JuTech
Jurnal Ilmiah Teknologi & Informasi

Website: http://ojs.itb-ad.ac.id/index.php/JUTECH
E-ISSN: 2797 - 4111

Pemanfaatan *Vector Space Model* Algoritma Nazief Andriani Pembobotan *Tfidf* Pada Prototipe Klasifikasi Teks Bahasa Indonesia

Reza Fitriansyah[1]

[1]*Institut Teknologi dan Bisnis Ahmad Dahlan, Jakarta*

ABSTRACT

Information Retrieval is one of the IR techniques that will be used to present the dummy profile is the vector space model where n is the sum of all the terms in the list. To overcome this problem, one technique that can be used is to classify the text of the document in accordance with characteristics, features, and classes based on the standard rules of the language to be processed. In this study, Indonesian is the language used as a reference source. The object of this research is the Indonesian Language Text document. This study will test the application of text classification engine of Indonesian language using Stemming Nazief Andriani algorithm, K-Nearest Neighbor algorithm and Vector Space Model Model based on frequency weighting of TFIDF number of words and Simpson functions. By using document news as document learning, as many as 15 (fifteen) documents with 3 (three) categories, yielding the average value of Precision and Recall of 81.33%.

Kata Kunci: *Information Retrieval, Vector Space Model, Fungtions Simpson, Precision dan Recall*

Januari – Juni 2021, Vol 2 (1) : hlm 37-47

*

(*) Korespondensi: rezafsyh@gmail.com *(Reza Fitriansyah)*

I. PENDAHULUAN

Information Retrieval adalah mesin pencari informasi, pada suatu *query* yang dapat memenuhi kebutuhan *user* dari kumpulan dokumen yang ada. Proses yang terjadi didalam *Information Retrieval* terdiri dari proses *Indexing* untuk memebentuk basisdata terhadap koleksi dokumen yang dimasukan kedalam aplikasi, *indexing* merupakan proses persiapan yang dilakukan terhadap suatu dokumen sehingga dokumen siap untuk dimasukan kedalam aplikasi. Pada proses tahapan *Indexing* didalam *Information Retrieval* terdapat proses *Stemming* yaitu proses mengubah suatu kata bentuk menjadi kata dasar (Asian & Williams, 2005). Proses *Stemming* sangat tergantung kepada Bahasa yang akan *stemming*. Algoritma *Stemming* untuk masing-masing Bahasa berbeda, seperti Algoritma *Stem*ming Bahasa inggris memiliki tata bahasa yang berbeda dengan Bahasa Indonesia Proses *Stemming* Bahasa Indonesia lebih rumit dibandingkan dengan Bahasa lainya. Dikarenakan dalam Bahasa Indonesia terdapat variasi imbuhan yang harus dibuang untuk mendapatkan kata dasar dari sebuah kata. Penggunaan algoritma stemming yang sesuai mempengaruhi *system IR*. Dalam penelitian ini akan di analisis tingkat relevansi suatu *search engine* berdasarkan *TFIDF* dengan fungsi *Simpson* dibandingkan dengan algoritma Nazief Andriani.

II.LANDASAN TEORI

2.1 *Tokenizing*

Tokenisasi adalah proses mengidentifikasi kata dalam karakter huruf, terutama memisahkan tanda baca, tetapi juga dengan mengidentifikasi kontraksi, perkata, dan sebagainya. (Ignatow, G, dan Mihalcea. R, 2017).

2.2 Algoritma *Stemming*

Algoritma *Stemming* adalah komputasi yang mencari asal kata dari suatu kata dalam kalimat yang dilakukan dengan cara memisahkan masing-masing kata dari kata dasar dan imbuhan. Sebagai contoh: *group, groups, grouped,* dan *grouping* dihasilkan dari kata dasar *group* (Asian & Williams, 2005). Saat ini hanya ada beberapa algoritma stemming untuk Bahasa Indonesia yang telah dikembangkan diantaranya yaitu : Algoritma Nazief Andriani.

