Vector space model, term frequency-inverse document frequency with linear search, and object-relational mapping Django on hadith data search

Ichsan Taufik, Agra, Yana Aditia Gerhana
Department of Informatics, Faculty Science and Technologi, UIN Sunan Gunung Djati, Bandung, Indonesia

Article Info

Article history:

Received Jun 2, 2024
Revised Oct 8, 2024
Accepted Oct 19, 2024

Keywords:

Hadith 9 imams
Linear search
Orm django
Search
Vector space model

ABSTRACT

For Muslims, the Hadith ranks as the secondary legal authority following the Quran. This research leverages hadith data to streamline the search process within the nine imams' compendium using the vector space model (VSM) approach. The primary objective of this research is to enhance the efficiency and effectiveness of the search process within Hadith collections by implementing pre-filtering techniques. This study aims to demonstrate the potential of linear search and Django object-relational mapping (ORM) filters in reducing search times and improving retrieval performance, thereby facilitating quicker and more accurate access to relevant Hadiths. Prior studies have indicated that VSM is efficient for large data sets because it assigns weights to every term across all documents, regardless of whether they include the search keywords. Consequently, the more documents there are, the more protracted the weighting phase becomes. To address this, the current research pre-filters documents prior to weighting, utilizing linear search and Django ORM as filters. Testing on 62,169 hadiths with 20 keywords revealed that the average VSM search duration was 51 seconds. However, with the implementation of linear and Django ORM filters, the times were reduced to 7.93 and 8.41 seconds, respectively. The recall@10 rates were 79% and 78.5%, with MAP scores of 0.819 and 0.814, accordingly.

Corresponding Author:

Yana Aditia Gerhana
Department of Informatics, Faculty Science and Technology, UIN Sunan Gunung Djati Bandung
Jl. A. H. Nasution 105, Bandung 40614 Indonesia
Email: yanagerhana@uinsgd.ac.id

1. INTRODUCTION

Information retrieval (IR), according to Salton in [1], is a system that can automatically retrieve information according to user needs from a set of data. One of the classic methods that is popular and has been tested in IR is the vector space model (VSM) [2]–[4]. VSM represents keywords in each document as a vector. The similarity values of the keywords and documents are then obtained from the angles formed by the two vectors.

In online news classification research [5], testing of the VSM method managed to achieve a high level of accuracy, namely 91.25% in four tests. The test was conducted by dividing the research object into training data and testing data. Other research showing the success of VSM in retrieving information can be found in journals [6]–[8].

To get the similarity value of two vectors in VSM [4], we first need to calculate the weight of each vector using the term weighting scheme. Methods used in word weighting include simple term frequency-inverse document frequency (TF-IDF), incremental TF-IDF, phrase-augmented TF-IDF, latent semantic index (LSI), and document-to-vector (D2V). A comparison of these methods is presented in a comparative study [9] for the case of text similarity to patent data managed by the United States Patent and Trademark Office (USPTO). For this research context, it is known that simple TF-IDF is a more suitable method because the objects used do not require complicated natural language processing (NLP) processes.

Furthermore, in the TF-IDF [5] algorithm, the weight value for each vector is obtained by combining the frequency of occurrence of a word in a document with the frequency of the inverse document (IDF) that contains the word. After that, we can rank the documents that have the greatest degree of similarity to keywords using text similarity calculations using the Cosine Similarity, Jaccard Similarity, or Euclidean Distance approaches. A Comparison of these three approaches [10] identifying top news items on news sites and measuring the similarity between the same two news items in two different languages based on the same event found that Cosine Similarity has the highest accuracy value compared to the other approaches.

Django is a web framework written in the Python programming language, rich in libraries [11]–[13]. These libraries facilitate the implementation of the VSM with tools for stopword removal, literary, numpy, and punkt, which are used for preprocessing -- removing meaningless words and punctuation marks before the TF-IDF calculation. Additionally, Django also supports ORM techniques, simplifying database queries.

Previous studies have highlighted that the VSM method tends to require less time to process searches with large amounts of data. In one study [14], the average search took 2.24 seconds, with a success rate of 81% on 75 hadith data entries. Another study [15] took 6.042 seconds, even after incorporating the Hamming distance algorithm to expedite the search.

This is caused by the VSM process of assigning weights to each word in all documents, even if the document does not contain the keyword being searched for. Thus, the more documents used for searching, the longer the weighting process will be. However, from the studies conducted, the data generated from searches using the VSM method is certain to contain at least one word of the keyword. Therefore, in this study, document filtering was carried out prior to the weighting process.

Linear search, also commonly known as sequential search, is a fairly simple and easy search algorithm. It works by comparing each element in succession, starting with the first element, until the sought element is found or all elements have been examined [16]. In contrast to Binary Search, which can only work if the data has been sorted (requiring a sorting process before the search can be carried out).

By using Linear Search for the filtering process, it can be ensured that the document to be weighted contains at least one word of the searched keyword. Additionally, Linear Search does not require prior sorting, saving time in the search process. Likewise, ORM simplifies making search queries in the filtering process, using linear search as a filter comparison. A similar study [17] explores the use of TF-IDF and VSM for efficient retrieval of medical records. However, unlike our research which focuses on performance improvements using linear and Django ORM filters, this study emphasizes the application of TF-IDF and VSM in medical document retrieval. Based on 1,000 medical record documents and tested with 20 search queries, the results showed an average precision value of 0.548 and an average recall value of 0.796.

