ANDROID BASED HATE SPEECH SEARCH APPLICATIONS USING TF-IDF ALGORITHM AND VECTOR SPACE MODELS

[1]R. DIMAS ADITYO, [2]TRI YOGI RIZQI, [3]MAS NURUL HAMIDAH

[1]Informatics Engineering Study Program, Faculty of Engineering

Bhayangkara University – Surabaya

e-mail: [1]dimas@ubhara.ac.id, [2]helloyogiriz@gmail.com,
[3]mas.nurul@ubhara.ac.id

ABSTRACT

The freedom in the use of social media becomes a common means in voicing opinions and expressions by issuing words or phrases of blasphemy (Hate Speech) in social media like Facebook. Hate speech and blasphemous words are easily spread across social and general media not found that have transgressed the limits and can trace the unrest. To find out what the users of social media especially Facebook we have issued words or phrases of blasphemy on the basis of this report occurred. The study was conducted by using TF-IDF weighting algorithm and VSM (Vector Space Model) calculation, while the data used was the 30 users post using Graph API. This research resulted in search with recall value: 2/2 = 1 which means the result of the relevant document and the precision value of the search result is 2/13 = 0,154 which means result also raises irrelevant search result. Percentage of hate speech equivalence value was 43%.

Keyword: *Hate speech, Android, Graph API, TF-IDF, Vector Space Model.*

I. INTRODUCTION

The development of information technology today has become very fast and become part of the lives of some people, especially social media information technology that allows people to share their news and thoughts in cyberspace in written form and can be read by many people. The use of social media now no longer has an age limit, ranging from children, adolescents, to adults today who have used social media to interact with each other.

Indonesia itself is a democratic country that frees every community to express their thoughts and opinions, and is free to express as long as it is still in the right corridor and still obeys applicable laws. But this freedom sometimes causes polemics because everyone has their own opinions, some people become excessive in voicing their opinions and expressions by issuing blasphemous words or sentences (Hate Speech) on social media such as Facebook.

Anxiety will be very easy to arise, even hate speech based on sara is easier to trigger mass riots. Prevention efforts by the state through law enforcement officials need to be done, no less important is the step to educate the public about the dangers of such behavior and its impact on the socio-economic life of the community (Rohman, 2016). Hate speech, blasphemy, defamation, racism and other harsh sentences can easily spread on social media and generally the perpetrators do not realize that they have crossed the line and violated applicable laws.

Thus, the description above becomes a consideration for making the title: "*Android-based Hate Speech Search Application Using TFIDF Algorithm and Vector Space Model*".

II LITERATURE REVIEW

Giat Karyono and Fandy Setyo Utomo, 2012, Information Retrieval System in Indonesian Language Text Documents with the Vector Space Retrieval Model Method.

The number of documents files in the form of text that is stored makes someone having trouble finding and getting the information you want because they have to look at the documents one by one to get the right information, and of course it takes a long time. So to overcome this we need a search engine that can determine and find relevant documents in accordance with user queries. In order to be realized perfectly, the Vector Space Retrieval Model method is used which is based on each document and reduces the term dominance that often appears in various documents. The results of this study are represented by the order / ranking of documents similarity to the query.

III SYSTEM DESIGN

3.1 Android

Android is an operating system for Linux-based mobile devices that includes an operating system, middleware and applications. Android provides an open platform for developers to create their applications.

3.2 Rest API

In the article that was written by Eka Y Saputra, explained that, REST (REpresentational State Transfer) is an architectural communication method that is often applied in the development of web-based services. REST architecture, which is generally run via HTTP (Hypertext Transfer Protocol), involves the process of reading certain web pages that contain an XML or JSON file. This file describes and contains the content to be presented. After going through a certain definition process, consumers will be able to access the intended application interface.

The specialty of REST lies in the interaction between the client and server which is facilitated by a number of operational types (verbs) and Universal Resource Identifiers (URIs) that are unique to each resource. Each verb - GET, POST, PUT and DELETE - has a special operational meaning to avoid ambiguity.

REST is often used in mobile applications, social networking websites, mashup tools, and automated business processes. The decoupled REST architecture and the light communication burden between producers and consumers makes it popular in the cloud-based API world, as presented by Amazon, Microsoft and Google.

