Jurnal Komunika

Jurnal Komunikasi, Media dan Informatika

ISSN 2579-5899 (Online) Vol. 8 No. 2/November 2019
ISSN 2303-1700 (print) DOI: 10.31504/komunika.v8i2.2016

Rekayasa Web Klasifikasi pada Data Tidak Terstruktur

Classification Web Engineering for Unstructured Data

Laili Wahyunita

IAIN Palangka Raya, Menteng, Kec. Jekan Raya, Palangka Raya, Kalimantan Tengah 73112
laili.wahyunita@iain-palangkaraya.ac.id

Diterima 14 Februari 2019, Revisi 15 November 2019, Diterbitkan 21 November 2019

Abstract

Unstructured data is an interesting research domain to study. Many data sources are available and can be easily taken. Examples of unstructured data include is online news. Data pre-processing process is needed to prepare unstructured data for the classification process. Rocchio classification is a classification algorithm that combines TF-IDF and cosine similarity. Web engineering using Rocchio classification for unstructured data is developed using PHP programming language and MySQL database. The web performance was measured using Black Box Testing method. The testing used five test criteria based on web functionality. The results indicated that the web that has been developed complied with the specifications. To evaluate the performance of Rocchio classification, precision and recall calculation was used. The validation testing method using Recall calculation has resulted the rate of 67,73% whereas Precision calculation has presented the rate of 72,01%, which demonstrated good performance classification.

***Keywords** : classification, rocchio, unstructured data, web engineering.*

Abstrak

Data tidak terstruktur merupakan domain penelitian yang menarik untuk diteliti. Banyaknya sumber data yang dapat diambil dengan mudah menjadi salah satu penyebabnya. Salah satu contoh data tidak terstruktur adalah berita *online*. Tahapan *pre-processing* diperlukan untuk menyiapkan data tidak terstruktur agar dapat diolah pada proses klasifikasi. Klasifikasi Rocchio adalah algoritma klasifikasi yang menggabungkan antara TF-IDF dan *cosine similarity.* Rekayasa web yang menggunakan klasifikasi occhio pada data tidak terstruktur dilakukan dengan menggunakan bahasa pemrograman PHP dan basis data MySQL. Kinerja hasil web diukur dengan menggunakan metode *Black Box Testing.* Pengujian dilakukan dengan menggunakan lima kriteria tes berdasarkan fungsionalitas web. Hasil pengujian menunjukkan bahwa web yang dihasilkan berfungsi sesuai dengan spesifikasi yang telah ditentukan. Adapun pengujian keakuratan klasifikasi Rocchio dilakukan dengan menggunakan perhitungan *precision* dan *recall*. Hasil perhitungan menunjukkan nilai *precision* sebesar 72,01%, dan *recall* sebesar 67,73%,.

Kata kunci : data tidak terstruktur, klasifikasi, rekayasa web, rocchio.

PENDAHULUAN

Data merupakan bagian terpenting dalam proses pengolahan informasi. Berkembangnya dunia internet menjadikan arus pertukaran data berjalan sangat dinamis. Salah satu sumber data dari internet adalah berita *online*. Selain menjadi sumber informasi, berita *online* juga bisa dijadikan bahan penelitian pada domain penggalian data atau *text mining*. Penggalian teks pada artikel berita akan menghasilkan informasi yang bermanfaat.

Data yang didapatkan dari berita *online* termasuk dalam golongan data tidak terstruktur. Data tidak terstruktur memerlukan pengolahan terlebih dahulu sebelum dilakukan proses lebih lanjut (Malarvizhi & Saraswathi, 2013). Banyak metode yang dapat digunakan untuk mengolah data yang ada di dalam halaman web. Salah satunya adalah dengan teknik prapemrosesan atau *pre-processing*. Data yang dihasilkan dari *pre-processing* dapat dianalisis dengan menggunakan berbagai macam metode klasifikasi. Antara lain, Rocchio, Naïve Bayes, *support vector machine,* dan metode lain untuk mendapatkan klasifikasi yang diinginkan dari suatu artikel berita.

