



Sistem Temu Kembali Informasi Pada Gejala Autisme Dengan Metode *Vector Space Model*

Bayu Sugara¹, Dody², Donny³¹Program Studi Magister Ilmu Komputer, STMIK Nusa Mandiri^{2,3}Program Studi Magister Ilmu Komputer, Fakultas Teknologi Sistem Informasi, Universitas Budi Luhur¹bayusugaraa@gmail.com, ²1711600559@student.budiluhur.ac.id, ³1811600459@student.budiluhur.ac.id

Abstract

Information is now very easy to get anywhere. Information technology, especially the internet, strongly supports the exchange of information very quickly. The internet has become an information and communication media that has been used by many people with many interests, especially in taking large-scale information data. Unfortunately the information presented is sometimes less relevant. Quality information is influenced by relevance, accuracy and on time. However, there are not many effective search systems available. This study discusses the implementation of an information retrieval system to find and find symptoms of autism disorders using the Vector Space Model (VSM) method. Vector Space Model (VSM) is a model used to measure the similarity between a document and a query. In this model, queries and documents are considered vectors in n dimensional space. Where n is the number of all terms listed. The purpose of this study was to design an information retrieval software to find and match the symptoms of autism disorders. By using Vector Space Model, it is hoped that it can provide a solution to the search engine to provide text matching information in the database using certain keywords, the results of the matching are presented in the form of ranks.

Keywords: Symptoms of Autism Disorders, Vector Space Models, Search Engines, Queries, Information Retrieval Systems

Abstrak

Informasi saat ini sangat mudah didapatkan oleh dimanapun berada. Teknologi informasi khususnya internet sangat mendukung terjadinya pertukaran informasi dengan sangat cepat. Internet menjadi media informasi dan komunikasi yang telah dimanfaatkan banyak orang dengan banyak kepentingan terutama dalam mengambil data informasi dari skala besar, Sayangnya informasi yang disajikan terkadang kurang relevan. Informasi yang berkualitas dipengaruhi oleh relevansi, keakuratan dan tepat waktu. Meskipun demikian, belum banyak tersedia sistem pencarian yang efektif. Penelitian ini membahas tentang implementasi sistem temu balik informasi untuk mencari dan menemukan gejala gangguan autisme menggunakan metode *Vector Space Model* (VSM). *Vector Space Model* (VSM) adalah suatu model yang digunakan untuk mengukur kemiripan antara suatu dokumen dengan suatu *query*. Pada model ini, *query* dan dokumen dianggap sebagai vektor-vektor pada ruang n dimensi. Dimana n adalah jumlah dari seluruh term yang ada didaftar. Tujuan dari penelitian ini adalah merancang sebuah perangkat lunak temu balik informasi untuk mencari dan mencocokkan gejala gangguan autisme. Dengan menggunakan *Vector Space Model* diharapkan dapat memberikan sebuah solusi pada mesin pencarian untuk memberikan informasi kecocokan teks dalam *database* dengan menggunakan kata kunci tertentu, hasil dari pencocokan tersebut disajikan dalam bentuk peringkat.

Kata Kunci: Gejala Gangguan Autisme, *Vector Space Model*, Mesin Pencarian, *Query*, Sistem Temu Balik Informasi

© 2019 Jurnal RESTI

1. Pendahuluan

Informasi saat ini sangat mudah didapatkan oleh setiap orang dimana pun berada. Teknologi informasi khususnya internet sangat mendukung terjadinya pertukaran informasi dengan sangat cepat. Internet menjadi media informasi dan komunikasi yang telah

dimanfaatkan banyak orang dengan banyak kepentingan terutama dalam mengambil data informasi dari skala besar, Sayangnya informasi yang disajikan terkadang kurang relevan [1]. Informasi yang berkualitas dipengaruhi oleh relevansi, keakuratan dan tepat waktu. Meskipun demikian, belum banyak

Diterima Redaksi : 19-07-2019 | Selesai Revisi : 06-08-2019 | Diterbitkan Online : 08-08-2019

tersedia sistem pencarian yang efektif. Sistem pencarian merupakan sistem yang dapat digunakan untuk menemukan informasi yang relevan dengan kebutuhan dari penggunaannya secara otomatis dari suatu koleksi informasi. Sistem ini akan menerima masukan berupa kata kunci dari informasi yang akan dicari. Dengan waktu yang relatif lebih singkat akan memberikan hasil beberapa dokumen atau informasi yang relevan terhadap kata kunci yang dimasukkan pengguna.

