ISSN: 2477-5126
e-ISSN: 2548-9356

Implementasi Vector Space Model Dengan Pembobotan Berbasis Kelas Pada Mesin Pencari Dokumen Skripsi

Ikwan Rizki Priandono[1], Maftahatul Hakimah[2], Nanang Fakhrur Rozi[3]
[1,2,3]Jurusan Teknik Informatika, Fakultas Teknik Elektro dan Teknologi Informasi, Institut Teknologi Adhi Tama Surabaya
[1,2,3]Jln. Arif Rahman Hakim No. 100, Kota Surabaya, 60117, Indonesia
email: [1]naxiaentertainment@gmail.com, [2]hakimah.mafta@itats.ac.id, [3]nanang@itats.ac.id

Abstrak – Skripsi merupakan salah satu persyaratan untuk kelulusan seorang mahasiswa. Untuk menentukan tema skripsi, mahasiswa dapat mencari referensi dari sumber eksternal dari *website* seperti *Research Gate*, *Springer*, IEEE dan *Science Direct*. Sedangkan salah satu sumber referensi internal yaitu website perpustakaan ITATS yang menyimpan dokumen skripsi yang sudah diselesaikan oleh Mahasiswa ITATS. Di Jurusan Teknik Informatika ITATS terdapat 3 bidang minat yang dapat dijadikan kelas pada dokumen skripsi yaitu Kecerdasan Buatan, Rekayasa Perangkat Lunak, dan Jaringan Komputer. Dengan adanya 3 bidang minat maka pembobotan kata yang diusulkan adalah TF.IDF.ICF dimana ICF melakukan pembobotan kata yang memperhatikan kelas (bidang minat) pada dokumen. Dengan pembobotan TF.IDF.ICF relevansi dari hasil pencarian lebih baik daripada menggunakan TF.IDF dengan nilai *mean average precision* masing-masing 72,39% dan 71,12%.

Kata kunci*: dokumen skripsi, mesin pencari, pembobotan term berbasis kelas, sumber referensi*

Abstract - Thesis is one of the requirements for graduation of a student. To determine the thesis theme, students can look for references from external sources from websites such as Research Gate, Springer, IEEE and Science Direct. Meanwhile, one of the internal reference sources is the ITATS library website which stores thesis documents that have been completed by ITATS students. In the ITATS Informatics Engineering Department there are 3 areas of interest that can be used as classes in the thesis document, namely Artificial Intelligence, Software Engineering, and Computer Networks. With the existence of 3 areas of interest, the proposed word weighting is TF.IDF.ICF where ICF carries out term weighting which paying attention to class (areas of interest) in the document. By weighting TF.IDF.ICF, the relevance of the search results is better than using TF.IDF with the mean average precision values 72.39% and 71.12%

Keyword*: class-based term weighting, search engine, thesis document, reference source*

I. PENDAHULUAN

Skripsi merupakan salah satu syarat untuk kelulusan seorang mahasiswa. Untuk menyelesaikan skripsi dibutuhkan tema untuk menentukan judul skripsi yang dapat ditemukan melalui sumber referensi internal maupun eksternal. Sumber referensi eksternal dapat berasal dari website seperti *Research Gate*, *Springer*, *IEEE* dan *Science Direct*. Sedangkan untuk sumber internal, ITATS memiliki sebuah website perpustakaan yang menyimpan dokumen skripsi yang sudah diselesaikan oleh Mahasiswa ITATS.

Pada website perpustakaan ITATS (ITATS Library) pencarian dokumen skripsi masih menggunakan pencarian berbasis *SQL query*. Pencarian dokumen skripsi dengan *SQL query* mengharuskan *user* untuk menulis *query* atau potongan kata kunci dengan tepat. Biasanya saat melakukan pencarian, hasil pencarian yang dimunculkan oleh *search engine* hanya berupa dokumen dimana dokumen tersebut memiliki kata atau potongan kata yang sama dengan *query*. Jika kata kunci yang dimasukkan kurang tepat maka hasil yang didapatkan oleh *user* menjadi kurang relevan atau bahkan tidak dapat menemukan dokumen skripsi.