Algoritma Rocchio pada awalnya dirancang untuk menyelesaikan persoalan penemuan kembali informasi (*information retrieval*) dan pencarian informasi yang terkait (*relevance feedback*) (Rocchio, J, n.d.). Dalam hal ini, algoritma Rocchio sangat handal dalam memproses data tidak

terstruktur. Pada perkembangan selanjutnya, algoritma Rocchio yang kemudian dinamakan klasifikasi Rocchio juga bisa digunakan untuk melakukan klasifikasi terhadap suatu kumpulan data (Joachims, n.d.). *Query* yang identik merupakan input pada persoalan penemuan kembali pada klasifikasi Rocchio. Hal ini dianggap merupakan isi dari dokumen uji yang dicari kemiripannya dengan dokumen latih. Algoritma Rocchio lazim dipilih dalam proses klasifikasi karena unggul dalam kecepatan proses. Selain itu, klasifikasi Rocchio juga menghasilkan sistem klasifikasi yang baik, yang ditunjukkan dengan capaian akurasi yang cukup tinggi (Widjojo, Rachmat C, & Santosa, 2014).

Penelitian terkait rekayasa web untuk proses klasifikasi banyak dilakukan. Berbagai metode algoritma untuk proses klasifikasi digunakan untuk data tidak terstruktur. Di antaranya adalah klasifikasi dengan algoritma *Naive Bayes* (Junianto & Riana, n.d.) (Darujati & Gumelar, n.d.; Wahyunita, 2017) dan *Ontology Based Classification* (Nidhi & Gupta, 2012).Adapun contoh penelitian tentang klasifikasi Rocchio antara lain adalah klasifikasi Rocchio untuk pengkategorian teks pada renungan harian Kristen (Widjojo *et al*., 2014). Metode Rocchio juga digunakan untuk klasifikasi teks dokumen kesehatan (Albitar, Fournier, & Espinasse, 2012). Sistem pencarian berbasis metode Rocchio juga pernah dikembangkan (Lumbanraja, 2013). Domain teks berbahasa Arab juga telah dikembangkan untuk kategorisasi teks menggunakan Rocchio (Mohammad, Al-Momani, & Alwada'n, 2016). Fokus penelitian ini adalah pembuatan rekayasa web untuk pengolahan data tidak terstruktur dengan klasifikasi Rocchio. Pengujian hasil rekayasa web dilakukan dengan menggunakan metode *Black Box Testing.* Adapun unjuk kerja klasifikasi Rocchio diukur dengan menggunakan perhitungan *precision* dan *recall.*

METODE

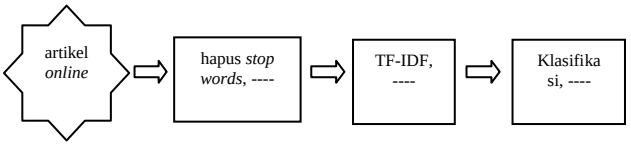
Dalam melakukan penelitian ini diperlukan perencanaan agar penelitian yang dilakukan dapat berjalan dengan baik, sistematis, dan efektif.

a. Data tidak terstruktur (*unstructured data*)

Data adalah serangkaian fakta atau symbol yang menerangkan sesuatu benda maupun kejadian yang terjadi dalam kehidupan sehari-hari (Hu & Liu, 2012). Data dapat dikelompokkan menjadi data terstruktur dan data tidak terstruktur. Data terstruktur (*structured data*) adalah data yang direpresentasikan ke dalam bentuk tabel atau relasi yang biasanya ditempatkan dalam suatu basis data. Adapun data tidak terstruktur (*unstructured data*) adalah data yang tidak dapat dengan mudah diinterpretasikan sebelum diolah terlebih dahulu. Contoh data tidak terstruktur adalah dokumen teks, multimedia, dan sebagainya. Data tidak terstruktur jika diolah sedemikian rupa bisa menjadi data terstruktur sesuai dengan kebutuhan (Aggarwal, 2012).

b. *Document Text Mining*

Document text mining juga merupakan analisis teks yang didefinisikan sebagai teknik pemodelan mesin pembelajaran untuk menemukan pengetahuan yang tersembunyi dari sebuah sumber data tekstual (Aggarwal, 2012).