Dalam sistem temu kembali diperlukan metode yang cocok diterapkan dalam sistem, salah satunya adalah Metode *Vector Space Model* (VSM). Metode *Vector Space Model* adalah suatu metode untuk merepresentasikan sistem temu kembali ke dalam vektor dan memperhitungkan fungsi *similarity* dalam proses pencocokan beberapa vektor. Suatu sistem temu kembali terdiri atas dua bagian, yaitu penyimpanan dokumen dan pemrosesan *query*, baik *query* maupun dokumen-dokumen yang disimpan dinyatakan dalam bentuk vektor. Metode VSM dipilih karena cara kerja model ini efisien, mudah dalam representasi dan dapat diimplementasikan pada *document-matching*. Metode VSM di juga digunakan untuk menghitung kesamaan antara teks yang masuk (tes baru kasus) dan teks yang dikategorikan sebelumnya dalam set data pelatihan [2].

Dalam *vector space model*, pesan dan *query* diwakili sebagai vektor. Kesamaan antara pesan dan antara pesan dan *query* didefinisikan dalam jarak antara dua vektor. Sebagai salah satu ukuran kesamaan umum adalah kesamaan kosinus, di mana perbedaan antara dua dokumen atau dokumen dan kueri sebagai kosinus sudut antara dua vektor ini [10].

Menurut (Priambodo, 2017), Hasil eksperimen ditunjukkan dengan pembobotan menggunakan pendekatan *vector space model*, sistem dapat memberikan jawaban ke hadits yang paling erat terkait dengan pertanyaan pengguna. Hadits yang paling terkait akan ditampilkan di bagian atas urutan. Namun, sistem tidak dapat memberikan jawaban dalam bentuk kalimat. Tetapi hanya memberikan jawaban daftar hadits yang paling relevan dengan pertanyaan itu [5].

Information Retrieval adalah salah satu topik penambangan teks. Banyak penelitian telah dilakukan untuk mencari jawaban untuk dokumen teks. Pencarian jawaban dengan *vector space model* sangat bagus dan efektif pada berbagai dokumen [5]. Tujuan dari penelitian ini adalah merancang sebuah perangkat lunak temu balik informasi untuk mencari dan mencocokkan gejala gangguan autisme menggunakan *vector space model* dengan bertujuan memberikan sebuah solusi pada mesin pencarian untuk memberikan informasi kecocokan teks dalam database dengan menggunakan kata kunci tertentu, hasil dari pencocokan tersebut disajikan dalam bentuk peringkat.

2. Metode Penelitian

2.1. Autisme

Autisme merupakan gangguan perkembangan mental pada anak yang menyebabkan seorang anak sulit untuk berinteraksi sosial. Diagnosa autisme biasanya dilakukan oleh seorang pakar/ahli dibidang tumbuh kembang anak [23]. Namun sebenarnya, orang tua juga dapat melakukan pengamatan perilaku anak dalam kesehariannya terutama dari cara berkomunikasi, berinteraksi sosial dengan anak sebayanya dan kemampuan berimajinasi pada anak.

2.2. Data Mining

Data mining adalah proses menemukan pengetahuan yang menarik, seperti asosiasi, pola, perubahan, struktur yang signifikan dan anomali, dari sejumlah besar data yang disimpan dalam *database* atau gudang data atau repositori informasi lainnya. Telah banyak digunakan dalam beberapa tahun terakhir karena ketersediaan data dalam jumlah besar dalam bentuk elektronik, dan ada kebutuhan untuk mengubah data tersebut menjadi informasi yang berguna dan pengetahuan untuk aplikasi yang besar. Aplikasi ini ditemukan di bidang-bidang seperti *Artificial Intelligence*, *Machine Learning*, Analisis Pasar, Statistik dan Sistem *Database*, Manajemen Bisnis dan Pendukung Keputusan [18].

Data mining adalah proses mempekerjakan satu atau lebih teknik *machine learning* untuk menganalisis dan mengekstraksi *knowledge* secara otomatis. Penggunaan *data mining* memiliki tujuan untuk mengetahui pola universal dari data-data yang ada. Dalam menghasilkan suatu *knowledge* dari pola yang ada, diperlukan penerapan metode *scientific* yang disebut dengan *Knowledge Discovery in Database* (KDD) [22].

