Indonesian Journal of Computing and Cybernetics Systems (IJCCS)
Vol.x, No.x, July xxxx, pp. 1~5
ISSN (print): 1978-1520, ISSN (online): 2460-7258
DOI: 10.22146/ijccs.xxxx

Online Learning Video Recommendation System Based on Course and Sylabus Using Content-Based Filtering

Faisal Ramadhan*[1], Aina Musdholifah[2]
[1] Bachelor Program of Computer Science, FMIPA UGM, Yogyakarta, Indonesia
[2] Department of Computer Science and Electronics, FMIPA UGM, Yogyakarta, Indonesia
e-mail: ***[1]faisalramadhan@mail.ugm.ac.id**, [2]aina_m@ugm.ac.id

Abstrak

Pembelajaran menggunakan media video seperti menonton video di YouTube menjadi salah satu alternatif cara belajar yang sering digunakan. Akan tetapi, video pembelajaran tersedia begitu melimpah sehingga mencari video yang kontennya tepat menjadi sulit dan memakan waktu. Oleh karena itu, penelitian ini membangun sistem rekomendasi yang dapat merekomendasikan video berdasarkan mata kuliah dan silabus. Sistem rekomendasi bekerja dengan mencari kedekatan antara mata kuliah dan silabus dengan anotasi video menggunakan metode cosine similarity. Anotasi video tersebut merupakan judul dan deksripsi video yang diambil secara real-time dari YouTube menggunakan YouTube API. Sistem rekomendasi ini akan menghasilkan rekomendasi berupa lima buah video berdasarkan mata kuliah dan silabus yang dipilih. Hasil pengujian menunjukkan persentase kinerja rata-rata adalah 81.13% dalam pencapaian tujuan sistem rekomendasi yaitu relevance, novelty, serendipity dan increasing recommendation diversity.

Kata kunci*—sistem rekomendasi, video pembelajaran, content-based filtering, cosine similarity*

Abstract

Learning using video media such as watching videos on YouTube is an alternative method of learning that is often used. However, there are so many learning videos available that finding videos with the right content is difficult and time-consuming. Therefore, this study builds a recommendation system that can recommend videos based on courses and syllabus. The recommendation system works by looking for similarity between courses and syllabus with video annotations using the cosine similarity method. The video annotation is the title and description of the video captured in real-time from YouTube using the YouTube API. This recommendation system will produce recommendations in the form of five videos based on the selected courses and syllabus. The test results show that the average performance percentage is 81.13% in achieving the recommendation system goals, namely relevance, novelty, serendipity and increasing recommendation diversity.

Keywords—*recommendation system, learning videos, content-based filtering, cosine similarity*

Received June 1st,2012; Revised June 25th, 2012; Accepted July 10th, 2012

1. INTRODUCTION

Learning through video is one of the learning alternatives that is often applied by college students. Learning videos also help lecturers who find it difficult to deliver material online. The use of video in the online learning process is the right step, because the ability of videos can visualize material very effectively and of course this is very helpful for educators in delivering dynamic material [1]. Several studies have shown that the use of video teaching media is more desirable and shows college student interest [2]. Scenarios in the form of videos also contain lots of information and triggers for students to determine their own learning methods and learning objectives that must be achieved [3]. Learning videos are abundantly available online such as on the YouTube video platform.

However, the abundance of available videos makes finding videos with the right topic or content quite difficult and time-consuming. Often the videos are incomplete and do not match the content of the topic. Therefore, it needs a video recommendation system that is capable of filtering videos by topic.

Recommender systems represent user preferences for the purpose of suggesting items to purchase or examine. They have become fundamental applications in electronic commerce and information access, providing suggestions that effectively prune large information spaces so that users are directed toward those items that best meet their needs and preferences [4].

In research conducted by Adam et al. [5], the researcher built a Calculus video recommendation system to help students find videos on Calculus topics with relevant content using content-based recommendation techniques. In this video recommendation system, the user first selects a Calculus topic in the system, then the system will make a query and search for several video lists via the YouTube API. The results of the video search will be displayed to the user as many as twenty videos and the user is asked to select several lists of these videos to be included in the favorite list. The recommendation system will then only work when at least one video is selected from the video list to be included in the favorite list. This makes the video recommendation system ineffective because the user must first select the right video list so that the results of the recommendations are more appropriate. In addition, the video content used in this recommendation system is the video title only, so the results of the recommendations may not be appropriate because the video title alone is not sufficient to represent the video as a whole.

