khazanah informatika

Jurnal Ilmu Komputer dan Informatika

Recommendation System to Propose Final Project Supervisor using Cosine Similarity Matrix

Zulfa Fajrul Falah[1], Fajar Suryawan[*2]
[1]Department of Informatics
[2] Department of Electrical Engineering
Universitas Muhammadiyah Surakarta
Central Java 57162, Indonesia
*Fajar.Suryawan@ums.ac.id

Abstract-The selection of a supervisor is an important thing and one of the determinants of whether or not a student's final project research is successful. At the location of this research, students select a supervisor by considering his academic records and recommendations from classmates or seniors. Words of mouth dominate their motivation, and many students do not have a basis for their choice. Selection of the best-fit supervisor significantly impacts a student's progression. Students will be more enthusiastic about doing the final project and may get facilitation in their research because the topics of the student projects match the supervisor's interests and ongoing work. This study aims to make a recommendation system that suggests a supervisor for a student. The student fills in the title, abstract, and keywords of his proposal. The system gives suggestions to prospective supervisors by calculating the similarity of the data with titles, abstracts, and keywords of published articles found in Google Scholar. The recommendation system uses the content-based filtering method to produce a list of recommendations. The cosine similarity algorithm calculates how similar the topic proposed by students is to the lecturers' interests. In building a website-based recommendation system, the authors use Django web framework as the backend and ReactJs as the frontend. The application succeeds in suggesting final project supervisors that match lecturers' interests and expertise with students' proposals.

Keywords: cosine similarity, recommendation system, web scraping, content-based filtering

Article info: *submitted November 9, 2021, revised May 4, 2022, accepted September 16, 2022*

1. Introduction

A final project is a common requirement for students to graduate from a university. This final project is typically the ultimate scientific work of a student in completing his or her undergraduate education study period. In the final project, students are accompanied by a supervisor. It is the supervising lecturer who will become a partner in collaboration between students and lecturers in carrying out the research that has been submitted. The supervising lecturer also functions as a facilitator for students if students experience difficulties or doubts in the research process. The supervising lecturer must also master the field that is in accordance with the topic taken by the student so that the research results are maximized, therefore the role and suitability of the research field of a supervisor is vitally important.

In the department where this research takes place, the selection of supervisors is mostly done manually and independently by students. Students choose their supervisors directly when the study planning phase takes place. The selection of supervisors is roughly based on personal knowledge related to the specialization of lecturers and is also based on research carried out by students themselves with minimal data sources, some of which are sourced from classmates or from seniors who have graduated. There are even students who choose a supervisor without a special reason regardless of whether the lecturer matches their interests or the topic they are going to propose.

From the problems above, the authors see the urgency of building a recommendation system to help students determine supervisors. A recommendation system serves to sort through large amounts of data to identify user interests and make it easier to find information and form decisions [1]. A recommendation system is also an information filtering system used to predict the rating or preference that will be given to a user on an item such as music, books, movies, and documents. The recommendation system model can be built from the characteristics of an item (content-based filtering) or with a user environment approach (collaborative filtering approaches)[2] which will also be used in this study.

Content-based filtering works using an item's feature to recommend other items that are similar to what the user likes. It is also one of the most successful recommendation techniques, which is based on correlation between content. It uses item information represented as attributes to calculate similarity between items [3]. However, this strategy also has weaknesses, one of which is that this method cannot produce appropriate recommendations if the content analyzed for an item does not contain information suitable for categorization [4]. There are several methods that can be used to calculate similarity between content such as Euclidean distance, cosine similarity and Manhattan distance. Research conducted by Fathin in 2019 which compared the results of calculations between cosine similarity and Euclidean distance showed similarity values and had the same level of accuracy [16]. In this study, the method used to compare the similarity between content using cosine similarity.

There have been several studies conducted on the topic of selecting a final project supervisor. One of them is done by Asrul in 2018 who used the Analytical Hierarchy Process (AHP) method to build a decision support system for selecting supervisors. In the AHP method, the weighting of the criteria is carried out by experts, which is very subjective because the scoring of the criteria depends on each expert [17]. There is another research conducted at the Department of Computer Science/Informatics, Faculty of Science and Mathematics, Diponegoro University regarding the supervisory lecturer selection system, in this study using the Vector Space Model (VSM) as a method of matching the strings of student research titles and research that has been carried out by lecturers [18].