The research object used is hadith data from the Hadith Encyclopedia of the Book of 9 Imams in the LIDWA Pusaka application. This application contains approximately 62,000 hadiths, which can be used as learning media or research objects [18]. Hadith is the second source of law for Muslims after the Quran. The use of hadith data [2] as an object of research is intended to make it easier for users who want to search hadith in the Book of Nine Imams by applying the VSM concept. As already explained, the advantage of VSM is its ability to return the most relevant information for the keywords you are searching for, which can be in the form of words, phrases or sentences. Currently, most hadith data searches can only be done using words or the concept of string matching, making it difficult for users to find the hadith in question.

The goal of this research is to improve the search function for hadiths in the LIDWA Pusaka app database using the VSM, TF-IDF with linear search, and ORM Django. This method helps users find relevant hadiths more easily by allowing searches with words, phrases, or sentences, unlike traditional methods that only use exact word matches. This makes it easier for users to access the hadiths they need for learning or research.

2. METHOD

2.1. VSM

In the journal [19], Anna and Hendini explained that the VSM is a method or algorithm that is often used for an information retrieval system. This algorithm is a model that is used to measure the similarity (Similarity terms) between a document and a query by means of term weighting. In the information retrieval system, similarities between documents are defined based on a bag-of-words representation and converted to a VSM. The relevance of a document to a query is based on the similarity between the document vector and the query vector, as seen in Figure 1.

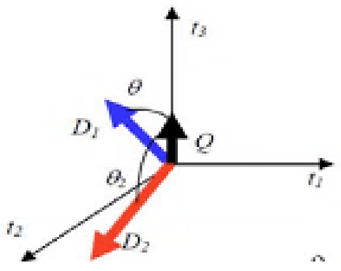


Figure 1. VSM illustration [20]

$$sim(q, d_j) = \frac{q.df_j}{|q|.|d_j|} = \frac{\sum_{i=1}^{t} w_{iq}.w_{ij}}{\sqrt[2]{\sum_{j=1}^{t}(w_{iq})^2}.\sqrt[2]{\sum_{i=1}^{t}(w_{ij})^2}} \quad (1)$$

Information:

Sim (di, dj)	= Similarity between query and document
||D1||	= Document vector length 1
||D2||	= Document vector length 2
Wij	= Term weight in the document
Wiq	= Query weight in the document

The basic concept of the VSM is to calculate the distance between documents and then sort them based on their closeness. The workings of the VSM begin with case folding, stopword removal, stemming, and tokenizing, which are stages of cutting the input string based on each word and breaking the document into word frequency tables. All the words in the document are combined into one, which is called a term. Each document is displayed as a vector, which will be compared with the terms that have been formed. Similarity analysis to measure the similarity of documents is done by calculating the cosine [21] of the distance between the documents.

2.2. Linear search

Linear search is one of the most popular and simple search algorithms [22], [23]. The search technique involves scanning data sequentially from start to finish based on the desired keywords. A significant advantage of this algorithm is that if the target data is located near the beginning, it can be found quickly. Conversely, a major disadvantage is that if the target data is near the end, the search can take a long time [24].

2.3. ORM

ORM is a programming method used to convert data from an object-oriented programming (OOP) language environment to a relational database environment [25], [26]. The advantages of using ORM include,

– It can speed up program development, such as query repetition, because tables are represented as objects.
– Data access becomes more abstract and portable based on the database vendor.
– It supports encapsulating business rules at the data access layer.
– It is able to generate boilerplate code for basic CRUD functions.

2.4. System analysis

The system to be built is a website-based application using Django [12] as the framework. The system architecture is shown in Figure 2. The client or user makes an HTTP request for a hadith search via a browser, which is then forwarded to a web server such as Apache, IIS, Nginx, Tomcat, and so on. The server processes the request and returns it as an HTTP response according to the user's request (in this research, it is a list of hadiths). Figure 3 shows the system flow to be developed in this study.

2.5. Confussion matrix

The confusion matrix is a crucial tool in evaluating of classification models in machine learning. It provides a detailed breakdown of the model’s performance by comparing actual and predicted classifications. The matrix consists of four key components: True positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). These components help calculate various performance metrics such as accuracy, precision, recall, and F1 score, which are essential for understanding a model's strengths and weaknesses. For instance, in a medical diagnosis scenario, a high recall indicates the model's effectiveness in identifying patients with a disease, while precision highlights its ability to avoid false alarms [27].