Oleh karena itu dibutuhkan sebuah *search engine* yang dapat membantu mahasiswa mencari dokumen skripsi untuk dapat dijadikan tema skripsi. Sehingga, usulan skripsi ini menggunakan *search engine* dengan metode *Vector Space Model* (VSM). Vector Space Model adalah suatu model aljabar untuk mewakili dokumen teks sebagai suatu vektor pengenal, contohnya indeks kata. VSM biasanya digunakan dalam penyaringan informasi, temu balik informasi, pengindeksan, dan perankingan relevansi[1]. Dengan menerapkan metode VSM diharapkan dapat meningkatkan kinerja *search engine* dokumen skripsi.

Search Engine atau mesin pencari adalah sebuah alat yang dapat mempermudah dan mempercepat pencarian. Mesin pencarian pertama kali dibuat oleh McBrian dengan nama World Wide Web Worm (WWWW) pada tahun 1993[2].

Di Jurusan Teknik Informatika ITATS terdapat 3 bidang minat yang dapat dijadikan kelas pada dokumen skripsi yaitu Kecerdasan Buatan, Rekayasa Perangkat Lunak, dan Jaringan Komputer. Pembobotan kata yang sering digunakan oleh peneliti adalah dengan pembobotan kata TF.IDF namun pembobotan ini tidak memperhatikan kelas pada dokumen. Sehingga diusulkan pembobotan TF.IDF.ICF dimana ICF melakukan pembobotan kata yang memperhatikan kelas pada dokumen.

II. PENELITIAN YANG TERKAIT

Penelitian terdahulu terkait metode pembobotan TF.IDF.ICF antara lain dilakukan oleh Rosid, tentang klasifikasi keluhan mahasiswa pada aplikasi e-complaint di Universitas Muhammadiyah Sidoarjo dengan metode *centroid based classifier* dengan menggunakan pembobotan kata TF.IDF.ICF untuk. Penelitian ini menggunakan metode dan Pembobotan TF.IDF.ICF dengan hasil yang diketahui dapat meningkatkan performa dengan rata rata akurasi 79.55%.[3]

Penelitian terdahulu terkait pencarian pada koleksi dokumen skripsi antara lain dilakukan oleh Yahya, tentang implementasi *web service* pada sistem pengindeksan dan pencarian dokumen skripsi pada kampus ITATS dengan metode *latent semantic indexing*. Dapat diketahui pada

*) **penulis korespondensi**: Ikwan Rizki Priandono
Email: naxiaentertainment@gmail.com

penelitian ini menghasilkan *recall* 37%, *precision* 97%, dan *accuracy* 64%[4].

Penelitian terkait judul dan abstrak skripsi dilakukan oleh Mas'udia, tentang pencarian dokumen yang mirip dengan kata kunci berdasarkan judul dan abstrak dokumen skripsi menggunakan metode *vector space model* dimana dengan pengujian menggunakan kata kunci "android" didapatkan hasil empat dokumen yang ditampilkan dengan urutan sesuai kemiripan yaitu docId 3 dengan tingkat kemiripan 0,.9512, docId 4 dengan tingkat kemiripan 0.5020, docId 2 dengan tingkat kemiripan 0.2671, docId 8 dengan tingkat kemiripan 0.1522[5].

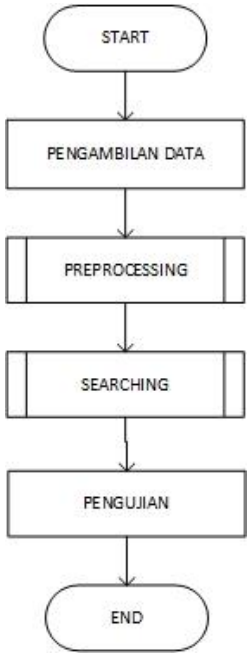
Penelitian terkait pencarian dokumen skripsi dengan metode VSM antara lain penelitian yang dilakukan oleh Kusuma, tentang perancangan *search engine* dengan pencarian full-text pada dokumen skripsi untuk perpustakaan Fakultas Teknik UHAMKA dengan metode VSM. Pembobotan yang digunakan pada penelitian ini adalah TF.IDF. Penelitian ini diketahui mampu menampilkan dokumen dengan relevan dari 10 query dengan nilai relevansi yang cukup baik, dengan nilai rata-rata *precision* 0.63, *recall* 0.92 dan *f-measure* 0.71[6].

