p - ISSN : 2407 – 1846
e - ISSN : 2460 – 8416
Website : jurnal.umj.ac.id/index.php/semnastek

Pengklasifikasian Berdasarkan Similiaritas pada Abstrak menggunakan *Algoritma Vektor Space* Model

Kristien Margi Suryaningrum[1]*
[1]Teknik Informatika, Universitas Bunda Mulia, Jl. Lodan Raya no. 2 Jakarta Utara 144340
[*]*Corresponding Author* : kristienmargi@gmail.com

Abstrak

Kemiripan dokumen pada abstrak dapat digunakan untuk menjadi petunjuk dan contoh mencari informasi yang sama. Tujuan utama aplikasi ini adalah mencari kesamaan sehingga dapat mengurangi waktu. Untuk menggambarkan tingkat kesamaan antar dokumen melalui abstrak, dapat diukur oleh Metode *Vector Space Model* dan Algoritma *Connected Component*. Berdasarkan tingkat kesamaan dokumen dapat diklasifikasikan dengan menggunakan Algoritma *Connected Component*. Aplikasi ini dibangun dengan menggunakan bahasa pemrograman PHP, web server Apache, dan MySql untuk database server. Objek penelitian ini adalah tentang deteksi kesamaan abstrak dari topik Jurnal Teknik Informatika yang telah dikelompokkan. Hasil penelitian ini akan menghasilkan tingkat kesamaan pada abstrak yang dibandingkan.

Kata kunci: *Vector Space Model, Connected Programming,* Klasifikasi

Abstract

The similarity of documents can be used to guide and sample the search for the same information. The ability to find this similarity can reduce time. To illustrate the level of similarity between documents can be measured by the Vector Space Model Method and the Connected Component Algorithm. Based on the level of similarity of documents can be classified using the Connected Component Algorithm. To detect the level of similarity of the object created an application using the PHP programming language, Apache 2.0 web server, and MySql 5 database server. The object of this research is the Detection of Similarity Abstractions from the Topics of Journal of Informatics Engineering which has been grouped. This application results can provide information about the similarity of abstract documents on research.

Keywords : *Vector Space Model, Connected Programming, Classification*

PENDAHULUAN

Pendeteksian kemiripan dokumen digunakan sebagai alat pencarian informasi lain yang sejenis, yang bertujuan untuk mempersingkat waktu (Strasberg, 2012). Metode *Vector Space Model* merupakan metode yang digunakan untuk menghitung tingkat kesamaan (similarity) antar dua buah dokumen.

Klastering didefinisikan sebagai upaya mengelompokkan data ke dalam klaster sedemikian sehingga data-data di dalam klaster yang sama lebih memiliki kesamaan dibandingkan dengan data-data pada klaster yang berbeda. Metode *Vector Space Model* adalah metode yang menggunakan strategi disain Bottom-Up yang dimulai dengan meletakkan setiap obyek sebagai sebuah klaster tersendiri dan selanjutnya menggabungkan klaster tersebut menjadi klaster yang lebih besar dan lebih besar lagi sampai akhirnya semua obyek menyatu dalam sebuah klaster atau proses dapat berhenti jika telah mencapai batasan kondisi tertentu. Tujuan dari penelitian ini adalah membuat sistem deteksi kemiripan dokumen menggunakan Algoritma *Vector Space Model* dan teknik mengelompokkan dokumen dengan Algoritma Connected Programming.

Ruang Lingkup

Ruang lingkup penelitian ini adalah :

1. Pencarian dilakukan dengan Algoritma *Porter Stemmer* dan *Vector Space Model.*

p - ISSN : 2407 – 1846
e - ISSN : 2460 – 8416
Website : jurnal.umj.ac.id/index.php/semnastek

2. Aplikasi hanya menampilkan dan mengurutkan hasil dari data-data yang telah diproses oleh algoritma.
3. Aplikasi hanya memproses abstrak

Tujuan Penelitian

1. Sistem dapat melakukan *text preprocessing* dalam mengidentifikasi dan mencari abstrak yang memiliki kesamaan antar kata.
2. Sistem dapat merekomendasikan abstrak yang mendekati kemiripan berdasarkan *inputan* dari *user*.
3. Sistem dapat menampilkan hasil dari perhitungan menggunakan *Algoritma Connected Component* dan *Vector Space Model*.

