Computer Science Research, Embedded Systems and Logic
p-ISSN: 2303-3304, e-ISSN: 2620-3553
Vol. 14 (1): 51–64 (March 2026)
The DOI 10.33558/piksel.v14i1.12078

Skill Recommendation System Using User-Based Collaborative Filtering Method

Fadli Yandra [1,*] , Muhammad Iqbal Fadillah [1] , Hari Soetanto [1]

*Corresponding Author: e-mail: 2311600684@student.budiluhur.ac.id

[1] Magister of Computer Science; Budi Luhur University; Jl. Ciledug Raya No.99, Petukangan Utara, Kec. Pesanggrahan, Kota Jakarta Selatan, Daerah Khusus Ibukota Jakarta 12260; e-mail: 2311600684@student.budiluhur.ac.id 2111600801@student.budiluhur.ac.id hari.soetanto@budiluhur.ac.id

Submitted : **30/12/2025**
Revised : **27/01/2026**
Accepted : **30/01/2026**
Published : **31/03/2026**

Abstract

The rapidly evolving job landscape in the digital era requires job seekers to continuously adapt to emerging skills driven by technological advancements across various industries. However, many job seekers struggle to keep up with these changing skill demands, while existing job portals often lack features that recommend relevant skills. To address this issue, this study proposes a skill recommendation system based on the User-Based Collaborative Filtering approach, which considers similarities between users' preferences. Two similarity measurement methods, Log-Likelihood Similarity and Cosine Similarity, are applied and compared to evaluate their effectiveness. The system matches user skill profiles with skill requirements extracted from job vacancy data, where job postings are also treated as user representations. The dataset was collected from the Jobstreet job portal using web scraping techniques, ensuring relevance to real-world job market conditions. The system performance was evaluated using the Hit Rate Matrix. The results show that the Log-Likelihood Similarity method achieved a hit rate of 0.73, outperforming the Cosine Similarity method, which obtained a score of 0.51. This indicates that Log-Likelihood Similarity provides more accurate and relevant skill recommendations. Overall, the proposed system demonstrates the potential to assist job seekers in identifying relevant skills aligned with current market demands, thereby supporting better career decision-making in a competitive and dynamic job environment.

Keywords: *Skill recommendation system, User-Based Collaborative Filtering, Hit Rate Matrix, Log-Likelihood Similarity, Cosine Similarity.*

1. Introduction

The rapid development of information technology has brought many benefits to everyday life, especially in facilitating quick and abundant access to information. One form of this progress is the availability of the internet as a huge source of data, allowing users to obtain various types of information according to their needs. However, there is too much information available on the internet. Therefore, it takes a long time to find the information that is needed. One example is in the field of job skills search (Muzdalifah et al., 2022). Specifically, in the field of information technology skills for software development, the skills required continue to evolve in line with technological developments. Job seekers often find it difficult to keep up with these developments.

As a result, their skills may become outdated compared to industry demands. This becomes a problem when eventually must search for new jobs. Currently there are many new jobs and skills are created, while some jobs become obsolete due to technological changes and so many software engineering graduates face a skills gap due to the mismatch between what is taught in educational institutions and what is needed by the industry (Garousi, 2020). This shows the importance of access to the latest information on the skills needed in the world of work. Based on data from the Central Statistics Agency (BPS), the national labor force in August 2024 was 152.11 million people, an increase of 4.40 million people compared to August 2023. There were 630,672 registered job vacancies and 509,237 registered job seekers. With such a large workforce, registered job vacancies, and registered job seekers, it is necessary to conduct a specific search for job vacancies that match the skills of job seekers and to learn the latest skills required (Statistik, 2024).

The search for job vacancies is currently also rapidly developing into online job searches through job search portals such as Jobstreet, JobsDB, and LinkedIn. With so much data on job vacancies and the skills required, it can be difficult for job seekers to stay up to date with the latest skills needed in a world of work that is constantly changing in line with technological developments. Due to the vast amount of information available, effective and relevant information searching has become crucial. Manual searching can be difficult and time-consuming, especially when searching for skills that job seekers need to learned (Shamsuddin et al., 2023).

To overcome this problem, a recommendation system can be a solution for conducting relevant searches for job seekers to update their skills. The recommendation system is designed to provide suggestions or recommendations to users based on information collected from user profiles and searched for similarities with job vacancy data profiles. Reference from (Suhada et al., 2023) stated that in recommendation systems, Collaborative Filtering is one of the most commonly used algorithms. Their research results stated that User-based Collaborative Filtering using Cosine Similarity calculations can be applied and produce 10 product recommendations with an RMSE of 0.9 (Al-otaibi et al., 2022).

The User-based Collaborative Filtering (Juan et al., 2019) approach provides recommendations to users based on items that are liked or considered to have similarities. Additionally, reference (Ro et al., 2021) stated that in their research, a job recommendation system using Content Based Similarity achieved an accuracy of 81% with a precision value of 0.77, recall of 0.81, and an F1-score of 0.76. By building a job recommendation system (Feng et al., 2021), users will find it easier to discover relevant job information.