Penelitian terkait *term weighting* dengan antara lain dilakukan oleh Rozi, tentang tujuan pembuatan pembobotan kata baru untuk mengoptimasi klasifikasi kategori dari jenis halaman web yang disebut pembobotan berbasis pada indeks jenis (TF-IGF). Pembobotan TF-IGF diketahui memiliki performa lebih baik dari TF-IDF dengan nilai accuracy, precision, recall dan f-measure tanpa menyebutkan kata kunci spesifik adalah 78%, 80,2%, 78% dan 77,4% dan jika menyebutkan kaca kunci spesifik adalah 78,9%, 78,7%, 78,9% dan 78,1%[7].

III. METODE PENELITIAN

A. *Alur Proses Sistem Search Engine*

Alur proses pencarian dokumen skripsi dilakukan dengan menerapkan *flowchart* berikut :



Gambar 1. *Flowchart* sistem *search engine* dokumen skripsi

Berdasarkan *flowchart* (Gambar 1.) dapat dijelaskan alur proses sistem pencarian dokumen skripsi dilakukan dari tahap pengambilan data dari *database*, kemudian data dokumen skripsi dilakukan proses *preprocessing* (*tokenize*, *filtering* dengan *stopword removal*, *stemming* dan *term weighting*). Setelah didapatkan preprocessing perhitungan vektor dari dokumen yang kemudian dilakukan pencarian dokumen skripsi menggunakan kata kunci yang dicari. Pencarian dilakukan dengan menghitung *cosine similarity* antara dokumen dengan kata kunci. Hasil *output* pencarian ditampilkan dari similaritas tertinggi. Pengujian relevansi dokumen skripsi dengan kata kunci dilakukan dengan menggunakan kuesioner dan kemudian dihitung *recall*, *precision*, *average precision* dan *mean average precision*.

B. *Pengambilan Data*

Pengambilan data abstrak dokumen skripsi dilakukan denggan mengambil data dari *database web* Perpustakaan ITATS yang kemdian setiap dokumen skripsi dilakukan klasifikasi dokumen skripsi berdasarkan bidang minatnya.

Dari proses pengambilan data didapatkan abstrak dokumen skripsi sebanyak 378 dokumen dimana dokumen dengan bidang minat kecerdasan buatan sebanyak 96 dokumen, rekayasa perangkat lunak 151 dokumen dan jaringan komputer sebanyak 131 dokumen.

C. *Preprocessing*

Proses Tokenizing, adalah proses memecah dokumen menjadi kumpulan kata[8]. Pada proses ini dilakukan pemecahan judul dan abstrak dokumen menjadi kata.

Proses *Filtering*, *filtering/stopwords removal* merupakan proses penghilangan kata tidak penting pada deskripsi melalui pengecekan kata-kata hasil *tokenizing* apakah termasuk di dalam daftar kata tidak penting (*stoplist*) atau tidak[9]. Proses filtering menggunakan library php Sastrawi karena library ini cukup populer, mudah diitegrasikan dengan *framework* / *package* lain, mempunyai API yang sederhana dan mudah digunakan.

Stemming merupakan proses untuk mendapatkan *root/stem* atau kata dasar dari suatu kata dalam kalimat dengan cara memisahkan masing-masing kata dari kata dasar dan imbuhannya baik prefiks maupun sufiks[10]. Proses *Stemming* yang digunakan yaitu algoritma *stemming* Nazief dan Adriani.

Proses *term weighting* menggunakan dua macam pembobotan yaitu TF.IDF dan TF.IDF.ICF. Rumus TF.IDF dapat dilihat pada Persamaan 1 dan rumus TF.IDF.ICF dapat dilihat pada Persamaan 2.

$$TF.IDF\ (d,t) = TF(d,t)\,x\ IDF(t) \quad (1)$$

dimana : TF(d,f) = nilai TF term t pada dokumen d
IDF(t) = nilai IDF term t

$$TF.IDF.ICF\ (d,t) = TF(d,t)\,x\ IDF(t)\,x\ ICF(t) \quad (2)$$

dimana : TF(d,f) = nilai TF term t pada dokumen d
IDF(t) = nilai IDF term t
ICF(t) = nilai ICF term t

D. Searching

Pada tahap ini dilakukan proses dari hitung panjang vektor dokumen, *preprocessing* kata kunci dan hitung *cosine similarity* antara dokumen skripsi dengan kata kunci.