2.3. Text Mining

Text mining adalah teknik yang membantu pengguna untuk menemukan informasi yang berguna dari sejumlah besar dokumen teks digital di web atau *database*. Oleh karena itu penting bahwa model penambangan teks yang baik harus mengambil informasi yang memenuhi kebutuhan pengguna dalam kerangka waktu yang relatif efisien [16].

Text mining dapat didefinisikan sebagai suatu proses menggali informasi dimana seseorang user berinteraksi dengan sekumpulan dokumen menggunakan *tool* analisis yang merupakan komponen-komponen dalam *data mining*.

Text mining digunakan untuk mengolah dokumen sebelum dilakukan proses *similarity*. Didalam proses *text mining* terdapat proses *preprocessing*. *Preprocessing text* merupakan tindakan menghilangkan karakter-karakter tertentu yang terkandung dalam dokumen, seperti koma, tanda petik dan lain-lain serta mengubah semua huruf kapital menjadi huruf kecil [3].

Text mining adalah salah satu bagian penambangan data yang didefinisikan sebagai proses pencarian informasi di mana pengguna berinteraksi dengan setumpuk dokumen menggunakan alat analitik yang sebenarnya adalah komponen dalam *data mining* [8].

Text mining dalam prakteknya mencari pola-pola tertentu, mengasosiasikan suatu bagian teks dengan yang lain berdasarkan aturan-aturan tertentu, kata-kata yang dapat mewakili sehingga dapat dilakukan analisa keterhubungan antar satu dengan yang lain.

Berikut tahapan-tahapan proses didalam *text mining/preprocessing* [19]:

- a) *Casefolding*
Casefolding adalah tahap mengubah huruf besar menjadi huruf kecil dalam dokumen maka penghapusan tanda baca selain huruf "a" to "z" yang dianggap sebagai karakter pembatas.
- b) *Tokenizing*
Tokenizing adalah tahap memecah kalimat menjadi kata-kata. Dengan kata terbelah pertama, string yang sudah ada masukan akan lebih sederhana karena ditampilkan dalam setiap kata menurut ruang yang membaginya, sehingga dengan bentuk itu, akan mempermudah proses perubahan menjadi batang kata.
- c) *Filtering*
Filtering adalah tahap penghapusan kata-kata tidak dianggap mengandung arti atau pemikiran yang seharusnya ada ada (*stopwords*). *Stopword* berisi kata-kata umum yang sering muncul dalam sebuah dokumen dalam jumlah banyak namun tidak memiliki kaitan dengan tema tertentu.
- d) *Stemming*
Stemming adalah mengembalikan kata-kata yang diperoleh dari hasil *filtering* ke bentuk dasarnya, menghilangkan imbuhan awal (*prefix*) dan imbuhan akhir (*suffix*) sehingga didapat kata dasar.

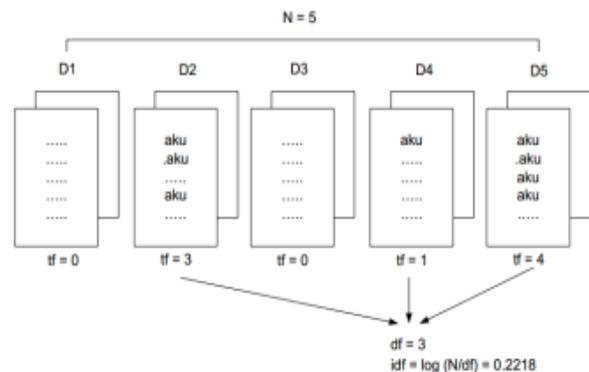
2.4. *Term Frequency (tf) – Inverse Document Frequency (idf)*

Term Frequency (tf) merupakan frekuensi kemunculan suatu kata (*term*) dalam dokumen. Oleh sebab itu, *tf* memiliki nilai yang bervariasi dari satu dokumen ke dokumen lain tergantung dari tingkat kepentingan sebuah *term* dalam sebuah dokumen. Semakin sering suatu *term* muncul dalam suatu dokumen, *term* tersebut akan memiliki nilai *tf* yang lebih besar daripada *term-term* lain yang jarang muncul. Ini adalah sebuah metode umum untuk menilai bagaimana suatu kata diperlukan dokumen. Ini biasanya digunakan sebagai faktor pembobotan pengambilan informasi[14].

