SCIENCE AND INFORMATION TECHNOLOGY
SINTECH
JOURNAL

SINTECH JOURNAL
Vol. 2, No 2, Oktober 2018
p-ISSN 2598-7305 (Print), e-ISSN 2598-9642 (Online)
Available Online at :
http://jurnal.stiki-indonesia.ac.id/index.php/sintechjournal

KLASIFIKASI MULTILABEL PADA ABSTRAK TUGAS AKHIR MENGGUNAKAN *VECTOR SPACE MODEL* DAN *K-NEAREST NEIGHBORS*

I Putu Yoga Indrawan[1], I Gede Indrawan[2], I Made Candiasa[3]

Program Studi Magister Ilmu Komputer, Program Pascasarjana
Universitas Pendidikan Ganesha
Jalan Udayana No.11 Singaraja, Bali 81116 - Indonesia

e-mail : yoga.indrawan@pasca.undiksha.ac.id[1], gindrawan@pasca.undiksha.ac.id[2],
made.candiasa@pasca.undiksha.ac.id[3]

Received : September, 2018	Accepted : October, 2019	Published : October, 2019

Abstract

The final project is one of the requirements of graduation students. Students who want to do the final project need to see the final project result on the same topic that has been done before. With a large number of end-task documents, it certainly takes a great effort to find the final project document on the same topic. The final grouping can be automated using the document classification method. The methods that can be used to classify documents are K-Nearest Neighbors as classifier and Vector Space Model to measure the distance between documents From the initial observation, the multilabel classification in the final abstract using Vector Sapce Model and K-Nearest Neighbors has not been evaluated. Because some previous studies have led to the testing of single labels and only lead to one method, as the method is tested. Classification of abstract document final task consists of 2 stages of making distance table using vector space model and multilabel classification using KNN. This method has not been able to predict the label accurately because the exact exact ratio of its optimum value is only 0.57 when m = 4 and k = 8. This method is good enough in predicting the label even though not precisely. Can be seen from the accuracy value of its optimum which is 0.74 when m = 4 and k = 9. The exact match ratio and accuracy value of this method has the optimum value at m = k / 3.

Keywords : *Multilabel, Classifier,K-Nearest Neighbors, Vector Space Model, Distance*

Abstrak

Tugas akhir merupakan salah satu persyaratan kelulusan mahasiswa. Mahasiswa yang ingin mengerjakan tugas akhir perlu melihat hasil tugas akhir dengan topik yang sama yang telah dikerjakan sebelumnya. Dengan jumlah dokumen tugas akhir yang banyak, tentunya membutuhkan usaha besar untuk menemukan dokumen tugas akhir dengan topik yang sama. Pengelompokan tugas akhir dapat diotomasi menggunakan metode klasifikasi dokumen. Metode yang dapat digunakan untuk mengklasifikasi dokumen adalah K-Nearest Neighbors sebagai classifier dan Vector Space Model untuk mengukur distance antar dokumen. Dari hasil pengamatan awal, klasifikasi multilabel pada abstrak tugas akhir menggunakan Vector Sapce Model dan K-Nearest Neighbors belum pernah dilakukan evaluasi. Karena beberapa penelitian sebelumnya lebih mengarah pada pengujian single label dan hanya menjurus kepada satu metode sebagai metode yang di ujikan. Klasifikasi dokumen abstrak tugas akhir terdiri dari 2 tahap yaitu membuat distance tabel menggunakan vector space model dan klasifikasi multilabel menggunakan KNN. Metode ini belum mampu memprediksi label secara tepat karena nilai exact match ratio optimum-nya hanya 0,57 saat m = 4 dan k = 8. Metode ini cukup baik dalam

memprediksi label walau tidak dengan tepat. Dapat dilihat dari nilai accuracy optimum-nya yang sebesar 0.74 saat m = 4 dan k = 9. Nilai exact match ratio dan accuracy metode ini memiliki nilai optimum pada saat m=k/3.