Based on the description of the problems previously described, this study created an online video learning recommendation system based on courses and syllabus to help students in their learning activities. The system will look for similarity between courses and syllabus with video annotations. The video annotations are video titles and descriptions obtained in real-time from YouTube with a maximum of fifty video annotations. The system then immediately displays the results of video recommendations based on the highest similarity value order.

2. METHODS

2.1 Data

This study uses two types of data to form recommendations on the system, namely learning video data and curriculum data.

2.1.1 Learning video data

Learning video data in this study is YouTube video data consisting of video annotations (video title and description), duration, publication date, thumbnail, and link. This data is obtained in real-time, when the system successfully performs a query to Youtube using the YouTube API.

2.1.2 Curriculum data

This research uses the curriculum 2016 of Computer Science Bachelor Degree Study Program UGM data consisting of courses and syllabus in English. Data is taken from the website of the Department of Computer Science and Electronics, Faculty of Mathematics and Natural Sciences. The website URL is https://dcse.fmipa.ugm.ac.id/site/en/undergraduate-computer-science/1999-2/. The data is finally stored in the system hard coded.

2.2 Text Preprocessing

Text processing is done to remove noise from text data before the data is used for further processing. The text preprocessing stages used in this study include case folding, tokenization, and stopword removal.

2.2.1 Case folding

In text preprocessing, the case folding process aims to change all letters in a text document to lowercase. In this process, characters other than letters such as numbers, punctuation marks, and symbols are removed so that what remains is the text of the alphabet a to z..

2.2.2 Tokenization

In text processing, tokenization is the procedure of splitting a text into words, phrases, or other meaningful parts, namely tokens. In other words, tokenization is a form of text segmentation. Typically, the segmentation is carried out considering only alphabetic or alphanumeric characters that are delimited by non-alphanumeric characters (e.g., punctuations, whitespace) [6].

2.2.3 Stop words removal

Many of the most frequently used words in English are useless in Information Retrieval (IR) and text mining. These words are called 'Stop words'. Stop-words, which are language-specific functional words, are frequent words that carry no information (i.e., pronouns, prepositions, conjunctions) [7].

2.3 Recommendation System

Recommender system works as a helper in finding relevant and related items by making relevant suggestions to the users [8]. In this study, the recommendation system was created through a content-based recommendation technique that used the similarity of video annotations (video titles and descriptions) taken from YouTube to make recommendations. The system flowchart is shown in the following Figure 2.

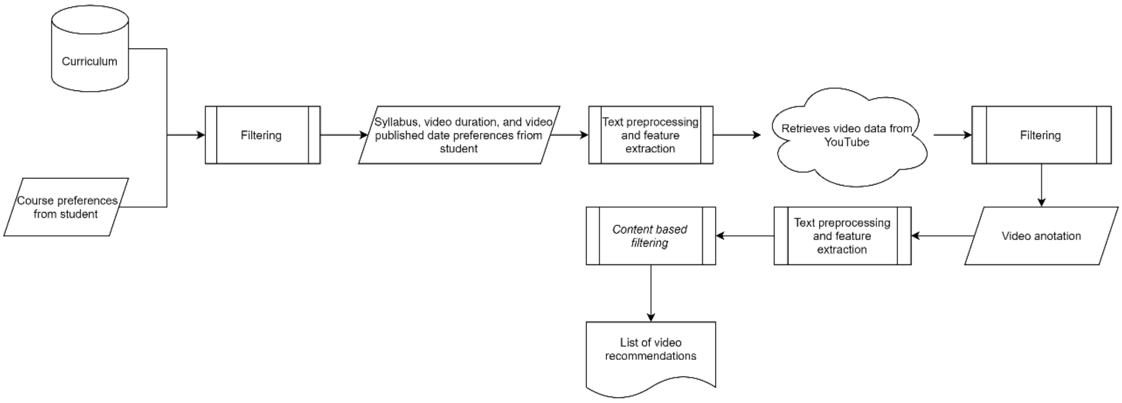

Curriculum
Course preferences from student
Filtering
Syllabus, video duration, and video published date preferences friom student
Text preprocessing and feature extraction
Retrieves video data from YouTube
Filtering
Video anotation
Text preprocessing and feature extraction
Content based filtering
List of video recommendations