2.3 Nazief Andriani Stemmer for Bahasa Indonesia

Nazief Andriani *Stemmer for* Bahasa Indonesia dibuat oleh Boby Nazief dan Mirna Andriani dari Fakultas Ilmu Komputer Universitas Indonesia tahun 1996, algoritma ini mengacu pada aturan morfologi Bahasa Indonesia yang mengelompokan imbuhan, yaitu imbuhan yang diperbolehkan atau imbuhan yang tidak diperbolehkan Pengelompokan ini termasuk imbuhan di depan (awalan), imbuhan kata belakang (akhiran), imbuhan kata tengah (sisipan), dam kombinasi imbuhan pada awal dan akhir kata (konfiks). Algoritma ini menggunakan kamus kata yang digunakan mengetahui bahwa proses *Stemming* telah mendapatkan kata dasar (Andriani, Nazief, Asian, & William, 2007).

2.4 *Vector Space Model*

dalam mencari informasi, ada berbagai macam teknik dan pendekatan yang dapat digunakan, seperti metode *Vector Space Model* dan metode lainnya. Namun yang paling banyak digunakan sekarang ini adalah *Vector Space Model*. Pada sistem *Information Retrieval*, kemiripan antar dokumen dikonversi ke dalam suatu model ruang vektor atau *Vector Space Model* (*VSM*). Model ini diperkenalkan oleh Salton (Salton, 1989). Pada *VSM*, setiap dokumen di dalam database dan query pengguna direpresentasikan oleh suatu *vector multi-dimensi* sesuai dengan jumlah *term* dalam dokumen yang terlibat. Pada model ini :

1. *Vocabulary* merupakan sebuah kumpulan term berbeda yang tersisa dari suatu dokumen setelah preprocessing dan mengandung *termindex*. *Term* ini membentuk suatu ruang vector.
2. Setiap *term* di sebuah dokumen atau *query* j, diberikan suatu bobot (*weight*) bernilai real wij.¬
3. Dokumen dan query diimplementasikan sebagai *vector* t dimensi dj=(w1, w2,…,wtj) dan terdapat n dokumen di dalam koleksi, yaitu j = 1,2, …,n.

2.5 *Term Frequency*

Term Frequency adalah sebuah metode untuk menghitung bobot tiap *term* dalam *text* (Yates dan Ribeiro-Neto, 1999).. Dalam metode ini, tiap *term* diasumsikan memiliki nilai yang sebanding dengan jumlah kemunculan *term* tersebut pada *text*. Bobot sebuah *term* t pada sebuah *text* d dirumuskan dalam persamaan berikut:

$$\boldsymbol{W(d,i) = TF(d,i)}$$

2.6 *Inverse Document Frequency*

Inverse Document Frequency (IDF) fokus pada kemunculan *term* pada keseluruhan *document text*. Pada *IDF*, *term* yang jarang muncul pada keseluruhan koleksi *term* dinilai lebih berharga. Nilai kepentingan tiap *term* diasumsikan berbanding terbalik dengan jumlah text yang mengandung *term* tersebut. Nilai *IDF* sebuah *term* i dirumuskan dalam persamaan berikut:

$$\boldsymbol{IDF_i = \log(\frac{D}{df_i})}$$

2.7 *Term Frequency* dan *Inverse Document Frequency* (*TFIDF*)

Pada pengindeksan yang berdasarkan pada frekuensi istilah diperkirakan bahwa definisi indeks terbaik adalah definisi yang sering muncul dalam dokumen tetapi jarang muncul dalam keseluruhan dokumen. Sebuah kepentingan jenis ini yang umum adalah perkalian *TFIDF* dengan bobot wij sebuah istilah Ti dalam dokumen Di, didefinisikan sebagai frekuensi istilah dikalikan dengan fungsi inverse document frequency (Baeza-Yates dan Ribeiro-Neto, 1999).

$$\boldsymbol{TF(d,i).IDF(i)}$$

2.8 *Simpson Similarity*

Kualitas dari sebuah mesin pencari dapat dilihat dari tingkat relevansi data yang didapatkan. Untuk mengetahui tingkat perbedaan antara *query* yang di*input*kan dengan dokumen. (D Ellis, J Fumer-Hines, & P Willett, 1993), Formula *Simpson* memberikan nilai bobot setiap *term* yang telah diperoleh akan dikalkulasikan dengan nilai seluruh *query* yang telah dihitung sebelumnya, sebagai berikut :