Gambar 1. Proses Penggalian Data Pada Dokumen Teks

Pengetahuan yang didapatkan dapat digunakan untuk keperluan intelijen bisnis, analisis data, penelitian dan penyelidikan. Penggalian data pada dokumen teks dibutuhkan untuk mengetahui pola-pola yang terkandung di dalamnya, sehingga didapatkan informasi yang dibutuhkan. Ada berbagai topik dalam penelitian penggalian data teks, di antaranya klasifikasi teks atau kategorisasi teks, *clustering* teks, *summarization*, dan sistem rekomendasi (Hu & Liu, 2012). Gambar 1 menunjukkan proses kerja penggalian data pada metode *document text mining*, yang dimulai dari pengolahan artikel dengan menghilangkan kata-kata yang tidak penting, kemudian dilakukan pembobotan kata-kata yang penting. Setelah itu, data teks bisa digunakan untuk proses klasifikasi.

c. *Klasifikasi Rocchio*

Klasifikasi adalah salah satu metode pembelajaran *data mining* kategori *supervised learning*. Artinya, penentuan golongan sudah dilakukan terlebih dahulu. Klasifikasi adalah penentuan sebuah *record* data baru untuk dikelompokkan ke salah satu dari beberapa kategori atau kelas yang telah ditentukan sebelumnya (Manning, Raghavan, & Schuetze, 2009). Proses klasifikasi adalah proses pembelajaran suatu fungsi tujuan atau (*f)* yang memetakan tiap himpunan

atribut atau (*x*) ke sebuah label kelas atau (*y*) yang telah didefinisikan sebelumnya.

d. Web

Dunia internet identik dengan istilah web. Web atau bisa disebut dengan WWW (*World Wide Web*) adalah sekumpulan halaman *hypertext* yang saling terhubung dan berisi instruksi tertentu dalam penggunaan internet. *Hypertext* adalah teks yang memiliki tautan atau *links* (sambungan). Halaman web dapat terdiri dari data teks, grafik, audio, dan video .

Halaman web dapat dibuat menggunakan berbagai macam bahasa pemrograman. Bahasa PERL (*Practical Extraction and Reporting Languange*) merupakan bahasa pemrograman web yang digunakan di awal perkembangan internet dan web. Kemudian, menyusul HTML, PHP, Javascript, Ajax, dan sebagainya.

Pada perkembangan berikutnya, banyak tersedia *framework* dalam membuat web. Hal ini sangat membantu tugas para pembuat web karenamemungkinkan web dibuat dengan lebih cepat. *Framework* yang banyak berkembang di dunia web umumnya berbasis MVC(*Model View Controller)*. MVC memisahkan antara koneksi basis data, tampilan, dan logika.Dengan demikian, penggunaan *framework* memungkinkan perhatian orang lebih terfokus pada logika daripada tampilan web.

e. Rekayasa Software/*Software Engineering*

Metode SDLC (*System Development Life Cycle*) merupakan metode pengembangan *software* yang sering digunakan oleh pengembang program atau *developer*. Secara garis besar, metode SDLC/*waterfall* terbagi menjadi beberapa tahapan antara lain sebagai berikut (Pinontoan, Rachmat, & Delima, 2019):

1. Tahapan analisis dan definisi kebutuhan (*requirements analysis and difinition)*
2. Desain sistem dan *software* (*system and software design)*
3. Implementasi dan pengujian (*implementation and testing)*

Metode SDLC juga biasa disebut metode pengembangan *step down*.

f. *Black Box Testing*

Tahap akhir dari desain penelitian ini adalah menentukan jenis pengujian yang akan diterapkan pada sistem yang dibuat. Sistem pengujian *Black Box* adalah teknik pengujian yang berfokus pada sisi fungsionalitas. Metode *Black Box Testing* menentukan keberhasilan sistem dengan membandingkan antara input yang diberikan kepada sistem dan *output* yang dihasilkan oleh sistem, apakah sudah sesuai dengan perencanaan atau belum.

Skenario pengujian ditentukan terlebih dahulu dengan menggunakan *tester*. *Tester* menentukan kondisi input dan melakukan pengetesan terhadap spesifikasi fungsional program yang sudah dibuat. *Black Box Testing* akan menemukan apakah terdapat kondisi yang tidak sesuai dengan fungsional program atau kesalahan *output* (Pinontoan *et al*., 2019).

g. *Precision dan Recall*

Pengujian keakuratan dan validitas sistem klasifikasi dapat menggunakan perhitungan *precision* dan *recall*. *Precision* menunjukkan persentase hubungan hasil *output* data yang benar dari total keseluruhan data yang diolah. *Recall* adalah pengambilan hasil data yang relevan[5]. Skala pengukuran nilai *recall* dan *precision* biasanya merupakan persentase dengan skala 1%-100 %.

