OPEN ACCESS
Indo-JC
Indonesian Journal on Computing
ISSN 2460-9056
socj.telkomuniversity.ac.id/indojc

Ind. Journal on Computing
Vol. 9, Issue. 1, April 2024. pp. 8-19
doi:10.34818/indojc.2024.9.1.907

Utilizing GP 2 for Restaurant Recommendation

Nitamayega #1, Gia Septiana Wulandari #2, Kemas Rahmat Saleh Wiharja #3

 # School of Computing, Telkom University
Jl. Telekomunikasi No. 1 Ters. Buah Batu Bandung 40257 Indonesia

1 nitamayega@student.telkomuniversity.ac.id
2 giaseptiana@telkomuniversity.ac.id
3 bagindokemas@telkomuniversity.ac.id

Abstract

The increasing diversity of food and beverage providers poses a challenge for people to find a restaurant that aligns with their preferences. Restaurant recommendation systems can address this problem by providing accurate and relevant suggestions. Although there are many previous studies have explored various recommendation methodologies, the utilization of knowledge graph implemented with GP 2 is still limited. Knowledge graphs can represent complex information in a structured way, while GP 2 is a graph-specific programming language that has a simple syntax. This research focuses on the implementation of a knowledge graph-based restaurant recommendation system with GP 2. The recommendation scheme built can provide the best accuracy, reaching 84.97%. This shows that the knowledge graph-based restaurant recommendation system with GP 2 can demonstrate the effectiveness of the system in providing accurate and relevant recommendations, showing the potential of knowledge graph and GP 2 for the development of recommendation systems in the future and being an effective solution to overcome recommendation problems.

Keywords: GP 2, knowledge graph, recommender system, restaurant recommendation

Abstrak

Meningkatnya keragaman penyedia makanan dan minuman menjadi tantangan tersendiri bagi seseorang dalam menemukan restoran yang sesuai dengan preferensi mereka. Sistem rekomendasi restoran dapat mengatasi masalah ini dengan memberikan rekomendasi yang akurat dan relevan. Meskipun ada banyak penelitian terdahulu terkait sistem rekomendasi, tetapi pemanfaatan knowledge graph yang diimplementasikan dengan GP 2 masih terbatas. Knowledge graph dapat merepresentasikan informasi yang kompleks secara terstruktur, sedangkan GP 2 merupakan bahasa pemrograman khusus graf yang memiliki sintaks sederhana. Penelitian ini berfokus pada implementasi sistem rekomendasi restoran berbasis knowledge graph dengan GP 2. Skema rekomendasi yang dibangun dapat memberikan akurasi terbaik, mencapai 84,97%. Hal ini menunjukkan bahwa sistem rekomendasi restoran berbasis knowledge graph dengan GP 2 dapat menunjukkan efektivitas sistem dalam memberikan rekomendasi yang akurat dan relevan, menunjukkan potensi knowledge graph dan GP 2 untuk pengembangan sistem rekomendasi di masa depan dan menjadi solusi yang efektif untuk mengatasi permasalahan rekomendasi.

Kata Kunci: GP 2, *knowledge graph*, rekomendasi restoran, sistem rekomendasi

I. Introduction

The availability of a diverse range of restaurants can make difficult for consumers to choose a place to eat that suits their preferences [1] [2], this potentially leads to frustration and dissatisfaction. With the

Received on January, 2024. Accepted on April, 2024

development of technology, restaurant recommendation systems have become an important solution to help consumers make the right decisions. However, conventional methods such as content-based [3], collaborative filtering [4] [5] [6], deep learning [7], and knowledge graph, although widely used, still have limitations in providing accurate and diverse recommendations.

Content-based analyzes the content or description of the item under study, but provides polarized and less diverse recommendations. Collaborative filtering involves tracking user interactions with items in the past to make recommendations, but cannot make recommendations for new users or items. Deep learning learns patterns from user data to compile recommendations, but requires complex training data. These limitations highlight the need for alternative approaches that leverage richer and more structured information. Knowledge graphs work by representing information in a structured graph format. While there have been efforts to combine methods to improve accuracy, there is still room for further development due to the limitations of the methods used today.

The combination of recommender systems with knowledge graphs is an interesting aspect of this research. Knowledge graphs allow for the representation of complex and structured information [8], while GP 2 as a graph programming language provides ease of program development [9]. This research aims to address the existing problems in restaurant recommendation systems by implementing knowledge graphs using GP 2. The recommendation system built in this research utilizes an input graph programmed by applying several rules, resulting in an output graph as a representation of the recommendation. By focusing on the Restaurant Data with Consumer Ratings dataset from UCI Machine Learning, this research not only presents a solution, but also faces the limitations of the existing data, with 138 users, 935 restaurants, and 1161 ratings.