This study uses data from lecturers' publications, the abstracts of which have been published on the Google Scholar page. Data retrieval is done by scraping each lecturer's Google Scholar profile page. This data will be the knowledge base in building a recommendation system model which will then produce information in determining the appropriate supervisor for students. Data taken from Google Scholar includes titles, abstracts and keywords from lecturer publications. To the best of the authors' knowledge, this research is the first research in Indonesia that uses data from Google Scholar to build a recommendation system.

The purpose of this recommendation system is to produce a list of final project supervisors that are in accordance with the topics proposed by students, and can make it easier for students to choose suitable supervisors based on the proposed topic.

2. Methods

In conducting this research, we first collect publication data from Google Scholar, followed by the initial preparation of the data. The data that is ready is then fed to the main algorithm. A website was built to embed the recommendation system, which is the main interface between the system and the user. The details are as follows.

a. Data Collection

The data was taken by doing web scraping on the Google Scholar profile page for each prospective supervisor.

Web scraping is the process of retrieving a semi-structured document from the internet, which is generally in the form of web pages in a markup language such as HTML or XHTML, then analyzing the document to retrieve certain data that is used for several purposes. In this study, the author uses Python to do web scraping with the help of third party libraries such as Selenium, BeautifulSoup, Requests, and CSV, which then the data from the web scraping is saved into a CSV file for processing to the next stage. This data set has 603 rows and 4 attributes as shown in Table 1.

Table 1. Description of the supervisor's research data attributes

Number	Attribute	Information
1	Name	Supervisor's name
2	Title	Research title
3	Abstract	Research abstract
4	Keyword	Research keywords

b. Data Preprocessing

Data preprocessing is one of the main stages in the knowledge discovery process, although this stage is not as popular as other stages such as data mining, data preprocessing actually involves more time and effort in the entire data analysis process [5]. Data from web scraping is generally in the form of raw data, such as there are still HTML elements that are accidentally taken. There are also non-alphanumeric characters and incomplete rows. The presence of data inconsistencies and noise contained in the dataset can affect the performance of the machine [6]. Data preprocessing also serves to improve the data format and to clean interference and noise from the raw data [7]. The following are some of the data preprocessing steps performed here.

Punctuation Removal is the process of removing characters that are not included in the letters of the alphabet because of other characters such as punctuation marks and non-alphanumeric characters (except spaces) such as !”#$%&’()*+,-./:;<=>? @[\]^_`{|}~ can affect the accuracy of the analysis. In this study the author uses python to clean punctuation and non-alphanumeric characters.

Case Folding, to change all uppercase letters in the document to lowercase letters.

Tokenization, which is shown in Table 2, is a process to divide or break texts in the form of sentences, paragraphs, or documents into tokens. In linguistics, token is the smallest unit in a text. This token will help to understand the context and for the development of the NLP model. Tokenization will also be useful for interpreting the meaning of the text by analyzing the order of words.

Table 2. Tokenization Process

Original Text	After Tokenization
'rancang bangun aplikasi pembelajaran hadist'	'rancang', 'bangun', 'aplikasi', 'pembelajaran', 'hadist'

Stop-words removal, which is the process of taking only important words. In general, in a text there are words that commonly appear such as prepositions, conjunctions, pronouns, and others. These words do not provide much information in the text. Removing less informative words from the text can give the engine more focus to process only the words that are important. In other words, removing less informative words will not have a negative impact on the vector, and will be more efficient from the processing side.

Word vectorization is a methodology in NLP to map words or phrases from vocabulary into a vector of real numbers and to determine word predictions or word/semantic similarity. By doing word vectorization on the supervisor's research text, the machine can process it as a vector of real numbers no longer as a collection of words. To represent a document, it is necessary to convert it into a vector form, so that it can be processed by machines [8]. The use of Term Frequency (TF) and Inverse Document Frequency (IDF) schemes has proven to be a powerful algorithm in processing text data or other purposes [9]. TF-IDF uses word frequency and document frequency to produce weighted words that are used to represent documents [8]. Terms of word frequency or document frequency in the TF-IDF approach are usually used to weigh each word in a text document according to its uniqueness [10].

c. Data Processing: Cosine similarity

Recommendation systems have several algorithms such as content-based filtering, collaborative filtering and a combination of the two [1], [11]. In this study, the author uses a content-based filtering algorithm as a method to determine the results of recommendations from supervisors. The content-base used is the text of titles, abstracts, and lecturers' research keywords. Recommendation systems using this technique have similarities with other techniques in terms of item descriptions, user profiles, and techniques for comparing profiles with items to identify the most suitable recommendation results for users [2]. One of the methods used to measure the closeness between texts is the cosine similarity method, which will be used here.