Tabel 1: Hasil *Information Retrival* dirangkum dalam tabel berikut :

		Counts, tfi						Weights, wi = tfi*IDFi			
Terms	Q	D1	D2	D3	DFI	D/DFI	IDFi	Q	D1	D2	D3
Sebuah	0	1	1	1	3	1	0	0	0	0	0
tiba	0	0	1	1	2	1.5	0.1761	0	0	0.1761	0.1761
rusak	0	1	0	0	1	3	0.4771	0	0.4771	0	0
pengiriman	0	0	1	0	1	3	0.4771	0	0	0.4771	0
api	0	1	0	0	1	3	0.4771	0	0.4771	0	0
emas	1	1	0	1	2	1.5	0.1761	0.1761	0.1761	0	0.1761
di	0	1	1	1	3	1	0	0	0	0	0
dari	0	1	1	1	3	1	0	0	0	0	0
perak	1	0	2	0	1	3	0.4771	0.4771	0	0.9542	0
pengiriman	0	1	0	1	2	1.5	0.1761	0	0.1761	0	0.1761
truk	1	0	1	1	2	1.5	0.1761	0.1761	0	0.1761	0.1761

Saya telah menambahkan empat kolom terakhir untuk menggambarkan semua perhitungan bobot pada vector space. Mari kita analisa data mentah, kolom demi kolom :

1. Kolom 1 - 5: Pertama, kita membuat indeks istilah dari dokumen dan menentukan istilah penghitungan tfi untuk query dan setiap dokumen Dj.
2. Kolom 6 - 8: Kedua, kita menghitung frekuensi dokumen di untuk setiap dokumen. Karena IDFi = log (D / dfi) dan D = 3.
3. Kolom 9-12: Ketiga, kita mengambil produk IDF dan menghitung bobot istilah. Kolom ini dapat dilihat sebagai matriks di mana sebagian besar nilai nol.

Similarity Analysis

Pertama untuk setiap dokumen dan *query*, kita menghitung semua panjang *vektor* (istilah nol diabaikan).

$$\mathbf{Di} = \sum W_{i,j}$$

D1 = 0.4771 + 0.4771 + 0.1761 + 0.1761 = 1.3064

D2 = 0.1761 + 0.4771 + 0.9542 + 0.1761 = 1.7835

D3 = 0.1761 + 0.1761 + 0.1761 + 0.1761 = 0.5283

Selanjutnya, kita menghitung semua nilai (nilai nol diabaikan)

$$\mathbf{Q} = \sum \boldsymbol{W_{Q,j}}$$

Q = 0.1761 + 0.4771 + 0.1761 = 0.6532

$$\mathbf{Q.Di} = \sum \mathbf{min}(\boldsymbol{W_{Q,j}} . \boldsymbol{W_{i,j}})$$

D1 = 0.4771 + 0.4771 + 0.1761 + 0.1761 = 1.3064

Q.D2 = (0.4771 * 0.9542) + (0.1761 * 0.1761) = 0.8075

Q.D3 = (0.1761 * 0.1761) + (0.1761 * 0.1761) = 0.0620

Sekarang kita hitung nilai kemiripannya

$$\frac{\sum \boldsymbol{min}(\boldsymbol{W_{Q,j}} . \boldsymbol{W_{i,j}})}{\boldsymbol{min}(\sum \boldsymbol{W_{Q,j}} . \sum \boldsymbol{W_{i,j}})}$$

Simpson D1 = $\frac{\boldsymbol{Q.D1}}{\sum(\boldsymbol{Q.D1})}$ $\frac{\mathbf{0.0310}}{\mathbf{0.6532 * 1.3064}}$ = 0.0363

Simpson D2 = $\frac{\boldsymbol{Q.D2}}{\sum(\boldsymbol{Q.D2})}$ $\frac{\mathbf{0.8075}}{\mathbf{0.6532 * 1.7835}}$ = 0.6931

Simpson D3 = $\frac{\boldsymbol{Q.D3}}{\sum(\boldsymbol{Q.D3})}$ $\frac{\mathbf{0.0620}}{\mathbf{0.6532 * 0.5283}}$ = 0.1797

.