Web-based services that use such REST architectures are called RESTful APIs (Application Programming Interfaces) or REST APIs.

3.3 Graph API

Graph API or Graph API is the main way to get data inside and outside the Facebook platform. The Graph API is a low-level, HTTP-based API that can be used by applications to program data queries, post new stories, manage ads, upload photos, and perform various other types of tasks.

The name of the Graph API is rooted in the idea of "social graph", which is a representation of information on Facebook. This API consists of:

- Nodes - are basically individual objects such as Users, Photos, Pages or Comments
- Edge - the relationship between a collection of objects and a single object, such as a number of Photos on a Page or a number of Comments on a Photo
- Column - data about an object, such as User's birthday or Page name

Usually you use nodes to get data about a particular object, use edges to get a collection of objects in a single object, and use columns to get data about a single object or each object contained in a collection.

3.4 Hate Speech

Hate speech is an act of communication carried out by an individual or group in the form of provocation, provocation, or insults to other individuals or groups in terms of various aspects such as race, color, ethnicity, gender, disability, sexual orientation, citizenship, religion, and so on.

According to R. Susilo, hate speeches or insults to one individual of its kind are divided into 6 types, namely:

1. Oral insults (smaad)
2. Menista with letters / written (smaadschirft)
3. Slander (laster)
4. Slight humiliation (eenvoudige belediging)
5. Defamating (lasterlijke aanklacht)
6. Defamation allegations (lasterlijke verdachtmaking)

According to critics, hate speech is a modern example of the Newspeak novel, when Hate speech is used to give tacit criticism to social policies that are poorly implemented and rushed as if the policies look politically correct.

According to the law, hate speeches are words, behavior, writing, or performances that are prohibited because they can trigger acts of violence and prejudice on the part of the perpetrators. The statement or the victim of the action. In Indonesia, there are articles that regulate acts of hate speech against individuals contained in Book I of the Criminal Code Chapter XVI, especially in Article 310, Article 311, Article 315, Article 317, and Article 318 of the Criminal

Code. Meanwhile, defamation or defamation of a government, organization, or group is regulated in specific articles, namely:

1. Humiliation of the head of a foreign country (Article 142 and Article 143 of the Criminal Code)
2. Supervision of a group of residents / groups / organizations (Article 156 and Article 157 of the Criminal Code)
3. Humiliation of religious employees (Article 177 of the Criminal Code)
4. Humiliation of power in Indonesia (Article 207 and article 208 of the Indonesian Criminal Code)

3.5 Term Frequency - Inverse Document Frequency (TFIDF).

TF-IDF method is a way to give the weight of the relationship of a word (term) to the document (Robertson, 2005). This method combines two concepts for weight calculation, namely the frequency of occurrence of a word in a particular document and the inverse frequency of the document containing the word.

The frequency with which words appear in a given document indicates how important that word is in the document. The frequency of documents containing the word indicates how common the word is. So the weight of the relationship between a word and a document will be high if the frequency of the word is high in the document and the overall frequency of the document containing the word is low in the document. The VSM calculation process goes through the term frequency (tf) calculation stage using the equation.

$$tf = tfij$$

Where:

tf : word frequency

Where tf is the term frequency, and $tf_{i,j}$ is the number of occurrences of the i *term* in the d_j document. After the results of tf are obtained, a frequency log weight calculation ($tf\_wt_{t,d}$) is calculated using the following equation:

$$tf\_wt_{t,d} = \begin{cases} 1+\log_{10}(tf_{t,d}) & , \text{jika } tf_{t,d} > 0 \\ 0 & , \text{jika } tf_{t,d} \le 0 \end{cases}$$

Where:

$tf\_wt_{t,d}$: frequency log weights

$tf_{t,d}$: word frequency

After the results of $tf\_wt_{t,d}$ are performed, the next calculation is Inverse Document Frequency (***idf***), using the following equation:

$$idf_i = log\ (\frac{N}{dfi})$$

Where:

idf_i : document frequency

N : number of documents

df_i : document

With idf_i is the inverse document frequency, N is the number of documents in the system, and df_i is the number of documents in the collection where the *i term* appears in it, then the idf_i calculation is used to find out the number of *terms* searched (df_i) that appear in other documents in the database (corpus).