Reference (’Alim, A. Solichin, A. and Painem, 2020) state that the User-based Collaborative Filtering method is one of the memory-based techniques in the Collaborative Filtering category. In the User-Based method, calculations are made based on the level of compatibility between users. The approach to users works by finding the level of similarity between users and other users who have the same preferences. The Log-Likelihood Similarity method is used to calculate similarity (Khusna et al., 2021; Suhada et al., 2023). The Log-Likelihood Similarity method is suitable because it does not require information in the form of preferences, making it suitable for use in skill recommendation searches.

This approach assumes that if something is liked by similar users, it will also have a high chance of being liked by people with similar interests. Previous studies have not compared similarity methods. To differentiate itself from existing studies, this study compares the Log-Likelihood Similarity and Cosine Similarity methods to find the best methods (Baruah et al., 2023; Mahendra et al., 2024). Based on the description and previous research explained above, this study analyzes and tests a recommendation system that provides suggestions on skills that job seekers need to learn in order to stay up to date with the developments in the world of work. The recommended skill (Pari et al., 2021) will take into account the skills that the user already has, so that they will be more relevant to the user. The recommendation system to be developed in this study uses the User-based Collaborative Filtering method with the Log-Likelihood Similarity and Cosine Similarity calculation methods. The use of these two similarity calculation methods aims to compare and deepen the research results.

2. Research Method

Methodology refers to the systematic steps or sets of methods used to solve a problem. This chapter explains how the research was conducted, including the stages of research that must be undertaken to create a skill recommendation systems (Barranco & Martínez, 2025) using methods and analyses appropriate to the problem at hand, and how to obtain the data needed to support the research.

2.1 Research Method

Research methods, also known as scientific methods, are procedures or steps in obtaining scientific knowledge, so that research methodology is a systematic way of compiling a scientific study to solve problems and test research hypotheses. In this study, an experimental research method was used to apply the User-Based Collaborative Filtering technique (Mondi & Wijayanto, 2019) to the recommendation process that will be carried out by the system, then create a web-based application that can recommend skills. The following is an overview of the process carried out by the system that will be developed in this study. Figure 1 shows the general research process.

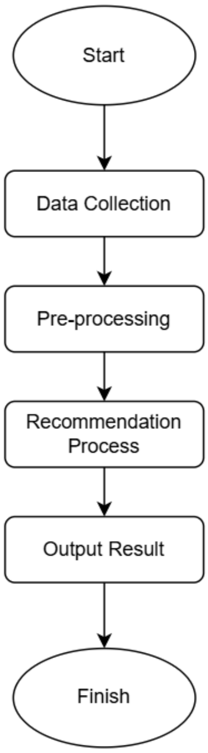

Figure 1. Conceptual Framework

2.2 Sampling Selection Method

This research is based on a sample recommendation system, referring to job vacancy data collected from the Jobstreet portal (Jobstreet, 2025). The data used is job vacancy information in the field of information technology, specifically in the field of software development support. Data selection was carried out by considering its suitability and relevance to the research objectives. The criteria for the data used include:

a) Job vacancies in the field of information technology, particularly in the field of software development support, namely Software Engineering.
b) Each job vacancy entry must have complete information about the job title, job description, and skills required for the position.

c) More than a thousand data points were collected to ensure diversity in the positions and skills required by the industry.

2.3 Data Structure and Processing

The scraped data was stored in tabular form and then entered into a relational database. The skill information contained in the job descriptions was extracted through a preprocessing process. The data structure included several main tables, as follows:

a) Skills table: Stores a list of unique skills required.
b) Jobs table: Stores job positions along with their titles and descriptions.
c) Job_skills table: Stores the relationship between job positions and the skills required

2.4 Data Collection Method

The data used in this study is derived from secondary data. Secondary data is data obtained by researchers from existing sources. Data collection was conducted through observation of the jobstreet.co.id website. Data collection on the Jobstreet website was used to obtain data in the form of information on skills, positions, companies, URLs, and content. Data collection was carried out starting in January 2025, resulting in 2,612 job vacancy data. The collection of information on the Jobstreet website was intended to serve as a reference for data on skills and positions. The following is an explanation of the data collection process on the Jobstreet website.