Evaluation of the recommendation system using a confusion matrix provides an objective basis for assessing the accuracy and performance of the proposed system. Thus, this research not only discusses practical implementation, but also contributes to the development of concepts and methods in the field of knowledge graph-based restaurant recommendation systems.

II. Literature Review

A. Restaurant Recommender System

Recommender systems are software tools or an a system that is created with the aim of helping users to find out items that they might be interested in [5], helping them make the right decision with a large number of choices. Approaches to recommender systems that are commonly used are collaborative filtering , content-based, and hybrid recommender systems.

Knowledge-based approaches are becoming the main focus, leveraging specific knowledge in determining recommendations. For example, a recommender system that uses knowledge about reviews or ratings can provide more relevant recommendations based on a user's preferences. All of these approaches provide a comprehensive framework for recommender systems that can be adapted and developed to meet research needs.

A restaurant recommendation is a suggestion for the best restaurants that may be of interest according to a user's preferences. Alif Azhar Fakhri [4] has conducted a related study using user-based collaborative filtering, focusing on the Bandung Raya area. This system utilizes information from other users who have similar preferences (called neighbors) to the target user to provide restaurant recommendations. This study has a MAE value of 1.492.

Another study related to restaurant recommendation systems was conducted by Kevin Hartanto Muliadi [5]. Using a development method based on user-based collaborative filtering, namely Typicality Based CF (TyCo), this study claims to be able to provide accurate predictions with limited data. Although it got an MAE value of 1.366 due to the poor quality of the training data.

B. Knowledge Graph

Knowledge Graph (KG) is a method of structured knowledge representation in the form of triples that present entities and the relationships between those entities [10]. KG can be seen as a combination of a data schema

(TBox) and the data graph itself (ABox). TBox includes information about entity types and relationships, while ABox includes concrete data or facts that are dynamic and can grow rapidly over time.

In KG, semantics are represented in the form of a graph, where nodes represent entities, and edges represent relationships between those nodes [11]. KG can store various information, including facts, attributes, and context related to entities, enabling complex data processing. Xiaohan Zou [11] notes that KG has become a focus of attention, especially after being proposed by Google, and a survey showed that recommendation systems are the second most common field for KG implementation, as shown in Figure 1.

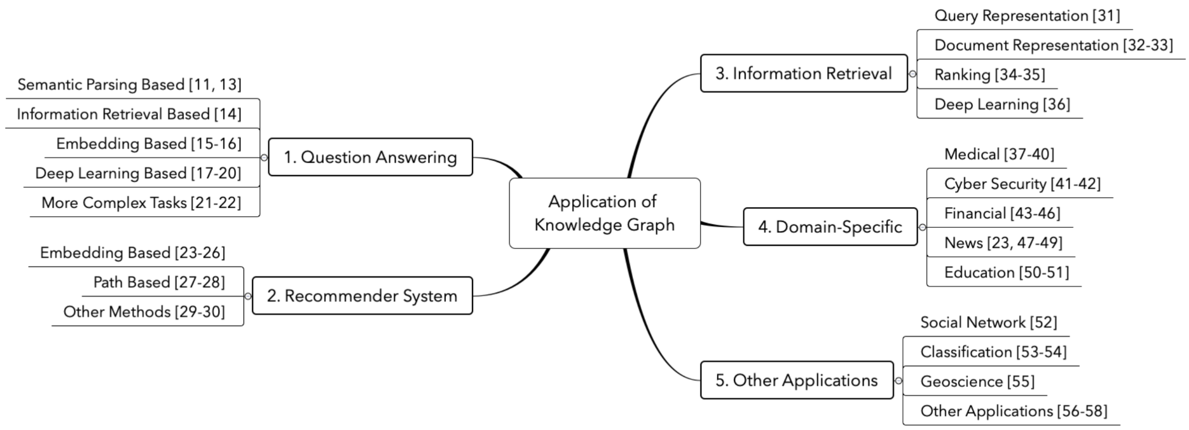

Fig. 1. Results of the KG Implementation Survey [11]

The implementation of KG depends on the ontology used, including rules for connecting one entity to another. For example, if a user is known to like European food, the system can use the relationship "restaurant A serves European food" to provide recommendations. Entities and relationships in KG can be extended according to the purpose of creating a recommendation system [12].

C. GP 2

GP 2 is a graph-specific programming language that implements graph transformation rules with a simple syntax and semantics, making it easy to develop the programs [9]. These transformations convert an input graph into an output graph, with the graph used being a directed graph that can have parallel edges and loops. In general, a graph system consists of two finite sets of nodes and edges, source and target functions, a partial node labeling function, a partial edge labeling function, and a partial root function [9].