Figure 1 Flowchart of the system

The system first displays a list of available courses based on curriculum data. Next, the system accepts input for one of the courses chosen by students based on their preferences. The system then performs filtering based on course input to display the syllabus. Filling in the syllabus is done by selecting the syllabus that is displayed by the system in the form of a checkbox or can be typed in the form provided. The syllabus must be selected or filled out, the system will give a warning if the syllabus is not filled out after submitting. The system also provides filling in the duration and date of the video publication. Video duration input aims to get a video with the desired duration range. The video duration consists of any (any video duration and is the default), short (less than four minutes), medium (between four to twenty minutes), and long (more than twenty minutes). The video publication date input aims to get videos with a publication date before or equal to the date that is inputted with the default being the date when the user inputs.

Furthermore, the system performs text preprocessing on course and syllabus input including case folding, deleting numbers, deleting certain symbols, deleting links, deleting html tags, deleting characters with only one length, tokenization, and deleting stop words. The input that has been preprocessed will be used as one of the query parameters to find and retrieve a maximum of fifty video lists on YouTube in real-time using the YouTube API. The system takes YouTube videos when the user submits data and there is a query to YouTube. Video retrieval is also filtered based on other input query parameters such as video duration and video publication date. The obtained videos are then stored in the system in the form of a list consisting of link, duration, date of publication, and video annotations consisting of video title and description. Video annotations will be preprocessed before being stored in the system.

The next stage is the system performs feature extraction on input text and video annotations using the TF-IDF method. After getting TF-IDF, the system then calculates the similarity using the cosine similarity method. Similarity calculations are carried out on the input text of the course name and syllabus with a collection of video annotations that have been preprocessed. Then, the results of the text similarity calculation will be sorted from largest to smallest (descending) and the system recommends five videos to the user.

2.3.1 Term Frequency – Inverse Document Frequency

Term Frequency (TF) is the number of occurrences of certain words or terms in a document [9]. Term Frequency has several solutions, one of which is used in this study is the Raw Term Frequency (Raw TF). The formula for Raw TF is shown in Equation 1, where *i* is the unique term that appears in document *j*.

$$tf(i,j) = f_{ij} \tag{1}$$

Inverse Document Frequency (IDF) is a reduction of the term domination that often appear in various documents by calculating the inverse frequency of document which contain the term [10]. The fewer the number of documents that contain the term, the greater the IDF value. IDF is calculated using Equation 2, where *D* is the total number of documents and DF_i is a document containing the unique term *i*.

$$idf_i = log\left(\frac{D}{df_i}\right) \tag{2}$$

The term frequency-inverse document frequency (TF-IDF) is a numerical statistic which reflects how important a word is to a document. The tf-idf value increases proportionally to the number of times a word appears in the document [11]. The formula used to calculate TF-IDF is in Equation 3, where tf_{ij} is the term frequency of term *i* in document *j* and idf_i is the inverse document frequency of term *i*.

$$\begin{gathered} w_{ij} = tf_{ij} \times idf_i \\ w_{ij} = tf_{ij} \times \left(log\frac{D}{df_i} + 1\right) \end{gathered} \tag{3}$$

2.3.2 Cosine Similarity

Cosine similarity is a method to calculate the similarity between two text documents. The concept of cosine similarity is to calculate the cosine angle between two vectors that if given a document which is represented by a vector d_j and documents or query is represented as a vector *q*, and *t* terms are extracted from a database or a collection of text, then the value of the cosine similarity is obtained using the formula in Equation 4.

$$Similarity(q, d_j) = \frac{q \bullet d_j}{|q||dj|} = \frac{\sum_{i=1}^{t} wiq \times wij}{\sqrt{\sum_{i=1}^{t}(wiq)^2} \times \sqrt{\sum_{i=1}^{t}(wij)^2}} \tag{4}$$