The data collection process on the Jobstreet website was carried out using web scraping technique (Flores et al., 2020). The web scraping process used the Python programming language. The web scraping process on the Jobstreet website is explained as follows:

a) Accessing the Jobstreet website, the process begins by accessing the Jobstreet portal URL at https://id.jobstreet.com/id/ using the Python programming language and the BeautifulSoup library.
b) Collecting information, this process retrieves web page content from job vacancy URLs in the form of the HTML of that page. Next, using the BeautifulSoup library, it extracts important information from each job card, then loops through each page to retrieve job vacancy data and combines all job vacancy data into a list. The scraped results are then converted into a Pandas data frame for analysis and storage. The scraped data is saved in Excel format for subsequent storage in a database.
c) Storing data in the database, after obtaining the data in Excel format, it is converted into a CSV file containing the job vacancy results and skills. The next step is to insert the data into the database table. Job data is inserted into the jobs table, skill data is inserted into the skills table, and the relationship between jobs and skills is inserted into the job-skills table.

2.5 Preprocessing Process

The preprocessing process is the process of preparing data, such as cleaning data noise. This process is carried out on the results of scraping the Jobstreet website after data collection is complete. The initial stage of the job vacancy preprocessing process is to retrieve all job vacancy information at the data collection stage, followed by preprocessing to clean the data of noise in the form of HTML tags, to be converted into plain text and saved as an Excel file. In this study, skill data extraction from job descriptions was performed using a specially designed Python script. The implementation utilizes DeepSeek's Large Language Model (LLM) via API to process text and identify skills. The Large Language Model (LLM) identifies and separates relevant skill keywords, such as Python, Java, React.Js, and various other skills that represent the skills required in job vacancies. The script workflow begins by reading an Excel file containing a dataset of job descriptions obtained from web scraping. This data is then iterated row by row. Each job description is sent to the DeepSeek API with specific instructions via a prompt that instructs the model to act as a skill parser (Deepseek, 2025).

The model is asked to return the output in pure Json format, which contains a skill data array (Bai et al., 2019). The extraction results for each row are immediately recorded in a CSV file. The result is a new dataset in CSV format, where each job description entry now has an additional column containing the list of extracted skills. The job_skills data table will be used to perform a suitable recommendation search based on a user-based approach using the Log-Likelihood Similarity and Cosine Similarity calculation techniques. In the recommendation search process, the application will read the job_skill table and the user_skills table to obtain existing skill data, then calculate the level of similarity between user skills and the skills listed in the job vacancy using the Log-Likelihood Similarity and Cosine Similarity methods. These two similarity calculation methods are used respectively to calculate the comparison of the evaluation result (Elahi & Zirak, 2024). The results are then sorted based on the skills in job vacancies that are most similar to the user, according to the highest similarity score. Skills that the user does not yet possess but are possessed by the job vacancies most similar to the user will be used as skill recommendations.

2.6 Conceptual Framework

This research proposes a conceptual framework for problem solving by building a recommendation system using the User-Based Collaborative Filtering method, comparing the Log-Likelihood similarity calculation with the Cosine Similarity calculation. The comparison of these similarity calculations aims to deepen the research results. The first step begins with identifying and reviewing the problems, followed by searching for literature references to select the methods to be used. After deciding on the method to be used, the next step is to collect the dataset and perform data preprocessing. Once the data requirements are met, a prototype is developed for testing and analysis of the results.

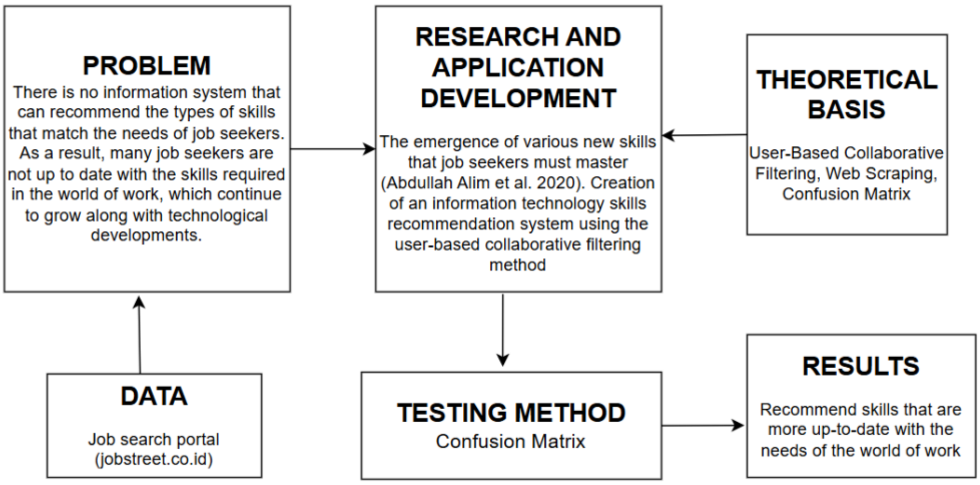

Figure 2. Conceptual Framework

2.7 User Based Collaborative Filtering

User-Based Collaborative Filtering is an approach in memory-based recommendation systems that provide recommendations to users based on similarities in preferences with other users. This approach works by identifying a group of users (neighborhood) who have similar rating or interaction patterns toward a number of items, then using information from that group to predict items that may be liked by the target user. (Khusna et al., 2021) state that the User-Based Collaborative Filtering approach provides recommendations to users based on items liked or rated by other users who have similarities. The advantage of this system is that it can produce better recommendations. According to (Februariyanti et al., 2021), User-Based Collaborative