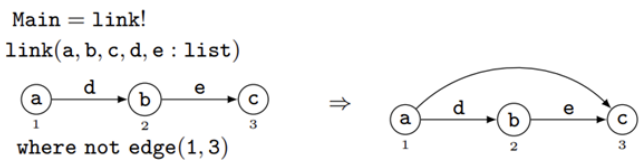

Fig. 2. Transitive-Closure Program [9]

The main programming construct in GP 2 is a conditional graph transformation rule labeled with an expression. This rule is the basis for graph manipulation [9], as seen in Figure 2, which shows the graph transformation rule for a transitive-closure program. In this example, the "link" rule has five formal parameters and is prefixed with the keyword where. Parameters can be of type list, atom, int, string, or char. Nodes and edges can be labeled with text, color, or other characteristics, as seen in Figure 3.

```
Main = try init then (DFS!; Check)
DFS = forward!; try back else break
Check = if match then fail
```

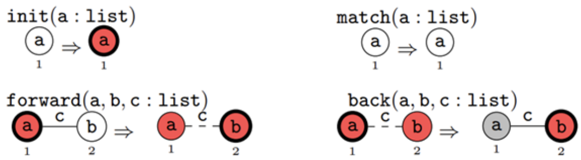

Fig. 3. Variations of Vertices and Edges in the GP 2 [9]

Although they can be visualized in the form of a graph, the rules used in GP 2 are written in text format using an abstract syntax that can be seen in Figure 4 [13]. For example, the operators indeg and outdeg specify the number of edges (relations between two nodes in the graph) that enter and exit a node (a node or point in the graph), while the operator length returns the length of a list variable. In graphical representation, nodes in the graph can be displayed in gray, while edges can be shown with dashed lines.

Program	::=	Declaration { Declaration }
Declaration	::=	MainDecl | ProcedureDecl | RuleDecl
MainDecl	::=	`Main` '=' CommandSeq
ProcedureDecl	::=	ProcedureID '=' ['[' LocalDecl ']'] CommandSeq
LocalDecl	::=	(RuleDecl | ProcedureDecl) { LocalDecl }
CommandSeq	::=	Command {';'Command}
Command	::=	Block
		| `if` Block `then` Block [`else` Block]
		| `try` Block [`then` Block] [`else` Block]
Block	::=	'(' CommandSeq ')' ['!']
		| SimpleCommand
		| Block `or` Block
SimpleCommand	::=	RuleSetCall ['!']
		| ProcedureCall ['!']
		| `break`
		| `skip`
		| `fail`
RuleSetCall	::=	RuleID | '{' [RuleID { ',' RuleID }] '}'
ProcedureCall	::=	ProcedureID

Fig. 4. Abstract Syntax for GP 2 [13]

III. RESEARCH METHOD

In this research, the initial phase involves the preparation of a restaurant dataset, which undergoes a meticulous preparation stage aimed at enhancing data quality. The construction of the knowledge graph and the recommendation system is initiated using the refined data. The system testing stage uses a confusion matrix. If the results are satisfactory, the process will continue to the evaluation stage. If not, then the construction of the knowledge graph and the recommendation system will be repeated. The flowchart of the restaurant recommendation system in this research can be seen in Figure 5 below.

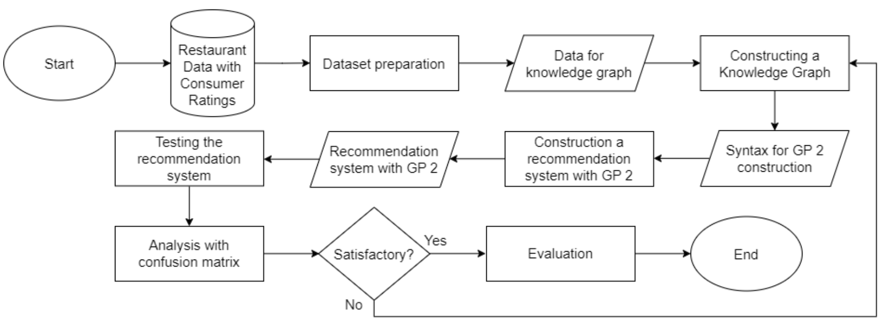

Fig. 5. System Flowchart

A. Dataset Preparation

In the preparation stage, several critical steps are taken to ensure the quality and usability of data in the development of the recommendation system. First, the representation of data in several columns is standardized to make it clearer and more intuitive. Next is change of rating from users to restaurants to a scale of 100 and the calculation of the average rating for each restaurant. The rating used in this recommendation system is rounded up, because GP 2 only accepts integers. Replacing the rating scale aims to produce a standard and easy-to-interpret representation.