After searching for IDF values, the next step is to calculate the *term frequency Inverse Document Frequency (tf-idf),* using the following equation:

$$w_{t,d} = \left(1+\log_{10}(tf_{t,d})\right)\times\log_{10}\left(\frac{N}{df_t}\right)$$

Where:

Wt,d : *tf-idf* weights

tf : word frequency

N : number of documents

dfi : document

With *Wt,d* is the weight of *tf-idf*, *N* is the number of documents taken by the system, $tf_{i,j}$ is the number of occurrences *ti* in *dj* documents, and dfi is the number of documents in the collection where *ti* appears in it. The *tf-idf* weight is calculated to get a multiplication weight or a combination of the frequency log weight ($tf\_wt_{t,d}$) and Inverse Document Frequency (*idfi*).

Distance calculation using query equations and documents, using equations.

$$|q| = \sqrt{\sum_{j=1}^{t} (w_{iq})^2}$$

Where:

|*q*| : distance *query*

Wiq : document *query* weights

With |*q*| is distance *query*, and *Wiq* is first document *query* weights, then the distance *query* (|*q*|) calculated to get distance *query* of document *query* weights (*Wiq*) which is taken by the system. The distance *query* can be calculated by the root equation of the number of squares of the *query*.

$$|d_j| = \sqrt{\sum_{i=1}^{t} (w_{ij})^2}$$

Where:

|*dj*|: distance document

Wij : document weights

3.6 Vector Space Model

Vector Space Model is a representation of a collection of documents as vectors in a vector space. Vector Space Model is a basic technique in information acquisition that can be used to research the relevance of documents to search keywords (query classification documents and grouping documents. Collection of words and documents represented in the form of a matrix of words.

3.6.1 Calculation of Cosine Similarity

With | *dj* | is document distance, and W_{ij} is the *i* document weight, document distance (| *dj* |) is calculated to obtain the document distance from the document weight (W_{ij}) taken by the system. Document spacing can be calculated by the equation of the square root number of documents.

Calculation of measurement similarity query document (inner product), using the equation.

$$sim\,(q.d_j) = \sum_{i=1}^{t} w_{iq}.w_{ij}$$

Where:

sim(*q,dj*): query and document similarity

Wiq: document query weight

Wij: document weight

With *Wij* is *term* weight in the document, *Wiq* is *query* weight, and *Sim* (*q, dj*) is Similarity between *query* and document. Similarity between *query* and document or *inner product/Sim* (*q, dj*) use to get weights based on weights *term* in the document (*Wij*) and *query* weight (*Wiq*) or by adding up the weight *q* multiplied by the weight of the document.

Cosine Similarity measurement (calculating the cosine value of an angle between two *vectors*) using the equation.

$$sim(q.d_j) = cos(\theta) = \frac{q.d_j}{|q|.|d_j|} = \frac{\sum_{i=1}^{t} w_{iq} \cdot w_{ij}}{\sqrt{\sum_{i=1}^{t}(w_{iq})^2} \cdot \sqrt{\sum_{i=1}^{t}(w_{ij})^2}}$$

Where:

sim(*q, dj*): similaritas *query* dan dokumen

q: *query*

dj: document

|*q*|: distance *query*

|*dj*|: document distance

Wiq : document *query* weights

Wij : document weight

Similarity between *query* and document or *Sim* (*q,dj*) directly proportional to the number of weights *query* (*q*) times the document weight (*dj*) and inversely proportional to the number square root *q* (|*q*|) times the square root of the number of documents (|*dj*|). Similarity calculations produce document weights that are close to value 1 or produce document weights that are greater than the values generated from calculations *inner product*.

IV SYSTEM ANALYSIS AND DESIGN

4.1 *Data Flow Diagram*

Figure 4.1 Data Flow Diagram

Explanation for figure 4.1 is where when users log in using their Facebook account into the application, the Facebook user status will be grabbed and made a document which will then be processed and then displayed again to the user after weighting and calculation.