Penggunaan faktor *tf* belum mencukupi dalam menentukan pembobotan. Untuk itu, diperlukan faktor *Inverse Document Frequency (idf)* yang merupakan sebuah statistik “global” yang mengkarakteristikan sebuah *term* dalam keseluruhan koleksi dokumen. *idf*

merupakan sebuah perhitungan dari bagaimana *term* didistribusikan secara luas pada koleksi dokumen yang bersangkutan. Semakin sedikit dokumen yang mengandung *term* yang dimaksud, maka nilai *idf* semakin besar. Jika setiap dokumen dalam koleksi mengandung *term* yang bersangkutan, maka nilai *idf* dari *term* tersebut adalah nol (0). Hal ini menunjukkan bahwa setiap *term* yang muncul pada dokumen dalam koleksi tidak berguna untuk membedakan dokumen berdasarkan topik tertentu.

TF-IDF (Gambar 1) juga merupakan cara yang sangat menarik untuk mengubah representasi tekstual informasi menjadi *Vector Space Model (VSM)*, atau menjadi fitur yang jarang [4].



Gambar 1. Ilustrasi Algoritma TF-IDF

Keterangan:

- D1 - D5 = dokumen
- tf = banyaknya term yang dicari pada setiap dokumen
- N = total dokumen
- Df = banyaknya dokumen yang mengandung term yang dicari

Dalam penelitian ini, algoritma *Term Frequency (TF) – Inverse Document Frequency (IDF)* ditetapkan pada tahap *similarity document*. Nilai TF-IDF diperoleh dengan menggunakan persamaan:

$$W_{i,j} = tf_{i,j} \times idf_j = tf_{i,j} \times \log\left(\frac{N}{df_j}\right) \quad (1)$$

Dimana:

- $W_{i,j}$ = bobot *term* ke-j terhadap dokumen ke-i
- tf_{ij} = jumlah kemunculan *term* j ke dalam dokumen i
- N = jumlah dokumen secara keseluruhan
- df_j = jumlah dokumen yang mengandung *term* j

Berdasarkan persamaan (1), berapapun besarnya nilai tf_{ij} apabila $N = df_j$ maka akan didapat hasil nol (0) untuk perhitungan *idf*. Untuk itu, dapat ditambahkan nilai 1 pada sisi *idf*, sehingga perhitungan perhitungan bobotnya menjadi:

$$W_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_j}\right) + 1 \quad (2)$$

Perhitungan bobot dari *term* tertentu dalam sebuah dokumen dengan menggunakan $tf \times idf$ menunjukkan bahwa deskripsi terbaik dari dokumen adalah *term* yang banyak muncul dalam dokumen tersebut dan sangat sedikit muncul pada dokumen lain.

Inverted Indexing adalah metode yang populer di sistem pengambilan informasi. Karena efisiensi yang tinggi dalam hal kecepatan dan penyimpanan hemat ruang terdiri dari dua bagian[9].

2.5. Vector Space Model (VSM)

Menurut (Singh dan Singh, 2016) menjelaskan VSM adalah model aljabar yang digunakan untuk Informasi Pengambilan kembali. Ini merupakan dokumen bahasa alami dalam cara formal dengan menggunakan vektor dalam multidimensional ruang. *Vector Space Model* (VSM) adalah cara mewakili dokumen melalui kata-kata itu berisi. Konsep di balik *vector space model* adalah dengan menempatkan istilah, dokumen, dan pertanyaan dalam ruang *term-document* adalah mungkin untuk menghitung persamaan antara kueri dan istilah atau dokumen, dan memungkinkan hasil perhitungan untuk diberi peringkat sesuai dengan ukuran kemiripan diantara mereka. VSM memungkinkan keputusan dibuat tentang dokumen mana yang mirip satu sama lain dan untuk pertanyaan[20].

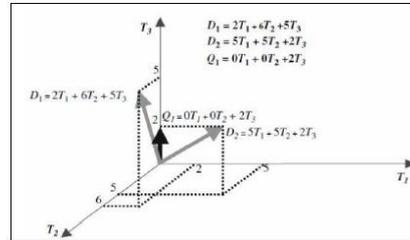
Menurut (Karyono dan Utomo, 2012) menjelaskan bahwa pada sistem IR, kemiripan antar dokumen didefinisikan berdasarkan representasi *bag of words* dan dikonversi ke suatu model ruang vektor (*vector space model*, VSM). Model ini diperkenalkan oleh Salton dan telah digunakan secara luas. Pada VSM, setiap dokumen di dalam database dan *query* pengguna direpresentasikan oleh suatu vektor multi-dimensi.