Cosine Similarity is a method used to measure the similarity between two text documents which are considered as vectors [12]. Cosine similarity is also a matrix that is widely implemented in information retrieval and is often applied in comparing the similarity of two texts (sentences, paragraphs or entire documents), the similarity between two documents is obtained by calculating the cosine value of the vectors between documents [13]. In this study, the method to calculate the similarity between the lecturer's research and the topic that will be proposed by students is by comparing the similarity of the title, abstract, keywords of potential supervisors' research with those of the student-submitted proposals. The value of cosine similarity between vectors can be calculated by the following equation:

$$cos\ a = \frac{A \cdot B}{|A||B|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}} \quad (1)$$

where:

A = Vector A, is the lecturer's research vector
B = Vector B, is a vector of topics proposed by students
A • B = dot product between vector A and vector B
|A| = vector length A
|B| = vector length B
|A||B| = cross product between |A| and |B|

Further explanation and illustration of this cosine similarity calculation can be seen in sections 3.A and 3.B.

A. Implementation and Interface

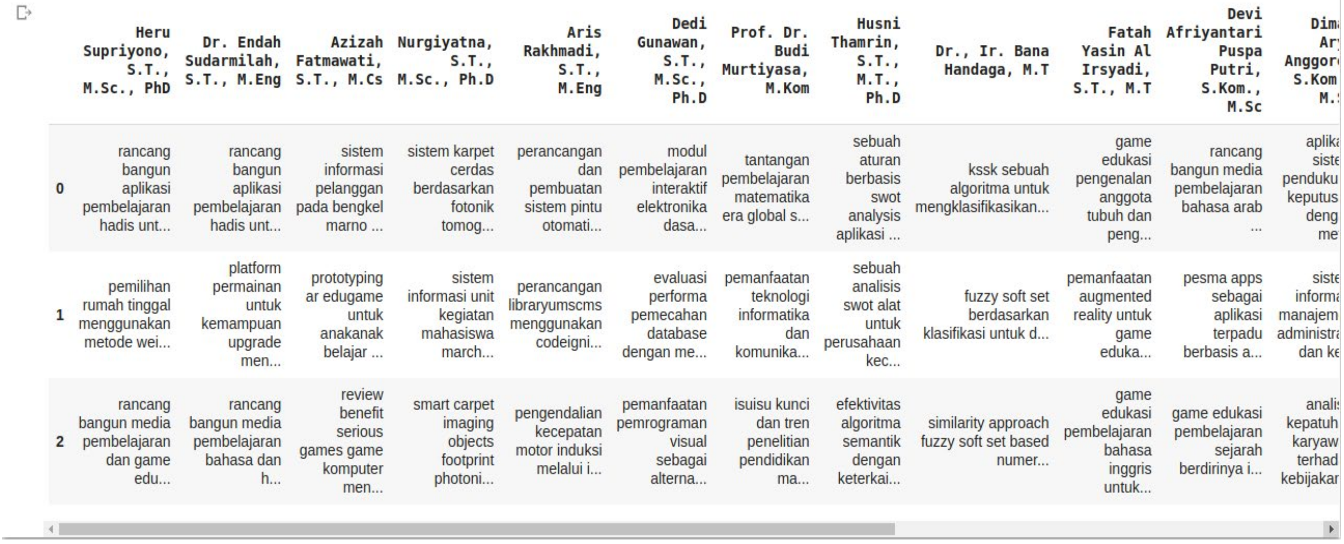
A website is created as an interface between the recommendation algorithm and users, namely students. The system is built using the Python programming language, and the interface is built using Django, a web framework built on top of Python. Here Django is used as the back-end of the web system, which is in charge of providing the data needed by the user. Testing has also been carried out on this information system [15]. A more detailed description can be found in the next section.

3. Development Process and Results

This section will detail the stages of development carried out and their results. The first is preprocessing, where the data is cleaned and formed into tokens, then the clean data is fed to the recommendation algorithm. The next stage is the implementation of this recommendation system into a web framework.

a. Preprocessing Stage

A total of 603 web scraping datasets from the Google Scholar page have 4 attributes. Then the dataset will be reduced to 2 attributes as shown in Figure 1, with the aim of simplifying the next preprocessing process. Before being used as a vector and measuring its proximity, the data is first cleaned of noise. Furthermore, the data will go through the tokenization process. Data in the form of long sentences will be broken down into words or into a token. Then after the data becomes a token, the data will enter the stopword process. Words that appear frequently in the document, and those words are listed in the stoplist, will be removed. For example the words 'at', 'and', 'to'.