The stage of selecting attributes that will be used as the basis for the construction of the recommendation system scheme is selecting the most relevant attribute so that it can describe user preferences. Thus, the dataset is ready to be used in the development of restaurant recommendation systems. Examples of the results of the preparation stage can be represented in the form of Table I for rating results, Table II for user attributes, and Table III for restaurant attributes.

TABLE I
PREPARATION DATA FOR RATINGS

userID	placeID	100 Scale	Rounded Rating	Rounded Mean
U1077	135085	100	100	67
U1043	132630	50	50	56
U1011	132715	33,5	34	42
U1068	132733	33,5	34	64

TABLE II
PREPARATION DATA FOR USER ATTRIBUTES

userID	Smoker	Drinker	Dress Preference	Birth Era	Interest	Personality
U1077	Non Smoker	Drinker	Formal	Milenial	Technology	Thrifty Protector
U1043	Non Smoker	Non Drinker	Informal	Milenial	Technology	Hard Worker
U1011	Non Smoker	Non Drinker	Informal	Milenial	Variety	Hard Worker
U1068	Non Smoker	Drinker	Informal	Milenial	Technology	Thrifty Protector

TABLE III
PREPARATION DATA FOR RESTAURANT ATTRIBUTES

placeID	Smoking Area	Alcohol Served	Dress Code	Price Range	Payment Method	Ambience
135085	Non Smoker	Non Drinker	Informal	Medium	Cash	Familiar
132630	None	Non Drinker	Informal	Low	None	Familiar
132715	None	Non Drinker	Informal	Low	Cash	Quiet
132733	Non Smoker	Non Drinker	Informal	Medium	Visa	Familiar

B. Construction a Knowledge Graph

The data from the previous preparation will be used in the construction of the knowledge graph, which consists of 138 user nodes, 935 restaurant nodes, and 1161 rating nodes. There are 9 attributes that are analyzed to determine similarities between user-restaurant, user-user, and restaurant-restaurant. User nodes are marked in green, restaurant nodes in red, and other nodes are not colored.

In the knowledge graph scheme, a comparison of similarities is made between the user and restaurant nodes by considering attributes such as smoker or not, drinker or not, and clothing preference, formal or informal. The comparison is also made between two user nodes by considering the era of birth, interests, and personality. In addition, two restaurant nodes are also compared by considering the budget price, payment method, and atmosphere.

The graph used for the input graph on GP 2 was built with the help of Python. The Python program will automatically generate syntax to become the left-hand graph (input graph) of GP 2, including the creation of nodes and their relationships. Node and relationship IDs are generated through a Python increment. Some examples of output from the Python program can be seen in Table IV.

TABLE IV
SYNTAX FROM THE OUTPUT OF THE PYTHON PROGRAM FOR THE GP 2 INPUT GRAPH

	Syntax	Meaning
Node	(45, "U1046"#green)	The node for user U1046 has node ID 45 and is colored green.
	(755, "132861"#red)	The node for restaurant 132861 has node ID 755 and is colored red.
	(1079, "Milenial")	The node for the Milenial attribute has node ID 1079 and has the default color (white).
Edge	(1945, 45, 1079, empty)	The relation with relation ID 1945 is a relation between node ID 45 and node ID 1079 with no relation name (empty).
	(4812, 45, 755, 84)	The relation with relation ID 4812 is a relation between node ID 45 and node ID 755 with relation name 84 (integer). In this study, this can be interpreted as node ID 45 rating node ID 755 with a value of 84.

C. Recommendation System Construction with GP 2

1) Recommendation System Scheme

The recommendation system built in this research includes three extended schemes. These schemes use user target preference similarity to generate recommendations. Each scheme is designed to consider certain factors that are relevant in determining more accurate recommendations. An overview of the recommendation system scheme used in this research is presented in Table V.

TABLE V
OVERVIEW OF THE RESTAURANT RECOMMENDATIONS SYSTEM SCHEME

Scheme 1	The system recommends restaurants based on the similarity of the target user's preferences to restaurants that are highly rated by neighboring users but have not been rated by the target user. The condition is that both users have rated a restaurant highly.
Scheme 2	The system recommends restaurants based on the similarity of the target user's preferences to restaurants that are highly rated by neighboring users but have not been rated by the target user. The condition is that both users have rated a restaurant highly, and the neighboring user has similar preferences to the target user.
Scheme 3	The system recommends restaurants based on the similarity of the target user's preferences to restaurants that are highly rated by neighboring users but have not been rated by the target user. The condition is that both users have rated a restaurant highly that has similar preferences to the recommended restaurant, and the neighboring user has similar preferences to the target user.