Kata kunci: *Multilabel, Classifier,K-Nearest Neighbors, Vector Space Model, Distance.*

1. PENDAHULUAN

Tugas Akhir adalah karya ilmiah yang disusun oleh mahasiswa setiap program studi berdasarkan hasil penelitian suatu masalah yang dilakukan secara seksama dengan bimbingan dosen pembimbing. Tugas akhir merupakan salah satu persyaratan kelulusan mahasiswa. Ketentuan-ketentuan mengenai tugas akhir diatur oleh masing-masing fakultas, dengan mengikuti standar universitas. Tugas akhir bagi mahasiswa program diploma III berbentuk paper atau proyek akhir. Untuk program sarjana berbentuk skripsi. Untuk program magister tugas akhirnya berbentuk Tesis dan tugas akhir untuk program doktoral berbentuk disertasi.

Tugas akhir yang telah dilakukan oleh mahasiswa menghasilkan dokumen tugas akhir. Misalnya di Program Studi Magister Ilmu Komputer Undiksha, setiap mahasiswa yang telah menyelesaikan tugas akhirnya diwajibkan membuat laporan tugas akhir yang secara umum terdiri dari abstrak, bab pendahuluan, bab kajian pustaka, bab metode penelitian, bab implementasi dan daftar pustaka.

Selain disimpan dalam bentuk cetak, dokumen tugas akhir juga dapat disimpan dalam bentuk digital pada sebuah sistem penyimpanan tugas akhir. Setiap tahun Program Studi Teknik Elektro meluluskan kurang lebih 60 orang mahasiswa yang artinya, terdapat kurang lebih 60 dokumen baru yang harus dikelola program studi. Semakin banyak jumlah dokumen yang ada maka pengelolaannya akan semakin susah.

Mahasiswa yang ingin mengerjakan tugas akhir perlu melihat hasil tugas akhir dengan topik yang sama yang telah dikerjakan sebelumnya. Dengan jumlah dokumen tugas akhir yang banyak, tentunya membutuhkan usaha besar untuk menemukan dokumen tugas akhir dengan topik yang sama. Untuk mempermudah pencarian, dokumen tugas akhir hendaknya dikelompokkan berdasarkan topik. Pengelompokan dokumen berdasarkan topik membutuhkan usaha yang besar pula sehingga pekerjaan ini harus diotomasi menggunakan sistem.

Pengelompokan tugas akhir dapat diotomasi menggunakan metode klasifikasi dokumen. Beberapa penelitian tentang klasifikasi dokumen tugas akhir pernah dilakukan antara lain: Penelitian tentang klasifikasi dokumen tugas akhir menggunakan algoritma *k-means* [9] Penelitian ini mengklasifikasikan dokumen tugas akhir kedalam sebuah label dengan 8 topik berbeda. Klasifikasi single-label seperti yang dilakukan oleh penelitian tersebut tidak sesuai dengan kenyataan karena dokumen tugas akhir dapat dikategorikan ke lebih dari satu kategori atau dengan kata lain, dokumen tugas akhir memiliki lebih dari satu label.

Penelitian yang kedua adalah penerapan *Learning Vector Quantization* untuk klasifikasi abstrak tesis [9] Penelitan tersebut mengklasifikasi abstrak tesis ke dalam 3 bidang minat dan menghasilkan akurasi yang baik. Abstrak adalah rangkuman yang isinya terkandung keseluruhan dokumen tugas akhir dengan jumlah kata yang tidak lebih dari 1 halaman. Oleh karena itu mengklasifikasikan dokumen tugas akhir berdasarkan abstraknya adalah cara yang baik karena kita memproses jumlah kata yang lebih sedikit namun mewakili makna tugas akhir secara menyeluruh. Namun, sama seperti sebelumnya, penelitian ini juga melakukan klasifikasi *single-label*.

Metode yang dapat digunakan untuk mengklasifikasi dokumen adalah *K-Nearest Neighbors.* Beberapa penelitian tentang klasifikasi dokumen menggunakan metode tersebut antara lain; implementasi algoritma *k-nearest neighbors* yang berdasarkan *one pass clustering* untuk kategorisasi teks [2], klasifikasi *newsgroup* menggunakan *vector space model* dan *novel k-nearest neighbors* [9], dan pemanfaatan *vector space model* pada penerapan algoritma Nazief-Adriani, *KNN* dan fungsi *similarity cosine* untuk pembobotan *IDF* dan *WIDF* pada *prototipe* sistem klasifikasi teks bahasa Indonesia [11]. Penggunaan *KNN* dan *Vector Space Model* untuk klasifikasi teks pada

penelitian-penelitian tersebut menghasilkan performa yang baik.

