the three recommendations. The explanations provided for each recommendation make it easier for students to compare these options.

TABLE XI
COMPARISON OF TOP-2 AND TOP-3 CONFIGURATIONS (N=60)

Configuration	Precision	Recall	F1-Score	MAP	MRR	NDCG
Top-2	0.800	0.533	0.640	0.792	0.967	0.834
Top-3	0.694	0.694	0.694	0.657	0.972	0.752

I. Usability Evaluation

The usability evaluation was done with 29 respondents using the System Usability Scale. The average SUS score was 69.57 This score suggests that the prototype is both understandable and usable, although there is room for improvement. The result is quite satisfactory, since the system already offers a structured interface, recommendation cards and database inspection features, but users may still need more explicit instructions, simpler language and better visual hierarchy. The number of respondents follows the same scoping decision: the usability evaluation was deliberately limited to students from the two targeted programs (RPL and TKJ) to keep the study focused and time-efficient, and because not all of them completed the questionnaire, the 29 valid responses represent the realistically obtainable sample within this scope.

The system benefits from its simple input structure from a user experience point of view. Students just need to pick their competence and write about their interests, likes and hobbies. You can optionally enter a report-card input, and this is in an expandable area, so students who do not want to use this can still get recommendations. The database tab also offers more transparency, as students can see available majors and see the information that the system is using.

TABLE XII
SUMMARY OF USABILITY EVALUATION

Aspect	Result	Meaning
Respondents	29 users	Usability questionnaire participants
Method	System Usability Scale	Measures perceived ease of use
Average score	69.57	Prototype is usable but can be improved
Improvement direction	Guidance, wording, visual layout	Enhances student understanding and confidence

J. Discussion of Strengths and Limitations

The system's main strength is its combination of flexible semantic understanding and explicit domain control. Sentence-Transformers allow the system to interpret students' interests in the form of free-text, while FAISS facilitates an efficient search process. In an educational setting, rule-based matching improves interpretation and relevance. Relying only on statistical rules or semantic similarity is not as appropriate for research problems as this combination.

Compared with prior work, major and study-program recommendation has typically relied on single methods or classical hybrids. A prior study combined Naïve Bayes and item-based collaborative filtering and reported a mean absolute error of 0.0385 [24]. On the semantic side, another study applied Sentence-BERT and FAISS for retrieval in an academic repository and obtained a MAP of about 0.39 without a rule-based layer [25]. The hybrid system proposed in this study reached MAP = 0.657, MRR = 0.972, and NDCG = 0.752. Although the different datasets mean these figures cannot be compared directly, the contrast supports the argument that fusing semantic search with explicit domain rules contributes meaningfully to ranking quality relative to a purely semantic approach.

However, the system is not without limitations. The major database contains only the dataset generated for this study; increasing the number of majors, universities, and job descriptions would improve recommendation coverage. The evaluation set, although enlarged to 60 labeled profiles (44 without and 16 with report-card input), is still relatively small, and the usability study involved 29 respondents, so the results should be interpreted as a proof of concept. The rule-based component requires complete domain rules and keyword lists, and the keyword layer may fail to capture meaning when a student uses unusual expressions. Finally, the report-card feature is optional and simplified and should be strengthened before being used as a rigorous academic-eligibility mechanism.

Future work could expand and diversify the labeled dataset and the number of usability respondents, add stronger single-method and learning-based baselines through controlled ablation, and incorporate feedback-based learning. The ground truth could be validated by multiple domain experts to measure inter-rater agreement, and the rule design could be refined to reduce the false positives and false negatives identified in the error analysis. The system could also be extended to vocational competencies beyond RPL and TKJ, and an explanation module could be added to generate clearer natural-language reasoning while keeping the explanations grounded in the rule and similarity results.

V. Conclusion

In this study, Sentence-Transformers, FAISS, and rule-based matching were integrated to create a hybrid college major recommendation system specifically for vocational high school students. This system uses FAISS to retrieve candidate majors, applies education domain rules to re-rank the candidates, and represents student profiles and major descriptions as semantic embeddings. Python and Streamlit were used to implement a prototype for students, specifically those in the RPL and TKJ programs at Telkom Vocational High School in Jakarta.