Tabel 1. Referensi Nilai Hasil Klasifikasi

Nilai Prediksi Klasifikasi	**Nilai Sebenarnya**	
	True	**False**
TRUE	TP (*True Positive*) Correct Result	FP (*False Positive*) *Unexpected result*
FALSE	FN (*False Negative*) *Missing result*	TN (*True Negative*) *Correct absence of result*

Tabel 1 memperlihatkan bahwa ada dua kriteria yang dihasilkan dari prediksi kelas oleh algoritma klasifikasi. Kriteria nilai *true* menunjukkan hasil yang diharapkan dalam proses algoritma klasifikasi, baik itu berupa TP (*True Positive*) maupun FP (*False Positif*). Adapun kriteria nilai *false* adalah hasil yang tidak diharapkan dari algoritma klasifikasi.

HASIL DAN PEMBAHASAN

Sumber Data dan Data Kelas

Data yang digunakan dalam penelitian ini bersumber dari Laboratorium Komputer Program Studi Ilmu Komputer pada Universitas Gadjah Mada. Data berupa hasil *crawling* dan *scrapping* artikel berita dari beberapa situs web berita *online.* Data yang dipakai berjumlah 768 artikel. Data

artikel dibagi menjadi data latih dan data uji dengan persentase 80 : 20 pada saat proses klasifikasi.

Proses klasifikasi menggunakan empat kelas yang mengidentifikasi kelas penyebab kasus penyalahgunaan narkoba. Dalam setiap kelas ditentukan fitur-fitur kata yang menjadi acuan dalam penentuan kelas.

Pembuatan Web

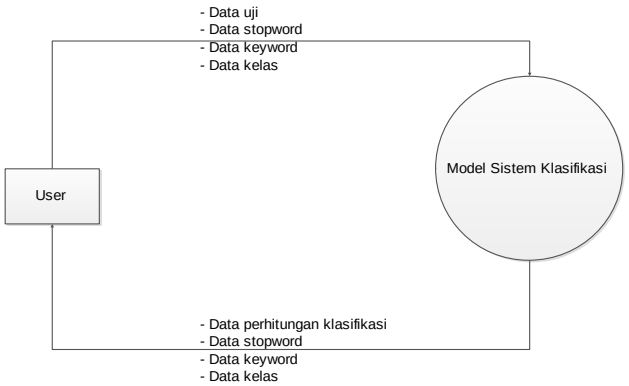
a. Tahapan analisis

Tahapan analisis dilakukan untuk menetapkan spesifikasi pembuatan web dalam penelitian ini. Adapun spesifikasi pembuatan web dalam penelitian ini adalah sebagai berikut:

- Pembuatan web menggunakan bahasa pemrograman PHP.
- Kerangka kerja menggunakan *Framework Fat Free*.
- Basis data menggunakan mySQL.
- Penulisan *script* atau perintah bahasa pemrograman PHP menggunakan Notepad ++ sebagai media.

b. Tahapan desain

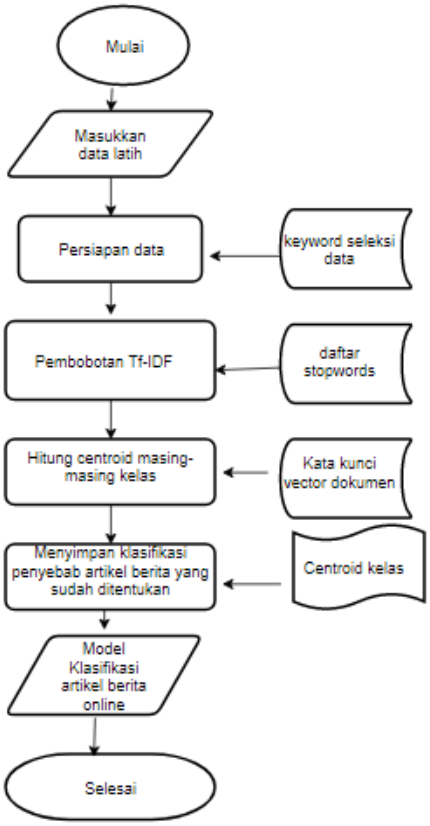
Tahapan desain adalah membuat gambaran sistem secara umum, merancang alur kerja web, dan merumuskan *flowchart* sistem. Adapun gambaran umum sistem dapat dilihat pada **Error! Reference source not found.** yang merupakan diagram konteks. Diagram konteks menunjukkan bahwa u*ser* atau pengguna dapat memasukkan data uji, data *stopword*, data *keyword*, dan data kelas ke dalam sistem. Sistem akan mengeluarkan hasil berupa informasi hasil klasifikasi diikuti dengan kelas hasil proses perhitungan klasifikasi.