III.METODE

3.1 Metode Penelitian

Dalam melakukan pengklasifikasian dokumen, diperlukan dokumen *classifier* yang dipakai untuk tolak ukur dokumen yang akan uji ketika akan dikalsifikasikan. Adapun metode yang akan dipakai untuk melakukan pengklasifikasian dokumen *classifier* dengan menggunakan pendekatan algoritma *stemming* Nazief Adriani dan pembobotan dokumen menggunakan fungsi *simpson*.

3.2 Prototipe Model

Model sistem pengklasifikasian pada penelitian ini menggunakan prototipe model, pedekatan yang dipilih karena mempunyai struktur yang sesuai dalam mengembangkan simulasi model sistem pengklasifikasian dokumen menggunakan *vector space model* algoritma *stemming* Nazief Andriani teks Bahasa Indonesia pembobotan frekuensi kata dalam dokumen dan fungsi *Simpson*.
Berikut penjelasan proses yang menjadi komponen utama proses dalam sebuah sistem model pengklasifikasian :

1. Unggah Dokumen
   Unggah Dokumen adalah proses dimana dokumen dimasukann ke aplikasi.
2. Konversi Dokumen
   Konversi Dokumen adalah proses dimana dokumen *PDF* yang di-Upload kemudian diubah formatnya menjadi teks, yang kemudian isi dokumen teks tersebut akan diproses lebih lanjut untuk mencari nilai token di dokumen tersebut.

3. Proses Tokenisasi
   Pada proses ini dokumen yang berisi sekumpulan kalimat, akan dipecah menjadi kata perkata. Tanda baca dan karakter khusus akan ikut terbuang dari kumpulan kata tersebut.
4. Proses *Stopword Removal*
   Stopword merupakan kumpulan kata yang tidak memiliki hubungan dengan topik dari suatu dokumen, misalnya kata pengganti, dia, mereka, aku, kamu, dll. Pada proses *stopword removal* semua kata-kata yang tidak mempunyai hubungan dengan isi dari topik akan dibuang.
5. Proses *Stemming*
   Merupakan proses perubahan kata berimbuhan menjadi kata dasar, imbuhan yang dapt berupa awalan, akhiran atau keduanya. Dalam penelitian ini menggunakan algoritma *stemming* Nazief Andriani.
6. Proses pembobotan kata (*Term Weight*)
   Proses pembobotan kata merupakan proses dimana setiap kata yang ada dokumen diberikan nilai bobot. Proses perhitungan kata yaitu dimana bobot tiap kata yang dihitung dari jumlah kemunculan kata dalam sebuah query maupun isi dari sebuah dokumen.
7. Proses perhitungan nilai Similarity *(Similarity Measurement*)
   Similarity Measurement merupakan proses pengukuran relevansi kesamaan dokumen yang dimiliki dengan query yang dimasukan .
   Menghitung kesamaan ini dilakukan setelah proses pembobotan kata.
   Pada penelitian ini menggunakan fungsi *Similarity Simpson*.
8. Proses perangkingan dokumen
   Proses perangkingan dokumen menggunakan Algoritma *K-Nearest Neighbor*. *K-Nearest Neighbor* (*KNN*) adalah suatu metode untuk melakukan klasifikasi terhadap objek berdasarkan data learning yang jaraknya paling dekat dengan objek tersebut

3.3 Pengujian Data

3.3.1 Pengujian Ketepatan

Precision adalah suatu tingkat ketepatan hasil terhadap dokumen yang relevan dengan dokumen yang ditemukan. Rumus untuk menghitung *precision* adalah :

$$Precision = \frac{Jumlah\ dokumen\ relevan\ yang\ terambil}{Jumlah\ dokumen\ yang\ terambil\ dalam\ pencarian}$$

3.3.2 Pengujian Kelengkapan

Recall adalah suatu tingkat keberhasilan untuk mengenali suatu dokumen dari seluruh dokumen yang seharusnya dikenali. Rumus untuk menghitung *recall* adalah

$$Recall = \frac{Jumlah\ dokumen\ relevan\ yang\ terambil}{Jumlah\ dokumen\ relevan\ yang\ ada\ dalam\ database}$$

Pengujian ketepatan dan kelengkapan tersebut, biasanya diberi nilai dalam bentuk persentase, 1 sampai 100%. Sebuah sistem informasi akan dianggap baik jika tingkat *recall* maupun *precision* nya tinggi. (Shafi, S.M and Rafiq A. Rather, 2005).