Filtering is intended to find interesting items by using a method that is suitable for specific users, namely by searching for other users who have similar interests or desires. At the beginning, User-Based Collaborative Filtering will be able to find the closest users (user neighbors) by finding similar users (user similarity), then the rating values obtained from the closest users will be used as recommendation material. According to (Ko et al., 2022) the User-Based Collaborative Filtering approach is one of the earliest models in the development of recommendation systems. This model compares user evaluation data for the same item, then compiles a list of recommendations based on the rating values of similar user groups. Although this method has proven to be effective, especially in the context of online services such as e-commerce.

2.8 Application Design

The application designed in this study was developed using the React.js programming language, Python, and the PostgreSql database. In this study, a portal was developed to present information and input information in the form of user profiles and skills already possessed by users. The recommendation process will be carried out by an engine running in the background. Figure 3 shows the developed application.

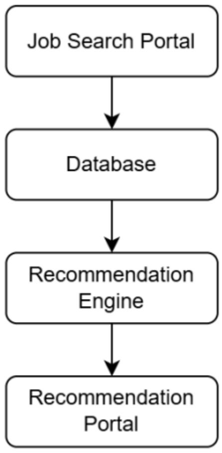

Figure 3. Application Architecture

The recommendation system in this study was built using the User-Based Collaborative Filtering method, which is a method in a recommendation system that works by finding similarities between users based on their preferences or interaction history, such as their skills. In the context of this study, users are defined as job seekers, while job vacancies are represented as users who have a list of skills as attributes. Thus, the system matches real users and job vacancies based on similarities in skills, in order to recommend skills that users do not yet have but are required by similar job vacancies. To measure the similarity between users and job vacancies (which are positioned as users), two similarity calculation methods are used, namely Log-Likelihood Similarity and Cosine Similarity (Permata et al., 2025). Log-Likelihood Similarity measures the strength of the association between two users based on the probability of having skills in common or not. This method is suitable for binary data and does not require explicit preference weights, while Cosine Similarity measures the extent to which the directions of two vectors are similar, without considering their lengths.

The recommendation process begins with the extraction of user and job vacancy data collected through web scraping. Job vacancy data is taken from the Jobstreet portal and user data is taken from LinkedIn. The skills possessed by users and job vacancies are converted into binary vectors, and the system calculates the similarity between users and each job vacancy using Log-Likelihood Similarity or Cosine Similarity.

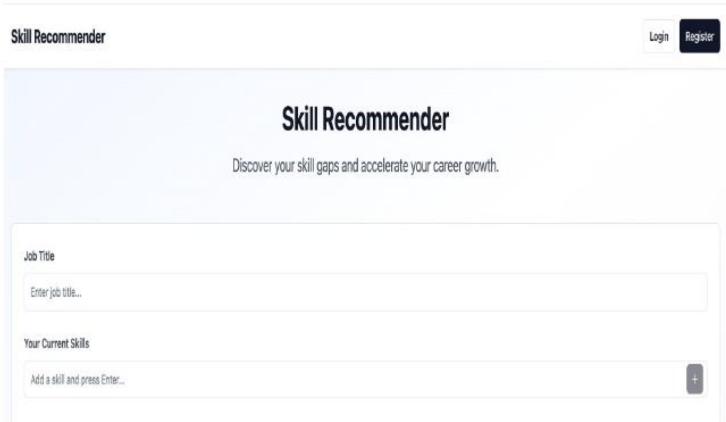

Figure 4. Prototype of the Skill Recommendation System Application

The system identifies skills possessed by job vacancies but not yet possessed by users. The selected skills are provided as suggestions for skills that users need to learn. User-Based Collaborative Filtering is suitable for use when explicit data in the form of ratings is not available, but only binary data about possessed skills is available. Therefore, this approach is capable of generating more contextual skill suggestions.

3. Results and Analysis

Data collection began with web scraping on the Jobstreet website using the Python programming language and the Beautiful Soup library with Google Colab tools. Python was chosen for its flexibility in processing text and data from the web. Meanwhile, the Beautiful Soup library was used because of its ease in extracting data from HTML structures. The aim was to retrieve data in the form of job titles, job detail links, companies, locations, and job details. The following is Table 1, which shows the results of web scraping using the Python programming language with the Beautiful Soup library.