The first scheme works by checking whether there are restaurants that are highly rated by the target user and those restaurants are also highly rated by other users, who are neighbors. Considering that the neighbor also highly rated other restaurants that have not been rated by the target user. If this condition is met, the system then evaluates the similarity in two attributes between the target user and the other restaurant, so it can recommend the restaurant. This scheme shows that the recommendation system will recommend restaurants based on the similarity of the target user's preferences to other restaurants, which is determined by the high level of rating for a particular restaurant. An illustration of the scheme can be seen in Figure 6 below, with user A as the target user, and resto B as the recommended restaurant.

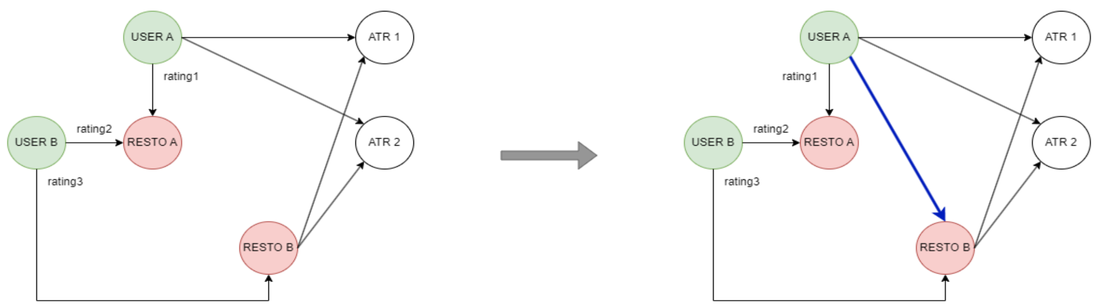

Fig. 6. First Recommendation System Construction Scheme

The second scheme is an extension of the first scheme, where attribute evaluation is not only limited to between the user and restaurant, but also involves neighbors. In other words, this second scheme resembles the first scheme, but the neighbors reviewed must have two attributes in common with the target user, namely era of birth, interests, or personality. The second scheme is visualized in Figure 7.

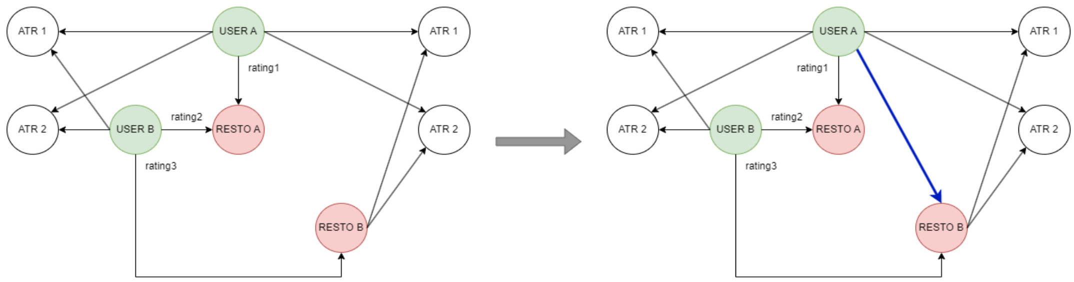

Fig. 7. Second Recommendation System Construction Scheme

The third scheme expands the analysis to the restaurant level. With this approach, the third scheme creates an additional level of complexity, combining information from the target user's preferences and the characteristics of restaurants that are rated well by the target user and their neighbors. Similarity in attributes such as price budget, payment method, and atmosphere can provide more specific and accurate restaurant recommendations. In this context, restaurants that are considered to be recommendation candidates must have similar attributes to restaurants that have been rated well by the target user and neighbors. The third scheme is visualized in Figure 8.

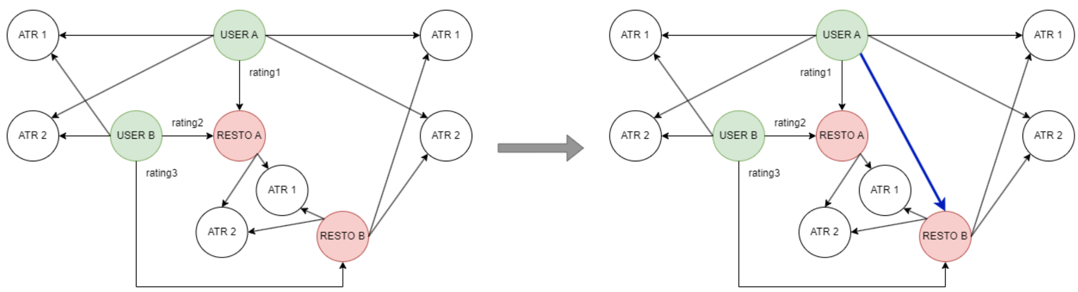

Fig. 8. Third Recommendation System Construction Scheme

Each scheme has a threshold for the rating value analyzed, namely 49, 59, and 69, with the reason that the median user rating for restaurants is 67 and the median average restaurant rating is 58. This threshold value is a reference limit that indicates the level of user tendency to "like" the restaurant. In addition, the second and third schemes use rooted nodes to optimize the program and accelerate the evaluation process.