Proses hitung panjang vektor dokumen skripsi, rumus hitung panjang vektor dokumen dapat dilihat pada Persamaan 3.

$$|d_i| = \sqrt{\sum_{i=0}^{n} t_i} \quad (3)$$

dimana : d_i = dokumen ke –i
t_i = term ke-i dari dokumen

Preprocessing pada tahap ini dilakukan pada *query* untuk dapat didapatkan nilai bobot kata dan panjang vektornya. Rumus untuk menghitung panjang vektor *query* dapat dilihat pada Persamaan 4.

$$|Q_i| = \sqrt{\sum_{i=1}^{n} t_i} \quad (4)$$

Dimana : Q_i = Query ke –i
t_i = term ke-i dari Query

Setelah didapatkan nilai panjang vektor dokumen dan *query*, dilakukan perhitungan *cosine similarity* antara *query* dengan dokumen. *Cosine similarity* digunakan untuk menentukan similarity antara dua vektor [11].

Perhitungan kesamaan antara *query* dengan dokumen dilakukan dengan menggunakan metode *Vector Space Model*. Metode *Vector Space Model* menghitung nilai sudut cosinus dari dua vektor yaitu vektor dari *term* pada *query* dan vektor dari *term* pada dokumen. Rumus perhitungan similaritas antara *query* dengan dokumen dapat dilihat pada Persamaan 5.

$$sim(dj, q) = \frac{dj.q}{\|dj\| \|q\|} = \frac{(\sum_{i=1}^{N} w_{i,j} w_{i,q})}{\sqrt{\sum_{i=1}^{N} w_{i,j}^2} \sqrt{\sum_{i=1}^{N} w_{i,q}^2}} \quad (5)$$

Dimana :
d*j* = Dokumen ke-j
q = query
w*i*,*j* = Bobot dari term-i dokumen ke-j
W*i*,q = Bobot dari term-i dokumen query

E. Pengujian

Dalam pengujian ini relevansi dokumen ditentukan dari kuesioner dengan 9 macam kata kunci dari 3 bidang minat (Tabel 1) yang diisi oleh 30 koresponden dengan berbagai kalangan profesi.

Tabel 1. Kata Kunci Yang Digunakan Untuk Menguji Relevansi Hasil Pencarian Dokumen Skripsi

NO	KATA KUNCI	BIDANG MINAT
1	Klasifikasi Kematangan	Kecerdasan Buatan
2	Optimasi Citra	Kecerdasan Buatan
3	Deteksi Tepi	Kecerdasan Buatan
4	Game Edukasi	Rekayasa Perangkat Lunak
5	Aplikasi Berbasis Android	Rekayasa Perangkat Lunak
6	Sistem Informasi	Rekayasa Perangkat Lunak
7	Keamanan Enkripsi	Jaringan Komputer
8	Transaksi Kriptografi	Jaringan Komputer
9	Wireless Microcontroller	Jaringan Komputer

Pengujian dilakukan dengan mencari *recall*, *precision*, *average precision* dan *mean average precision*. *Recall* adalah proporsi jumlah dokumen yang dapat ditemukan-kembali oleh sebuah proses pencarian di sistem IR, sedangkan *Precision* dapat diartikan sebagai kepersisan atau kecocokan (antara permintaan informasi dengan jawaban terhadap permintaan itu)[12].

IV. HASIL DAN PEMBAHASAN

Masing-masing kata kunci diuji dengan menggunakan pembobotan TF.IDF.ICF dan TF.IDF. Setelah didapatkan dokumen mana saja yang relevan menurut koresponden, selanjutnya diuji dengan menghitung *recall*@k, *precision*@k, *average precision* dan *mean average precision*.

Pengertian dari *average precision* itu sendiri adalah jumlah nilai *precision* berdasarkan obyek terpilih yang bernilai true/relevant dibagi dengan jumlah semua item terpilih yang bernilai true/relevant[13].Contoh perhitungan *average precision* dari kata kunci “Game Edukasi” dengan TF.IDF.ICF dapat dilihat pada Tabel 2 dan dengan TF.IDF dapat dilihat pada Tabel 3.