Table 1. Example of Preprocessing Output Table

Id	Job_title	Job_detail_link	Company	Locations	Extracted_skills
1	Frontend Engineer (React.js/Next.js)	https://...	Geniebook Pte. Ltd.	Surabaya	React.js', 'Next.js', 'RESTful API integration', 'Crossbrowser compatibility', 'Mobile responsiveness', 'Clean code
2	AI Trainer - Frontend Developer (ID)	https://...	Ebit Co., Ltd	West Java	HTML, CSS, JavaScript, React, Next.js, 'Vue', 'UI/UX', 'Responsive design', 'Accessibility', 'Front-end development'
3	Front End Developer (PHP Programmer)	https://...	Ebit Co., Ltd	Greater Jakarta	PHP', 'MVC', 'Laravel', 'CodeIgniter', 'Angular', 'HTML', 'JavaScript', 'Node.js', 'CSS', 'OOP', 'SQL

Id	Job_title	Job_detail_link	Company	Locations	Extracted_skills
4	Front End React JS Next JS Developer	https://...	Steradian Data Optima	South Jakarta	HTML, CSS, 'JavaScript', 'NodeJS', 'Vue.js', 'NextJS', 'ReactJS', 'npm', 'Yarn', 'Responsive design
5	Front-End Programmer	https://....	PT Trimitra Putra Mandiri	South Jakarta	HTML5, CSS3, 'JavaScript', 'React', 'Vue.js', 'Angular', 'Responsive web design', 'RESTful APIs', 'Git'
6	Frontend Developer (Web / Mobile)	https://...	PT Insura Media Solusi (Jakarta)	Greater Jakarta	'Frontend Development', 'Angular', 'Vue', 'React', 'Java', 'Kotlin', 'Flutter', 'Swift', 'Web Development
7	Junior Programmer / Developer	https://....	PT Datacaraka Solusindo	West Jakarta	.NET, NodeJS, MVP development', 'NoSQL', 'PostgreSQL', 'NestJS', 'Express', 'NextJS', 'React

Next, skills are selected for recommendation purposes related to the job_skills table. Table 2 shows the skill table.

Table 2. Skill Table

skill_id	skill_name
1	redis
2	production-ready web interfaces
3	scalable systems
4	jquery
5	react
6	rest api
7	website deployment
8	mongodb
9	kotlin
10	swift

Next, a job_skills table is created containing job_id and skill_id. This table is related to the job table and the skill table. Table 3 shows the job_skills table used to obtain skill recommendations.

Table 3. Job Skill Table

job_id	skill_id
1	15
1	2
1	17
1	18
1	21
1	23
1	26
2	27
2	28
2	29
2	30

With this relational table, each job_id has all the skills with their skill_id according to the web scraping data result (Prehanto et al., 2018). Furthermore, the job_id in the job_skills table is also related to the job_id in the jobs table. The recommendation process generates skills that the system recommends for users to learn. These recommended skills are suitable for complementing the skills already possessed in the user's profile [15]. The system compiles the recommendation results in JSON format to be displayed in the application. The system records each recommendation process in the audit_history table, which records:

a) User ID
b) Request time
c) Recommendation results in JSON format

Figure 5 shows the skill recommendation application interface using the log-likelihood similarity calculation, which generates recommended skills and compares job vacancies whose skills match the user's, complete with scores and percentages.

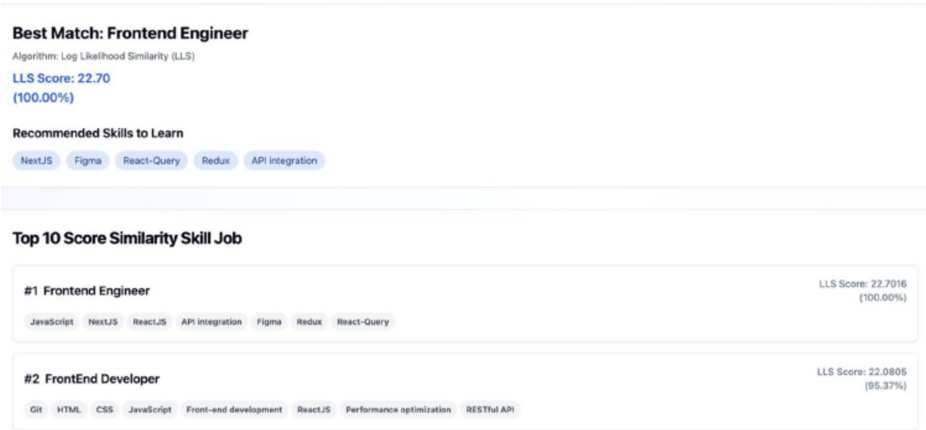

Figure 5. Skill Recommendation Result Using the Log-likelihood Similarity Method

Figure 6 shows the skill recommendation application display using the cosine similarity calculation method, which generates skill recommendations and compares job vacancies with the user's skills, complete with scores and percentages.