Overall, the recommendation system in this research uses a series of extended schemes, focusing on preference similarity and relevant attribute evaluation, with the aim of providing more personalized recommendations that are in line with the unique preferences of each user in the context of restaurant search. In addition, the schemes in this recommendation system provide an overview of the level of complexity that can improve recommendation accuracy.

2) Use of Rooted Node

A rooted node is a special node on a graph that is used to speed up pattern matching in graph transformation. Rooted node serves as a focus point for matching, so it only needs to check around the root, not the entire graph. This makes the graph transformation program more efficient and faster.

The use of a rooted node is to overcome the inefficiency of graph pattern matching. The slow pattern matching process because it must check the entire graph can be accelerated by utilizing rooted node. The presence of rooted node allows the program to focus on a specific point (root) so it has the potential for the development of complex computational programs with high performance.

In this research, the use of rooted node is implemented in scheme 2 and scheme 3 for the construction of the recommendation system. This is based on the complexity of the program built, where the second scheme has a higher level of complexity than the first scheme, and the third scheme becomes the most complex. To overcome the long program execution time, both schemes utilize rooted node as a solution, while the first scheme can still be overcome by using a regular program. The rule used for rooted node implementation can be seen in Figure 9.

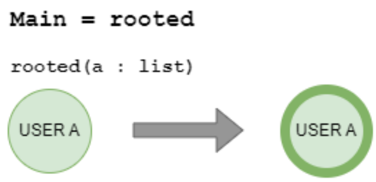

Fig. 9. Rule for Rooted and Unrooted Node

The process begins with searching for a specific node, such as one with a green color. Once the node is located, it is then converted into a rooted node, visualized with double or thick edges to distinguish it from a regular node. This approach enhances the efficiency of both search and transformation operations on the graph, thereby making a positive contribution to the overall performance of the developed recommendation system.

D. Recommendation System Testing

The evaluation method used at the testing stage is a confusion matrix, which involves the calculation of accuracy and recall. Accuracy is used to measure how well the recommendation system is able to predict restaurants that are in line with user preferences. This accuracy is used for threshold 49 because the number of False Negative and False Positive data is symmetrical. Recall is used to measure the percentage of restaurants that are recommended compared to the total number of restaurants that should be recommended. In addition, there is an assumption that the user will truly like a restaurant if the average value of the restaurant is above the threshold.

The True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) values are determined as the first step in calculating the evaluation matrix. TP is the number of recommended restaurants that have an average value above the threshold. TN is the number of restaurants that have an average value below the threshold and are not included in recommendations. FP is the number of recommended restaurants that have an average value below the threshold. FN is the number of restaurants that have an average value above the threshold but are not included in the recommendations.

The evaluation process is carried out using Python. Python is used to compare the results of restaurant recommendations by GP 2 with the average value of the restaurant. Python is also used to calculate the values of TP, TN, FP, and FN. These values are used to support the analysis of the results of recommendations by the system. Next, Python also performs the calculation to get the values of accuracy and recall.

IV. Results And Discussion

The results of the restaurant recommendation system testing can be classified in more detail by limiting the rating value to 49, 59, and 69. The recommendation system was able to achieve good performance, producing positive and expected results. The details of the accuracy and recall values from this testing can be seen in Table VI.

TABLE VI
Evaluation value of recommendation system testing results

Scheme	Threshold	Accuracy	Recall
1	49	81.78%	100%
	59	48.29%	100%
	69	27.84%	100%
2	49	85.57%	99.18%
	59	52.65%	98%
	69	34.81%	94.06%
3	49	84.97%	98.77%
	59	53.22%	97.63%
	69	35.52%	94.01%

The evaluation values of the results of the restaurant recommendation system testing have sensitivity to the performance of the system against different threshold values. When using a threshold value of 49, it can be seen that the accuracy of the recommendation system reached a high level, namely 81.78%, 85.57%, and 84.97% for scheme 1, scheme 2, and scheme 3, respectively.

However, when the threshold value is increased to 59 and 69, the accuracy value decreases significantly, even not reaching 60%. This is because the increase in threshold puts pressure on the evaluation of more complex attributes. In addition, it can be influenced by the existing rating data, where the rating data given to a restaurant tends to be low. In this study, the trade-off between accuracy and recall becomes relevant, where increasing the threshold value can increase recall, but at the same time can cause a decrease in accuracy.