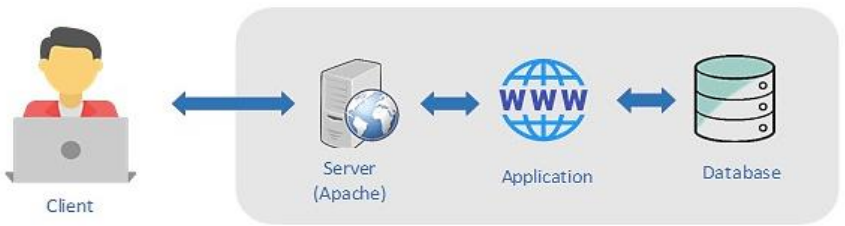


Figure 2. System arhitecture

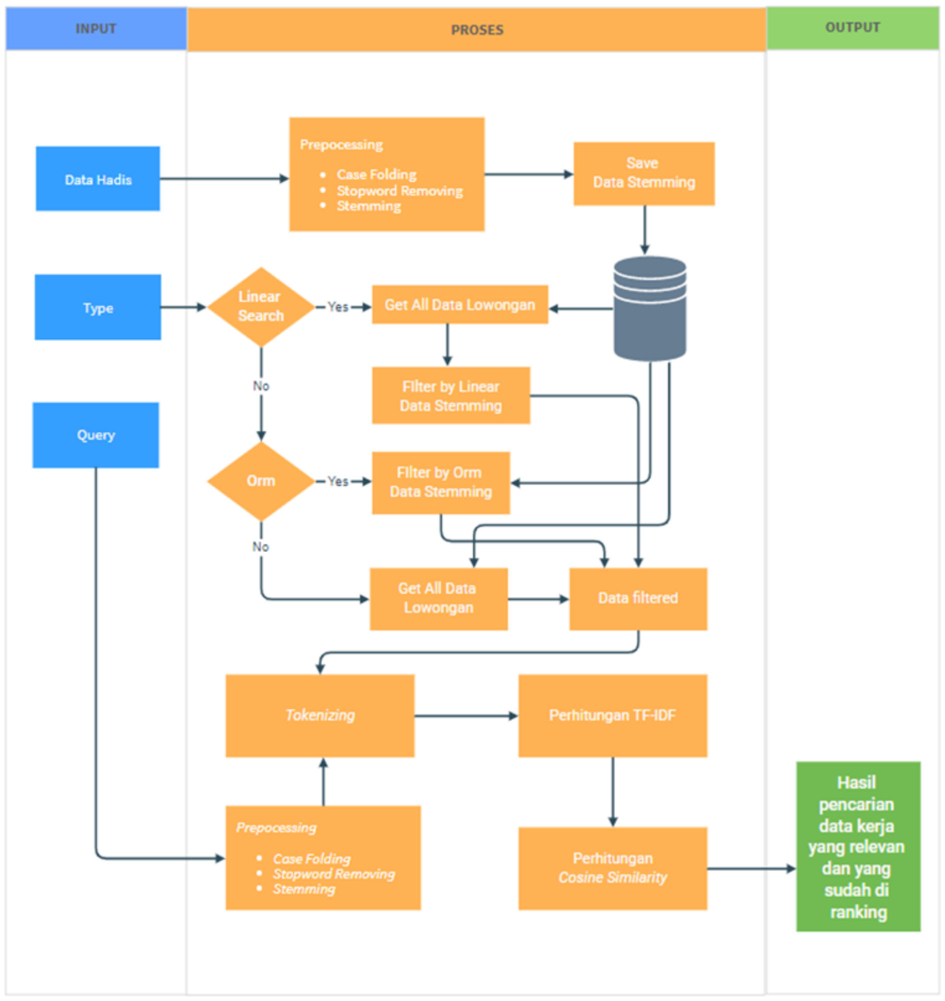
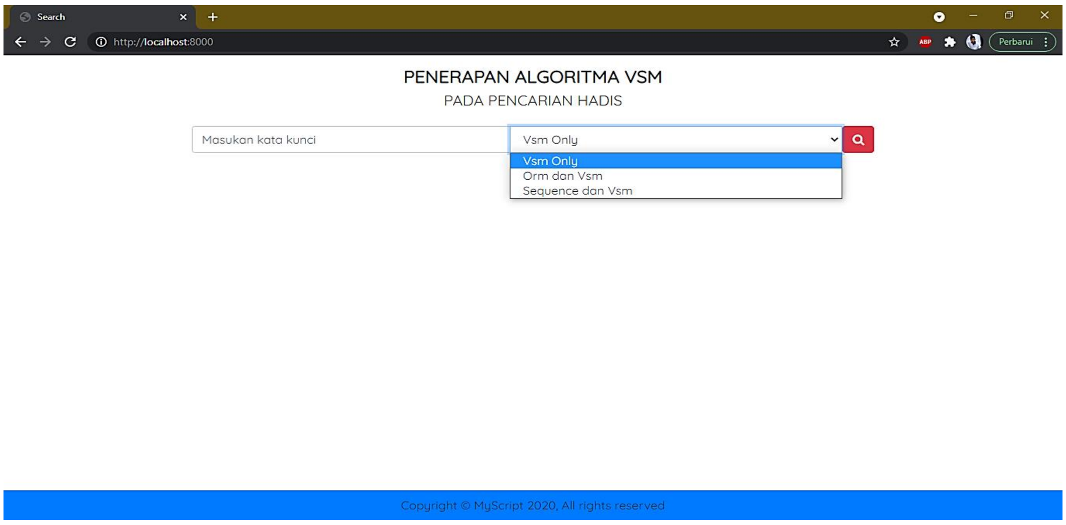


Figure 3. System flowchart

3. RESULT AND DISCUSSION

3.1. Implementation

In its implementation, this website has three interface pages, including the home page, search page, and detail page as seen in Figures 4 to 6. The home page serves as the main entry point, providing an overview and easy navigation. The search page allows users to find specific content quickly and efficiently. The detail page offers in-depth information and additional resources related to the selected content.