Dari latar belakang tersebut maka penulis bermaksud melakukan penelitian tentang klasifikasi dokumen secara multilabel pada abstrak tugas akhir yang ada di program Studi Teknik Magister Ilmu Komputer, Universitas Pendidikan Ganesha menggukanan *Vector Space Model* untuk mengukur *distance* antar dokumen dan K-Nearest Neighbors sebagai *classifier.*

2. Metode Penelitian

2.1 Studi Pustaka

Pengumpulan data dan informasi sekunder diperlukan untuk menghimpun informasi yang relevan dengan rancang bangun sistem. Informasi diperoleh dari sumber-sumber tertulis baik tercetak maupun elektronik. Data dan informasi sekunder untuk membantu menganalisis.[7]

2.2 Ekstraksi Term

Ekstraksi Term adalah proses mengolah data yang berupa teks dokumen menjadi *array* dari kata-kata (*term*) atau yang ada dalam dokumen tersebut yang dapat kita sebut dengan *term per document*. Setiap kata yang digunakan di seluruh dokumen juga disimpan tanpa duplikat yang dapat kita sebut dengan *all words*. Proses ekstraksi *term* sendiri ada tiga tahap yaitu, *tokenization*, *stopwords removal* dan *stemming*. Proses tokenization adalah mengubah paragraf dalam dokumen menjadi *array term*, dimana paragraf akan di *split* berdasarkan spasi. Proses *stopwords removal* adalah penghilangan kata yang sama dengan daftar *stopwords* di array *term.* Sedangkan *stemming* adalah proses mencari kata dasar dari setiap kata yang ada pada array *term.*

2.3 Hitung Tf/Idf

Hasil dari proses ekstraksi term adalah daftar term yang ada di setiap dokumen yang kita sebut dengan *term per document* dan semua kata yang ada di seluruh dokumen yang dapat kita sebut dengan *all words.* Proses pertama adalah menghitung nilai *tf* setiap word di setiap dokumen berdasarkan term yang ada di satu dokumen. Proses yang kedua adalah menghitung nilai idf setiap *word* di setiap dokumen berdasarkan *terms per document.* Perkalian antara nilai *tf* dan *idf* menghasilkan nilai *tf/idf* dan disimpan di *tf/idf vector per document*.

2.4 Hitung Cosine Similarity

Tf/idf vector per document dari hasil sebelumnya akan di proses di tahap ini. Masing masing *tf/idf vector* akan di hitung jaraknya satu sama lain menggunakan *cosine similarity*. Nilai jarak yang didapatkan kemudian disimpan di *distance table* dengan kolom sebagai berikut: id dokumen a, id dokumen b, jarak, kelas dan *isTrain*. *Flowchart* proses hitung *cosine similarity* Proses klasifikasi multilabel dalam penelitian ini menggunakan k-NN. Proses klasifikasi didahului dengan mengalokasikan data mana yang akan digunakan sebagai data *tesing* dan *data training*.

2.5 Mencari Kelas Nearst Neighbhor

Mencari *list class* dari *nearest neighbhor* dari data yang akan kita prediksi adalah dengan memilih *k* data dengal label *isTrain* sama dengan *true* yang memiliki *distance* terbesar pada *distance table.* Masing-masing kelas dari *k* data yang dipilih kemudian di *split* dan disimpan di *list class*.

2.6 Prediksi Single Data

List class yang diperoleh dari *nearest neighbhor* suatu akan digunakan untuk memprediksi kelas dari data tersebut. Parmater yang digunakan dalam memprediksi kelas multilabel selain k adalah *minimum class frequency* (*m*) yaitu frekuensi minimum dari kelas yang muncul di k *nearest neighbhor* yang akan digunakan sebagai kelas prediksi*.* Tahapan prediksi *single data* diawali dengan mengambil *list class* dari *k nearest neighbhor* seperti yang dijelaskan sebelumnya. *List class* kemudian dihitung frekuensi kemunculanya dan jika class tersebut frekuensinya lebih besar atau sama dengan *m,* maka kelas tersebut akan golongkan sebagai *prediction class.*

2.7 Prediksi Single Batch

Prediksi *single batch* pada dasarnya adalah melakukan prediksi *single data* secara berulang-ulang sebanyak jumlah data yang ada pada *batch*.