Gambar 2. Diagram Konteks

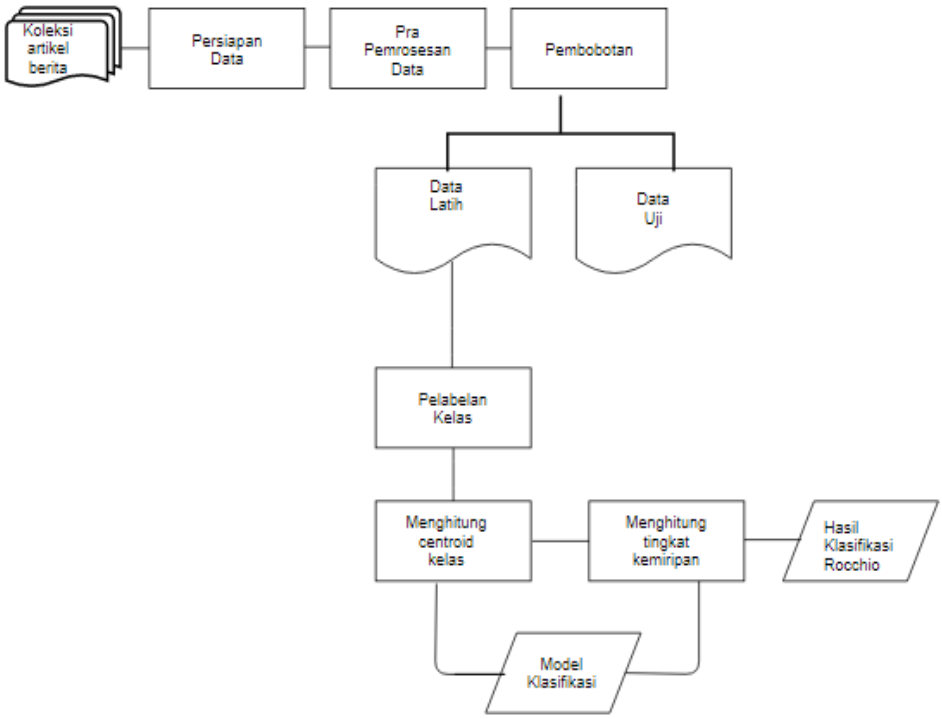
Rancangan alur proses kerja web yang dibuat dalam penelitian ini tampak pada Gambar . Mula-mula akan dilakukan proses pembersihan dan pembobotan terhadap data uji dan data latih terlebih dahulu. Setelah data latih dibersihkan kemudian dilakukan perhitungan TF-IDF, yang dilanjutkan dengan proses pelabelan sesuai dengan klasifikasi kelas data. Hasil dari pelabelan kelas pada data latih ini disimpan dalam basis data.

Proses klasifikasi akan dilakukan saat data uji dimasukkan. Kemudian, dengan prosedur yang sama seperti data latih, data uji melalui proses persiapan data dan pembobotan terlebih dahulu. Selanjutnya, dilakukan perhitungan *centroid* dan *cosine similarity* untuk menentukan kelas klasifikasi data uji yang dimasukkan. Kelas klasifikasi yang dipilih adalah kelas yang memiliki tingkat *cosine similarity* tertinggi.



Gambar 3. *Flowchart* Proses Pengolahan Data

Gambar 4 menunjukkan alur proses web. Proses yang dilakukan pertama kali adalah *data cleaning* atau pembersihan isi data dari bagian yang tidak dibutuhkan untuk proses klasifikasi. Proses ini berupa pembersihan *noise* data, misalnya data alamat web atau *url*, isi gambar dan lainnya. Data yang sudah dibersihkan kemudian masuk ke proses prapemrosesan pengolahan teks yaitu tokenisasi, *stemming*, dan *stopwords*.