Dimensi sesuai dengan jumlah term dalam dokumen yang terlibat[21]. Pada model ini:

- Vocabulary* merupakan kumpulan semua *term* berbeda yang tersisa dari dokumen setelah preprocessing dan mengandung t *term index*. *Term-term* ini membentuk suatu ruang vektor.
- Setiap *term* i di dalam dokumen atau *query* j , diberikan suatu bobot (weight) bernilai *real* w_{ij} .
- Dokumen dan *query* diekspresikan sebagai vektor t dimensi $d_j = (w_{j1}, w_{j2}, \dots, w_{jt})$ dan terdapat n dokumen di dalam koleksi, yaitu $j = 1, 2, \dots, n$.

Vector Space Model (VSM) dapat dibagi menjadi tiga tahap. Tahap pertama adalah pengindeksan pesan di mana konten yang mengandung istilah diambil dari teks dokumen. Tahap kedua adalah pembobotan dari istilah yang diindeks untuk meningkatkan pengambilan pesan sebagai spam. Tahap terakhir memeringkat pesan sehubungan dengan permintaan menurut ukuran kesamaan [10].

Setiap dokumen dianggap sebagai vektor as- D_1, D_2, \dots, D_r ; dimana r adalah jumlah dokumen di corpora[11]. Contoh dari *vector space model* tiga dimensi untuk dua dokumen D_1 dan D_2 , satu *query* pengguna Q_1 , dan tiga term T_1, T_2 dan T_3 diperlihatkan pada gambar 2 berikut ini:



Gambar 2. Contoh Model Ruang Vektor

Dalam *vector space model*, koleksi dokumen direpresentasikan oleh matriks *term document* (atau matriks *term-frequency*). Setiap sel dalam matriks bersesuaian dengan bobot yang diberikan dari suatu term dalam dokumen yang ditentukan. Nilai nol berarti bahwa term tersebut tidak hadir di dalam dokumen. Akan diperjelas dengan gambar berikut:

$$\begin{bmatrix}
 & T_1 & T_2 & \dots & T_t \\
 D_1 & w_{11} & w_{21} & \dots & w_{t1} \\
 D_2 & w_{12} & w_{22} & \dots & w_{t2} \\
 \dots & \dots & \dots & \dots & \dots \\
 D_n & w_{1n} & w_{2n} & \dots & w_{tn}
 \end{bmatrix}$$

Gambar 3. Contoh Matriks Term Document

2.6. Matching Document

Matching Documents adalah tahap dimana akan dilakukan perhitungan tingkat kemiripan (*similarity*) antara dokumen acuan (kunci jawaban) dengan kandidat dokumen (jawaban siswa). Perhitungan *matching documents* ini menggunakan metode *Cosine Similarity*.

Cosine Similarity adalah salah satu ukuran kesamaan yang paling populer yang diterapkan pada teks dokumen, dalam berbagai aplikasi pengambilan informasi. Ketika dokumen direpresentasikan sebagai vektor jangka, kesamaan dua dokumen sesuai dengan korelasi antara vektor. Pada dasarnya, kesamaan kosinus adalah ukuran kesamaan itu dapat digunakan untuk membandingkan dokumen sehubungan dengan vektor kata kueri tertentu [13]. Ukuran *cosine* sering digunakan untuk menghitung ukuran kemiripan dan untuk menentukan sudut antara vektor dokumen dan vektor kueri [6].

Rumus dari *Cosine Similarity* itu sendiri adalah sebagai berikut:

$$Sim(D, D_i) = \cos \theta = \frac{D \cdot D_i}{|D| |D_i|} = \frac{\sum W_{q,j} W_{i,j}}{\sqrt{\sum W_{q,j}^2} \sqrt{\sum W_{i,j}^2}} \quad (3)$$

Dimana :

- D = Dokumen acuan
- D_i = Dokumen ke-i
- $W_{q,j}$ = Bobot *term* j terhadap dokumen acuan
- $W_{i,j}$ = Bobot *term* j terhadap dokumen ke-i

Similarity atau $Sim(D, D_i) = 1$ jika $D = D_i$, dan $Sim(D, D_i) = 0$ ketika D dan D_i tidak memiliki kesamaan sama sekali (benar-benar berbeda). Dokumen diurutkan berdasarkan nilai cosinus yang menurun, $CosSim(d, q) = 1$ ketika $d = q$, $CosSim(d, q) = 0$ ketika d dan q berbagi tanpa syarat[17].