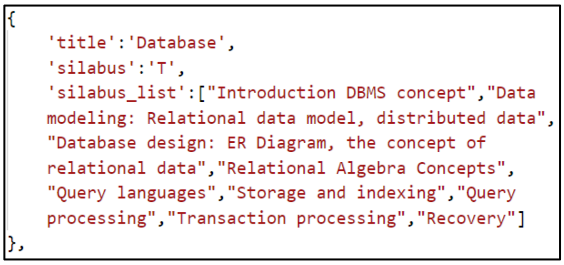
2.4 Evaluation

System evaluation performed to measure the system objective which consists of relevance, novelty, serendipity and increasing recommendation diversity.

3. RESULT AND DISCUSSION

3.1 Curriculum Data Collection Results

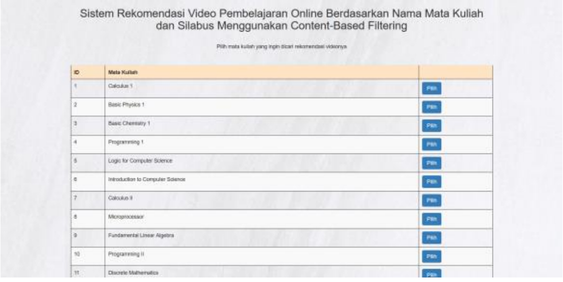
As previously explained, curriculum data in the form of course and syllabus used in this system are the curriculum 2016 of Computer Science Bachelor Degree Study Program UGM. The data is stored in the system as hard code in the form of a list of dictionaries data structure. The following is a snippet of curriculum data stored in Figure 2..



```
{
    'title':'Database',
    'silabus':'T',
    'silabus_list':["Introduction DBMS concept","Data
    modeling: Relational data model, distributed data",
    "Database design: ER Diagram, the concept of
    relational data","Relational Algebra Concepts",
    "Query languages","Storage and indexing","Query
    processing","Transaction processing","Recovery"]
},
```

Figure 2 Snippet of curriculum data

Curriculum data that has been stored will be displayed to users in stages starting with displaying courses with an interface as shown in Figure 3.



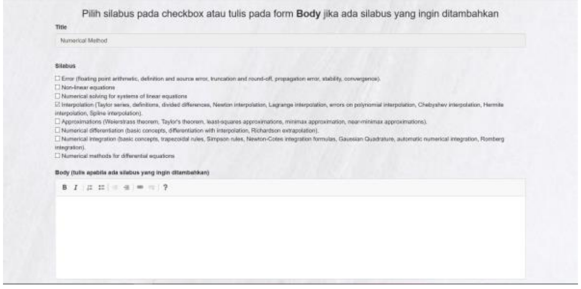
Sistem Rekomendasi Video Pembelajaran Online Berdasarkan Nama Mata Kuliah dan Silabus Menggunakan Content-Based Filtering

Pilih mata kuliah yang ingin dicari rekomendasi videonya

ID	Mata Kuliah	
1	Calculus 1	Pilih
2	Basic Physics 1	Pilih
3	Basic Chemistry 1	Pilih
4	Programming 1	Pilih
5	Logic for Computer Science	Pilih
6	Introduction to Computer Science	Pilih
7	Calculus II	Pilih
8	Microprocessor	Pilih
9	Fundamental Linear Algebra	Pilih
10	Programming II	Pilih
11	Discrete Mathematics	Pilih

Figure 3 Course selection interface

After selecting courses, the system then displays the syllabus with an interface as in Figure 4 below.