3. Hasil dan Pembahasan

Pengujian suatu algoritma pembelajaran adalah mengukur sejauh mana sistem pembelajaran dapat memprediksi kelas label yang sebenarnya, yang diuji pada data yang tidak

diketahui labelnya. Untuk menangkap aspek kebenaran parsial hasil prediksi system yang dibangun, strategi yang dapat dilakukan adalah menguji perbedaan rata-rata antara label hasil prediksi dan label yang sebenarnya untuk data pengujian. Pendekatan ini disebut dengan pengujian berbasis contoh. Ada 4 metode pengujian yang akan digunakan antara lain, *Exact Match Ratio* (*MR*), *Accuracy* (*A*), *Precision* (*P*), dan *Recall* (*R*) [10]. Penjelasan masing-masing metode pengujian adalah sebagai berikut.

3.1 Exact Macth Ratio (MR)

Seperti yang dijelaskan sebelumnya bahwa pengujian klasifikasi multilabel agak susah karena mengandung aspek kebenaran parsial. Namun ada metode sederhana yang dapat dilakukan yaitu dengan mengabaikan kebenaran parsial dan mengganggap kasusnya adalah prediksi *single label.* Metode ini disebut dengan *Exact Match Ratio* (*MR*) dengan rumus sebagai berikut:

$$MR = \frac{1}{n}\sum_{i=1}^{n} I(Y_i = Z_i)$$

Dimana I adalah fungsi indicator akan bernilai 1 jika kelas prediksi ke-I (Y_i) sama dengan kelas sebenarnya (Z_i), dan bernilai 0 jika sebaliknya.

Metode ini tidak bisa membedakan salah keseluruhan dan benar parsial.

3.2 Accuracy (A)

Dalam *Accuracy* untuk setiap data adalah proporsi dari label yang diprediksi benar dibagi dengan jumlah total label prediksi dan label actual dari data tersebut. *Overall accuracy* adalah rata-rata nilai akurasi untuk setiap data dibagi jumlah data. Rumusnya adalah sebagai berikut;

$$A = \frac{1}{n}\sum_{i=1}^{n} \frac{|Y_i \cap Z_i|}{|Y_i \cup Z_i|}$$

Ada 2 parameter yang akan digunakan dalam menguji metode yang diusulkan yaitu *k* dan *m*. *k* adalah jumlah *nearest neighbhor* dan *m* adalah minimum frekuensi. Kombinasi nilai *k* dan *m* yang akan diuji adalah di rentang *k*=3 sampai *k*=10 dan *m*=1 sampai *m*=10. masing masing kombinsi parameter akan dihitung *Exact Match Ratio* (*MR*), *Accuracy* (*A*),

Tabel 1 Nilai Exact Match Ratio

	k=3	k=4	k=5	k=6	k=7	k=8	k=9	k=10
m=1	0.207143	0.192857	0.142857	0.078571	0.042857	0.014286	0.007143	0.014286
m=2	00.55	0.514286	0.528571	00.45	0.385714	0.307143	0.214286	0.107143
m=3	0.357143	0.492857	0.507143	0.521429	0.564286	0.535714	0.471429	0.378571
m=4	0	0.314286	0.435714	0.485714	0.514286	0.571429	0.557143	0.535714
m=5	0	0	0.278571	0.392857	0.442857	0.478571	0.535714	0.564286
m=6	0	0	0	0.221429	0.342857	0.385714	00.04	0.478571
m=7	0	0	0	0	0.157143	0.292857	00.35	0.371429
m=8	0	0	0	0	0	00.01	0.207143	0.292857
m=9	0	0	0	0	0	0	0.071429	0.164286
m=10	0	0	0	0	0	0	0	00.05

(*P*), dan *Recall* (*R*)-nya menggunakan 10-*fold cross validation.* Selain itu, juga dijelaskan bagaimana implementasi dalam melakukan *k-fold cross validation* Masing masing metode pengujian diimplementasikan dengan sebuah fungsi. Nilai *exact match ratio* dicari menggunakan fungsi getExactMatchRatio().Nilai *accuracy* dicari menggunakan fungsi getAccuracy().Sedangkan fungsi getUnion() dan getIntersection() digunakan untuk mencari irisan dan gabungan dari himpunan label yang ada. Pengujian menggunakan *k-fold cross validation* diawali dengan memberi indeks pada data dan selanjutnya diacak. Indeks untuk *data testing* kemudian dipilih menggunakan fungsi getTestFold() dan *data training* dipilih menggunakan fungsi getTrainFold(). Setiap *data training* dan *data testing* terdiri dari 10 *fold*. Pengujian kemudian dilakukan untuk setiap *fold* nya dan dan performa rata-rata dari masing-masing *fold* akan dihitung.