2.7. Information Retrieval

Information Retrieval adalah sistem yang digunakan untuk mencari kemiripan suatu dokumen[15]. Ada proses mendasar dalam sistem pencarian informasi untuk mendapatkan informasi yang memenuhi kebutuhan pengguna dengan mencari kesamaan antara *query* dan dokumen yang ada. Di bab ini, kita akan menunjukkan beberapa proses dasar pencocokan permintaan dengan dokumen, tergantung pada *Vector Space Model* (VSM) di mana baik dokumen dan *query* direpresentasikan sebagai vector [7].

Ada beberapa proses sistem temu kembali informasi sebagai berikut:

- 1) Menghapus tanda baca dari setiap dokumen.
- 2) Menghapus kata-kata penghenti dari setiap dokumen, seperti kata depan, artikel dan kata lain yang muncul sering dalam dokumen tanpa menambahkan makna apa pun.
- 3) *Stemming* kata-kata menggunakan *porter stemmer* yang paling sering digunakan.
- 4) Indeks Terbalik, memberikan setiap dokumen nomor seri yang unik, yang dikenal sebagai pengenalan dokumen dan menghubungkan mereka dengan elemen-elemen dalam dokumen apa pun (ID dokumen).

3. Hasil dan Pembahasan

3.1. Implementasi

Pada penelitian ini pembobotan terhadap data uji dilakukan dengan menggunakan metode *Term Frequency – Inverse Document Frequency* (TF-IDF) dan dilanjutkan dengan metode *Vector Space Model* (VSM). Penelitian ini bertujuan untuk mengetahui dokumen dengan kemiripan tertinggi dari *query* yang diuji yaitu: “Anak Tidak Suka Diam dan Berbicara Aneh”. Sebelum melakukan proses pembobotan, setiap data uji harus terlebih dahulu melalui tahap *preprocessing*, yang terdiri dari *casefolding*, *tokenizing*, *filtering* dan *stemming*. Adapun tahap *preprocessing* adalah sebagai berikut:

- 1) Data Uji Penelitian

Penelitian ini menggunakan data uji sebanyak 10 data gejala gangguan autisme. Lihat Tabel 1.

Tabel 1. Tabel gejala autisme

Kode Gejala	Nama Gejala
GJ01	Tidak dapat berbicara
GJ02	Bisa berbicara namun tidak jelas
GJ03	Sering berbicara berlebihan
GJ04	Suka mengucapkan bahasa/kata-kata yang aneh secara berulang-ulang
GJ05	Tidak dapat menunjuk sesuatu dengan jari sendiri
GJ06	Tidak dapat menunjukkan keinginan dengan kata-kata
GJ07	Suka menarik-narik orang lain jika menginginkan sesuatu
GJ08	Tidak ada usaha dalam berkomunikasi
GJ09	Menghindar jika didekati
GJ10	Tidak dapat berinteraksi dengan lingkungan sekitar

2) Casefolding

Tahap *casefolding* ini adalah mengubah huruf besar pada semua data gejala gangguan autisme menjadi huruf kecil.

tidak dapat berbicara bisa berbicara namun tidak jelas sering berbicara berlebihan suka mengucapkan bahasa atau kata-kata yang aneh secara berulang-ulang tidak dapat menunjuk sesuatu dengan jari sendiri tidak dapat menunjukkan keinginan dengan kata-kata suka menarik-narik orang lain jika menginginkan sesuatu tidak ada usaha dalam berkomunikasi menghindar jika didekati tidak dapat berinteraksi dengan lingkungan sekitar
--

Gambar 4. Casefolding

3) Tokenizing

Tahap *tokenizing* ini adalah memecah kalimat pada satu gejala gangguan autisme menjadi kata-kata.

tidak dapat berbicara	bisa berbicara namun tidak jelas	sering berbicara berlebihan	suka mengucapkan bahasa atau kata-kata yang aneh secara berulang-ulang	tidak dapat menunjuk sesuatu dengan jari sendiri
tidak dapat menunjukkan keinginan dengan kata-kata	suka menarik-narik orang lain jika menginginkan sesuatu	tidak ada usaha dalam berkomunikasi	menghindar jika didekati	tidak dapat berinteraksi dengan lingkungan sekitar

Gambar 5. Tokenizing

4) Filtering

Tahap *filtering* ini adalah menghapus kata-kata yang dianggap tidak berkaitan dengan tema.

tidak dapat berbicara	bisa berbicara tidak jelas	sering berbicara berlebihan	suka mengucapkan bahasa kata aneh secara ulang	tidak dapat menunjuk sesuatu jari sendiri
tidak dapat menunjukkan keinginan kata	suka menarik orang lain menginginkan sesuatu	tidak ada usaha berkomunikasi	menghindar didekati	tidak dapat berinteraksi lingkungan sekitar

Gambar 6. Filtering

Hasil perhitungan *cosine similarity* dengan persamaan ke-3 dapat dilihat pada tabel 4.