Pilih silabus pada checkbox atau tulis pada form **Body** jika ada silabus yang ingin ditambahkan

Title

Numerical Method

Silabus

☐ Error (floating point arithmetic, definition and source error, truncation and round-off, propagation error, stability, convergence).
☐ Non-linear equations
☐ Numerical solving for systems of linear equations
☑ Interpolation (Taylor series, definitions, divided differences, Newton interpolation, Lagrange interpolation, errors on polynomial interpolation, Chebyshev interpolation, Hermite interpolation, Spline interpolation).
☐ Approximations (Weierstrass theorem, Taylor's theorem, least-squares approximations, minimax approximation, near-minimax approximations).
☐ Numerical differentiation (basic concepts, differentiation with interpolation, Richardson extrapolation).
☐ Numerical integration (basic concepts, trapezoidal rules, Simpson rules, Newton-Cotes integration formulas, Gaussian Quadrature, automatic numerical integration, Romberg integration).
☐ Numerical methods for differential equations

Body (tulis apabila ada silabus yang ingin ditambahkan)

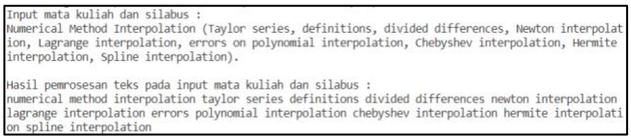
Figure 4. Syllabus selection interface

3.2 Video Data Retrieval

Before retrieving video data, the system first accepts input consist of courses and syllabus, duration, and publication date. The input of courses and the syllabus is preprocessed first before being used as one of the parameters in the query..

3.2.1 The results of text processing on the input of courses and syllabus

After the system receives the input, the system will first perform text preprocessing on the input of the course and syllabus before being used as one of the parameters in the query. For example, the system accepts input the numerical method and one of its syllabus, then the text preprocessing results is in Figure 5.



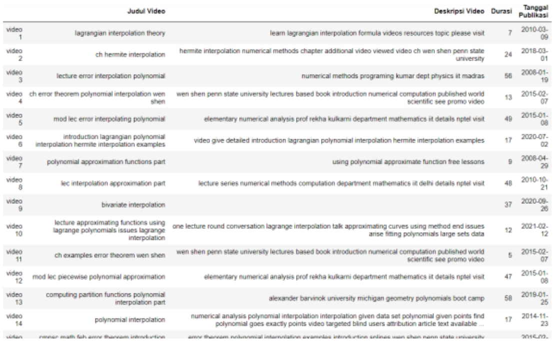
```
Input mata kuliah dan silabus :
Numerical Method Interpolation (Taylor series, definitions, divided differences, Newton interpolat
ion, Lagrange interpolation, errors on polynomial interpolation, Chebyshev interpolation, Hermite
interpolation, Spline interpolation).

Hasil pemrosesan teks pada input mata kuliah dan silabus :
numerical method interpolation taylor series definitions divided differences newton interpolation
lagrange interpolation errors polynomial interpolation chebyshev interpolation hermite interpolati
on spline interpolation
```

Figure 5 The results of text preprocessing on the input

3.2.2 The results of video data retrieval

Video data retrieval is done by making a request to YouTube using YouTube API with query parameters consisting of: courses and syllabus as shown in Figure 5, the video duration is any (any video duration), and the video publication date is the default (the date when making a request to YouTube). The video data obtained, such as the title and description, will be preprocessed first before being saved. The following are the results of some video data that have been taken as in Figure 6.