Tabel 2. Perhitungan *Average Precision* Menggunakan TF.IDF.ICF

TF.IDF.ICF					
rank	Jumlah Terpilih	Presentase	Relevansi	r@k	p@k
1	28	93	Relevan	0,1000	1,0000
2	23	77	Relevan	0,2000	1,0000
3	23	77	Relevan	0,3000	1,0000
4	2	7	Tidak Relevan	0,3000	0,7500
5	6	20	Tidak Relevan	0,3000	0,6000
6	7	23	Tidak Relevan	0,3000	0,5000
7	23	77	Relevan	0,4000	0,5714
8	1	3	Tidak Relevan	0,4000	0,5000
9	21	70	Relevan	0,5000	0,5556
10	20	67	Relevan	0,6000	0,6000
11	21	70	Relevan	0,7000	0,6364
12	2	7	Tidak Relevan	0,7000	0,5833
13	21	70	Relevan	0,8000	0,6154
14	15	50	Tidak Relevan	0,8000	0,5714
15	1	3	Tidak Relevan	0,8000	0,5333
16	4	13	Tidak Relevan	0,8000	0,5000
17	6	20	Tidak Relevan	0,8000	0,4706
18	19	63	Relevan	0,9000	0,5000
19	0	0	Tidak Relevan	0,9000	0,4737
20	19	63	Relevan	1,0000	0,5000
	average precision (%)				69,79

Tabel 3. Perhitungan *Average Precision* Menggunakan TF.IDF

TF.IDF					
rank	Jumlah Terpilih	Presentase	Relevansi	r@k	p@k
1	27	90	Relevan	0,1000	1,0000
2	26	87	Relevan	0,2000	1,0000
3	24	80	Relevan	0,3000	1,0000
4	4	13	Tidak Relevan	0,3000	0,7500
5	5	17	Tidak Relevan	0,3000	0,6000
6	5	17	Tidak Relevan	0,3000	0,5000
7	24	80	Relevan	0,4000	0,5714
8	6	20	Tidak Relevan	0,4000	0,5000
9	22	73	Relevan	0,5000	0,5556
10	21	70	Relevan	0,6000	0,6000
11	21	70	Relevan	0,7000	0,6364
12	2	7	Tidak Relevan	0,7000	0,5833
13	23	77	Relevan	0,8000	0,6154
14	3	10	Tidak Relevan	0,8000	0,5714
15	5	17	Tidak Relevan	0,8000	0,5333
16	4	13	Tidak Relevan	0,8000	0,5000
17	4	13	Tidak Relevan	0,8000	0,4706
18	21	70	Relevan	0,9000	0,5000
19	2	7	Tidak Relevan	0,9000	0,4737
20	20	67	Relevan	1,0000	0,5000
	average precision (%)				69,79

Dari semua percobaan didapatkan hasil *average precision* menggunakan TF.IDF.ICF yang dapat dilihat pada Tabel 4 dan menggunakan TF.IDF yang dapat dilihat pada Tabel 5.

Tabel 4. Hasil Perhitungan *Average Precision* Setiap Kata Kunci Dengan TF.IDF.ICF

Percobaan	Average Precision
Klasifikasi Kematangan	49,93
Optimasi Citra	59,09
Deteksi Tepi	70,93
Game Edukasi	69,79
Aplikasi Berbasis Android	99,40
Sistem Informasi	51,68
Keamanan Enkripsi	94,70
Transaksi Kriptografi	58,60
Wireless Microcontroller	97,38

Tabel 5. Hasil Perhitungan *Average Precision* Setiap Kata Kunci Dengan TF.IDF

Percobaan	Average Precision
Klasifikasi Kematangan	52,22
Optimasi Citra	59,09
Deteksi Tepi	68,54
Game Edukasi	69,79
Aplikasi Berbasis Android	99,40
Sistem Informasi	43,31
Keamanan Enkripsi	92,22
Transaksi Kriptografi	58,64
Wireless Microcontroller	96,85

Dari beberapa pengujian, hasil *average precision* tiap kata kunci dihitung *mean average precision* pada semua kata kunci dan masing-masing bidang minat. Hasil perhitungan *mean average precision* dengan TF.IDF.ICF dan TF.IDF dapat dilihat pada Tabel 6.