Manfaat Penelitian

Penelitian ini memiliki manfaat yakni Aplikasi dapat membantu mengidentifikasi jurnal TI yang memiliki kesamaan abstrak, seingga dapat memberikan rekomendasi pemilihan jurnal yang telah dikelompokkan berdasarkan kata kunci yang *user* inginkan.

Landasan Teori

Text Mining

Text mining merupakan proses analisis dalam data yang berupa teks dimana sumber data didapatkan dari dokumen (Bening, Dian dan Lailil, 2018). Konsep *text mining* biasanya digunakan dalam klasifikasi dokumen tekstual dimana dokumen-dokumen tersebut akan diklasifikasikan sesuai dengan topik dokumen tersebut. Dengan bantuan *text mining* suatu artikel dapat diketahui jenis kategorinya melalui kata-kata yang terdapat pada artikel tersebut. Kata-kata yang dapat mewakili isi dari artikel tersebut dianalisa dan dicocokkan pada basis data kata kunci yang telah ditentukan sebelumnya. Sehingga dengan adanya *text mining* dapat membantu melakukan pengelompokkan suatu dokumen dengan waktu yang singkat.

Tahapan dalam melakukan analisa pada *text mining* yaitu melakukan pengumpulan data kemudian melakukan ekstraksi terhadap fitur yang akan digunakan (Bening, Dian dan Lailil, 2018) Adapun teknik yang digunakan dalam ekstraksi fitur yaitu melakukan pembersihan data mulai dari *tokenizing*, *stopwords removal* dan *stemming*. Selanjutnya yaitu melakukan *transform* data dengan pembobotan terhadap *term* yang telah dibersihkan. Kemudian dilanjutkan dengan reduksi data. Tahap terakhir yaitu melakukan analisis terhadap proses klasifikasi untuk merepresentasikan hasil informasi yang ditemukan.

Information Retrieval

Information retrieval (IR) system atau sistem temu kembali informasi digunakan untuk menemukan kembali informasi-informasi yang relevan terhadap kebutuhan pengguna dari suatu kumpulan informasi secara otomatis. (Wahyudi, Wurijanto dan Sulistiowati, 2017) Salah satu aplikasi umum dari IR *system* adalah *search engine* atau mesin pencari yang terdapat pada jaringan internet. Pengguna dapat mencari halaman-halaman *web* yang dibutuhkannya melalui *search engine*.

Tujuan dari sistem temu kembali informasi yang ideal adalah:

1. Menemukan seluruh dokumen yang relevan terhadap suatu *query*.
2. Hanya menemukan dokumen relevan saja, artinya tidak terdapat dokumen yang tidak relevan pada dokumen hasil pencarian.

Dua keadaan tersebut digunakan untuk menghitung performansi sistem temu kembali, yaitu *recall* dan *precision*. *Recall* dinyatakan sebagai bagian dari dokumen relevan dalam dokumen yang ditemukan. *Recall* dapat dihitung dengan persamaan:

$$Recall\ (r) = \frac{Jumlah\ Dokumen\ Relevan\ ditemukan}{Jumlah\ Dokumen\ Relevan\ dalam\ koleksi}$$

Rumus 1. Rumus Persamaan *Recall*

Nilai *recall* tertinggi adalah 1, yang berarti seluruh dokumen dalam koleksi berhasil ditemukan. (Novianti dan Diaz, 2017). *Precision* dinyatakan sebagai bagian dokumen relevan yang ditemukan. *Precision* dapat dihitung dengan persamaan:

$$Precision\ (P) = \frac{Jumlah\ Dokumen\ Relevan\ ditemukan}{Jumlah\ Dokumen\ ditemukan}$$

Rumus 2 .Rumus Persamaan *Precision*

p - ISSN : 2407 – 1846
e - ISSN : 2460 – 8416
Website : jurnal.umj.ac.id/index.php/semnastek

Nilai *precision* tertinggi adalah 1, yang berarti seluruh dokumen yang ditemukan adalah relevan. (Novianti dan Diaz, 2017)

Pengukuran *recall* dan *precision* ini merupakan perhitungan yang dilakukan terhadap kumpulan dokumen hasil pencarian (*set based measure*) secara keseluruhan. Pengukuran dengan menggunakan *set based measure* ini tidak dapat menggambarkan performansi sistem temu kembali informasi mengenai urutan dari dokumen relevan. Pengukuran performansi dengan mempertimbangkan aspek keterurutan atau *ranking* dapat dilakukan dengan melakukan interpolasi antara *precision* dan *recall*. Nilai rata-rata *interpolated* precision dapat mencerminkan urutan dari dokumen yang relevan pada perangkingan.