Tabel 4. Perhitungan *Cosine Similarity*

GJ01	0.077171 / 0.69352)	(1.3458	*	0.08268	RANK 3
GJ02	0.077171 / 1.52401)	(1.3458	*	0.03763	RANK 4
GJ03	0.074749 / 1.50778)	(1.3458	*	0.03684	RANK 5
GJ04	1.238689 / 2.44481)	(1.3458	*	0.37647	RANK 1
GJ05	0.002422 / 1.78457)	(1.3458	*	0.00101	RANK 8
GJ06	0.002422 / 1.29354)	(1.3458	*	0.00139	RANK 6
GJ07	0.238689 / 2.11321)	(1.3458	*	0.08393	RANK 2
GJ08	0.002422 / 1.7462)	(1.3458	*	0.00103	RANK 7
GJ09	0 / (1.3458 * 1.41421)			0	RANK 10
GJ10	0.002422 / 1.79097)	1.3458	*	0.001	RANK 9

Dari perhitungan tabel 4 tersebut diatas, Dapat dirangking tingkat kemiripan dokumen dengan query “Tidak Suka Diam dan Berbicara Aneh” adalah sebagai berikut:

GJ04, GJ07, GJ01, GJ02, GJ03, GJ06, GJ08, GJ05,
 GJ10, GJ09

Dan dapat diketahui bahwa GJ04 memiliki tingkat similaritas tertinggi terhadap data uji (DU) sebesar 0.37647.

4. Kesimpulan

Dari hasil penelitian ini, maka diambil kesimpulan bahwa metode *Vector Space Model* (VSM) dapat digunakan sebagai mesin pencarian untuk pencarian data gejala gangguan autisme.

Hasil pengujian menggunakan *Cosine Similarity* dengan menggunakan 10 data gejala gangguan autisme menunjukkan bahwa GJ04 dengan nama gejala Suka mengucapkan bahasa/kata-kata yang aneh secara berulang-ulang memiliki tingkat kemiripan yang tinggi dengan *query* sebesar 0.37647.

