Vol 1 No. 1, November 2020, pp. 83-88 **Jurnal Pepadun**

KLASIFIKASI ABSTRAK JURNAL KOMPUTASI MENGGUNAKAN METODE *TEXT MINING* DAN ALGORITMA *SUPPORT VECTOR MACHINE*

[1]Eliza Fitri, [2]Favorisen Rosyking Lumbanraja, dan [3]Ardiansyah

[1,2,3] Jurusan Ilmu Komputer
Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung
[1]Elizafitr@gmail.com, [2]Favorisen@gmail.com, [3]Ardiansyah@fmipa.unila.ac.id

Abstract *— The University of Lampung especially Computer Science Departement has an online journal that publishes various scientific articles written by researchers both students and lecturers. This scientific article is called the online Computating Journal which is published once every 6 months. But, this online Computating Journal has not been structured and classified into the category of science that more specific. Therefore, in this research the abstract Computating Journal will be classified using text mining techniques to process the abstract become more structured and retrieve information in it. Then, the information in the abstract is extracted as a feature by the TFIDF weighting technique. The proposed classification model uses the support vector machine algorithm that has strong consistency. The model classification will be validated by applying the 10-Fold Cross Validation technique.* ***Keywords:*** *Text Mining, TFIDF, Support Vector Machine, 10-Fold Cross Validation.*

1. PENDAHULUAN

Menurut Kamus Besar Bahasa Indonesia (KBBI) *online* [1], jurnal merupakan majalah yang khusus memuat artikel dalam satu bidang ilmu tertentu. Salah satu bidang ilmu yang dimaksud yaitu bidang komputasi yang khusus ditujukan bagi peneliti di bidang Ilmu Komputer. Universitas Lampung khususnya Jurusan Ilmu Komputer memiliki jurnal *online* yang menerbitkan berbagai artikel ilmiah yang ditulis oleh peneliti baik mahasiswa maupun dosen. Artikel ilmiah ini dinamakan Jurnal Komputasi *online* yang diterbitkan setiap 6 bulan sekali. Jurnal Komputasi *online* belum terstruktur dan terklasifikasi ke dalam kategori keilmuan yang lebih khusus.

Khalid et al [2] melakukan penelitian dalam kategorisasi teks berbasis algoritma SVM dengan metode *feture ranking* IG (*Information Gain*) dan *feature selection* ABC (*Artificial Bee Colony*). Berdasarkan 100 dan 200 fitur yang diuji menggunakan kedua metode tersebut, diperoleh peningkatan *precision* sebesar 15% dan *recall* sebesar 13% dibandingkan metode pembanding PSO (*Particle Swarm Optimization*) SVM.

Penelitian selanjutnya dilakukan oleh Pratama [3] untuk mengimplemetasikan *kernel* Linear SVM untuk klasifikasi dokumen teks berbahasa Indonesia mengenai tanaman holtikultura. Seleksi fitur yang digunakan dalam penelitiannya yaitu chi-kuadrat agar dihasilkan fitur yang lebih sedikit untuk meringankan proses komputasi. Terdapat dua faktor yang digunakan sebagai parameter penilaian hasil klasifikasi, yaitu panjang dokumen dan nilai *epsilon*. Dari kedua faktor tersebut, diperoleh hasil akurasi tertinggi pada implementasi SVM menggunakan kernel linear dengan persentase sebesar 76% pada nilai *epsilon* 0.01