4.2 *Data Flow Diagram level 1*

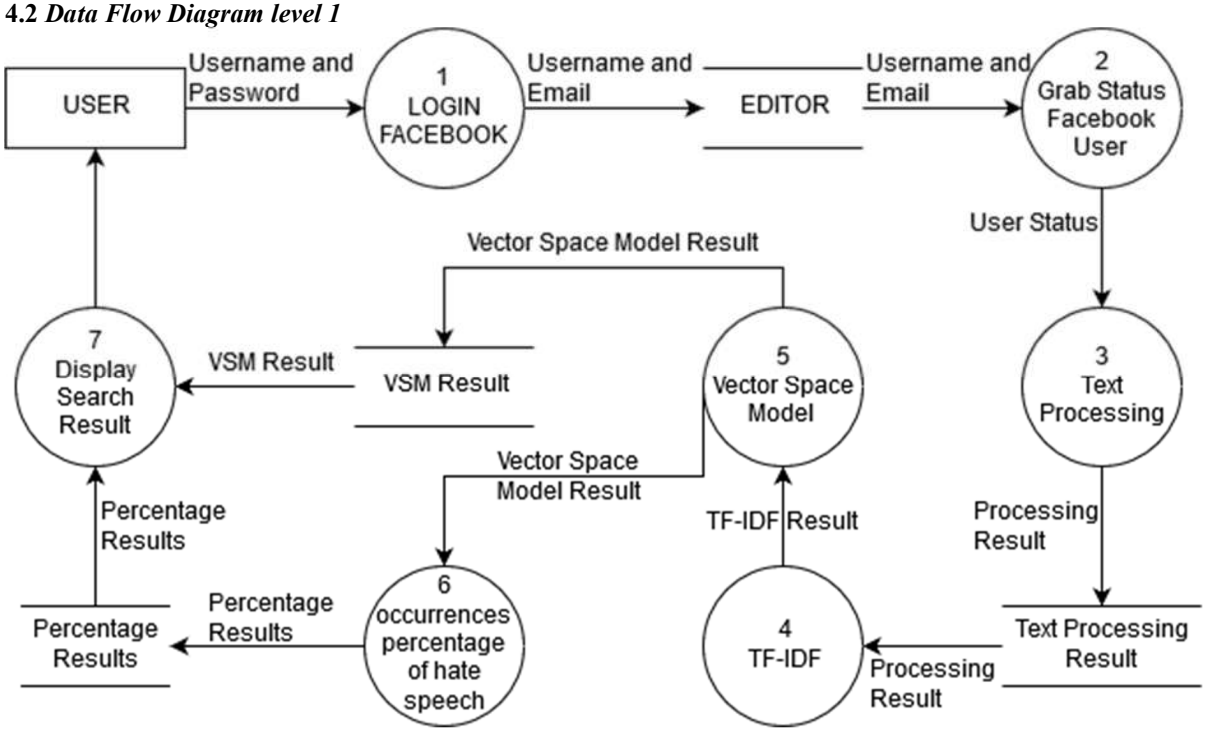

Figure 4.2 Data Flow Diagram level 1

Explanation of figure 4.2 Data Flow Diagram level 1:

1) Users are required to log in using their Facebook account.
2) After logging in, the 30 most recent Facebook user statuses will be retrieved and made into documents (data).
3) The document (data) that has been grabbed then performs text processing.
4) Data that has passed through the next stage of text processing will be assigned a value to each term using TF-IDF which will be used for vector space model calculations.
5) The results of the TF-IDF are then calculated using the vector space model method.
6) The whole document will be calculated the percentage value to find out what percentage of hate words contained in the latest 30 Facebook user status.
7) The results of vector space model calculations and percentages are displayed to the user.