Gambar 4. Alur Proses Web

Selanjutnya, dilakukan proses pembobotan dengan metode TF-IDF. Setelah didapatkan *feature* hasil dari proses pembobotan, hasil dari pembobotan disimpan dalam basis data. Berikutnya, dilakukan pelabelan nama kelas sesuai dengan kata kunci atau *keyword* yang telah ditentukan. Setelah itu, dilakukan perhitungan nilai *centroid* masing-masing kelas.

Pengolahan Data Tidak Terstruktur

Pembersihan data dilakukan terlebih dahulu dengan tujuan untuk menghilangkan isi yang dianggap tidak perlu untuk diolah dan mengubah isi dokumen menjadi huruf kecil semua. Contoh isi yang dihapus adalah label-label HTML. Proses pembersihan data juga bertujuan untuk menghilangkan tanda baca, misalnya titik (.) dan menyeragamkan huruf menjadi huruf kecil semua.

Setelah data dibersihkan, proses selanjutnya adalah prapemprosesan data (*pre-processing data*). Proses prapemrosesan yang digunakan dalam penelitian ini antara lain:

a. Tokenisasi
Tokenisasi adalah proses pemenggalan atau pemecahan kalimat menjadi kata tunggal atau menjadi token-token.

b. *Stopwords*
Proses *stopword* dimaksudkan untuk menghapus kata-kata yang tidak bermakna dalam proses klasifikasi.

c. *Stemming*
S*temming* adalah proses menghilangkan imbuhan pada tiap token. Proses *stemming* bertujuan untuk mengubah token menjadi bentuk kata dasar.

Tokenisasi:
wacana gubernur dki jakarta basuki ahok tjahaja purnama yang hendak mewajibkan seluruh minimarket yang menjual minuman keras memasang kamera pengawas cctv dinilai hanya akan memboroskan anggaran

Stopwords:
wacana gubernur dki jakarta basuki ahok tjahaja purnama mewajibkan minimartket menjual minuman keras memasang kamera pengawas cctv dinilai memboroskan anggaran

Stemming:
wacana gubernur dki jakarta basuki ahok tjahaja purnama wajib minimartket jual minum keras masang kamera pengawas cctv nilai boros anggaran

Proses Klasifikasi Rocchio

Klasifikasi Rocchio dimulai dengan memberikan bobot kepada masing-masing term atau token yang muncul pada suatu dokumen. Pembobotan dilakukan dengan menggunakan perhitungan TF-IDF (*Term Frequency-Inverse Document Frequency*). TF-IDF merupakan pembobotan yang umum digunakan pada data tidak terstruktur.

W = TF*IDF (1)

Persamaan (1) merupakan formula untuk mencari nilai *W*. *W* adalah bobot dari suatu term yang merupakan hasil dari perhitungan TF-IDF sebagaimana dirumuskan dalam Persamaan (2) atau Persamaan (3) berikut ini:

TF-IDF = TF x IDF (2)

$= TF * \log\frac{N}{dft}$ (3)

N adalah jumlah dokumen latih dan *dft* adalah jumlah dokumen yang memuat suatu *term*. Perhitungan TF-IDF diperlihatkan pada Tabel 2. *D1*, *D2*, dan *D3* notasi untuk data latih. Adapun DU adalah data uji yang akan ditentukan kelas klasifikasinya oleh sistem.

Tabel 2. Perhitungan Pembobotan

Term	Tf				dft	N/dft
	D1	D2	D3	DU		
wacana	1	0	1	1	1	1,5
gubernur	1	0	1	0	1	1,5
dki	1	0	0	0	1	3
jakarta	1	0	0	0	1	3
basuki	1	0	0	0	1	3
ahok	0	1	0	0	1	3
tjahaja	0	1	0	0	1	3
purnama	0	1	0	0	1	3
wajib	0	1	0	0	1	3
minimarket	0	1	0	0	1	3
jual	0	0	1	0	1	3
minum	0	0	1	0	1	3
keras	0	0	1	0	1	3
pasang	0	0	1	0	1	3
kamera	0	0	0	1	1	3
awas	0	0	0	1	1	3
cctv	0	1	0	0	1	3
nilai	0	1	0	0	1	3
boros	0	1	0	0	1	3
anggar	0	1	0	0	1	3