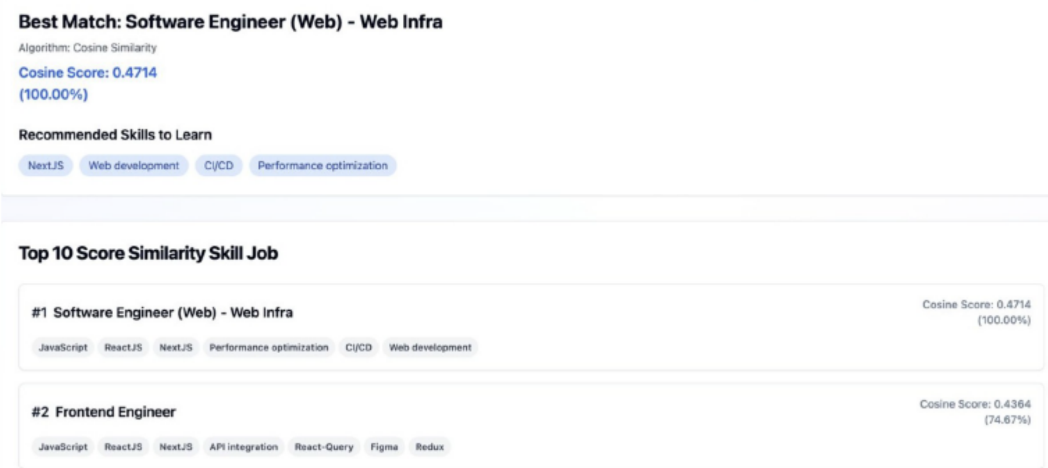

Figure 6. Skill Recommendation Result Using the Cosine Similarity Method

Testing Results of the User-Based Collaborative Filtering Recommendation Method with the Log-Likelihood Similarity Method first was conducted comparison between the top skills generated by the system with suitable skills according to the experts using method similarity Log-Likelihood Similarity. The test also considers the initial skills and profession or job desired by the user. The following is Table 4 of the actual skills and professions desired by users.

Table 4. Actual Skill and Desired User Profession

Id	Profession	Actual Skills
1	Backend Developer	GitLab, Programming, Java, Spring Framework, Spring MVC, Spring Boot
2	Frontend Developer	Bootstrap (Framework), Web Experience, User Experience (UX), JavaScript, React.js, Front-End Development, Object-Oriented Programming (OOP)
3	Backend Developer	Back-End Web Development, PostgreSQL, REST APIs, Teamwork, Leadership, Docker, API Development, Postman API
4	Frontend Developer	styled-components, Tailwind CSS, Git, TypeScript, Next.js, React.js
5	Backend Developer	Application Programming Interfaces (API), MySQL, MongoDB, PostgreSQL
6	Android Developer/Flutter Developer	Web Development, Android Development, Application Development, Mobile Application Development, Android
7	Backend	Object-Oriented Programming (OOP), Spring Boot, Java, Spring Framework
8	Backend	SQL, Java, HTML, Bootstrap, JavaScript, Jasper Reports, Problem Solving
9	Backend Developer	Java, Spring Boot, Hibernate, Software Development, Agile Methodologies, Object-Oriented Programming (OOP)
10	Java Backend	PostgreSQL, Kubernetes, Continuous Integration and Continuous Delivery, Google Kubernetes Engine (GKE), Google Cloud Platform (GCP), Apache Kafka, Eureka, Jhipster
…	…	…
100	Mobile Developer	Android app Development, Flutter, Mobile App Development, iOS Development

The following is Table 5, which shows the results of testing the User-Based Collaborative Filtering recommendation method using the Log-Likelihood Similarity method. The testing still considers the actual profession and skills possessed by the user.

Table 5. Table Result of the User-Based Collaborative Filtering Recommendation Method with The Log-Likelihood Similarity

id	Expert Opinion	System Recommendation Occurence	Hit 1	Hit 0
1	Docker & Containerization, RESTful API Design Best Practices, microservices architecture	Microservices Architecture	1	
2	State Management (Redux/Zustand/Context API), Component Design & Reusability, HTML	HTML	1	
3	System Design & Scalability, Application Architecture	Application Architecture	1	
4	State Management (Redux/Zustand), Performance Optimization, Docker	Docker	1	
5	Docker & Containerization, Unit Testing & Test Coverage, Java	-		0
6	Flutter & Dart Mastery, State Management (Bloc/Provider/GetX)	-		0
7	Database Design & Optimization, Rest API Development	Rest API	1	

id	Expert Opinion	System Recommendation Occurence	Hit 1	Hit 0
8	Spring Boot Framework, RESTful API Design	Restful API	1	
9	Database Design & Rest API, Unit Testing & Test Automation	Rest API	1	
10	Spring Cloud & Docker,Observability (Logging, Tracing, Monitoring)	Docker	1	
…	…	…	…	…
11	State Management (Bloc/Provider), Kotlin, Cross-Platform Performance, Optimization	Kotlin	1	
	Total		73	27

From the total Hit Rate calculation results, we obtained a total of 73 hits and 27 misses. Therefore, the overall Hit Rate score can be calculated as 0.73. This means that 73% of users received at least one relevant recommendation in the top skill recommendations provided by the system compared to the expert opinion recommendations. The second test compared the top skill recommendations generated by the system with the skills deemed suitable by expert opinion using the Cosine Similarity method.