Daftar Rujukan

- [1] Yulianto, Budi., Widodo Budiharto and Iman H. Kartowisastro. *The Performance of Boolean Retrieval and Vector Space Model in Textual Information Retrieval*. CommIT (Communication & Information Technology) Journal. 2017. 33-39.
- [2] Hadi, Musa Wa'el., Fadi Thabtah and Hussein Abdel-Jaber. *A Comparative Study Using Vector Space Model With K-Nearest Neighbour on Text Categorization Data*. Proceedings of The World Congress on Engineering. 2017. ISBN:978-988-98671-5-7.
- [3] Man Yua, Yuan Xin Ouyang and Zhang Xiong. *A Text Categorization Methode Using Extended Vectore Space Model By Frequent Term Sets*. Journal of Information Science and Engineering 29. 2013. 99-114.
- [4] Singh, Nath Jitendra and Sanjay Kumar Dwivedi. *Analysis of Vector Space Model in Information Retrieval*. International Journal of Computer Application (IJCA). 2012. 14-18.
- [5] Priambodo, Bagus. *Answer Search Indonesian Language Hadith Using Vector Space Model in PDF Document*. International Research Journal of Computer Science (IRJCS). 2017. ISSN: 2393-9842. Vol.4, Issue 08. 1-5.
- [6] Ogheneovo, E. E. and R. B. Japheth. *Application of Vector Space Model to Query Ranking and Information Retrieval*. International Journal of Advanced Research in Computer Science and Software Engineering. 2016. ISSN: 2277 128X. Vol.6, Issue 5. 42-47.
- [7] Laith Mohammad Qasim Abualigah. *Applying Genetic Algorithms To Information Retrieval Using Vector Space Model*. International Journal of Computer Science, Engineering and Application (IJCSEA). 2015. DOI: 10.5121/ijcsea.2015.5102. Vol.6, No.1. 19-28.
- [8] Slamet, C. A R Atmadja, D S Maylawati, R S Lestari, W Darmalaksana and M A Ramdhani. *Automated Text Summarization for Indonesian Article Using Vector Space Model*. The 2nd Annual Applied Science and Engineering Conference (AASEC). 2017. DOI: 10.1088/1757-899X/288/1/012037.
- [9] Lartariyatham, Sasithorn., Pongpisit Wuttidittachotti, Somchai Prakancharoen and Sakda Arj-ong Vallipakorn. *Comparative Weighting Methods of Vector Space Model*. ARPN Journal of Engineering and Applied Sciences. 2017. ISSN: 1819-6608. Vol.10, No.3. 1316-1323.
- [10] Bansal S. *Comparison Between The Probabilistic and Vector Space Model For Spam Filtering*. Internation Journal of Computational Intelligence Techniques. 2012. ISSN: 0976-0466. Vol.3, Issue 2. 82-85.
- [11] R. K. Makhijani and I. N. Bharambe. *Design of Search Engine Using Vector Space Model for Personalized Search*. Internation Journal of Computer Science and Mobile Computing (IJCSMC). 2013. ISSN: 2320-088X. Vol.2, Issue 3. 63-66.
- [12] Eman Al Mashagba, Feras Al Mashagba and Mohammad Othman Nassar. *Query Optimization Using Genetic Algorithms in The Vector Space Model*. International Journal of Computer Science Issues (IJCSI). 2011. ISSN: 1694-0814. Vol.8, Issue 5. 450-457.
- [13] J. Usharani. *A Genetic Algorithm Bases on Cosine Similarity for Relevant Document Retrieval*. International Journal of Research & Technology (IJERT). 2013. ISSN: 2278-0181. Vo.2, Issue 2. 1-5.
- [14] Mesariya, Priyanka and Nidhi Madia. *Document Ranking Using Costumizes Vector Method*. International Journal of Trend is Scientific Research and Development (IJTSRD). 2017. ISSN: 2456-6470. Vol.1(4).
- [15] Minh Chau Huynh, Pham Duy Thanh Le and Trong Hai Doung. *Improved Vector Space Model TF/IDF Using Lexical Relations*. International Journal of Advance Computer Research. 2015. ISSN: 2249-7277. Vol.5. 334-346.
- [16] Gawande, Prashant. S. and Prof. Ashish Suryawanshi. *Improving Web Page Classification by Vector Space Model*. International Journal of Innovative Research in Computer and Communication Engineering. 2015. ISSN: 2320-9801. Vol.3, Issue 2. 1252-1257.
- [17] Ravindran, R. Malathi and Dr. Anthony Selvadoss Thanamani. *K-Means Document Clustering Using Vector Space Model*. Bonfring International Journal of Data Mining. 2015. ISSN: 2277-5048. Vol.5, No.2. 10-14.
- [18] Singh, Jitendra Nath and Sanjay K. Dwivendi. *Performance Evaluation of Search Engines Using Enhanced Vector Space Model*. Journal of Computer Science. 2015. DOI:10.3844/jcssp.2015.692.698. 692-698.
- [19] Ni Made Ari Lestari, I Ketut Gede Darma Putra and AA Ketut Agung Cahyawan. *Personlaity Types Classification for Indonesian Text in Partnets Searching Websites Using Naïve Bayes Methods*. International Journal of Computer Science Issues (IJCSI). 2013. ISSN: 1694-0784. Vol. 10, Issue 1. 1-7.
- [20] Jovita, Linda, Andrei Hartawan and Derwin Suhartono. *Using Vector Space Model in Question Answering System*.

- International Conference on Computer Science and Computational Intelligence (ICCCSCI). 2015. ISSN: 1877-0509. 305-311.
- [21] Singh, Vaibhav Kant and Vinay Kumar Singh. *Vector Space Model: An Information Retrival System*. International Journal of Advanced Engineering and Studies. 2015. E-ISSN: 2249-8974. 141-143.
- [22] Jayawardanu, Ivana Herliana and Seng Hansun. 2015. Rancang Bangun Sistem Pakar Untuk Deteksi Dini Katarak Menggunakan Algoritma C4.5. *Ultima Computing*. Vol. VII, No.2, ISSN: 2355-3286. 48-58.
- [23] Tutik A, Gusti Ayu Kadek, Rosa Delima dan umi Proboyekti. 2009. Penerapan *Forward Chaining* Pada Program Diagnosa Anak Penderita Autisme. Yogyakarta. *Jurnal Informatika* Volume 5 Nomor 2 November 2009: 46-59. Diambil dari: <http://ti.ukdw.ac.id/ojs/index.php/informatika/article/viewFile/73/35>. (20 Oktober 2014).