4.3 Entire System Diagram Blocks

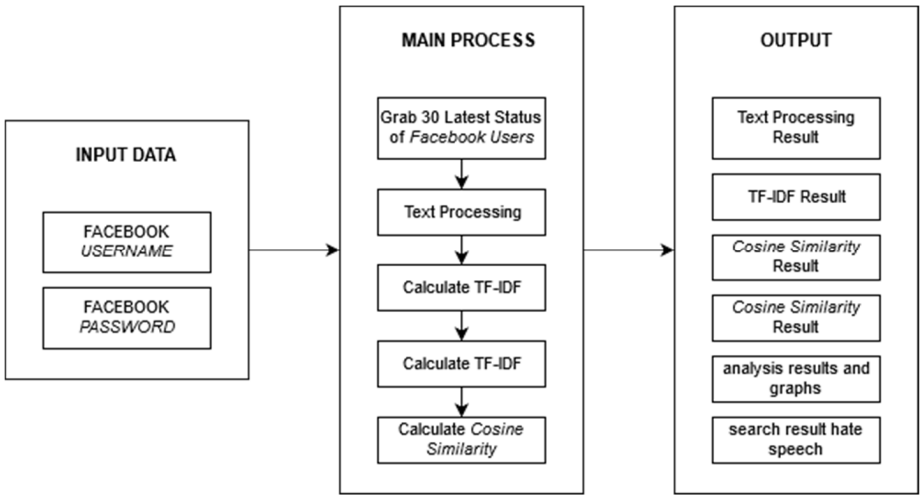

Figure 4.3 Block Diagram of the Whole System

P-ISSN: 2528-0260 E-ISSN: 2579-5392
p.611-624

In the main process of Input Data, the user is asked to login with a personal Facebook account to be able to use the hate speech search application. Then in the second process, 30 of the user's latest status will be automatically grabbed by the application to be used as a document and look for whether the status contains hate speech or not by going through the stages of text processing, calculating TF-IDF and calculating the cosine similarity value. In the third process, which is the Output process, all the results of the text processing and calculation will be displayed together with the results of the search and analysis.

4.3.1 Block Diagram of Determination of Hate Speech

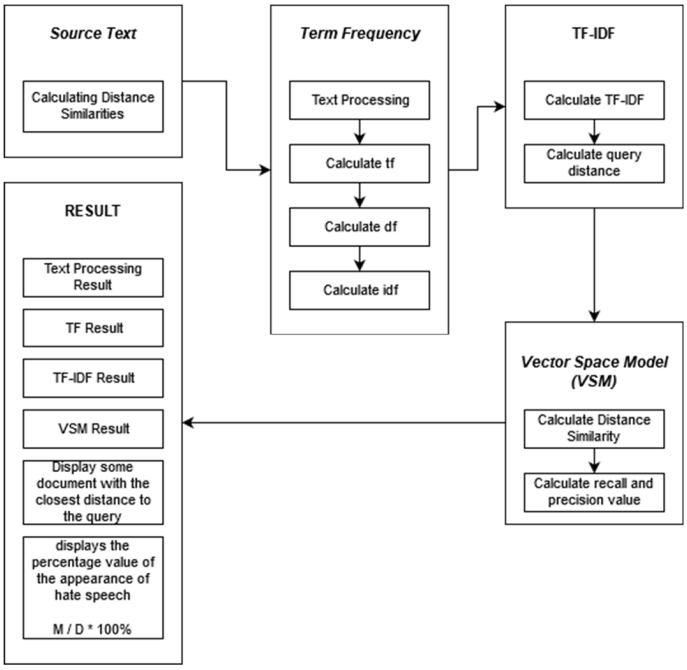

Figure 4.4 Block Diagram of Determination of Hate Speech

To be able to determine the percentage of occurrences of hate speech, the value of documents that have a similarity value and similarity value is obtained from the VSM (Vector Space Model), which is a document that has a cosine similarity value divided by the total number of documents.

4.4 Flowchart

Flowcharts are charts that have flows that can describe the steps for solving a problem. In the process, users log in with their Facebook account, after logging in automatically the program will retrieve some of the latest statuses to be used as documents, then do text processing such as tokenization, case folding, stopword and stemming.