From the total Hit Rate calculation results, we obtained a total of 51 hits and 49 misses. Therefore, the overall Hit Rate score can be calculated as 0.51. This means that 51% of users received at least one relevant recommendation in the top recommendations provided by the system compared to the expert recommendations. The skill recommendation system was developed using a User-Based Collaborative Filtering approach with two different methods to calculate the similarity between users and users of the job vacancy application, which are also represented as users. First, with the Log-Likelihood Similarity method, and second, with the Cosine Similarity method. These two methods are used to identify other users who have similar skill patterns, so that new relevant skills can be recommended to users based on the skills of those similar users. In other words, skills that are not yet possessed will be recommended.

The Log-Likelihood Similarity method measures the strength of association between two sets of skills using a probabilistic approach. In this context, the system calculates the probability that the appearance of certain skills in two users is not a coincidence. This method is suitable for use when the data is binary. The Cosine Similarity method measures the angular proximity between two skill vectors. In its implementation, each user's skill vector is formed based on a binary representation, where a value of 1 indicates that the skill is possessed and 0 indicates that it is not. The smaller the angle between the two vectors, the greater the cosine similarity value obtained, which means that the two users have similar skills.

To assess the effectiveness of the skill recommendation system that was built, the recommendation results were tested using the Hit Rate metric, which measures the proportion of relevant skills from all skills recommended by the system compared to expert recommendations. From the total Hit Rate calculation results for the User-Based Collaborative Filtering recommendation method with the Similarity Log-Likelihood Similarity method, the results obtained were 73 hits and 27 misses. The overall Hit Rate score was 0.73. This means that 73% of users received at least one relevant recommendation in the top skill recommendations provided by the system compared to the expert recommendations. Meanwhile, the User-Based Collaborative Filtering method with the Cosine Similarity method resulted in 51 hits and 49 misses. Therefore, the overall Hit Rate score was calculated to be 0.51. This means that 51% of users received at least one relevant recommendation. From this test, there was a significant difference between users of the Log-Likelihood Similarity method and the Cosine Similarity method. The Log-Likelihood Similarity method produced more relevant results.

4. Conclusion

Based on the overall research process, starting from problem formulation, data collection through web scraping, data preprocessing, implementation of the recommendation system algorithm, and system evaluation using the Hit Rate Matrix, several conclusions can be drawn. The results indicate that the User-Based Collaborative Filtering method combined with the Log-Likelihood Similarity approach outperforms the Cosine Similarity method. This is supported by the evaluation results, where the Log-Likelihood Similarity achieved a Hit Rate of 0.73, compared to 0.51 obtained by the Cosine Similarity, demonstrating a significant performance difference. For future research, several improvements are suggested to enhance the skill recommendation system. First, incorporating additional data sources from various job portals such as LinkedIn Jobs, Glassdoor, Kalibrr, and Karir.com would provide more diverse and representative data, leading to more relevant and accurate recommendations aligned with current job market demands. Second, further studies should explore and compare other recommendation approaches, such as Item-Based Collaborative Filtering and Content-Based Filtering, to provide a more comprehensive evaluation and deeper understanding of each method’s strengths and limitations. Such comparisons would also strengthen the justification for selecting the most suitable algorithm for skill recommendation systems.

Acknowledgments

We would like to express our deepest gratitude to the Computer Sains of Budi Luhur University lecturer, friends and family for their support and trust to support this research. And for those who involved in helping us complete this research.

Contributions of the Authors

Fadli Yandra proposed the idea and topic; Muhammad Iqbal Fadillah designed the model and experiments; Fadli Yandra and M. Iqbal Fadillah designed the hybrid detection algorithm and application; and analyzed the research results in consultation with Mr. Hari Soetanto.

Conflicts of Interest

The authors declare no conflict of interest

References

’Alim, A. Solichin, A. and Painem, P. (2020). *Rekomendasi Keterampilan Teknologi Informasi Menggunakan Metode User-Based Collaborative Filtering dan Log-Likelihood Similarity*. *6*(2).