IV.HASIL DAN PEMBAHASAN

4.1 Analisis Koleksi Dokumen

Pada penelitian ini digunakan dokumen dengan format *pdf* sebagai dokumen data, Dokumen tersebut terdiri daari 15 dokumen yang dibagi menjadi tiga kategori, dari masing-masing terdiri dari 5 dokumen dengan tiga jenis kategori, yaitu otomotif, teknologi, dan keuangan. Dokumen ini bersumber dari media berita *online*, berikut masing-masing dokumen dari setiap kategori.

Tabel 2 : Dokumen Eksperimen

No	Judul	Kategori	Sumber
1	Bikin Jok Sopir Xpander Pakai Sandaran Tangan	Otomotif	Kompas.com
2	Datsun Indonesia Mulai Buka Identitas Cross	Otomotif	Kompas.com
3	Iannone Mengaku Banyak Belajar dari Tim Suzuki	Otomotif	Kompas.com
4	Lebih Bertenaga, BMW M2 "Competition" Mengaspal Tahun Ini	Otomotif	Kompas.com
5	Pelek Pinggir Jalan di Tanah Abang Bisa Tukar Tambah	Otomotif	Kompas.com
6	2018, Produsen Jepang Janjikan Produk Elektronik Teknologi IoT	Teknologi	Sindo.com
7	Hadapi Persaingan 2018, HTC Justru Pangkas Jumlah Smartphone	Teknologi	Sindo.com
8	Panel Smart TV Samsung Diduga Alami Overheating	Teknologi	Sindo.com
9	Pertama di Dunia, Pria Tewas Karena Main Game Virtual Reality	Teknologi	Sindo.com
10	Terjual 233 Juta Unit, IPhone Produk Teknologi Terlaris di 2017	Teknologi	Sindo.com
11	Biaya Umrah Rp 20 Juta, Menag Supaya Jemaah Tidak Jadi Korban Travel	Keuangan	Detik.com
12	Kerugian Ambruknya Girder di Tol Desari Ditanggung Asuransi	Keuangan	Detik.com
13	Pemerintah Jual Surat Utang Rp 25,5 Triliun	Keuangan	Detik.com
14	Soal Utang Subsidi PLN dan Pertamina, Sri Mulyani Tunggu Audit	Keuangan	Detik.com
15	Tiket Kereta Bandara Soetta Mau Naik Jadi Rp 100	Keuangan	Detik.com

Dari table diatas terlihat judul, kategori dan sumber dari dokumen pengujian yang terdiri dari 15 dokumen didapat dari tiga media *online*, diantaranya : kompas.com, sindo.com dan detik.com. berdasarkan perolehan dokumen, pembagian sumber dapet di lihat pada tabel 3 berikut.

Tabel 3 : Dokumen Eksperimen

No	Sumber	Jumlah Dokumen
1	Kompas.com	5
2	Sindo.com	5
3	Detik.com	5

4.2 Hasil Perangkingan *KNN*

4.2.1 Hasil Perangkingan Q1 Untuk K=5

Tabel 4 : Hasil Perangkingan Q1 Untuk K=5

Katergori Query	Dokumen Query	Data	Hasil Similarity	KNN (K=5)	Kategori	Keterangan
Otomotif	Q1	D1	1	1	Otomotif	Relevan
		D4	0.126400341019	2	Otomotif	Relevan
		D3	0.101138468879	3	Otomotif	Relevan
		D5	0.0809655407072	4	Otomotif	Relevan
		D6	0.0443590942249	5	Teknologi	Tidak Relevan

Penjelasan Tabel 4

Sebelum melakukan pengujian, ada beberapa proses melakukan perangkingan :

1. Menentukan kategori yang akan di pilih
2. Memasukan 15 dokumen yang sudah disiapkan ke Document Learning.
3. Menguji satu persatu dokumen dengan membandingkan 15 dokumen di dalam aplikasi Document Classification, dengan menggunakan metode *TFIDF* dan fungsi similarity simpson.
4. Setelah semua proses dari no.1 sampai no.3 selesai, barulah mendapatkan hasil perangkingan (*K-Nearest Neighbour*). Maka dapat terlihat hasil perangkingan yang ada pada tabel.