Standar yang biasa digunakan adalah 11 standar tingkat *recall*, yaitu rj Є {0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9,1.0}. Misalkan rj, j Є {0,1,2,.10} adalah tingkat standar *recall* ke-j maka:

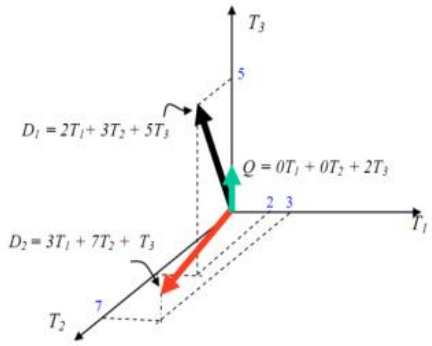
$$P_{(r_j)} = \max r_j \leq r \leq r_{j+1} P_{(r)}$$

Rumus 3. Standar Tingkat *Recall*

Aturan interpolasi adalah recall standard ke-j memiliki nilai *inter polated precision* sebesar maksimum *precision* pada *recall* yang lebih besar dari *recall* standard ke-j.

Metode *Vector Space Model*

Vector Space Model (VSM) merupakan model yang digunakan untuk mengukur kemiripan antara dokumen dan *query* yang mewakili setiap dokumen dalam sebuah koleksi sebagai sebuah titik dalam ruang (Wicaksono, Saptono dan Sihwi, 2015). Dalam metode *Vector Space Model* dihitung *weighted* dari setiap *term* yang terdapat dalam semua dokumen dan *query* dari *user*. *Term* adalah kata atau kumpulan kata yang merupakan ekspresi verbal dari suatu pengertian. Penentuan relevansi dokumen dengan *query* dipandang sebagai pengukuran kesamaan antara vektor dokumen dengan vektor *query*. Contoh representasi relevansi antara dokumen dan query dapat digambarkan pada Gambar 2.2. Q merupakan *query* pembanding, D1 dan D2 adalah dua dokumen yang akan dibandingkan, sedangkan T1, T2 dan T3 adalah tiga term pada dokumen tersebut.



Gambar 2. Representasi dokumen dan *query* pada ruang vector

Prosedur Penelitian

Penelitian ini bersifat terapan, yaitu mencoba untuk menghasilkan sebuah aplikasi/sistem yang dapat membantu user dalam memberikan output jurnal yang memiliki abstract sesuai kemiripannya. Untuk metode penelitian yang dipilih disesuaikan dengan kebutuhan, yaitu metode pengembangan aplikasi/sistem yang mengikuti kaidah-kaidah SDLC atau *System Development Life Cycle*.

Pengembangan Instrumen (Metode Pemecahan Masalah)

Langkah pemecahan masalahnya adalah sebagai berikut :

1) Tokenizing

Proses *tokenizing* dilakukan untuk membuang kata spesial seperti +, -, ! dan lainnya.

2) Stopword

Proses *stopword* dilakukan dengan menghilangkan kata-kata yang tidak digunakan pada proses pembobotan.

3) Stemming

Stemming digunakan untuk mencari kata dasar yang sudah difilter dari proses *stopword*, proses *stemming* ini juga akan menentukan perhitungan bobot dan kesamaan setiap kata.

4) Hitung Similaritas

Similaritas digunakan untuk mencari nilai kesamaan antar abstrak dan kata yang akan dihitung dan diproses.