PENERAPAN ALGORITMA VSM

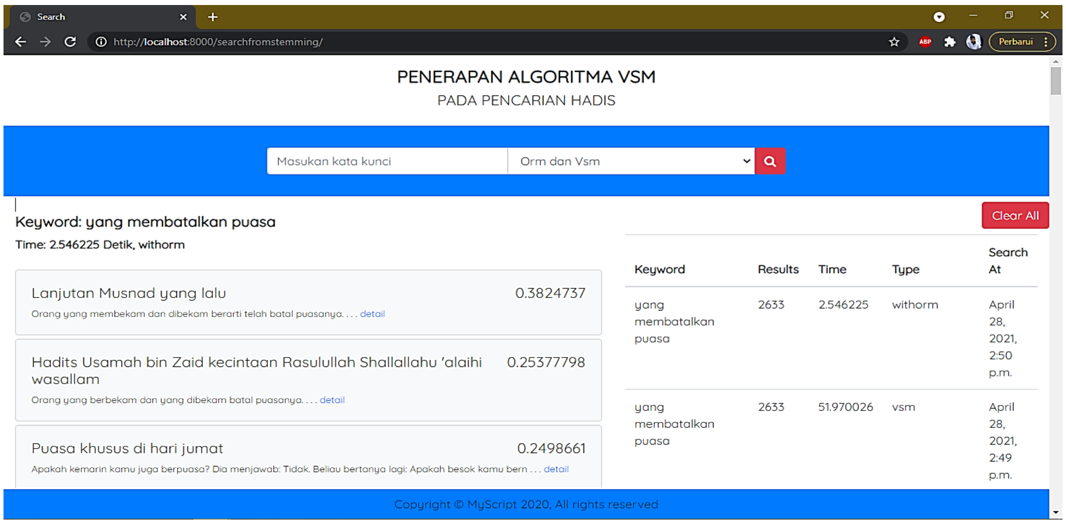
PADA PENCARIAN HADIS

Masukan kata kunci | Vsm Only

Vsm Only
Orm dan Vsm
Sequence dan Vsm

Copyright © MyScript 2020, All rights reserved

Figure 4. Home page



PENERAPAN ALGORITMA VSM

PADA PENCARIAN HADIS

Masukan kata kunci | Orm dan Vsm

Clear All

Keyword: yang membatalkan puasa

Time: 2.546225 Detik, withorm

Lanjutan Musnad yang lalu 0.3824737
Orang yang membekam dan dibekam berarti telah batal puasanya. . . . detail

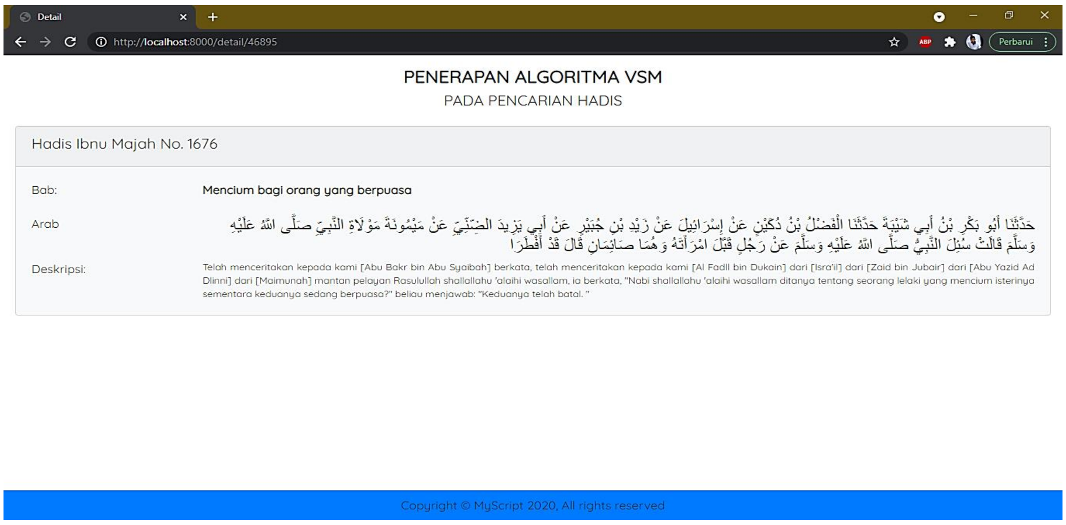
Hadits Usamah bin Zaid kecintaan Rasulullah Shallallahu 'alaihi wasallam 0.25377798
Orang yang berbekam dan yang dibekam batal puasanya. . . . detail

Puasa khusus di hari jumat 0.2498661
Apakah kemarin kamu juga berpuasa? Dia menjawab: Tidak. Beliau bertanya lagi: Apakah besok kamu bern . . . detail

Keyword	Results	Time	Type	Search At
yang membatalkan puasa	2633	2.546225	withorm	April 28, 2021, 2:50 p.m.
yang membatalkan puasa	2633	51.970026	vsm	April 28, 2021, 2:49 p.m.