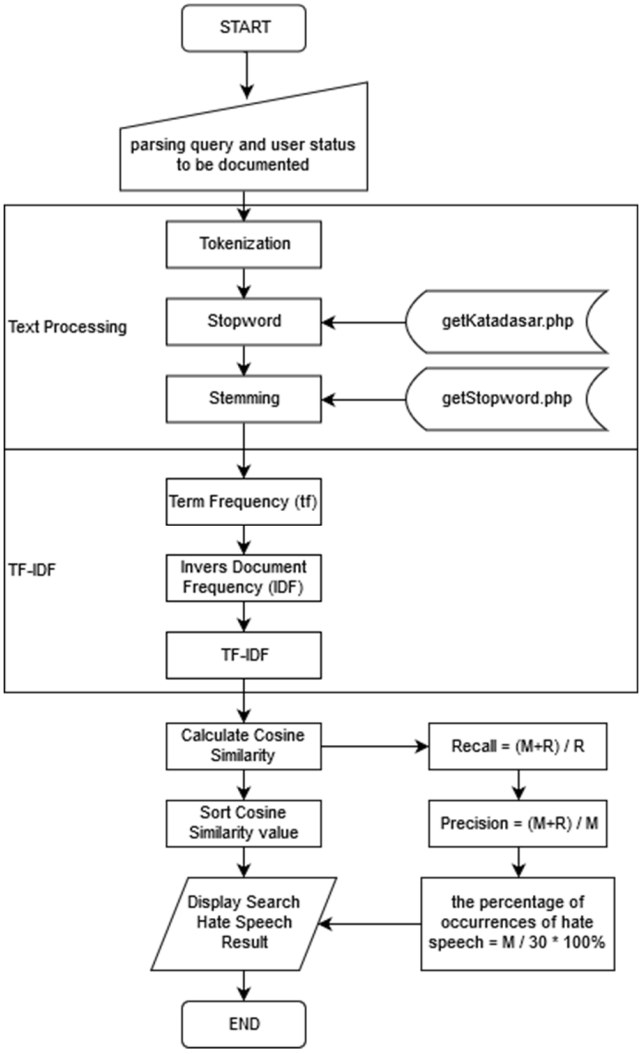

Figure 4.5 Flowchart

V RESULTS AND DISCUSSION

After going through word processing then the weighting of words will be done using the TF-IDF method and calculations using the vector space model method.

Process 1: Case Folding and Tokenization

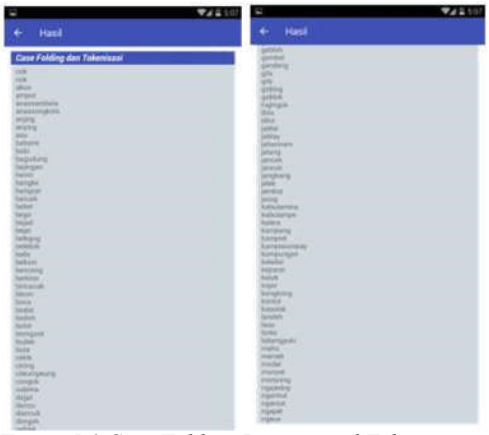

Figure 5.1 Case Folding Process and Tokenization

P-ISSN: 2528-0260 E-ISSN: 2579-5392
p.611-624

In Figure 5.1 is when the query and document have gone through a case folding and tokenization process, that is, the process in which all letters are converted to lowercase and the separation of a sentence into words.

Process 2: Filtering

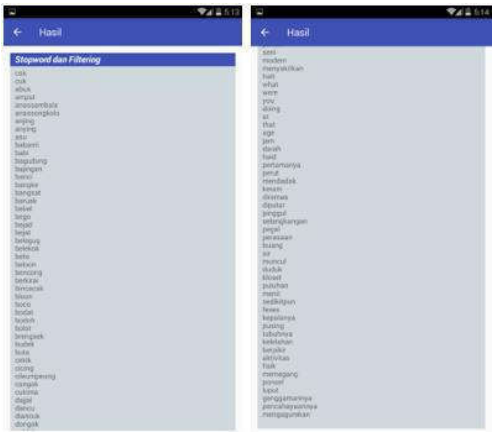

Figure 5.2 Filtering Process

In Figure 5.2, is when the term has been through the filtering process, which is a process to take important words in a sentence and eliminate the less important words that have a high likelihood of contained in each document such as the words "in", "the", " and "," from "and so on.

Process 3: Stemming

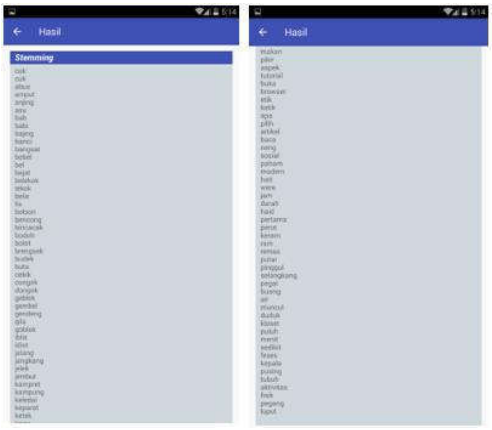

Figure 5.3 The stemming process

In Figure 5.3 is when the term has been through the process of stemming, which is the process of returning a word that contains the prefix and suffix affixes into basic words.