Of the three schemes, a satisfactory accuracy value is seen at threshold 49, with a level above 80%. However, when using thresholds 59 and 69, the accuracy becomes lower because it involves the evaluation of more complex attributes by considering more factors. The difference in evaluation results between schemes is due to differences in the approaches used. Scheme 1 uses a simple approach that only considers the rating value and user-restaurant similarity. Scheme 2 uses a more complex approach that considers the rating value, user-restaurant similarity, and user-user similarity. Scheme 3 uses the most complex approach by considering the rating value, user-restaurant similarity, user-user similarity, and restaurant-restaurant similarity.

The trade-off between accuracy and recall is a critical aspect of evaluating a recommendation system. In choosing the optimal threshold value, it is necessary to consider the priority of the system that is being built. For example, if you want to provide very accurate recommendations, a threshold adjustment may be required to improve accuracy, but it could potentially reduce recall. Conversely, if the priority is to identify the correct number of recommendations, then the threshold setting can be adjusted, even if there is a possible decrease in accuracy.

The complexity factor of the rule or implementation method can also contribute to the difference in execution time between schemes. A scheme with a higher level of similarity or evaluation may require a longer execution time. Scheme 2 and scheme 3, which involve a high level of similarity with neighbors in a particular attribute, can be negatively affected if the data does not support this level of similarity. So, it is necessary to consider the quality and representativeness of the data in the design of the scheme.

Although accuracy provides a general overview of the quality of a recommendation system, the evaluation of the recall results can provide additional comparison. By measuring the percentage of restaurants that were successfully recommended compared to the total number of restaurants that should be recommended, all schemes showed good recall results, which are above 90%. This indicates that even though there is a decrease in accuracy at higher thresholds, the system is still effective in recommending restaurants that are truly in line with the user's preferences.

The influence of the data on the evaluation results also needs to be considered. Unrepresentative data can cause a recommendation system to generate inaccurate recommendations. This is because unrepresentative data does not adequately reflect the variation in user preferences. To improve the accuracy of the system, it is necessary to improve the data that is used. The data used needs to be expanded and balanced to adequately reflect the variation in user preferences.

TABLE VII
EXECUTION TIME OF RECOMMENDATION SYSTEM TEST RESULTS

Scheme	Threshold	Execution Time
1	49	2 min 23.6 sec
	59	1 min 30.3 sec
	69	55.5 sec
2	49	3.9 sec
	59	3.2 sec
	69	2.5 sec
3	49	13 min 34.7 sec
	59	9 min 58.4 sec
	69	7 min 5.6 sec

In addition, the aspect of execution time of the recommendation system is also a focus in performance evaluation. Table VII presents information on the execution time for each scheme that was tested. The smaller the threshold value, the longer the program execution time. For example, in scheme 1, with a threshold value of 69, the execution time is only 55.5 seconds, while at threshold 49, the execution time reaches 2 minutes and 23.6 seconds. The factors that influence the execution time are the complexity of the rule or implementation method used in each scheme.

In terms of execution time, threshold 69 shows the fastest execution time compared to other limits due to data filtering, namely the amount of data with threshold 69 is less. Overall, scheme 3 is the system with the longest execution time. This is because scheme 3 has a higher rule complexity than scheme 1 and scheme 2.

This research utilizes the use of rooted node to overcome the execution time problem. The application of rooted node helps the GP 2 compiler to find the appropriate subgraph to be used as the left-hand graph in a rule by giving priority first to the rooted node. In GP 2, it should be noted that only one rooted node can be used. Therefore, the use of rooted node provides a significant advantage in terms of execution time, even though more and more complex rules are applied.

In the case of scheme 2, the recorded execution time is faster than scheme 1, even though scheme 2 has a higher level of complexity. This phenomenon can be explained by the use of rooted node which is only applied to scheme 2 and scheme 3, while scheme 1 only uses ordinary rules. The execution of scheme 1 can be completed only by using ordinary rules, so in the context of this research, the rooted node is not applied to scheme 1. Therefore, there is a possibility of longer execution time for scheme 1, even though using simpler rules, due to the difference in the application of rooted node.

Based on the results of the evaluation, it can be concluded that the restaurant recommendation system based on knowledge graph is able to achieve good performance. The system is able to generate recommendations that are in line with the user's preferences, with an accuracy of above 80% for threshold 49. However, the system also has some limitations, such as the sensitivity of accuracy to the threshold value and the longer execution time for more complex schemes.

V. CONCLUSION

Based on the results of testing and analysis, a knowledge graph-based restaurant recommendation system implemented using the GP 2 programming language has achieved good performance. The evaluation was carried out by considering the accuracy and recall values for each scheme and threshold, as well as the execution time aspect. The system was able to generate restaurant recommendations that were in line with the user's preferences with an accuracy rate of over 80% for a threshold of 49, namely scheme 1 reaching 81.78%, scheme 2 reaching 85.57%, and scheme 3 reaching 84.97%. However, the accuracy value decreased when using thresholds of 59 and 69. This is due to the more complex attribute evaluation by considering more factors.