Copyright © MyScript 2020, All rights reserved

Figure 5. Search page



PENERAPAN ALGORITMA VSM

PADA PENCARIAN HADIS

Hadis Ibnu Majah No. 1676

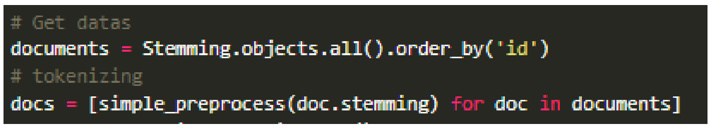
Bab:	**Mencium bagi orang yang berpuasa**
Arab	حَدَّثَنَا أَبُو بَكْرِ بْنُ أَبِي شَيْبَةَ حَدَّثَنَا الْفَضْلُ بْنُ دُكَيْنٍ عَنْ إِسْرَائِيلَ عَنْ زَيْدِ بْنِ جُبَيْرٍ عَنْ أَبِي يَزِيدَ الضِّنِّيِّ عَنْ مَيْمُونَةَ مَوْلَاةِ النَّبِيِّ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ قَالَتْ سُئِلَ النَّبِيُّ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ عَنْ رَجُلٍ قَبَّلَ امْرَأَتَهُ وَهُمَا صَائِمَانِ قَالَ قَدْ أَفْطَرَا
Deskripsi:	Telah menceritakan kepada kami [Abu Bakr bin Abu Syaibah] berkata, telah menceritakan kepada kami [Al Fadll bin Dukain] dari [Isra'il] dari [Zaid bin Jubair] dari [Abu Yazid Ad Dlinni] dari [Maimunah] mantan pelayan Rasulullah shallallahu 'alaihi wasallam, ia berkata, "Nabi shallallahu 'alaihi wasallam ditanya tentang seorang lelaki yang mencium isterinya sementara keduanya sedang berpuasa?" beliau menjawab: "Keduanya telah batal. "

Copyright © MyScript 2020, All rights reserved

Figure 6. Detail page

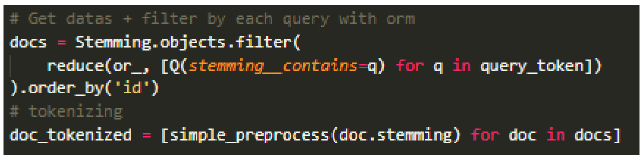
On the search page, this algorithm is applied. Figures 7 to Figure 15 are details of the programming syntax using the Python language. These figures illustrate the step-by-step implementation of the algorithm. Additionally, they highlight key functions and methods used to achieve the desired search functionality.

a) Function gets documents and tokenizing



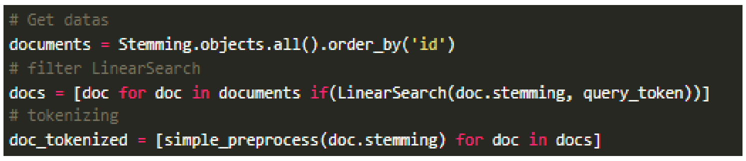
```
# Get datas
documents = Stemming.objects.all().order_by('id')
# tokenizing
docs = [simple_preprocess(doc.stemming) for doc in documents]
```

Figure 7. Get documents and tokenizing VSM



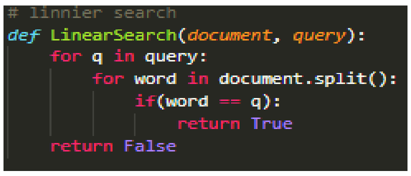
```
# Get datas + filter by each query with orm
docs = Stemming.objects.filter(
    reduce(or_, [Q(stemming__contains=q) for q in query_token])
).order_by('id')
# tokenizing
doc_tokenized = [simple_preprocess(doc.stemming) for doc in docs]
```

Figure 8. Get documents and tokenizing VSM with filter ORM



```
# Get datas
documents = Stemming.objects.all().order_by('id')
# filter LinearSearch
docs = [doc for doc in documents if(LinearSearch(doc.stemming, query_token))]
# tokenizing
doc_tokenized = [simple_preprocess(doc.stemming) for doc in docs]
```

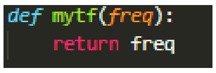
Figure 9. Get documents and tokenizing with filter linear search



```
# linnier search
def LinearSearch(document, query):
    for q in query:
        for word in document.split():
            if(word == q):
                return True
    return False
```

Figure 10. Linear search function

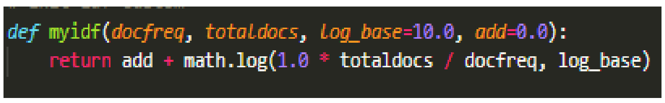
b) Term frequency



```
def mytf(freq):
    return freq
```

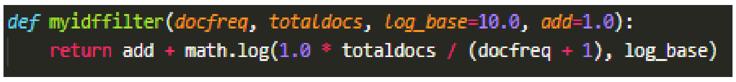
Figure 11. Term frequency

Inverse document frequency



```
def myidf(docfreq, totaldocs, log_base=10.0, add=0.0):
    return add + math.log(1.0 * totaldocs / docfreq, log_base)
```