	Heru Supriyono, S.T., M.Sc., PhD	Dr. Endah Sudarmilah, S.T., M.Eng	Azizah Fatmawati, S.T., M.Cs	Nurgiyatna, S.T., M.Sc., Ph.D	Aris Rakhmadi, S.T., M.Eng	Dedi Gunawan, S.T., M.Sc., Ph.D	Prof. Dr. Budi Murtiyasa, M.Kom	Husni Thamrin, S.T., M.T., Ph.D	Dr., Ir. Bana Handaga, M.T	Fatah Yasin Al Irsyadi, S.T., M.T	Devi Afriyantari Puspa Putri, S.Kom., M.Sc	Dim Ar Anggor S.Kom M.
0	rancang bangun aplikasi pembelajaran hadis unt...	rancang bangun aplikasi pembelajaran hadis unt...	sistem informasi pelanggan pada bengkel marno ...	sistem karpet cerdas berdasarkan fotonik tomog...	perancangan dan pembuatan sistem pintu otomati...	modul pembelajaran interaktif elektronika dasa...	tantangan pembelajaran matematika era global s...	sebuah aturan berbasis swot analysis aplikasi ...	kssk sebuah algoritma untuk mengklasifikasikan...	game edukasi pengenalan anggota tubuh dan peng...	rancang bangun media pembelajaran bahasa arab ...	aplik siste penduku keputus deng me
1	pemilihan rumah tinggal menggunakan metode wei...	platform permainan untuk kemampuan upgrade men...	prototyping ar edugame untuk anakanak belajar ...	sistem informasi unit kegiatan mahasiswa march...	perancangan libraryumscms menggunakan codeigni...	evaluasi performa pemecahan database dengan me...	pemanfaatan teknologi informatika dan komunika...	sebuah analisis swot alat untuk perusahaan kec...	fuzzy soft set berdasarkan klasifikasi untuk d...	pemanfaatan augmented reality untuk game eduka...	pesma apps sebagai aplikasi terpadu berbasis a...	siste informa manajem administra dan ke
2	rancang bangun media pembelajaran dan game edu...	rancang bangun media pembelajaran bahasa dan h...	review benefit serious games game komputer men...	smart carpet imaging objects footprint photoni...	pengendalian kecepatan motor induksi melalui i...	pemanfaatan pemrograman visual sebagai alterna...	isuisu kunci dan tren penelitian pendidikan ma...	efektivitas algoritma semantik dengan keterkai...	similarity approach fuzzy soft set based numer...	game edukasi pembelajaran bahasa inggris untuk...	game edukasi pembelajaran sejarah berdirinya i...	anali kepatuh karyaw terhad kebijakar

Figure 1. Dataset before and after preprocessing stage

After the data undergoes several processes until the data becomes a token, the next step is the data will be converted into a vector using the TF-IDF method with an n-gram range of 1-2 words. In linguistics and computational probability, an n-gram is a contiguous sequence of n items from a text. The N-gram will give the probability of the next word that can help in understanding the meaning of a text. The essence of this method is to calculate the TF and IDF values of each keyword against each document. The TF-IDF value can be calculated using the equation:

$$w_{i,j} = \ tf_{i,j} \times ln\ ln\ \left(\frac{N+1}{df_i+1}\right) + 1 \tag{2}$$

Note:
$tf_{i,j}$ = Many *i*-words on document *j*
N = Total documents
df_i = Many documents contain the word *i*

There are 2 abstracts as follows:
d1 = "Design and build a Hadith learning application"
d2 = "child-based learning application"

Suppose we want to calculate the weight of the word "child" in abstract d2, because the word "child" appears once in abstract d2, the calculation of the weight of the word "child" becomes:

$$w_{anak,d2} = ln\ ln\ \left(\frac{N+1}{df_i+1}\right) + 1$$
$$= \ 1 \times \left[ln\ ln\ \left(\frac{2+1}{1+1}\right) + 1\right]$$
$$= \ 1 \times 1.40$$
$$= 1.40 \tag{3}$$

After all the words are weighted, the results are as shown in table 3, so that the abstract vector d1 is [0.0, 0.0, 1.0, 1.0, 1.4, 1.4, 0.0, 1.4, 1.0, 0.0, 1.4, 1.4, 1.4, 0.0, 0.0] and the abstract vector d2 is [1.4, 1.4, 1.0, 1.0, 0.0, 0.0, 1.4, 0.0, 1.0, 1.4, 0.0, 0.0, 0.0, 1.4, 1.4]. After converting abstract d1 and abstract d2 into vectors, the next step is to measure the closeness between vectors which is discussed in more detail in subsection 3.B.