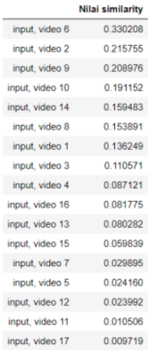


	Judul Video	Deskripsi Video	Durasi	Tanggal Publikasi
video 1	lagrangian interpolation theory	learn lagrangian interpolation formula videos resources topic please visit	7	2010-03-09
video 2	ch hermite interpolation	hermite interpolation numerical methods chapter additional video viewed video ch wen shen penn state university	24	2018-03-01
video 3	lecture error interpolation polynomial	numerical methods programing kumar dept physics iit madras	56	2008-01-19
video 4	ch error theorem polynomial interpolation wen shen	wen shen penn state university lectures based book introduction numerical computation published world scientific see promo video	13	2015-02-07
video 5	mod lec error interpolating polynomial	elementary numerical analysis prof rekha kulkarni department mathematics iit details nptel visit	49	2015-01-08
video 6	introduction lagrangian polynomial interpolation hermite interpolation examples	video give detailed introduction lagrangian polynomial interpolation hermite interpolation examples	17	2020-07-02
video 7	polynomial approximation functions part	using polynomial approximate function free lessons	9	2008-04-29
video 8	lec interpolation approximation part	lecture series numerical methods computation department mathematics iit delhi details nptel visit	48	2010-10-21
video 9	bivariate interpolation		37	2020-09-26
video 10	lecture approximating functions using lagrange polynomials issues lagrange interpolation	one lecture round conversation lagrange interpolation talk approximating curves using method end issues arise fitting polynomials large sets data	12	2021-02-12
video 11	ch examples error theorem wen shen	wen shen penn state university lectures based book introduction numerical computation published world scientific see promo video	5	2015-02-07
video 12	mod lec piecewise polynomial approximation	elementary numerical analysis prof rekha kulkarni department mathematics iit details nptel visit	47	2015-01-08
video 13	computing partition functions polynomial interpolation part	alexander barvinok university michigan geometry polynomials boot camp	58	2019-01-25
video 14	polynomial interpolation	numerical analysis polynomial interpolation interpolation given data set polynomial given points find polynomial goes exactly points video targeted blind users attribution article text available ...	17	2014-11-23
video	cmnec math feb error theorem introduction	error theorem polynomial interpolation examples introduction splines wen shen penn state university		2015-02-

Figure 6 Snippets of video data retrieval results

3.3 The Result of Cosine Similarity Calculation

After the system performs TF-IDF calculations from input text (courses and syllabus) and video annotation text obtained from video data retrieval, the system then calculates the cosine similarity between the input text and the video annotation text. The calculation results are then sorted based on the highest similarity value as shown in Figure 7.

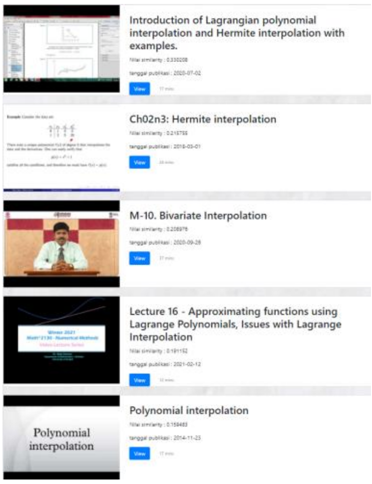


	Nilai similarity
input, video 6	0.330208
input, video 2	0.215755
input, video 9	0.208976
input, video 10	0.191152
input, video 14	0.159483
input, video 8	0.153891
input, video 1	0.136249
input, video 3	0.110571
input, video 4	0.087121
input, video 16	0.081775
input, video 13	0.080282
input, video 15	0.059839
input, video 7	0.029895
input, video 5	0.024160
input, video 12	0.023992
input, video 11	0.010506
input, video 17	0.009719

Figure 7 The result of cosine similarity calculation

3.4 Video Recommendation

Once sorted, the system then recommends five videos that have the highest value of the similarity. The following is the video recommendation interface in Figure 8.



Introduction of Lagrangian polynomial interpolation and Hermite interpolation with examples.
Nilai similarity : 0.330208
tanggal publikasi : 2020-07-02
View 17 mins

Ch02n3: Hermite interpolation
Nilai similarity : 0.215755
tanggal publikasi : 2018-03-01
View 24 mins

M-10. Bivariate Interpolation
Nilai similarity : 0.208976
tanggal publikasi : 2020-09-26
View 37 mins

Lecture 16 - Approximating functions using Lagrange Polynomials, Issues with Lagrange Interpolation
Nilai similarity : 0.191152
tanggal publikasi : 2021-02-12
View 12 mins

Polynomial interpolation
Nilai similarity : 0.159483
tanggal publikasi : 2014-11-23
View 17 mins

Figure 8 Results of video recommendation

3.5 System Evaluation

Evaluation of system objective was carried out by conducting a survey of 40 UGM Computer Science students. Before filling out the form, students are asked to try the recommendation system first by entering the name of the course and the syllabus. After trying the system, students are then asked to fill in the form that has been provided in relation to the evaluation of the recommendation system that has been built.

Each question regarding the recommendation system will have 5 values to choose from. Value 5 means strongly agree, value 4 means agree, value 3 means neutral, value 2 means disagree, and value 1 means strongly disagree. After getting all the scores, then for each attribute the value will be added up and then divided by the maximum score to get the percentage of goal achievement. Following are the results of the survey for the achievement of the system objectives in Table 1 below.