The fastest execution time of the recommendation system occurred in scheme 1 with a threshold of 69, due to the simple system construction. Conversely, the longest execution time of the recommendation system occurred in scheme 3 with a threshold of 49.

Although there was a decrease in accuracy at higher thresholds, the evaluation of the recall results showed that all schemes remained effective in recommending restaurants that were in line with the user's preferences, with a recall value of over 90%.

In related research [3] that used a content-based approach, the system provided an accuracy of 51.18%, while in related research [6] to collaborative filtering, the system provided an accuracy of 80.63%. In this case, the application of GP 2 in the development of knowledge graph-based restaurant recommendation systems indicates that the recommendation system has good quality in reaching user preferences, although further research is needed.

To improve the accuracy of the system, it is necessary to improve the quality and representativeness of the data to be used, such as by deleting irrelevant data and updating data sources (datasets) to overcome the limitations of the dataset used in this study. In addition, future research can explore other schemes, such as integrating restaurant genre information or leveraging the time recording when users provide information in the dataset to get recommendations based on the user's most recent preferences. This can help in coping with changes in user preferences over time.

In summary, our research contributes to advancing the scientific understanding of restaurant recommendation systems by demonstrating the effectiveness of a GP 2-based knowledge graph approach. By providing accurate recommendations and identifying avenues for further improvement, we aim to facilitate better decision-making for consumers in the realm of dining choices.

ACKNOWLEDGMENT

The authors would like to thank the many people who helped us during the research process, especially mentors and colleagues at Telkom University. Your support and guidance were invaluable and enabled us to complete this project.

REFERENCES

[1] Editor, “Suka Bingung Saat Pilih Menu Makanan? Begini Penjelasan Psikologisnya,” *GenerasiMilenial,* 28 Oktober 2022.

[2] V. A. Dhini, “Jumlah Usaha Penyedia Makanan & Minuman di DKI Jakarta Terbanyak se-Indonesia,” *Databoks,* 20 Juni 2022.

[3] R. H. Mondi, A. Wijayanto dan Winarno, “Recommendation System With Content-Based Filtering Method for Culinary Tourism In Mangan Application,” *ITSMART: Jurnal Ilmiah Teknologi dan Informasi,* 2019.

[4] A. A. Fakhri, Z. K. A. Baizal dan E. B. Setiawan, “Restaurant Recommender System Using User-Based Collaborative Filtering Approach: A Case Study at Bandung Raya Region,” *Journal of Physics: Conference Series,* 2019.

[5] K. H. Muliadi dan C. C. Lestari, “Rancang Bangun Sistem Rekomendasi Tempat Makan Menggunakan Algoritma Typicality Based Collaborative Filtering,” *Techno.Com,* 2019.

[6] M. N. S. Bahri, I. P. Y. D. Jaya, B. Dirgantoro, Istikmal, U. A. Ahmad dan R. R. Septiawan, “Implementasi Sistem Rekomendasi Makanan pada Aplikasi EatAja Menggunakan Algoritma Collaborative Filtering,” *Jurnal Multinetics,* 2021.

[7] S. Krishnaraj, “Restaurant Recommendation System Using Machine Learning Algorithms,” *RIET-IJSET International Journal of Science Engineering and Technology,* 2021.

[8] L. Miao, X. Li, D. Yu, Y. Ren, Y. Huang and S. Cao, "Integrating Users’ Long-Term and Short-Term Interests With Knowledge," *Journal of King Saud University – Computer and Information Sciences,* 2023.

[9] G. S. Wulandari dan D. Plump, “Verifying Graph Programs with Monadic Second-Order Logic,” *Springer-Verlag,* 2021.

[10] K. R. S. Wiharja, “An iterative approach for schema aware knowledge graph completion,” *Doctoral dissertation, University of Aberdeen,* 2020.

[11] X. Zou, “A Survey on Application of Knowledge Graph,” *Journal of Physics: Conference Series,* 2020.

[12] M. H. Syed, T. Q. B. Huy dan S.-T. Chung, “Context-Aware Explainable Recommendation Based on Domain Knowledge Graph,” *Big Data and Cognitive Computing,* 2022.

[13] B. Courtehoute dan D. Plump, “A Small-Step Operational Semantics fo GP 2,” *Electronic Proceedings in Theoretical Computer Science,* 2021.

[14] G. Campbell, B. Courtehoute dan D. Plump, “Fast Rule-based Graph Programs,” *Science of Computer Programming,* 2022.