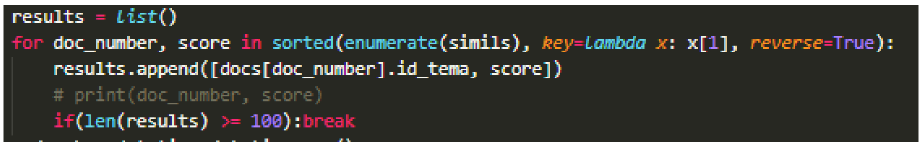
Figure 12. IDF without filter (VSM only)



```
def myidffilter(docfreq, totaldocs, log_base=10.0, add=1.0):
    return add + math.log(1.0 * totaldocs / (docfreq + 1), log_base)
```

Figure 13. IDF with filter (linear or ORM)

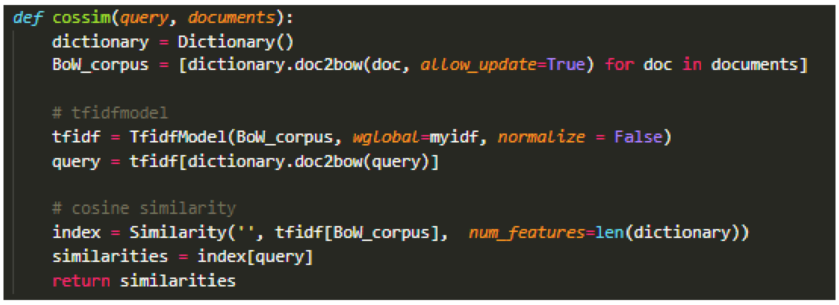
c) Ranking of the 100 highest data



```
results = list()
for doc_number, score in sorted(enumerate(simils), key=lambda x: x[1], reverse=True):
    results.append([docs[doc_number].id_tema, score])
    # print(doc_number, score)
    if(len(results) >= 100):break
```

Figure 14. Ranking of the 100 highest data

d) TF-IDF and cosine similarity



```
def cossim(query, documents):
    dictionary = Dictionary()
    BoW_corpus = [dictionary.doc2bow(doc, allow_update=True) for doc in documents]

    # tfidfmodel
    tfidf = TfidfModel(BoW_corpus, wglobal=myidf, normalize = False)
    query = tfidf[dictionary.doc2bow(query)]

    # cosine similarity
    index = Similarity('', tfidf[BoW_corpus],  num_features=len(dictionary))
    similarities = index[query]
    return similarities
```

Figure 15. TF-IDF and cosine Similarity

3.2. Testing

Algorithm testing was conducted to determine the performance of the VSM algorithm after adding a filter by using the recall@k (recall top k), precision@k (precision top k), and mean average precision (MAP) approaches with a value of k=10 to the performance of the VSM algorithm without filters. In this research, testing was carried out using 20 keywords in the form of words, phrases, and sentences for each filter method. The use of TF-IDF in testing results transformed textual data into numerical values, highlighting the importance of each term within the dataset. The TF-IDF values were crucial in creating document vectors for the VSM, which were then used to compute similarities between documents and search queries, resulting in performance metrics such as recall@k, precision@k, and MAP. The value of k=10 was chosen to evaluate the top 10 results returned by the VSM algorithm, providing practical insights into the algorithm's effectiveness in returning the most relevant results within a manageable number of top results. The selection of 20 keywords ensured a diverse set of search queries, including words, phrases, and sentences, to comprehensively evaluate the algorithm's performance across different query complexities and lengths, ensuring robustness and generalizability of the results.

Algorithm testing with recall@10 using Linear Search and ORM filters is valued 79%. This means that about 8 of the top 10 hadiths predicted by VSM appear in searches using the filter in the top 10 orders. MAP evaluates the average precision across several keywords, with a value of 0.814 for the VSM method with the ORM filter and 0.819 for the VSM method with Linear Search filter. It can be concluded that both methods work well in terms of sorting.

4. CONCLUSION

From the results of the Black Box testing that has been carried out, the system search using the VSM method, both with the Linear Search filter and the Django ORM, has been successfully implemented and works well functionally, as expected. The testing confirmed that the search functionality is robust and efficient. Additionally, the integration with Django ORM ensures seamless database interactions. Overall, the system meets the desired performance criteria and user expectations. VSM performance in this research was carried out with 20 sample queries. For searches using the VSM method alone, an average time of 51 seconds was obtained with 62,169 hadith documents. This is quite long for a search method. Therefore, VSM with filters can be used as an option. The filter here is applied before carrying out similarity calculations because it can reduce the accumulation of words during weighting, thus speeding up the process. Tests such as recall@k, precision@k, and MAP tests were also carried out to assess the relevance of the relevance of the algorithm after applying the filter. The results show that adding filters to VSM can work well and is relevant. Data filters can be developed further by exploring other word filter algorithms. Additionally, the VSM in this research performs exceptionally well for searches using keywords in words, phrases, or sentences. However, it does not include searches using synonyms for these keywords. This can be addressed through a query expansion approach to increase the relevance of search results or by other methods.