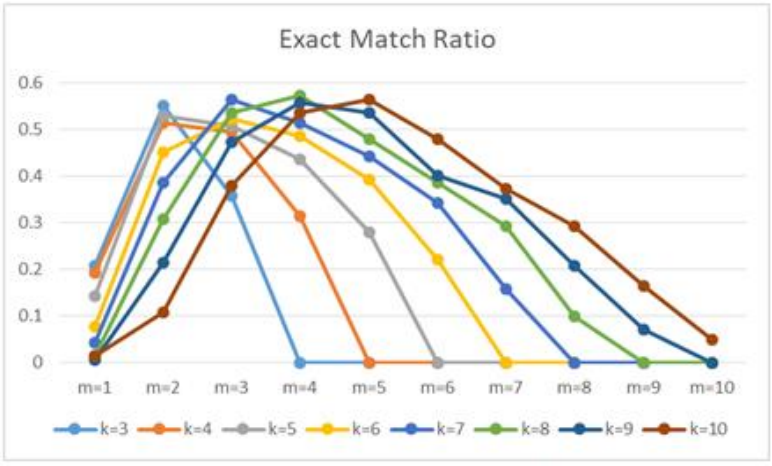
3.3 Pembahasan

Performa metode hasil pengujian yang terdiri dari *exact match ratio* dan *accuracy* yang ditunjulan oleh tabel 1 dan 2 akan ditampilkan dalam betuk *line chart.* Tiap nilai pengujian ditampilkan dalam 2 bentuk yaitu, *chart* terhadap *m* dan chart terhadap *k.* Pada *chart* terhadap *m,*

sumbu *y*-nya adalah nilai performa metode, sumbu *x* adalah nilai *m* dan garis menyatakan nilai *k* yang dibedakan dengan warna. Pada *chart* terhadap *k,* sumbu *y*-nya adalah nilai performa metode, sumbu *x* adalah nilai *k* dan garis menyatakan nilai *m* yang dibedakan dengan warna.

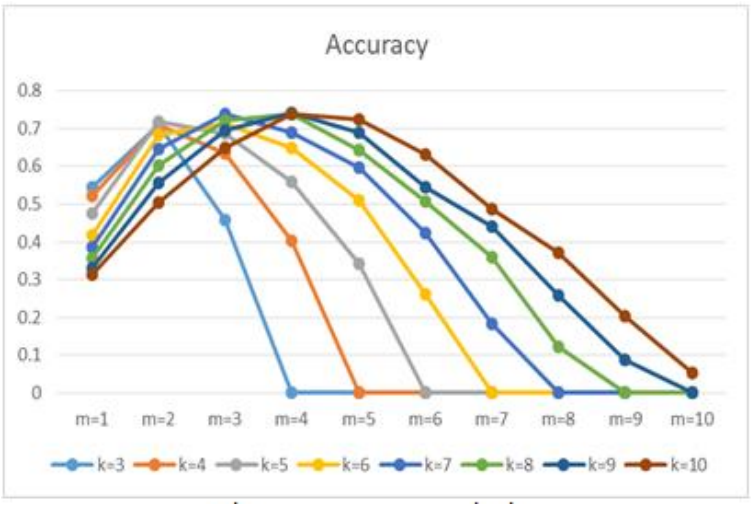
Tabel 2 Nilai Accuracy

	k=3	k=4	k=5	k=6	k=7	k=8	k=9	k=10
m=1	0.544524	0.520952	0.474524	0.41869	0.385969	0.355986	0.329949	0.314405
m=2	0.709524	0.709524	0.719048	0.684524	0.647024	0.602619	0.556429	0.504881
m=3	0.457143	0.634524	0.68631	0.71369	0.739881	0.722024	0.694048	0.649405
m=4	0	0.403571	0.559524	0,045138889	0.689286	0.739286	0.741667	0.738095
m=5	0	0	0.342857	0.509524	0.596429	0.642857	0.689286	0,503472222
m=6	0	0	0	0.261905	0.422619	0.507143	0.545238	0.632143
m=7	0	0	0	0	0.184524	0.360714	0.441667	0.486905