P-ISSN: 2528-0260 E-ISSN: 2579-53920
p.611-624

Process 4: Perform term frequency, df and idf calculations

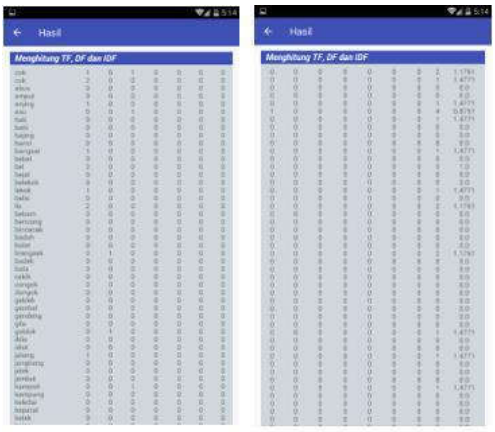

Figure 5.4 The process of calculating the term frequency, df and idf

In Figure 5.4 above is when the term has been through the calculation of the appearance of the term in each document and calculate the weight for the df and idf values.

Process 5: Perform Tf_Wt calculation

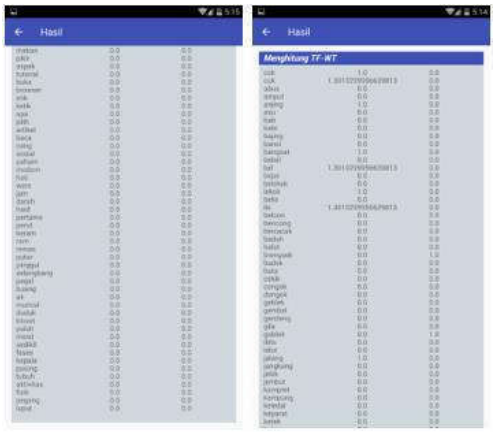

Figure 5.5 The Tf_Wt calculation process

Figure 5.5 is the result of calculating the Tf_Wt frequency log weight.

Process 6: Perform TF-IDF calculation

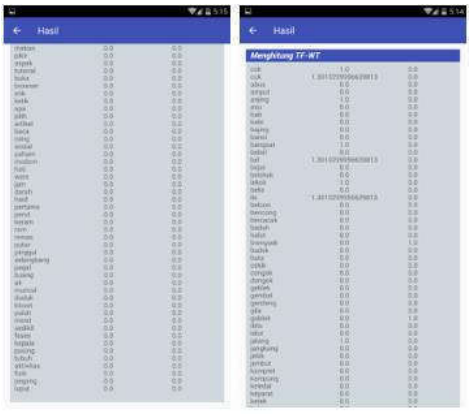

Figure 5.6 TF-IDF calculation process

P-ISSN: 2528-0260 E-ISSN: 2579-5392
p.611-624

Figure 5.6 shows the TF-IDF calculation results. Namely the Tf_Wt calculation result times IDF.

Process 7: Perform Total Wiq * Calculation Wij

*Figure 5.7 The process of calculating Total Wiq * Wij*

Figure 5.7 is the result of the TF-IDF (query) calculation process multiplied by TF-IDF (document).

Process 8: Perform Wij2 Calculation

Figure 5.8 is the result of the calculation process which squares the TF-IDF results.

Figure 5.8 Wij^2 Calculation Process

Process 9: Perform Cosine Similarity Calculations

Figure 5.9 The process of calculating cosine similarity

P-ISSN: 2528-0260 E-ISSN: 2579-53920
p.611-624

Process 10: Display analysis results and search results

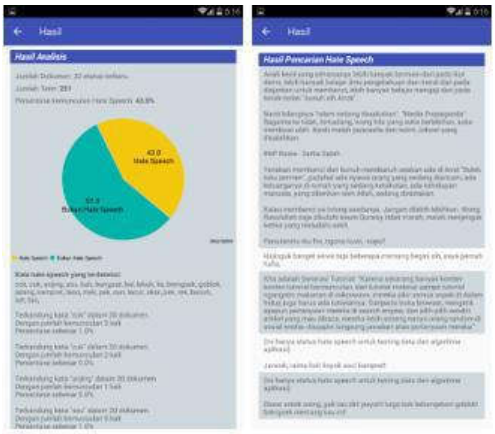

Figure 5.10 Displays the analysis results and search results

In Figure 5.10 is the result of a search analysis of documents and document search results that display hate speech. To be more precise it can be seen in table 5.1.