ACKNOWLEDGEMENTS
The authors thank the Center for Research and Publications, Institute for Research and Community Service at UIN Sunan Gunung Djati Bandung. In most cases, sponsor and financial support acknowledgments.

REFERENCES
[1] M. Mayeni, W. W. Winarno, and A. Sunyoto, “Information retrieval of thesis documents to determine their similarity to existing research" *(Indoensian)* "Information retrieval dokumen tesis untuk mengetahui kemiripannya dengan penelitian yang telah ada,” *Jurnal TRANSFORMASI*, vol. 12, no. 2, pp. 105–115, 2016.
[2] S. E. Pratama, W. Darmalaksana, D. Sa’adillah Maylawati, H. Sugilar, T. Mantoro, and M. A. Ramdhani, “Weighted inverse document frequency and vector space model for hadith search engine,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 18, no. 2, pp. 1004–1014, 2020, doi: 10.11591/ijeecs.v18.i2.pp1004-1014.
[3] A. Fauzi and G. Ginabila, “Online news classification using TF-IDF weighting and cosine similarity,” “Online news classification using TF-IDF weighting and cosine similarity,” *(Indonesian)* “Information retrieval system pada file pencarian dokumen tesis berbasis text menggunakan metode vector space model,” *Jurnal Pilar Nusa Mandiri*, vol. 15, no. 1, pp. 41–46, 2019, doi: 10.33480/pilar.v15i1.61.
[4] J. E. Dobson, “Vector hermeneutics: On the interpretation of vector space models of text,” *Digital Scholarship in the Humanities*, vol. 37, no. 1, pp. 81–93, 2022, doi: 10.1093/llc/fqab079.
[5] B. Herwijayanti, D. E. Ratnawati, and L. Muflikhah, “Online news classification using TF-IDF weighting and cosine similarity,” *(Indonesian)* “Klasifikasi berita online dengan menggunakan pembobotan TF-IDF dan cosine similarity,” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 2, no. 1, pp. 306–312, 2018, [Online]. Available: http://j-ptiik.ub.ac.id.
[6] T. Hariguna, B. Fandy, S. Utomo, J. Pol, S. Watumas, and P. J. Pol, “Implementation of information retrieval Indonesian text documents using the vector space model,” 2012, [Online]. Available: https://www.researchgate.net/publication/304605781.
[7] S. Salmon, D. Paseru, and V. Kumenap, “Implementasi metode vector space model pada search engine perpustakaan,” *Prosiding SISFOTEK*, vol. 4, no. 1, pp. 84–92, 2020, [Online]. Available: http://seminar.iaii.or.id/index.php/SISFOTEK/article/view/158.
[8] D. B. Lomet, L. Gravano, A. Halevy, and G. Weikum, “Editorial board (Associate Editors),” *Journal of the Franklin Institute*, vol. 358, no. 4, p. i, 2021, doi: 10.1016/s0016-0032(21)00098-3.
[9] O. Shahmirzadi, A. Lugowski, and K. Younge, “Text similarity in vector space models: A comparative study,” *Proceedings - 18th IEEE International Conference on Machine Learning and Applications, ICMLA 2019*, pp. 659–666, 2019, doi: 10.1109/ICMLA.2019.00120.
[10] R. Singh and S. Singh, “Text similarity measures in news articles by vector space model using NLP,” *Journal of The Institution of Engineers (India): Series B*, vol. 102, no. 2, pp. 329–338, 2021, doi: 10.1007/s40031-020-00501-5.
[11] S. Jaiswal and R. Kumar, *Learning Django web development : from idea to prototype, a learner’s guide for web development wih the Django application framework*. 2015.
[12] K. M. Vamsi, P. Lokesh, K. N. Reddy, and P. Swetha, “Visualization of real-world enterprise data using Python Django framework,” *IOP Conference Series: Materials Science and Engineering*, vol. 1042, no. 1, p. 012019, 2021, doi: 10.1088/1757-899x/1042/1/012019.
[13] A. N. Rahimah, D. S. Rusdianto, and M. T. Ananta, "Development of a web-based reading room management system using the Django framework (Case study: reading room, Faculty of Computer Science, Brawijaya University)" *(Indonesian)* “Pengembangan sistem pengelolaan ruang baca berbasis web dengan menggunakan Django framework (Studi kasus: ruang baca fakultas ilmu komputer Universitas Brawijaya),” *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer,* vol. 2, no. 5, pp 4439-4446, 2019.
[14] I. Taufik, M. Jaenudin, F. U. Badriyah, B. Subaeki, and O. T. Kurahman, “The search for science and technology verses in qur’an and hadith,” *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 2, pp. 1008–1014, 2021, doi: 10.11591/eei.v10i2.2629.
[15] V. Basmalah Wicaksono, R. Saptono, and S. Widya Sihwi, “Comparative analysis of vector space model and weighted tree similarity methods with cosine similarity in the case of searching for basic treatment guideline information at community health centers" *(Indonesian)* "Analisis perbandingan metode vector space model dan weighted tree similarity dengan cosine similarity pada kasus pencarian informasi pedoman pengobatan dasar di puskesmas,” *Jurnal Teknologi & Informasi ITSmart*, vol. 4, no. 2, p. 73, 2016, doi: 10.20961/its.v4i2.1768.
[16] A. Sonita and M. Sari, ““Implementation of sequential searching algorithm for searching letter numbers in electronic archive systems" *(Indonesian)* "Implementasi algoritma sequential searching untuk pencarian nomor surat pada sistem arsip elektronik,” *Pseudocode*, vol. 5, no. 1, pp. 1–9, 2018, doi: 10.33369/pseudocode.5.1.1-9.