Tabel 6. Perhitungan Pengujian Dengan *Mean Average Precision*

Perhitungan Berdasarkan	Mean Average Precision	
	TF.IDF	TF.IDF.ICF
Semua Kata Kunci	71,12	72,39
Kecerdasan Buatan	59,95	59,98
Rekayasa Perangkat Lunak	70,83	73,62
Jaringan Komputer	82,57	83,56

Sehingga dapat diketahui hasil pengujian relevansi menggunakan *Mean Average Precision* dengan pembobotan TF.IDF.ICF lebih relevan dibandingkan dengan pembobotan tanpa ICF.

V. KESIMPULAN

Berdasarkan percobaan dan pengujian yang sudah penulis lakukan, dapat disimpulkan pencarian dokumen skripsi dengan menggunakan metode TF.IDF.ICF dapat meningkatkan relevansi dibandingkan hanya menggunakan TF.IDF dengan hasil *Mean Average Precision* TF.IDF sebanyak 71,12% dan *Mean Average Precision* TF.IDF.ICF 72,39%. Hasil Pengujian *Mean Average Precision* dari 3 Bidang Minat dengan TF.IDF lebih kecil relevansinya dibandingkan dengan menggunakan TF.IDF.ICF.

DAFTAR PUSTAKA

[1] Hongdan, et al, "A Document-Based Information Retrieval Model Vector Space," IEEE, pp 65-68, 2011.

[2] O. A. McBryan, "GENVL and WWWW: Tools for Taming the Web," First International Conference on the World Wide Web, Geneva, 1994.

[3] M. A. Rosid, Gunawan and E. Pramana, "Centroid Based Classifier dengan Fitur TF-IDF-ICF untuk Klasifikasi Keluhan Mahasiswa pada Aplikasi e-complaint di Universitas Muhammadiyah Sidoarjo," jTE-U, vol. 1, no 1, 2015

[4] Achmad Adin Yahya, "Implementasi Web Service pada Sistem Pengindeksan dan Pencarian Dokumen Skripsi pada Kampus ITATS dengan Metode Latent Semantic Indexing," Skripsi, FTETIF, Teknik Informatika, Insitut Teknologi Adhi Tama Surabaya, 2019

[5] M. A. Kusuma, M. Kamayani, and A. Avorizano, "Pencarian Full Text Pada Koleksi Skripsi Fakultas Teknik UHAMKA Menggunakan Metode Vector Space Model." FT-UHAMKA, vol. 2, ISSN No. 2502-8782, 2017

[6] 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," Jurnal SIMETRIS, vol. 8, no. 1, 2017

[7] Sugiyanto, Nanang F. Rozi, T. E. Putri, and A. Z. Arifin, "Term Weighting Based on Index of Genre for Web Page Genre Classification," JUTI, vol. 12, no. 1, pp. 27-34, 2014

[8] R. Melita, V. Amrizal, H. B. Suseno, and T. Dirjam, "Penerapan Metode Term Frequency Inverse Document Frequency (TF-IDF) dan Cosine Similarity pada Sistem Temu Kembali Informasi untuk Mengetahui Syarah Hadits Berbasis

Web (Studi Kasus: Syarah Umdatil Ahkam)," Jurnal Teknik Informatika, vol. 11, no. 2, 2018

[9] O. Nurdiana, Jumadi, and D. Nursantika, "Perbandingan Metode Cosine Similarity dengan Metode Jaccard Similarity pada Aplikasi Pencarian Terjemah Al-Qur'an dalam Bahasa Indonesia," JOIN, vol. 1, no. 1, 2016

[10] D. Wahyudi, T. Susyanto, and D. Nugroho, "Implementasi dan Analisis Algoritma Stemming Nazief & Adriani dan Porter pada Dokumen Berbahasa Indonesia," Jurnal Ilmiah SINUS

[11] http://www10.org/cdrom/papers/519/node12.html dilihat pada 16-08-2020.

[12] N. P. Lestari, "Uji Recall and Precision Sistem Temu Kembali Informasi OPAC Perpustakaan ITS Surabaya," OPAC

[13] A. R. F. Rahman, M. A. Bijaksana, and A. Romadhony, "Perankingan Jawaban yang Terklasifikasi pada Komunitas Tanya-Jawab dengan Term Frequency dan Similarity Measure Features," e-Proceeding of Engineering, vol.4, no.1, pp 1093, 2017