5) Identifikasi antar Abstrak

Proses identifikasi dilakukan untuk mencari abstrak-abstrak yang memiliki kesamaan satu sama lain dengan menampung hasil sementara dari tabel preferensi.

p - ISSN : 2407 – 1846
e - ISSN : 2460 – 8416
Website : jurnal.umj.ac.id/index.php/semnastek

6) Pencarian

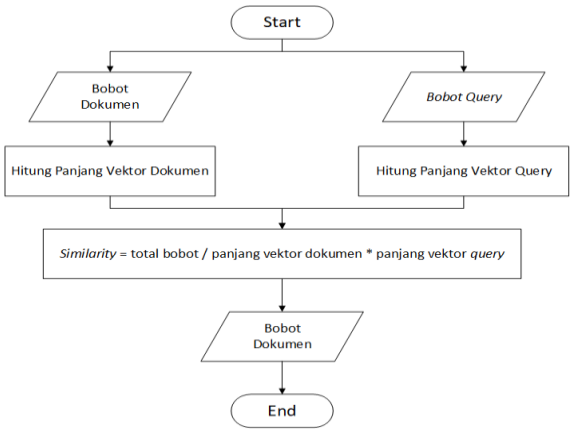
Proses pencarian dapat dilakukan dengan menggunakan data abstrak yang akan diproses berdasarkan *query* yang *diinput*.

Pembobotan similiarity

Bobot *term* tersebut di gunakan dalam proses perhitungan jarak kemiripan dengan kata pada *query* atau dokumen. Setiap kata yang ditemukan akan dibobotkan dan dikalikan dengan panjang vektor *query* dan dibagi ke panjang vektor dokumen pada setiap lirik yang ada. Setelah itu akan didapatkan nilai dari bobot dokumen tersebut dan diurutkan dari nilai tertinggi hingga terendah. Dalam kasus ini nilai 0 akan dihilangkan dari *list* karena dianggap tidak memiliki similaritas sama sekali.

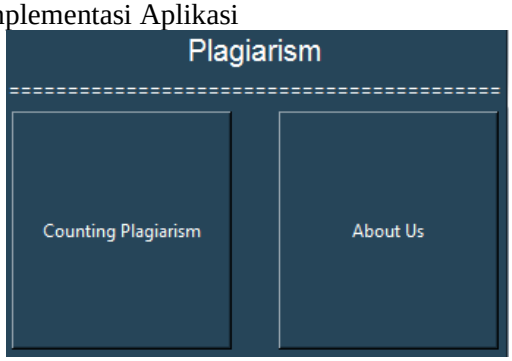
Teknik Analisis Data

Rancangan gambarnya ditunjukkan pada Gambar 4.



Gambar 4. Rancangan Teknik Analisis Data

Implementasi Aplikasi



Gambar 5. Antarmuka Halaman Home

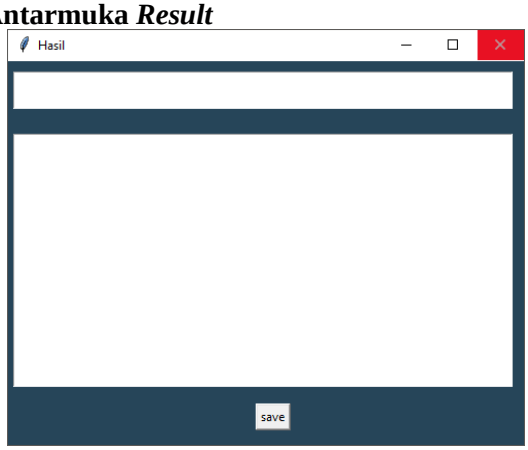
Halaman Menu Utama merupakan halaman utama yang ditamplikan pada saat pengguna menjalankan aplikasi yang dapat dilihat pada Gambar 5.

Antarmuka Halaman *Deteksi*

Gambar 6. Antarmuka Halaman deteksi

Halaman *deteksit* ini merupakan Halaman yang memproseskan data masukan dalam melakukannya perhitungan *Vector Space Model* pada satu *input* dengan membandingkan otomatis dengan data lainnya.