Table 1 System objective survey result

Student	Question about			
	Relevance	Novelty	Serendipity	Diversity
1	5	5	4	5
2	4	3	2	4
3	4	3	4	4
4	4	4	4	4
5	4	4	3	5
6	2	3	3	4
7	4	4	5	3
8	5	4	4	5
9	4	3	3	5
10	5	4	4	4
11	4	5	5	5
12	5	4	3	5
13	4	4	4	3
14	5	4	4	5
15	4	3	4	5
16	4	3	5	5
17	4	5	4	4
18	2	5	4	4
19	3	4	4	2
20	4	5	5	4
21	4	4	4	4
22	4	4	3	4
23	5	5	4	4
24	4	5	3	5
25	5	4	4	3
26	5	4	4	4
27	4	4	5	4
28	5	2	2	3
29	5	5	5	5
30	4	3	4	4
31	4	4	4	4
32	5	4	4	5
33	4	4	5	5
34	5	3	4	4
35	3	5	3	5

36	4	4	4	5
37	4	2	3	4
38	4	3	4	5
39	5	3	3	4
40	5	5	4	5
Total Score	168	156	154	171
Maximum Score	200	200	200	200
Achievement Percentage	84%	78%	77%	85.5%

4. CONCLUSIONS

The recommendation system that has been built is able to achieve the system objective with a good score for relevance of 84%, novelty of 78%, serendipity of 77%, and 85.5% of increasing recommendation diversity. The system has an average 81.13% of the system objective.

5. FUTURE WORKS

Adding features to filter certain videos such as filtering videos with Indian accents because some users feel uninterested in these types of videos and hopefully it can increase the system objective value.

REFERENCES

[1] Ammy, P. M. and Wahyuni, S., 2020, Analisis Motivasi Belajar Mahasiswa Menggunakan Video Pembelajaran Sebagai Alternatif Pembelajaran Jarak Jauh (PJJ), *Jurnal Mathematic Paedagogic*, *5*(1), 27-35.

[2] Persson, A.C., Fyrenius, A. and Bergdahl, B., 2010, Perspectives on using multimedia scenarios in a PBL medical curriculum, *Medical teacher, 32*(9), pp.766-772.

[3] Van Den Hurk, M. M., Wolfhagen, I. H., Dolmans, D. H., and Van Der Vleuten, C. P. (1999), The impact of student- generated learning issues on individual study time and academic achievement, *Medical Education*, *33*(11), 808-814.

[4] Burke, R., 2002, Hybrid recommender systems: Survey and experiments, *User modeling and user-adapted interaction*, *12*(4), pp.331-370.

[5] Adam, N. L., Sulaiman, M. S. A., and Soh, S. C., 2019, Calculus video recommender system, *Journal of Physics: Conference Series*, 1366, 1-8.

[6] Uysal, A.K. and Gunal, S., 2014, The impact of preprocessing on text classification, *Information Processing & Management*, *50*(1), pp.104-112.

[7] Srividhya, V. and Anitha, R., 2010, Evaluating preprocessing techniques in text categorization, *International journal of computer science and application*, *47*(11), pp.49-51.

[8] Khusro, S., Ali, Z. and Ullah, I., 2016. Recommender systems: issues, challenges, and research opportunities. In *Information Science and Applications (ICISA) 2016* (pp. 1179-1189). Springer, Singapore.

[9] Rahman, A., Wiranto, W. and Doewes, A., 2017. Online news classification using multinomial naive bayes. *ITSMART: Jurnal Teknologi dan Informasi*, *6*(1), pp.32-38.

[10] Nurjannah, M., Hamdani, H., and Astuti, I. F., 2016, Penerapan Algoritma Term Frequency-Inverse Document Frequency (TF-IDF) untuk Text Mining, *Informatika Mulawarman: Jurnal Ilmiah Ilmu Komputer*, *8*(3), 110-113.

[11] Munot, N. and Govilkar, S.S., 2014, Comparative study of text summarization methods, *International Journal of Computer Applications*, *102*(12).