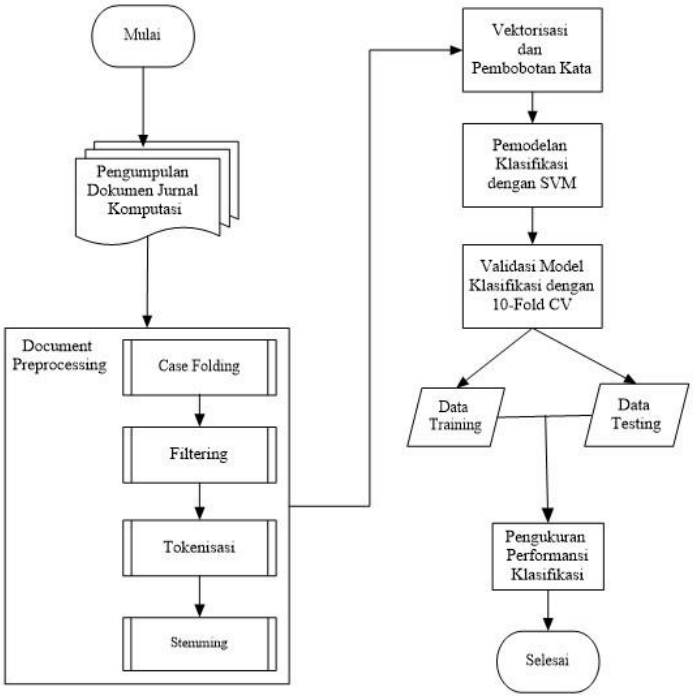
Penelitian klasifikasi teks lainnya dilakukan oleh Somantri dan Apriliani [4] dengan objek penelitian yaitu *sentiment* tingkat kepuasan pelanggan terhadap restoran dan warung makan di Kota Tegal. Klasifikasi yang dilakukan menggunakan algoritma SVM berbasis *feature selection* IG dan *Chi Squared Statistic*. Model SVM dengan *feature selection* IG menghasilkan tingkat akurasi terbaik sebesar 72,45% dan rata-rata kenaikan

tingkat akurasi sebesar 2,514% sedangkan model SVM dengan *Chi Squared Statistic* yang menghasilkan akurasi terbaik sebesar 70.09%.

Dari beberapa penelitian tersebut disimpulkan bahwa algoritma SVM yang ditambahkan dengan metode seleksi fitur IG (*information gain*) dapat menghasilkan fitur yang lebih baik untuk proses klasifikasi. Oleh karena itu, dalam penelitian ini akan dilakukan klasifikasi abstrak Jurnal Komputasi Universitas Lampung berdasarkan subbidang Ilmu Komputer. Metode *text mining* digunakan untuk mengolah abstrak Jurnal Komputasi menjadi lebih terstruktur dan mengambil informasi pada abstrak. Kemudian informasi di dalam abstrak diekstrak sebagai fitur dengan teknik pembobotan TF-IDF (*Term Frequency-Inverse Document Frequency*). Model klasifikasi yang diajukan menggunakan pendekatan algoritma *Support Vector Machine* yang memiliki konsistensi kuat. Model klasifikasi kemudian akan divalidasi dengan menerapkan teknik 10*Fold Cross Validation*.

2. METODOLOGI PENELITIAN

Implementasi *text mining* pada klasifikasi abstrak Jurnal Komputasi menggunakan *tools Natural LanguageToolkit* (NLTK) sedangkan proses pemodelan klasifikasi hingga tahap evaluasi menggunakan *library* Scikit-Learn yang tersedia pada Python. Alur penelitian yang dilakukan dapat dilihat pada Gambar 1.



Gambar 1. Alur Penelitian Implementasi *Text Mining* untuk Klasifikasi Jurnal

Berikut ini merupakan tahapan yang dilalui dalam penelitian ini:

2.1 Pengumpulan Data Abstrak

Data pada penelitian ini berupa abstrak Jurnal Komputasi Jurusan Ilmu Komputer Universitas Lampung berbahasa Inggris dengan format CSV berjumlah 144 dokumen jurnal (volume I-VI) tahun terbit 2013-2018. Abstrak-abstrak tersebut diklasifikasikan menjadi 12 kategori keilmuan berdasarkan matriks Dennings dengan cara *majority voting* yang dilakukan oleh 3 orang narasumber.