The evaluation shows that the proposed hybrid strategy generates relevant recommendations. On 60 labeled profiles, the Top-3 configuration achieved a precision, recall, and F1-score of 0.6944 (0.6818 without report-card input and 0.7292 with report-card input), together with MAP = 0.657, MRR = 0.972, and NDCG = 0.752. A Top-2 configuration traded recall for a higher precision of 0.80. The usability evaluation produced an average SUS score of 69.57, indicating that the prototype is understandable and usable, although the user interface still needs improvement.

Overall, the study shows that students can choose their university majors more methodically when semantic retrieval and rule-based educational constraints are combined. The system can offer a preliminary recommendation and explanation to facilitate discussion, but it should not take the place of the judgment of educators or counselors. The main database should be expanded in the future, academic rules should be improved, more users and experts should be involved in validation, and the user interface should be improved for wider deployment.

Acknowledgment

The author would like to thank Telkom University, the supervisors, and all respondents who contributed to the evaluation of the prototype. The author also thanks SMK Telkom Jakarta as the educational context used in this research.

References

[1] V. Nayak and N. Vora, “A Machine Learning-based Career Recommendation,” *Journal of Trends in Computer Science and Smart Technology*, vol. 6, no. 4, pp. 374–390, Dec. 2024, doi: 10.36548/jtcsst.2024.4.004.

[2] T. Amelia, Prajoko, and A. Pambudi, "Rekomendasi Jurusan Kuliah Berdasarkan Minat dan Kemampuan Menggunakan Metode Content Based Filtering," Technologia: Jurnal Ilmiah, vol. 14, no. 3, pp. 245–253, 2023, doi: 10.31602/tji.v14i3.11336.

[3] C. Lahoud, S. Moussa, C. Obeid, H. El Khoury, and P. A. Champin, “A comparative analysis of different recommender systems for university major and career domain guidance,” *Educ. Inf. Technol. (Dordr).*, vol. 28, no. 7, pp. 8733–8759, Jul. 2023, doi: 10.1007/s10639-022-11541-3.

[4] D. Ramadhan and L. Marlinda, “Decision Support System for selecting study programs using the AHP method,” *Sinkron*, vol. 7, no. 4, pp. 2547–2555, Oct. 2022, doi: 10.33395/sinkron.v7i4.11821.

[5] L. Guder, J. P. Aires, H. U. da Silva, F. Meneguzzi, and D. Griebler, "Sentence Representations for Semantic Textual Similarity: A Systematic Review," Computer Speech & Language, vol. 100, Art. no. 101970, 2026, doi: 10.1016/j.csl.2026.101970.

[6] M. Douze et al., "The FAISS Library," IEEE Transactions on Big Data, 2025, doi: 10.1109/TBDATA.2025.3618474.

[7] H. Jégou, M. Douze, and C. Schmid, “Product quantization for nearest neighbor search,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 1, pp. 117–128, 2011, doi: 10.1109/TPAMI.2010.57.

[8] M. Guesmi, M. A. Chatti, L. Kadhim, S. Joarder, and Q. U. Ain, “Semantic Interest Modeling and Content-Based Scientific Publication Recommendation Using Word Embeddings and Sentence Encoders,” *Multimodal Technologies and Interaction*, vol. 7, no. 9, Sep. 2023, doi: 10.3390/mti7090091.

[9] T. C. Uzoma, C. O. Ikerionwu, M. E. Nwanga, and C. Etus, “Machine Learning Recommender System for an Enhanced Students Course Selection,” *International Journal of Computer Trends and Technology*, vol. 72, no. 10, pp. 31–35, Oct. 2024, doi: 10.14445/22312803/ijctt-v72i10p106.

[10] A. Jadon and A. Patil, "A Comprehensive Survey of Evaluation Techniques for Recommendation Systems," arXiv preprint arXiv:2312.16015, 2024.

[11] N. Kamal, F. Sarker, A. Rahman, S. Hossain, and K. A. Mamun, “Recommender System in Academic Choices of Higher Education: A Systematic Review,” *IEEE Access*, vol. 12, pp. 35475–35501, 2024, doi: 10.1109/ACCESS.2024.3368058.