Table 3. TF-IDF calculation results

Term	tf		df	tf-idf $tf_{i,j} \times ln(\frac{N+1}{df_i+1}) + 1$	
	d1	d2		d1	d2
anak	0	1	1	0.00	1.40
anak usia	0	1	1	0.00	1.40
aplikasi	1	1	2	1.0	1.0
aplikasi pembelajaran	1	1	2	1.0	1.0
bangun	1	0	1	1.40	0.00
bangun aplikasi	1	0	1	1.40	0.00
berbasis	0	1	1	0.00	1.40
hadist	1	0	1	1.40	0.00
pembelajaran	1	1	2	1.0	1.0
pembelajaran anak	0	1	1	0.00	1.40
pembelajaran hadist	1	0	1	1.40	0.00
rancang	1	0	1	1.40	0.00
rancang bangun	1	0	1	1.40	0.00
usia	0	1	1	0.00	1.40
usia berbasis	0	1	1	0.00	1.40

b. Model Recommendation System

Data that has become a vector will be measured for its proximity to the input vector of the title, abstract and keywords of the students. In this study, the measurement of proximity between vectors is calculated using the cosine similarity method, which is the method used to measure the similarity between vectors. In the previous stage, the vector values for each lecturer's research have been represented by vector A (taken from column d1 in Table 3) and the value of each student input will be represented by vector B (taken from column d2 in Table 3).

vector A = [0.0, 0.0, 1.0, 1.0, 1.4, 1.4, 0.0, 1.4, 1.0, 0.0, 1.4, 1.4, 1.4, 0.0, 0.0]
vector B = [1.4, 1.4, 1.0, 1.0, 0.0, 0.0, 1.4, 0.0, 1.0, 1.4, 0.0, 0.0, 0.0, 1.4, 1.4]

The two vectors are processed with the cosine similarity equation:

$$\cos a = \frac{A \cdot B}{|A||B|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n}(A_i)^2} \times \sqrt{\sum_{i=1}^{n}(B_i)^2}} \quad (4)$$

$$= (0 + 0 + 1 + 1 + 0 + 0 + 0 + 0 + 1 + 0 + 0 + 0 + 0 + 0 + 0)/(\sqrt{(14.75)} \times \sqrt{(14.75)})$$

$$= \frac{3}{14.75}$$

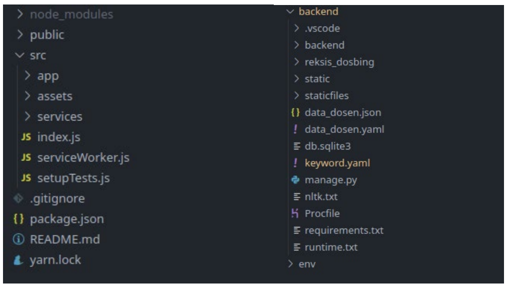
$$= 0.20$$

The measurement results of these vectors will be sorted based on their cosine similarity values. If the cosine similarity value is close to 1, then the vector has a tendency to be similar to the student input vector, and if the cosine similarity value is close to 0, then the vector has a tendency not to be similar to the student input vector. After getting the sorting results, the next step is to wrap all the cosine similarity calculation processes and sorting processes into a python class model which will later be installed into the website system.

c. System Implementation

The website system is designed as an interface for students, as well as being used as an implementation of a recommendation system to select project supervisors. The flow of the system runs in one direction starting from students entering the title, abstract and keywords. Then the system will perform computations to produce suitable recommendation lecturers based on input from students.

The website system is built using Django as the back-end and ReactJs as the front-end. Inside Django there is a project directory structure which as shown in Figure 2, each directory has its own function, the backend directory as the base project which contains the config and settings of the Django apps, then in the reksis_back-end directory functions as apps that will handle requests- requests and data processing from the front-end. Likewise in ReactJs there is also a directory structure that has its own function, the node_modules directory serves as a place to store packages needed by ReactJs such as bootstrap, multiple select, redux, etc. Then the public directory is used to store assets such as images, icons, and html files, the last is the src directory, in this directory there are javascript files that are useful for handling the components needed in making the user interface.



```
> node_modules
> public
v src
  > app
  > assets
  > services
  JS index.js
  JS serviceWorker.js
  JS setupTests.js
.gitignore
{} package.json
README.md
yarn.lock
```

```
v backend
  > .vscode
  > backend
  > reksis_dosbing
  > static
  > staticfiles
  {} data_dosen.json
  ! data_dosen.yaml
  db.sqlite3
  ! keyword.yaml
  manage.py
  nltk.txt
  Procfile
  requirements.txt
  runtime.txt
> env
```

Figure 2. ReactJs and Django directory structure

In Tables 4 and 5 there are 6 endpoints consisting of 2 POST methods and 4 GET methods, each endpoint has its own function, the rest-auth/google endpoint serves as a path to use OAuth2 google authentication to enter the website. Then the api/keyword endpoint serves to provide keywords that will be used by students, there are 2466 keywords that come from the research of the supervisor on the Google Scholar page. Furthermore, the api/rexis endpoint serves as a pathway to process data input from students, which will then be forwarded to the recommendation model in the back-end system and will be returned with the recommendation results.