4.2.2 Hasil Perangkingan Q6 Untuk K=5

Tabel 5 : Hasil Perangkingan Q6 Untuk K=5

Katergori Query	Dokumen Query	Data	Hasil Similarity	KNN (K=5)	Kategori	Keterangan
Teknologi	Q6	D6	1	1	Teknologi	Relevan
		D11	0.099720114351 3	2	Keuangan	Tidak Relevan
		D12	0.0901781859403	3	Keuangan	Tidak Relevan
		D15	0.0653754943032	4	Keuangan	Tidak Relevan
		D13	0.0579848888425	5	Keuangan	Tidak Relevan

Penjelasan Tabel 5

Sebelum melakukan pengujian, ada beberapa proses melakukan perangkingan :

1. Menentukan kategori yang akan di pilih
2. Memasukan 15 dokumen yang sudah disiapkan ke Document Learning.
3. Menguji satu persatu dokumen dengan membandingkan 15 dokumen di dalam aplikasi Document Classification, dengan menggunakan metode *TFIDF* dan fungsi similarity simpson.
4. Setelah semua proses dari no.1 sampai no.3 selesai, barulah mendapatkan hasil perangkingan (*K-Nearest Neighbour*). Maka dapat terlihat hasil perangkingan yang ada pada tabel.

4.2.3 Hasil Perangkingan Q11 Untuk K=5

Tabel 6 : Hasil Perangkingan Q11 Untuk K=5

Katergori Query	Dokumen Query	Data	Hasil Similarity	KNN (K=5)	Kategori	Keterangan
Keuangan	Q11	D11	1	1	Keuangan	Relevan
		D15	0.715628217596	2	Keuangan	Relevan
		D12	0.634583008757	3	Keuangan	Relevan
		D14	0.420862811463	4	Keuangan	Relevan
		D13	0.338986323642	5	Keuangan	Relevan

Penjelasan Tabel 6

Sebelum melakukan pengujian, ada beberapa proses melakukan perangkingan :

1. Menentukan kategori yang akan di pilih
2. Memasukan 15 dokumen yang sudah disiapkan ke Document Learning.
3. Menguji satu persatu dokumen dengan membandingkan 15 dokumen di dalam aplikasi Document Classification, dengan menggunakan metode *TFIDF* dan fungsi similarity simpson
4. Setelah semua proses dari no.1 sampai no.3 selesai, barulah mendapatkan hasil perangkingan (*K-Nearest Neighbour*). Maka dapat terlihat hasil perangkingan yang ada pada tabel

4.3 Hasil *Precision* dan *Recall*

Tabel 7 :. Hasil *Precision* dan *Recall* untuk Untuk K=5

No	Kategori	Query	Precision	Recall
1	Otomotif	Q1	80%	80%
2	Otomotif	Q2	80%	80%
3	Otomotif	Q3	100%	100%
4	Otomotif	Q4	100%	100%
5	Otomotif	Q5	100%	100%
6	Teknologi	Q6	20%	20%
7	Teknologi	Q7	100%	100%
8	Teknologi	Q8	80%	80%
9	Teknologi	Q9	60%	60%
10	Teknologi	Q10	60%	60%
11	Keuangan	Q11	100%	100%
12	Keuangan	Q12	100%	100%
13	Keuangan	Q13	40%	40%
14	Keuangan	Q14	100%	100%
15	Keuangan	Q15	100%	100%
Hasil Penjumlahan P dan R			1220%	1220%
Pengujian			1220%/15= 81.33	1220%/15= 81.33

Setelah melakukan pengujian maka dapat mehasilkan nilai *precision* dan *recall* yang relevan, berikut ini penjelasan tabel hasil *precision* dan *recall* diatas :

1. Setelah di uji dokumen Q1(D1), maka kategori Otomotif menghasilkan nilai yang relevan *precision* 80% dan *recall* 80%.
2. Setelah di uji dokumen Q2(D2), maka kategori Otomotif menghasilkan nilai yang relevan *precision* 80% dan *recall* 80%.
3. Setelah di uji dokumen Q3(D3), maka kategori Otomotif menghasilkan nilai yang relevan *precision* 100% dan *recall* 100%.