Tabel 1. Hasil Voting Klasifikasi Kelas Keilmuan Jurnal Komputasi.

id	Kelas Keilmuan	Jumlah Dokumen
1	Algoritma dan Struktur Data	27
2	Bahasa Pemrograman	0
3	Arsitektur	1
4	Sistem Operasi dan Jaringan	5
5	*Software Engineering*	57
6	*Database* dan Sistem Temu Kembali Informasi	10
7	*Artificial Intelligence* dan Robotik	15
8	Grafik	2
9	*Human Computer Interaction* (HCI)	2
10	Komputasi	1
11	*Organizational Informatics*	23
12	Bioinformatika	1
	TOTAL	144

Berdasarkan hasil klasifikasi abstrak pada Tabel 1 diperoleh fakta bahwa terjadi ketidakseimbangan jumlah anggota antar kelas. Kelas *Software Engineering* memiliki 57 anggota kelas sedangkan kelas Bahasa Pemrograman tidak memiliki anggota kelas sama sekali. Sementara itu kelas Arsitektur, Komputasi, dan Bioinformatika hanya memiliki satu anggota kelas.

2.2 Praproses Data Abstrak

Praproses abstrak terdiri dari beberapa subproses yang diuraikan sebagai berikut. Pertama, tahap *case folding* untuk mengubah semua huruf dalam abstrak Jurnal Komputasi menjadi huruf kecil sehingga menghasilkan bentuk abstrak yang standar dan seragam. Kedua, mengimplementasikan fungsi *filtering* yang terdiri dari tahap *remove number* untuk menghapus angka pada abstrak kemudian diganti menjadi *whitespace*, tahap *remove whitespace* untuk menghapus spasi yang berlebih, tahap *remove punctuation* untuk menghapus tanda baca, dan tahap *remove stopwords* untuk menghilangkan kata yang tidak memiliki makna sehingga menghasilkan *array* kata yang bermakna saja. Ketiga yaitu mengimplementasikan tokenisasi untuk memotong kalimat yang sudah melalui kelima proses sebelumnya menjadi kata/token. Dan tahap terakhir yaitu *stemming* untuk mencari kata dasar dari setiap kata hasil tokenisasi dengan membuang imbuhan di awal (*prefix*) dan diakhir (*sufix*) kata tersebut.

2.3 Transformasi Hasil Praproses Abstrak

Dari tahap praproses diperoleh token-token sebagai fitur dalam bentuk *string*. Selanjutnya, dilakukan transformasi dokumen dari tipe *string* menjadi bentuk vektor dengan menggunakan teknik pembobotan TFIDF. Pada *library* Scikit-Learn diimplementasikan parameter *max_feature* untuk membatasi jumlah kata yang dibobotkan dengan tujuan untuk melihat pengaruh jumlah kata terhadap akurasi *testing*.

2.4 Pemodelan dan Validasi Model Klasifikasi Abstrak

Setelah fitur dari hasil preproses dibobotkan, maka dibuat model prediksi menggunakan 3 *kernel Support Vector Machine* (SVM) yaitu Linear, Polynomial, dan RBF. Kemudian, model prediksi divalidasi menggunakan teknik 10-*Fold Cross Validation* untuk mendapatkan model prediksi terbaik. Sebanyak 144 data abstrak dibagi kedalam 10 *fold* berukuran sama dimana 9 *fold* (129 abstrak) digunakan untuk pelatihan dan 1 *fold* (15 abstrak) digunakan untuk pengujian. Ilustrasi penerapan teknik 10-*Fold* CV dalam penelitian ini dapat dilihat pada Gambar 2.