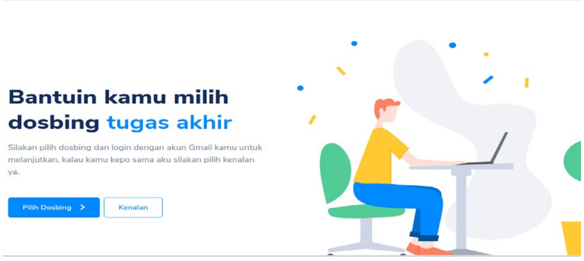
Table 4. List of endpoints on back-end

Method	Endpoint	Information
GET	api/keyword	Provide keyword data
GET	api/dosen	Provide supervisor lecturer data
POST	api/reksis	Perform data processing and provide recommendation data
POST	api/auth/google	Sign in with OAuth2 Google
GET	api/auth/logout	Log out of the system

Table 5. List of routes on the front-end

Method	Route	Information
GET	/	Show main page
GET	/reksis	Filling in data by the user and displaying recommendation results
GET	/dosbing	Displays a list of supervisors
GET	/about	Showing the about page

Figure 3 is the main page when the website is accessed by students. This page is accessed using route/, on this page there will be two buttons, the “select dosbing” button and the “acquaintance” button. the “select dosbing” button will then be redirected to google’s OAuth2 system, to authenticate.

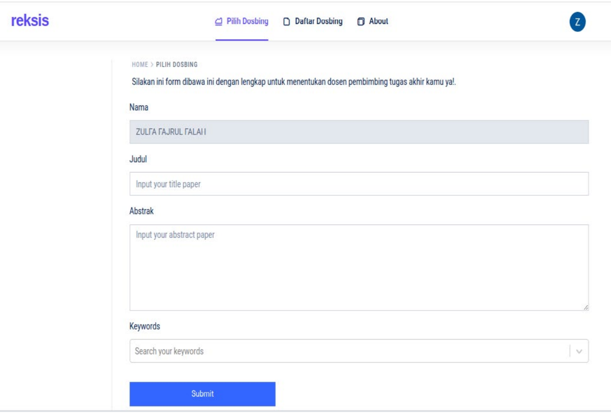


Bantuin kamu milih dosbing tugas akhir

Silakan pilih dosbing dan login dengan akun Gmail kamu untuk melanjutkan, kalau kamu kepo sama aku silakan pilih kenalan ya.

Figure 3. Website main page

The route/resis function is to handle when students successfully authenticate, as well as handle students in filling out the title, abstract, research keyword forms as shown in Figure 4. In this route students will also get results from the recommendation system. The results of this recommendation system can be seen in Figure 5.



HOME > PILIH DOSBING

Silakan ini form dibawa ini dengan lengkap untuk menentukan dosen pembimbing tugas akhir kamu ya!.

Nama: ZULFA FAJRUL FALAH

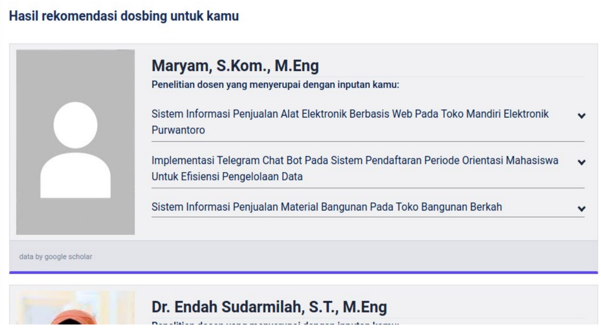
Judul: Input your title paper

Abstrak: Input your abstract paper

Keywords: Search your keywords

Submit

Figure 4. Input page title, abstract and keywords



Hasil rekomendasi dosbing untuk kamu

Maryam, S.Kom., M.Eng

Penelitian dosen yang menyerupai dengan inputan kamu:

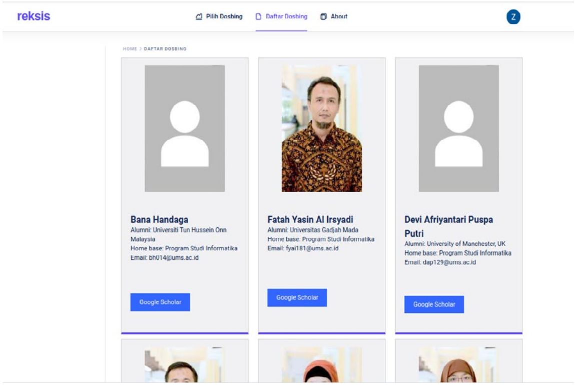
- Sistem Informasi Penjualan Alat Elektronik Berbasis Web Pada Toko Mandiri Elektronik Purwantoro
- Implementasi Telegram Chat Bot Pada Sistem Pendaftaran Periode Orientasi Mahasiswa Untuk Efisiensi Pengelolaan Data
- Sistem Informasi Penjualan Material Bangunan Pada Toko Bangunan Berkah

data by google scholar

Dr. Endah Sudarmilah, S.T., M.Eng

Figure 5. Results page of the supervisor's recommendation

Route/dosbing which is shown in Figure 6, serves to display a list of available final project supervisors, then in Figure 7 is the route / about which serves to display writings about the data used by the recommendation system, including the data sources and at a glance how the recommendation system works.



HOME > DAFTAR DOSBING

Bana Handaga
Alumni: Universiti Tun Hussein Onn Malaysia
Home base: Program Studi Informatika
Email: bh014@ums.ac.id

Google Scholar

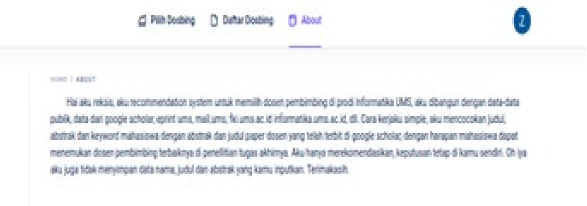
Fatah Yasin Al Irsyadi
Alumni: Universitas Gadjah Mada
Home base: Program Studi Informatika
Email: fyai181@ums.ac.id

Google Scholar

Devi Afriyantari Puspa Putri
Alumni: University of Manchester, UK
Home base: Program Studi Informatika
Email: dap129@ums.ac.id

Google Scholar

Figure 6. Final project supervisor list page



HOME / ABOUT

Hai aku reksis, aku recommendation system untuk memilih dosen pembimbing di prodi Informatika UMS, aku dibangun dengan data-data publik, data dari google scholar, eprint ums, mail.ums, fki.ums.ac.id informatika.ums.ac.id, dll. Cara kerjaku simple, aku mencocokan judul, abstrak dan keyword mahasiswa dengan abstrak dan judul paper dosen yang telah terbit di google scholar, dengan harapan mahasiswa dapat menemukan dosen pembimbing terbaiknya di penelitian tugas akhirnya. Aku hanya merekomendasikan, keputusan tetap di kamu sendiri. Oh iya aku juga tidak menyimpan data nama, judul dan abstrak yang kamu inputkan. Terimakasih.

Figure 7. Page about website

The system testing stage is the last stage that focuses on the final result and the features contained in the system. Table 6 shows the system testing with a black box which is a test where the system is directly faced with the user to interact and the system is able to respond properly and as planned.

Table 6. Black Box Testing

Function	Input	Output	Status
Main page	Access the website	Show main page	Valid
Login page	Enter Gmail and Password	Displays the page for filling out the recommendation system form	Valid
Recommendation page	Input data such as title, abstract and keywords	displaying the results of the supervisor's recommendation	Valid
Supervisor list page	Access the supervisor list page	Displays a list of supervisors	Valid

4. Conclusion

There have been several previous studies with the same topic as decision support systems using the Analytical Hierarchy Process (AHP) method in which the method is subjective depending on the expert in weighting the predetermined criteria [17]. Then there is also research conducted at the Department of Computer Science/Informatics, Faculty of Science and Mathematics, Diponegoro University with the Vector Space Model (VSM) method which is used to compare the strings of research titles between lecturers and students to build a recommendation system [18]. While the recommendation system built in this study uses the basis of comparison between title strings, abstracts, keywords from lecturer research and topics to be proposed by students with data sources from Google Scholar.

Data from web scraping from Google Scholar is still raw and rather polluted data, it still has to go through various processes before the data can really be used. Then the functionality of the recommendation system in general functions as planned. In the UMS Informatics study program, there are lecturers who have a tendency to have interests and expertise in the field of networking and this recommendation system will also recommend the lecturer if given input on topics about networking. Thus, it can be concluded that this research is in accordance with the objectives.