Al-otaibi, S., Altwoijry, N., Alqahtani, A., Aldheem, L., & Alqhatani, M. (2022). *Cosine similarity-based algorithm for social networking recommendation*. *12*(2), 1881–1892. https://doi.org/10.11591/ijece.v12i2.pp1881-1892

Bai, X., Wang, M., Lee, I., Member, S., Yang, Z., Kong, X., Member, S., Xia, F., & Member, S. (2019). Scientific Paper Recommendation : A Survey. *IEEE Access*, *7*, 9324–9339. https://doi.org/10.1109/ACCESS.2018.2890388

Barranco, M. J., & Martínez, F. J. (2025). *A preference-based daily meal recommendation framework for patients with diabetes*. *31*(3), 239–259. https://doi.org/10.3897/jucs.150833

Baruah, N., Gupta, S., Ghosh, S., & Afrid, S. N. (2023). *Exploring Jaccard Similarity and*

Cosine Similarity for Developing an Assamese Question Answering System. 1–11. https://doi.org/10.1007/978-981-99-5881-8_8
Deepseek. (2025). *Deepseek*. https://chat.deepseek.com/
Elahi, A., & Zirak, A. (2024). *Online and Offline Evaluations of Collaborative Filtering and Content Based Recommender Systems* (Vol. 1, Issue 1). Association for Computing Machinery.
Farikhah, I., Mahmudi, A. A., & Sodiq, F. (2026). *Implementation of the TOPSIS Method in a Website-Based Hospital Employee Performance Evaluation System to Improve Assessment Objectivity*. *12*(1), 1–18.
Februariyanti, H., Laksono, A. D., Wibowo, J. S., & Utomo, M. S. (2021). *IMPLEMENTASI METODE COLLABORATIVE FILTERING UNTUK SISTEM Diterima : Diterbitkan : IX*(I), 43–50.
Feng, S., Meng, J., & Zhang, J. (2021). *News Recommendation Systems in the Era of Information Overload*. *20*, 459–470. https://doi.org/10.13052/jwe1540-9589.20210
Flores, V. A., Permatasari, P. A., & Jasa, L. (2020). *Penerapan Web Scraping Sebagai Media Pencarian dan*. *19*(2), 157–162.
Garousi, V. (2020). *Closing the gap between software engineering education and industrial needs*. *37*, 68–77.
Jobstreet. (2025). *Jobstreet*. https://id.jobstreet.com/
Juan, W., Yue-xin, L., & Chun-ying, W. (2019). *Survey of Recommendation Based on Collaborative Filtering*. https://doi.org/10.1088/1742-6596/1314/1/012078
Khusna, A. N., Delasano, K. P., Chaerul, D., & Saputra, E. (2021). *Penerapan User-Based Collaborative Filtering Algorithm Studi Kasus Sistem Rekomendasi untuk Menentukan Gadget Shield Application of User-Based Collaborative Filtering Algorithm : Case Study of Recommendation System for Determining Gadget Shield*. *20*(2), 293–304. https://doi.org/10.30812/matrik.v20i2.1124
Ko, H., Lee, S., Park, Y., & Choi, A. (2022). *A Survey of Recommendation Systems : Recommendation*.
Mahendra, R. R., Anggraeny, F. T., & Wahanani, H. E. (2024). *Implementasi Item-Based Collaborative Filtering Untuk Rekomendasi Film*. *3*.
Mondi, R. H., & Wijayanto, A. (2019). *RECOMMENDATION SYSTEM WITH CONTENT-BASED FILTERING METHOD FOR CULINARY TOURISM IN*. *8*(2), 65–72.
Muzdalifah, N., Mohd, B., & Chew, K. W. (2022). *Change in Labour Force Skillset for the Fourth Industrial Revolution : A Literature Review*. *13*(August), 969–978. https://doi.org/10.14716/ijtech.v13i5.5875
Pari, R. A., Kurniawan, D., Komputer, M. I., Luhur, U. B., Acceptance, T., & Tam, M. (2021). *Sistem Rekomendasi Keterampilan dengan Metode User-Based Collaborative Filtering dan Tanimoto Coefficient Similarity*. *XVI*, 52–62.
Permata, R. P., Suharto, R. N., Candra, L., & Julianty, A. (2025). *Towards an Automated Essay Evaluation System NLP Based Text Embeddings and Similarity Metrics*. *16*(1), 37–46.
Prehanto, D. R., Indriyanti, A. D., Nuryana, K. D., Koniyo, M. H., & Lamusu, S. (2018). *Web Scraping and Naïve Bayes Classification for Job Search Engine Web Scraping and Naïve Bayes Classification for Job Search Engine*. *288*. https://doi.org/10.1088/1757-899X/288/1/012038
Ro, A., Pranoto, Y. M., & Setyati, E. (2021). *Sistem Rekomendasi Pekerjaan Menggunakan Content Based Similarity*.
Shamsuddin, F., Zabidi, A., & Razak, A. (2023). *Development of a Model for Data-Driven Decision Making : Critical Skills for School Leaders*. *8*(12), 1–20.
Statistik, B. P. (2024). *BPS 2024*. https://jakarta.bps.go.id/en/pressrelease/2024/11/05/1170/jakarta-s-current-labor-force-condition-in-august-2024.html
Suhada, S., Bahri, S., Nugraha, S. B., Hidayatulloh, T., Wintana, D., Artikel, I., & Kunci,

K. (2023). *Sistem Rekomendasi Produk Menggunakan Metode User-Based Collaborative Filtering Pada Digital Marketing*. *204*, 158–166.