Antarmuka *Result*



Gambar 7. Antarmuka Result

Gambar menunjukkan hasil deteksi dan analisa apakah kedua abstract tersebut mengalami plagiarism atau tidak

Implementasi Algoritma dan Proses

Implementasi Algoritma *SVM*

Algoritma *Stemming* digunakan untuk mengubah kata turunan menjadi kata dasar, contoh *walking* menjadi *walk*, dan *beautiful* menjadi *beauty*. Pada sistem ini, proses *stemming* melakukan proses *text pre-processing*.

p - ISSN : 2407 – 1846
e - ISSN : 2460 – 8416
Website : jurnal.umj.ac.id/index.php/semnastek

Proses *stemming* hanya dilakukan pada *verb* dasar (kata kerja dasar) dan kata lain yang dapat dapat di lakukan prosesnya terdapat pada *file input*. Berikut ini adalah implementasi *stemming algorithm* pada sistem ini :

```
def stem(word):
    """Returns the stemmed version of the argument string.
    """
    return fix_ending(remove_ending(word))



s1 = Text_area1.get("1.0","end-1c")
s2 = Text_area2.get("1.0","end-1c")
a1 = s1.split(" ")
a2 = s2.split(" ")
Text_area1.delete("1.0", END)
Text_area2.delete("1.0", END)


n=0
while n < len(a1):

    b1= stem(a1[n])
    b2= stem(a2[n])
    Text_area1.insert(INSERT, b1)
    '''print (b1)'''
    Text_area1.insert(INSERT, " ")
    Text_area2.insert(INSERT, b2)
    '''print (b2 + ' ')'''
    Text_area2.insert(INSERT, " ")
    n += 1
```

Gambar 8. Proses Stemming

Proses pada Gambar 8 melakukan adanya pengambilan data *input* yang terdapat dari *text*box 1 dan 2. Yang akan di olah atau memisahkan (*command split*) data *input* dengan *command* 'a1 = s1.split(" ")', yang akan dipisisahkan ketika terdapat spasi pada data *input* (" "). *Command* 'Text_*area*1.insert(INSERT, b1)' adalah sebuah *command* yang berfungsi untuk mengisi data *input* yang akan ditempati di *Text area*. *Command stem* adalah sebuah perintah yang membuat panggilan dimana aturan – aturan yang telah disimpan di *dict* (tersimpan) pada sistem.
Aturan – aturan:

```
#kondisi
    def A(base):
        #A No restrictions on stem
        return True
    def B(base):
        #B Minimum stem length = 3
        return len(base) >2
    def C(base):
        # C  Minimum stem length = 4
        return len(base) > 3
    def D(base):
        # D  Minimum stem length = 5
        return len(base) > 4
    def E(base):
        # E  Do not remove ending after e
        return base[-1] != "e"
    def F(base):
        # F  Minimum stem length = 3 and do not remove ending after e
        return len(base) > 2 and base[-1] != "e"
    def G(base):
        # G  Minimum stem length = 3 and remove ending only after f
        return len(base) > 2 and base[-1] == "f"
    def H(base):
        # H  Remove ending only after t or ll
        c1, c2 = base[-2:]
        return c2 == "t" or (c2 == "l" and c1 == "l")
    def I(base):
        # I  Do not remove ending after o or e
        c = base[-1]
        return c != "o" and c != "e"
    def J(base):
        # J  Do not remove ending after a or e
        c = base[-1]
        return c != "a" and c != "e"
    def K(base):
        # K  Minimum stem length = 3 and remove ending only after l, i or u*e
        c = base[-1]
        cc = base[-3]
        return len(base) > 2 and (c == "l" or c == "i" or (c == "e" and cc == "u"))
```

Gambar 9. Kondisi aturan

Pada Gambar 9 Kondisi aturan, adalah sebuah aturan – aturan yang dimana dideteksi pada panjang kata untuk dimasukan ke dalam aturan tersebut.

SIMPULAN DAN SARAN

SIMPULAN

Simpulan dari penelitian ini adalah:

1. Algoritma SVM dan *Connected Component* dapat diterapkan untuk menerapkan plagiarism pada sebuah abstract
2. Algoritma SVM dapat memberikan validitas 89% pada plagiarism sebuah abstrak
3. Aplikasi yang dibuat ini, sangat berpengaruh pada tenses dan kata yang di stemmer.

SARAN

Saran untuk penelitian ini adalah:

1. Menambahkan *dictionary* baru pada kamus kata untuk meningkatkan akurasi presentase *stemming* pada kata – kata.
2. Dalam penggunaan pada perhitungan similarity, supaya nilainya lebih akurat

UCAPAN TERIMAKASIH

Terimakasih kepada Prodi Teknik Informatika dan Manajemen Universitas Bunda Mulia yang tela membiayai penelitian ini sehingga dapat digunakan sebagai mana mestinya.