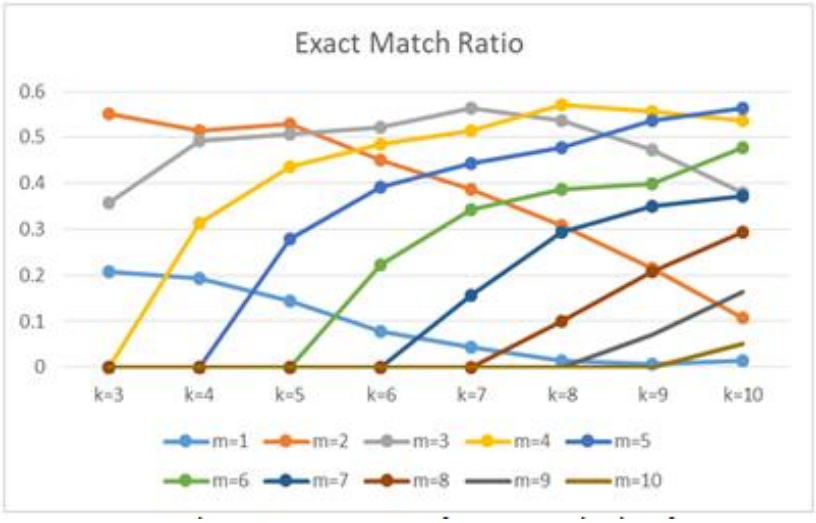
Gambar. 1 Exact Match Ratio Terhadap m

Gambar 1 menunjukan hubungan nilai *exact match ratio* terhadap *m* pada *k* yang berbeda. Dapat dilihat bahwa nilai *exact match ratio* meningkat saat nilai *m* meningkat sampai pada puncaknya yaitu saat nilai *m* sepertiga dari nilai *k* dan dilanjutkan dengan penurunan dan menyentuh nilai 0 saat nilai *m* sama dengan *k+1.*



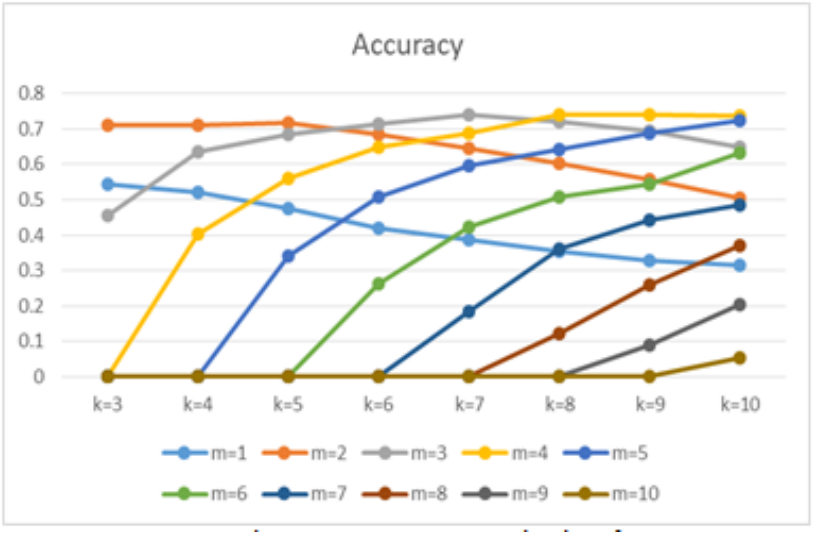
Gambar. 2 Accuracy Terhadap m

Gambar 2 menunjukan hubungan nilai *accuracy* terhadap *m* pada *k* yang berbeda. Gambar2 menunjukan pola yang sama dengan gambar 1, namun nilai yang berupa *accuracy* lebih tinggi dari nilai *exact match ratio*. *Chart accuracy* menunjukan pola yang lebih harus dari *chart exact match ratio*.