Selanjutnya, dilakukan perhitungan *centroid* di masing-masing kelas klasifikasi. Persamaan (4) memperlihatkan perhitungan untuk mencari nilai *centroid. Centroid* pada suatu kelas *C* atau $\vec{\mu}(C)$ adalah rata-rata vektor atau pusat massa dari anggota-anggota di kelas *C*. Adapun $\vec{v}(C)$adalah vektor dokumen *(tf)* pada kelas C [9].

$$\vec{\mu}(C) = \frac{1}{|D_c|}\sum d \in D_c \vec{v}(d) \quad (4)$$

Untuk penentuan kelas klasifikasi pada dokumen uji, Rocchio menggunakan formulasi *cosine similarity*. *Cosine similarity* biasa digunakan pada *relevance feedback* dengan menggunakan data input *query* untuk mencari kesamaan pada data latih. Pada klasifikasi *cosine similarity*,term atau token yang disebut vektor pada dokumen uji dibandingkan dengan dokumen latih. Persamaan (5) menunjukkan rumus untuk mencari nilai *cosine similarity.*

$$cosine\,(U,L) = \frac{\sum_{i=1}^{k} f(ui).f(li)}{\sqrt{\sum_{i=1}^{k} f(ui)^2}.\sqrt{\sum_{i=1}^{k} f(li)^2}} \quad (5)$$

Pengujian Web

Pengujian dilakukan dengan menggunakan metode *Black Box*. Metode *Black Box* digunakan untuk menguji unjuk kerja sistem web. Adapun kriteria poin pengujian yang telah ditetapkan berdasarkan spesifikasi sistem yang dibuat dapat dilihat pada Tabel 3.

Tabel 3. Kriteria Penilaian

No.	Poin Pengujian	Kondisi Syarat
1.	Input data uji	1. Input benar 2. Input salah
2.	Melihat hasil klasifikasi/ pembobotan	Pengguna memilih masuk dalam pilihan pembobotan
3.	Menghapus data uji pada daftar	Pengguna memilih menu gambar / Pilihan hapus pada tampilan kanan samping nama *file*
4.	Melihat isi data uji	Pengguna memilih menu gambar folder/pilihan detail data uji pada tampilan kanan samping nama *file*
5.	Menambahkan data pada *stopwords*, *keyword* dan kategori	Pengguna memilih Menu lalu memilih *keyword*, kategori, atau *stopwords*

Hasil pengujian dengan metode *Black Box Testing* dapat dilihat pada Tabel 4. Hasil pengujian dinilai sukses jika *output* yang dihasilkan oleh sistem telah sesuai dengan spesifikasi yang telah ditentukan.

Tabel 4. Hasil Pengujian

No.	Poin Pengujian	Hasil
1.	Input data uji	1.(sukses/~~tidak sukses~~) 2.(sukses/~~tidak sukses~~)
2.	Melihat hasil klasifikasi/pembobotan	(sukses/~~tidak sukses~~)
3.	Menghapus data uji pada daftar	(sukses/~~tidak sukses~~)
4.	Melihat isi data uji	(sukses/~~tidak sukses~~)
5.	Menambahkan data pada *stopwords*, *keyword* dan kategori	(sukses/tidak sukses)

Pengujian Keakuratan Algoritma Klasifikasi Rocchio

Pengujian unjuk kerja (*performance)* algoritma klasifikasi Rocchio pada sistem yang telah dibuat dilakukan dengan menggunakan perhitungan *precision* dan *recall.* Nilai *precision* dan *recall* biasa digunakan untuk mengetahui unjuk kerja hasil klasifikasi algoritma yang dipakai. Sebanyak 50 data uji digunakan untuk mengukur unjuk kerja algoritma klasifikasi Rocchio. Hasil dari pengukuran *recall* dan *precision* tampak pada Tabel 5.

Tabel 5. Hasil *Precision* dan *Recall*

Uji	Nilai
Recall	67,73%
Precision	72,01%

Tabel 5 menunjukkan bahwa nilai *recall* dan *precision* dari proses klasifikasi pada sistem yang dibuat melampaui 50%. Artinya, nilai *precision* yang dihasilkan dari 50 data uji yang dimasukkan mencakup 30 data lebih yang berhasil ditentukan kelasnya dengan benar oleh sistem.