[17] L. Heryawan, D. Novitaningrum, K. R. Nastiti, and S. N. Mahmudah, “Medical record document search with TF-IDF and vector space model (VSM),” *International Journal on Advanced Science, Engineering and Information Technology*, vol. 14, no. 3, pp. 847–852, 2024, doi: 10.18517/ijaseit.14.3.19606.

[18] H. H. Batubara, "Utilization of the encyclopedia of hadith books of the 9 imams as a medium and source for learning hadith," *(Indonesian)* “Pemanfaatan ensiklopedi hadis kitab 9 imam sebagai media dan sumber belajar hadis,” *Muallimuna : Jurnal Madrasah Ibtidaiyah*, vol. 2, no. 2, p. 63, 2017, doi: 10.31602/muallimuna.v2i2.769.

[19] Anna and A. Hendini, “Implementation of vector space model in karaoke machine search system,” *(Indonesian)* “Implementasi vector space model pada sistem pencarian mesin karaoke,” *Evolusi : Jurnal Sains dan Manajemen*, vol. 6, no. 1, 2018, doi: 10.31294/evolusi.v6i1.3535.

[20] Bania Amburika, Y. H. Chrisnanto, and W. Uriawan, "Vector space model (VSM) technique in determining the handling of the impact of online games on children," *(Indonesian)* “Teknik vector space model (VSM) dalam penentuan penanganan dampak game online pada anak,” *Prosiding SNST ke-7 Tahun 2016*, vol. 1, no. 1, pp. 10–27, 2016, [Online]. Available: http://cogsys.imm.dtu.dk/thor/projects/multimedia/textmining/node5.html.

[21] H. Hu *et al.*, “Horizontal and vertical crossover of sine cosine algorithm with quick moves for optimization and feature selection,” *Journal of Computational Design and Engineering*, vol. 9, no. 6, pp. 2524–2555, 2022, doi: 10.1093/jcde/qwac119.

[22] S. Saha, M. K.Bhaumik, and S. Das, “A new modified linear search algorithm,” *International Journal of Mathematics Trends and Technology*, vol. 65, no. 12, pp. 148–152, 2019, doi: 10.14445/22315373/ijmtt-v65i12p516.

[23] L. Goyal, K. Sharma, T. Rai, B. Gupta, and N. Arora, “Left-right mid way linear searching algorithm,” *International Journal of Computer Applications*, vol. 177, no. 45, pp. 22–26, 2020, doi: 10.5120/ijca2020919939.

[24] A. Muhazir, M. Fakhriza, and E. Sutejo, "Implementation of sequential method in searching for goods distribution in cargo integration system," *(Indonesian)* “Implementasi metode sequential dalam pencarian pendistribusian barang pada cargo integration sistem,” *Jurnal dan Penelitian Teknik Informatika*, vol. 2, no. 2, pp. 24–30, 2017.

[25] O. Lukyanchikov, E. Pluzhnik, S. Payain, and E. Nikulchev, “Using object-relational mapping to create the distributed databases in a hybrid cloud infrastructure,” *International Journal of Advanced Computer Science and Applications*, vol. 5, no. 12, 2014, doi: 10.14569/ijacsa.2014.051208.

[26] A. Joshi and S. Kukreti, “Object relational mapping in comparison to traditional data access techniques,” *International Journal of Scientific & Engineering Research*, vol. 5, no. 6, pp. 540–543, 2014, [Online]. Available: http://www.ijser.org.

[27] K. M. Ting, “Confusion matrix,” *Encyclopedia of Machine Learning*, pp. 209–209, 2011, doi: 10.1007/978-0-387-30164-8_157.

BIOGRAPHIES OF AUTHORS

Ichsan Taufik is a lecturer in Department Informatics at UIN Sunan Gunung Djati Bandung. Ichsan has graduated from Informatics Department at IAIN Sunan Gunung Djati. He received his M.T. Degree from Institut Teknologi Bandung. Research areas: Information System, Algorithm and Programming, Software Engineering. He can be contacted at email: ichsan@uinsgd.ac.id.

Agra received his first degree from UIN Sunan Gunung Djati Bandung. His academic qualification is S.T. His working at consultant IT in Jakarta. He can be contacted at email: agra0108@gmail.com.

Yana Aditia Gerhana received his first degree from Sekolah Tinggi Teknologi Garut, Informatic, Garut, in 2003. He has also Master degree from STMIK Likmi, Information System, Bandung, in 2009. The Dr. degree from the Vocation and Technology Education in Universitas Pendidikan Indonesia (UPI), Indonesia in 2016. He is currently a computer science lecuturer. His main research interests focus on Information System, Multimedia, Vocational and Tehcnology Educataion, Information Retrieval Data Mining, and Text Mining. He can be contacted at email: yanagerhana@uinsgd.ac.id.