Surat Penelitian : 263/PE/XI/2018

DAFTAR PUSTAKA

Fathansyah. 2012. Basis Data, Informatika Bandung, Bandung.

Harison dan Ahmad Syarif. 2016. Sistem Informasi Geografis Sarana pada Kabupaten Pasamat Barat. Padang. ISSN: 2338-2724.

Herwijayanti, Bening, Dian Eka Ratnawati dan Lailil Muflikhah. 2018. Klasifikasi Berita Online dengan menggunakan Pembobotan TF-IDF dan *Cosine Similarity*

Hidayat Rahmat. 2014. Analisi Setiomika Makna Motivasi pada Lirik Lagu "Laskar Pelangi" Karya Nidji. 243-258

Hudin, Jamal Maulana dan Achmad Rifai. 2017. Perancangan Sistem Temu Kembali Informasi menggunakan *Vector Space Model* pada Pencarian Dokumen berbasis Teks Berita. Jurnal Sistem Informasi STMIK Antar Bangsa. ISSN: 2098-8711. Vol. VI No.2 Agustus 2017.

Karhendana, A. 2008. Pemanfaatan Document Clustering pada Agregator Berita. Bandung: Program Studi Teknik Informatika ITB.

Indriyono, Bonifacius Vicky, Ema Utama dan Andi Sunyoto. 2015. Pemanfaatan Algoritma *Porter Stemmer* Untuk Bahasa Indonesia Dalam Proses Klasifikasi Jenis Buku. Yogyakarta: STMIK AMIKOM.

Muslih, Muhammad Taufiq dan Bambang Eka Purnama. 2013. Pengembangan Aplikasi SMS *Gateway* Untuk Informasi Pendafataran Pesrta Didik Baru di SMAN 1 Jepara, *Indonesian Jurnal on Networking* and *Security* (IJNS). Vol. 2. No.1.

Napsiyanan, Azka.G. 2013, Perencanaan dan Pengendalian Jadwal Dengan Menggunakan Program *Microsoft Project Professional* 2013 Dalam Pengelolaan Proyek, Jurnal Tugas Akhir UNSIL.

Novianti, K. D. P., & Diaz, R. A. N. 2017. Sistem Pencarian Program Studi Pada Perguruan Tinggi di Bali Berbasis Semantik. JST (Jurnal Sains dan Teknologi), 6(1).

Pressman, Roger S. 2012. Rekayasa Perangkat Lunak – Buku Satu, Pendekatan Praktisi (Edisi 7). Yogyakarta: Andi.

Sibero, Alexander F.K. 2013. *Web programming power pack*. MediaKom, Yogyakarta.

Suprapto. 2008. Bahasa Pemrograman untuk Sekolah Menengah Kejuruan. Departemen Pendidikan Nasional.

Swara, Ganda Yoga, M.Kom dan Yunes Pebriadi. 2016. Rekayasa Perangkat Lunak Pemesanan Tiket Bioskop berbasis Web. Padang: 2338-2724.

Tala, F.Z. 2013. *A Study of Stemming Efects on Information Retrieval in Bahasa Indonesia*. *Thesis. Institute for Logic Language and Computation Universiteit van Amsterdam TheNetherlands.*

Tan, P. N., Steinbach, M., Kumar, V. 2009. *Introduction to Data Mining. London: Pearson Education Inc.*

Turney, P. D., & Pantel, P. 2010. *From Frequency to Meaning: Vector Space. Journal of Artificial Intelligence Research*, 141 -188

Wahyudi, I. P., Wurijanto, T., & Sulistiowati, S. 2017. *Design Aplications To Increase Relevance To Search Thesis (A Case Study In Institute of Business and Information Stikom Surabaya Library*). Jurnal JSIKA, 5.

Wicaksono, Viko Basmalah, Ristu Saptono dan Sari Widya Sihwi. 2015. Analisis Perbandingan Metode *Vector Space Model* dan *Weighted Tree Similarity* dengan Cosine Similarity pada kasus Pencarian Informasi Pedoman Pengobatan Dasar di Puskesmas. JURNAL ITSMART ISSN: 2301–7201

Yuhefizar. 2008. 10 Jam Menguasai Internet: Teknologi dan Aplikasinya. Jakarta