Gambar. 3 Exact Match Ratio Terhadap k

Gambar 3 menunjukan hubungan nilai *exact match ratio* terhadap *k* pada *m* yang berbeda. Di nilai *m* yang kurang dari 3, semakin besar nilai *k* menghasilkan nilai *exact match ratio* yang semakin menurun. dengan 3, nilai *exact match ratio*-nya mengalami penurunan dan peningkatan seiring peningkatan nilai *k*.
Sedangkan di nilai *m* yang lebih dari 3, semakin besar nilai *k* menghasilkan nilai exact match ratio yang semakin meningkat. Dan pada saat *m* sama



Gambar. 4 Accuracy Terhadap k

Gambar 4 menunjukan hubungan nilai *accuracy* terhadap *k* pada *m* yang berbeda. Gambar 4 menunjukan pola yang sama dengan gambar 3, namun nilai yang berupa *accuracy* lebih tinggi dari nilai *exact match ratio*. *Chart accuracy* menunjukan pola yang lebih harus dari *chart exact match ratio*

4. Kesimpulan

Dari hasil penelitian, penulis mengambil kesimpulan :

1. Klasifikasi dokumen abstrak tugas akhir terdiri dari 2 tahap yaitu membuat distance tabel menggunakan vector space model dan klasifikasi multilabel menggunakan KNN..
2. Secara umum Metode ini belum mampu memprediksi label secara tepat karena nilai exact match ratio optimum-nya hanya 0,57 saat m = 4 dan k = 8..
3. Metode ini cukup baik dalam memprediksi label walau tidak dengan tepat. Dapat

dilihat dari nilai accuracy optimum-nya yang sebesar 0.74 saat m = 4 dan k = 9.

4. Nilai exact match ratio dan accuracy metode ini memiliki nilai optimum pada saat m=k/3

DAFTAR PUSTAKA

[1] Adriani, M., Asian, J., Nazief, B., Tahaghoghi, S. M., & Williams, H. E. Stemming Indonesian: A confix-stripping approach. *ACM*, 33, 2007.

[2] Arifin, A. D., Arieshanti, I., & Arifin, A. Z. (8, Mei 2017). *Implementasi Algoritma K-Nearest Neighbour Yang Berdasarkan One Pass Clustering Untuk Kategorisasi Teks.* Retrieved from Digital Library Institut Teknologi Sepuluh November : http://digilib.its.ac.id/public/ITS-paper-20008-5108100132-Paper.pdf

[3] Baeza, R., & Neto, R. *Modern Information Retrieval.* Boston,: Addison Wesley-Pearson International Edition, 1999.

[4] Goller. Automatic Document Classification: A Thorough Evaluation of Various Methods. *Proceedings of International Symposium on Information Theory and Its Application*, pp. 145-162, 2000.

[5] Hariri, F. R., Utami, E., & Amborowati, A. Learning Vector Quantization untuk Klasifikasi Abstrak Tesis. *Citec Journal Vol. 2*, 2015.

[6] Manning, C. D., Raghavan, P., & Schütze, H. *Introduction to Information Retrieval,.* Cambridge: Cambridge University Press, 2009.

[7] Minartiningtyas, B., & Sumariata, A. Rancang bangun sistem informasi perhitungan angka kredit dosen stmik stikom Indonesia, 2018.

[8] Naf'an, M. Z. *Segmentasi karakter pada citra manuskrip jawa menggunakan projection analysis dan connected component analysis.* Depok: Universitas Indonesia, 2015.

[9] Nasyuha, W. F., Husaini, H., & Mursyidah, M. Klasifikasi Dokumen Tugas Akhir menggunakan Algoritma K-Means. *Jurnal Infomedia Vol. 1*, 2016.

[10] Suryani, M., Nasuha, A. M., Yulita, I. N., & Paulus, E. Klasifikasi Newsgroup Menggunakan Vector Space Model Dan Novel K Nearest Neighbors. *Jurnal Informatika Universitas Padjadjaran Vol. 1, 2016.*

[11] Susandi, D., & Sholahudin, U. Pemanfaatan Vector Space Model pada Penerapan Algoritma Nazief Adriani, KNN dan Fungsi Similarity Cosine untuk Pembobotan IDF dan WIDF pada Prototipe Sistem Klasifikasi Teks Bahasa Indonesia. *Jurnal ProTekInfo Vol. 3 No. 1*, 2016.

[12] Tala, F. *A Study of Stemming Effects on Information Retrieval in bahasa Indonesia.* Master Thesis, Institut for logic, Language and Computation Universiteit van Amsterdam The Netherlands, 2003.