4. Setelah di uji dokumen Q4(D4), maka kategori Otomotif menghasilkan nilai yang relevan *precision* 100% dan *recall* 100%.
5. Setelah di uji dokumen Q5(D6), maka kategori Otomotif menghasilkan nilai yang relevan *precision* 100% dan *recall* 100%.
6. Setelah di uji dokumen Q6(D6), maka kategori Teknonogi menghasilkan nilai yang relevan *precision* 20% dan *recall* 20%.
7. Setelah di uji dokumen Q7(D7), maka kategori Teknologi menghasilkan nilai yang relevan *precision* 100% dan *recall* 100%.
8. Setelah di uji dokumen Q8(D8), maka kategori Teknologi menghasilkan nilai yang relevan *precision* 80% dan *recall* 80%.
9. Setelah di uji dokumen Q9(D9), maka kategori Teknologi menghasilkan nilai yang relevan *precision* 60% dan *recall* 60%.
10. Setelah di uji dokumen Q10(D10), maka kategori Teknologi menghasilkan nilai yang relevan *precision* 60% dan *recall* 60%.
11. Setelah di uji dokumen Q11(D11), maka kategori Keuangan menghasilkan nilai yang relevan *precision* 100% dan *recall* 100%.
12. Setelah di uji dokumen Q12(D12), maka kategori Keuangan menghasilkan nilai yang relevan *precision* 100% dan *recall* 100%.
13. Setelah di uji dokumen Q13(D13), maka kategori Keuangan menghasilkan nilai yang relevan *precision* 40% dan *recall* 40%.
14. Setelah di uji dokumen Q14(D14), maka kategori Otomotif menghasilkan nilai yang relevan *precision* 100% dan *recall* 100%.
15. Setelah di uji dokumen Q15(D15), maka kategori Otomotif menghasilkan nilai yang relevan *precision* 100% dan *recall* 100%.

Penjelasan Tabel 7 di bagian pengujian,
Setelah semua tabel diuji maka, seluruh tabel di jumlah menghasilkan nilai 1220 dan akan di bagi dengan 15 dokumen maka akan menghasilkan nilai 81.33%.

V.KESIMPULAN

Dari hasil penelitian yang telah dilakukan, maka dapat kesimpulan sebagai berikut,

1. Dengan menerapkan metode pembobotan frekuensi kata dalam dokumen dengan fungsi *simpson* pada prototipe model sistem dapat mengklasifikasikan dokumen teks berbahasa Indonesia
2. Dari hasil pengujian yang di di dapat nilai efektifitas *precesion* dan *recall* dengan nilai rata-rata 81.33% untuk k=5.

Hasil nilai *precesion* dan *recall* pada proses pengklasifikasi dokumen sangat dipengaruhi oleh frekuensi kata yang ada pada dokumen, dimana proses stemming dan *stopword removal* sangat mempengaruhi nilai pembobotan pada tiap dokumen.

DAFTAR PUSTAKA

Adriani 2007 Adriani, M., NAzief, B., Asian J., & Williaws, H. E. *Stemming* Indonesia A *Confixs Stripping Approach*. ACMTransactions on Asian *Language Information*.

Asian, 2005 Asian, J., & Williams, H. E. *Stemming* Indonesia. Australia *Computer Science Conference*.

Barakbah, Ali Ridho 2010, *Instance Base Classifier (Nearest Neightbour)*.

Ellis D, Fummer-Hines J, Willett P 1993, *Measuring the degree of similarity between objects in text retrieval systems*, Perspectives in *Information Manajement*, 3(2), 128-149.

Ignatow, G, dan Mihalcea. R, 2017, Text Mining A Guidebook for the Social Sciences, SAGE Publication, Inc, London, UK.

Salton, G. 1983, *Introduction to Modern Information Retrieval*. McGraw Hill.