Table 5.1 Search results that have similar values

Document	Relevant	Result
D1	R	M
D2	R	M
D5	TR	M
D6	TR	M
D9	TR	M
D11	TR	M
D12	TR	M
D13	TR	M
D14	TR	M
D15	TR	M
D16	TR	M
D26	TR	M
D29	TR	M

Based on the search results, documents that have similar values and relevance have been displayed in the table above. Generate a search with a recall value: 2/2 = 1 which means that the search results found relevant documents and the precision value of the search results is 2/13 = 0.154 which means the search results also bring up irrelevant search results. With the percentage value of the appearance of hate speech similarity value of (13/30) * 100% = 43%.

VI SUGGESTIONS

6.1 Conclusion

Based on the test results of the hate speech search application with the status of Facebook users used as an android-based document, it can be concluded several things as follows:

1) This application managed to find abusive, abusive words that are considered as hate speech with a percentage of 43%.
2) Based on the search results using 30 documents. There are 13 documents that have a similarity to the query value. Generate a search that has a recall value: 2/2 = 1 which means that the search results found relevant documents and the precision value of the search results is 2/13 = 0.154 which means the search results also bring up irrelevant search results.

6.2 Suggestions

- Word processing in stemming is expected to be further optimized because in this research it is still far from perfect and cannot process non-standard words.
- The documents used in this study are still static and only a total of 30 recent statuses can be processed.
- Can process data and words in search of hate speech even faster with the same method or not.

REFERENCES

[1] F. Amin, "Sistem Temu Kembali Informasi dengan Metode Vector Space Model," *J. Sist. Inf. Bisnis JSINBIS*, vol. 2, no. 2, 2012.
[2] O. Karmayasa and I. Bagus Mahendra, "Implementasi Vector Space Model Metode Term Frequency Inverse Document Frequency (TF-IDF) pada Sistem Temu Kembali Informasi," *Univ. Udayana, Denpasar*, 2012.
[3] G. Karyono, F. S. Utomo, A. Sistem, and T. Balik, "Temu Balik Informasi Pada Dokumen Teks Berbahasa Indonesia Dengan Metode Vector Space Retrieval Model," *Semin. Nas. Teknol. Inf. dan Terap. 2012*, vol. 2012, no. Semantik, pp. 282–289, 2012.
[4] A. A. Abdillah, I. B. Muktyas, P. Studi, P. Matematika, and S. Surya, "Implementasi Vector Space Model Untuk Pencarian Dokumen," pp. 1–7, 2008.
[5] Tudesman, E. Oktalina, Tinaliah, and Yoannita, "SISTEM DETEKSI PLAGIARISME DOKUMEN BAHASA INDONESIA MENGGUNAKAN METODE VECTOR SPACE MODEL," *STMIK Glob. Inform. MDP, Palembang*, pp. 1–10, 2011.
[6] K. D. Putung, A. S. M. Lumenta, and A. Jacobus, "PENERAPAN SISTEM TEMU KEMBALI INFORMASI PADA KUMPULAN DOKUMEN SKRIPSI," *Tek. Inform. Univ. Sam Ratulagi, Manad.*, 2016.
[7] KBBI, "Daftar Kata Kasar dalam Bahasa Indonesia," *kbbi.kata.web.id*. .
[8] P. E. Mas'udia, M. D. Atmadja, and L. D. Mustafa, "Information Retrieval Tugas Akhir Dan Perhitungan Kemiripan Dokumen Mengacu Pada Abstrak Menggunakan Vector Space Model," *Simetris J. Tek. Mesin, Elektro dan Ilmu Komput.*, vol. 8, no. 1, pp. 355–362, 2017.

P-ISSN: 2528-0260 E-ISSN: 2579-53920
p.611-624