[12] K. Crandall, K. S. Crandall, and M. North, "A Decade Perspective of Recommender Systems Research: A Systematic Review," *Issues in Information Systems*, vol. 25, no. 3, pp. 496–507, 2024.

[13] M. Aucancela, A. González Briones, and P. Chamoso, "Educational Recommender Systems: A Systematic Literature Review," in The Barcelona Conference on Education 2023: Official Conference Proceedings, 2023, pp. 933–951, doi: 10.22492/issn.2435-9467.2023.74.

[14] F. L. da Silva, B. K. Slodkowski, K. K. A. da Silva, and S. C. Cazella, “A systematic literature review on educational recommender systems for teaching and learning: research trends, limitations and opportunities,” *Educ. Inf. Technol. (Dordr).*, vol. 28, no. 3, pp. 3289–3328, Mar. 2023, doi: 10.1007/s10639-022-11341-9.

[15] K. Khatimah and M. A. Gustalika, “Implementation of A Decision Support System for Major Selection using AHP and TOPSIS Method,” *Building of Informatics, Technology and Science (BITS)*, vol. 6, no. 1, Jun. 2024, doi: 10.47065/bits.v6i1.5256.

[16] S. El-Keiey, D. ElMenshawy, and E. Hassanein, "Career Recommendation Based on Feature Selection for Undergraduate Students Using Machine Learning Techniques," International Journal of Advanced Computer Science and Applications (IJACSA), vol. 16, no. 3, pp. 234–244, 2025, doi: 10.14569/IJACSA.2025.0160323.

[17] Y. Han, C. Liu, P. Wang, J. Huang, and B. Li, "A Comprehensive Survey on Vector Database: Storage and Retrieval Technique, Challenge," arXiv preprint arXiv:2310.11703, 2023, doi: 10.48550/arXiv.2310.11703.

[18] H. Liu, A. Gegov, and M. Cocea, “Rule-based systems: a granular computing perspective,” *Granular Computing*, vol. 1, no. 4, pp. 259–274, Dec. 2016, doi: 10.1007/s41066-016-0021-6.

[19] M. Siino, I. Tinnirello, and M. La Cascia, “Is text preprocessing still worth the time? A comparative survey on the influence of popular preprocessing methods on Transformers and traditional classifiers,” *Inf. Syst.*, vol. 121, Mar. 2024, doi: 10.1016/j.is.2023.102342.

[20] Z. Miao, Q. Wu, K. Zhao, Z. Wu, and Y. Tsuruoka, "Enhancing Cross-lingual Sentence Embedding for Low-resource Languages with Word Alignment," in Findings of the Association for Computational Linguistics: NAACL 2024, 2024, pp. 3225–3236, doi: 10.18653/v1/2024.findings-naacl.20.

[21] M. Danish, S. I. Ansarullah, S. Khan, and S. Alshmrany, “A Sentiment-Aware and Explainable Hybrid Recommender System Based on Ratings and Transformer Embeddings,” *International Journal of Computational Intelligence Systems*, vol. 19, no. 1, Dec. 2026, doi: 10.1007/s44196-026-01305-z.

[22] M. Abbad, R. Alfath, and A. N. Rohman, "Content-Based Filtering Recommendation System for E-Commerce Products Using Sentence-BERT and Cosine Similarity," Journal of Applied Informatics and Computing, 2026.

[23] A. Klimashevskaia, D. Jannach, M. Elahi, and C. Trattner, "A survey on popularity bias in recommender systems," *User Model. User-adapt. Interact.*, vol. 34, no. 5, pp. 1777–1834, Nov. 2024, doi: 10.1007/s11257-024-09406-0.

[24] J. Aisyiah, M. Risnasari, and A. T. Ni'mah, "Sistem Rekomendasi Program Studi Menggunakan Metode Hybrid Recommendation (Studi Kasus: MAN Sumenep)," Journal of Education and Informatics Research, vol. 4, no. 1, 2023.

[25] I. Lubis, H. Lubis, and I. Nur Wahidah, "Implementation of Semantic Search in an Academic Repository Using Sentence-BERT and FAISS," *sinkron*, vol. 10, no. 2, pp. 1060–1069, Apr. 2026, doi: 10.33395/sinkron.v10i2.15940.