Recommendation systems utilizing content-based filtering method will depend heavily on the content in each item. More contents in the item, generally leads to better recommendation results. On the other hand, having more content will affect the execution time: the greater the content, the greater the time required by the system to perform calculations.

Content-based filtering also has weaknesses, because this method is very dependent on the content of the item. It is possible that this method cannot produce appropriate recommendations if the content analyzed for an item does not contain information suitable for categorization, or the item does not have enough content to categorize.

Reference

[1] P. Nagarnaik and A. Thomas, "Survey on recommendation system methods," in *2nd International Conference on Electronics and Communication Systems (ICECS)*, Coimbatore, India, Feb. 2015, pp. 1603–1608.

[2] L. Sharma and A. Gera, "A survey of recommendation system: research challenges," *International Journal of Engineering Trends and Technology.*, vol. 4 no. 5, pp. 1989–1992, 2013.

[3] J. Son and S. B. Kim, "Content-based filtering for recommendation systems using multi-attribute networks," *Expert Syst. Appl.*, vol. 89, pp. 404–412, 2017.

[4] S. Debnath, N. Ganguly, and P. Mitra, "Feature weighting in content based recommendation system using social network analysis," in *Proceeding of the 17th international conference on World Wide Web*, Beijing, China, 2008, pp. 1041–1042.

[5] D. Pyle, *Data Preparation for Data Mining.* Morgan Kaufmann, 1999.

[6] Z. Jianqiang, G. Xiaolin, and Z. Xuejun, "Deep convolution neural networks for Twitter sentiment analysis," *IEEE Access*, vol. 6, pp. 23253–23260, 2018.

[7] D. Gunawan, "Evaluasi performa pemecahan database dengan metode klasifikasi pada data preprocessing data mining," *Khazanah Informatika: Jurnal Ilmu Komputer dan Informatika*, vol. 2, no. 1. pp 10 – 13, 2016.

[8] R. K. Roul, J. K. Sahoo, and K. Arora, "Modified TF-IDF term weighting strategies for text categorization," in *14th IEEE India Council International Conference (INDICON)*, 2017.

[9] S. Robertson, "Understanding inverse document frequency: on theoretical arguments for IDF," *Journal of Documentation*, vol. 60 no. 5, pp. 503–520, 2004.

[10] Z. Yun-Tao, G. Ling, and W. Yong-Cheng, "An improved TF-IDF approach for text classification." *Journal of Zheijang University Science – A*, vol. 6, no. 1, pp. 49–55, 2005.

[11] M. Nilashi, K. Bagherifard, O. Ibrahim, H. Alizadeh, L. A. Nojeem, and N. Roozegar, "Collaborative filtering recommender systems," *Research Journal of Applied Science, Engineering, and Technology*, vol. 5, no. 16, pp. 4168–4182, 2013.

[12] R. Samuel, R. Natan, and U. Syafiqoh, "Penerapan cosine similarity dan K-Nearest Neighbor (K-NN) pada klasifikasi dan pencarian buku," *Journal of Big Data Analytic and Artificial Intelligence*, vol. 4, no. 1, pp. 9 – 14, 2018.

[13] F. Rahutomo, T. Kitasuka, and M. Aritsugi, "Semantic cosine similarity". *The 7th International Student Conference on Advanced Science and Technology (ICAST)*, vol. 4, no. 1, 2012.

[14] I. Sommerville, *Software engineering, 9th ed.* Pearson Education, 2011.

[15] Mohd. Ehmer Khan *et al.*, "Different approaches to black box testing technique for finding errors," *International Journal of Software Engineering and Applications*, vol. 2, no. 4, pp. 31–40, 2011.

[16] F. Mubarak, "Perbandingan cosine similarity dan euclidean distance pada sistem rekomendasi film menggunakan metode item based multi criteria collaborative filtering," Bachelor's thesis, Universitas Sebelas Maret, 2019.

[17] A. Abdullah and M. W. Pangestika, "Perancangan sistem pendukung keputusan dalam pemilihan dosen pembimbing skripsi berdasarkan minat mahasiswa dengan metode AHP (analytical hierarchy process) di Universitas Muhammadiyah Pontianak," *J. Edukasi Dan Penelit. Inform.*, vol. 4, no. 2, pp. 184–191, 2018

[18] N. Amalina, & S. Sutikno "Sistem rekomendasi dosen pembimbing tugas akhir berbasis text mining menggunakan vector space model", Bachelor's thesis, Universitas Diponegoro, 2017.