KESIMPULAN

Penelitian ini membuat rekayasa web yang menghasilkan klasifikasi Rocchio pada data tidak terstruktur. Data yang digunakan berupa artikel berita *online*. Proses *pre-processing* berhasil membuat data artikel berita menjadi token, yang kemudian dijadikan term. Selanjutnya, dilakukan proses klasifikasi dengan menggunakan algoritma Rocchio. Dari hasil pengujian terhadap fungsionalitas sistem dengan menggunakan metode *Black Box Testing* tampak bahwa web yang telah direkayasa berfungsi sesuai dengan spesifikasi kebutuhan. Pengujian terhadap performa klasifikasi Rocchio yang dibuat dengan menggunakan perhitungan *recall* dan *precision* menunjukkan nilai yang cukup baik, yaitu di atas 50 %. Dengan demikian dapat disimpulkan bahwa algoritma Rocchio juga dapat diterapkan pada proses klasifikasi dokumen untuk data tidak terstruktur.

Ucapan Terima Kasih

Terima kasih disampaikan kepada Bapak I Gede Mujiyatna, S.Kom., M.Kom, selaku Kepala Laboratorium Komputer yang telah memfasilitasi pengumpulan data yang digunakan dalam penelitian ini.

DAFTAR PUSTAKA

Aggarwal, C.C. (Ed.). (2012). *Mining text data*. New York, NY: Springer.

Albitar, S., Fournier, S., & Espinasse, B. (2012). Conceptualization Effects on MEDLINE Documents Classification Using Rocchio Method. *2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, 462–466. https://doi.org/10.1109/WI-IAT.2012.210

Darujati, C., & Gumelar, A.B. (n.d.). *PEMANFAATAN TEKNIK SUPERVISED UNTUK KLASIFIKASI TEKS BAHASA INDONESIA*. 9.

Hu, X., & Liu, H. (2012). Text Analytics in Social Media. In C. C. Aggarwal & C. Zhai (Eds.), *Mining Text Data* (pp. 385–414). https://doi.org/10.1007/978-1-4614-3223-4_12

Joachims, T. (n.d.). *A Probabilistic AnalysisToexf ttthCeaRteogcocrhizioatAiolngorithm with TFIDF for*. 9.

Junianto, E., & Riana, D. (n.d.). *Penerapan PSO Untuk Seleksi Fitur Pada Klasifikasi Dokumen Berita Menggunakan NBC*. 8.

Lumbanraja, F.R. (2013). *Sistem Pencarian Data Teks dengan Menggunakan Metode Klasifikasi Rocchio(Studi Kasus:Dokumen Teks Skripsi)*. 8.

Malarvizhi, R., & Saraswathi, K. (2013). Web Content Mining Techniques Tools & Algorithms – A Comprehensive Study. *International Journal of Computer Trends and Technology*, *4*(8), 6.

Manning, C., Raghavan, P., & Schuetze, H. (2009). *Introduction to Information Retrieval*. 581.

Mohammad, A.H., Al-Momani, O., & Alwada'n, T. (2016). *Arabic Text Categorization using k-nearest neighbour, Decision Trees (C4.5) and Rocchio Classifier: A Comparative Study*. 6.

Nidhi, & Gupta, V. (2012). Algorithm for Punjabi Text Classification. *International Journal of Computer Applications*, *37*, 6.

Pinontoan, M.S., Rachmat, A., & Delima, R. (2019). Penerapan Metode Waterfall Dan Webqual 4.0 Pada Pengembangan Website Dealer Asa Mandiri Motor. *Jurnal Teknik Informatika dan Sistem Informasi*, *5*(2). https://doi.org/10.28932/jutisi.v5i2.1729

Rocchio, J. (n.d.). *The Smart Retrieval System-Experiments in Automatic Document Processing*.

Wahyunita, L. (2017). Klasifikasi Penyebab Penyalahgunaan Narkoba Dari Berita Online Dengan Menggunakan Naive Bayes. *Jurnal ELTIKOM*, *1*(1), 23–30. https://doi.org/10.31961/eltikom.v1i1.12

Widjojo, E.A., Rachmat C, A., & Santosa, R.G. (2014). Implementasi Rocchio's Classification dalam Mengkategorikan Renungan Harian Kristen. *Jurnal ULTIMATICS*, *6*(1), 1–8. https://doi.org/10.31937/ti.v6i1.325