Gambar 2. Ilustrasi pembagian data training dan testing pada 10-fold CV

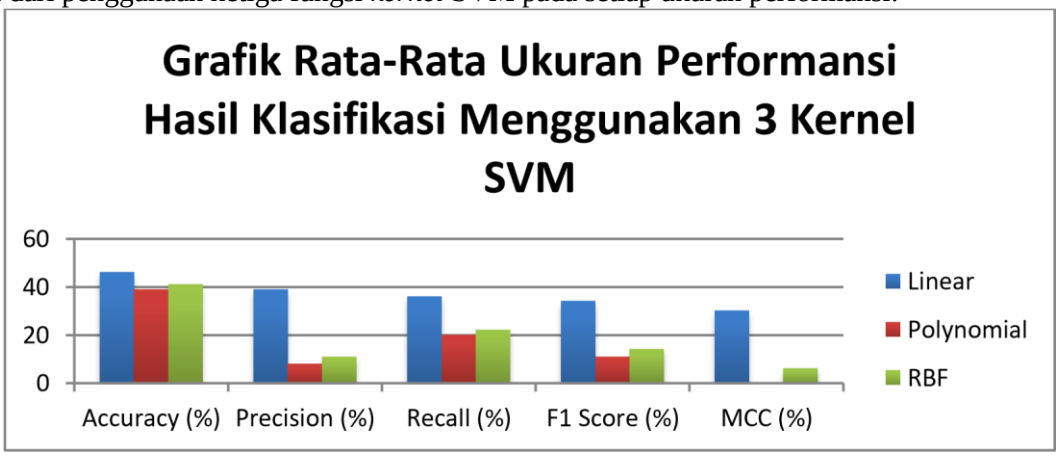
Dengan menggunakan parameter *Shuffle=False* pada Python, dihasilkan pasangan data *training* dan data *testing* sesuai urutan indeks abstrak jurnal menggunakan pengindeksan Numpy. *Fold* yang dihasilkan tidak terpengaruh oleh kelas atau grup, sehingga setiap fold tidak mewakili semua kelas keilmuan. Pada Gambar 2 ditampilkan implementasi 10-*Fold Cross Validation* dalam pembagian data *training* dan *testing*.

2.5 Pengukuran Performansi Klasifikasi

Tahap terakhir yaitu mengukur performansi hasil klasifikasi berdasarkan 5 ukuran performansi yaitu *accuracy*, *precision*, *recall*, *f-measure*, dan MCC. Kemudian akurasi *testing* diperoleh dengan mencari rata-rata nilai *testing* dari 10-*fold cross validation*.

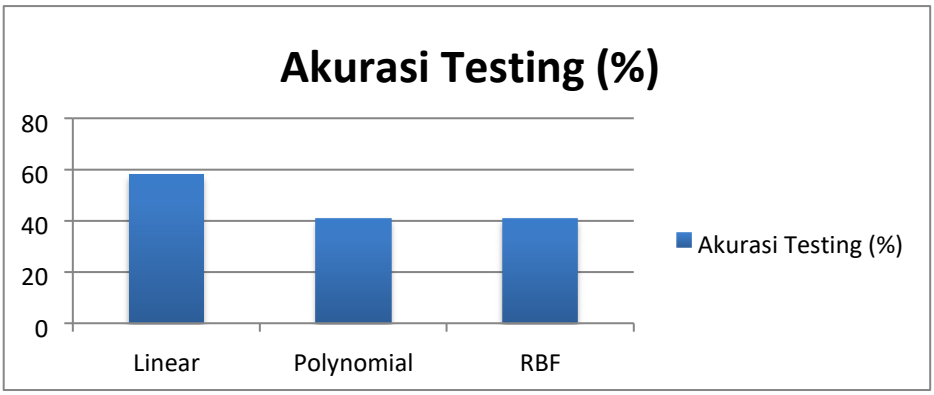
3. HASIL DAN PEMBAHASAN

Akurasi *testing* diperoleh dari nilai rata-rata akurasi *testing* selama 10 kali iterasi pada tahap validasi menggunakan 10-*fold cross validation*. Nilai untuk masing-masing ukuran performansi juga diperoleh dari masing-masing percobaan pada setiap *fold* sehingga diperoleh 10 nilai yang kemudian dicari rata-ratanya. Nilai rata-rata ini kemudian menjadi nilai akhir yang digunakan untuk menarik kesimpulan sebagai hasil penilaian dari proses klasifikasi yang dilakukan. Gambar 3 berikut merepresentasikan perbandingan rata-rata nilai yang diperoleh dari penggunaan ketiga fungsi *kernel* SVM pada setiap ukuran performansi.



Gambar 3. Grafik Nilai Rata-Rata Ukuran Performansi Hasil Klasifikasi

Berdasarkan grafik batang pada Gambar 3 diperoleh fakta bahwa nilai tertinggi pada masing-masing ukuran performansi dicapai pada penggunaan fungsi *kernel* Linear dengan rincian nilai *accuracy* sebesar 58%, *precision* sebesar 39%, *recall* sebesar 36%, F1 *Score* sebesar 34%, dan MCC sebesar 30%. Sedangkan nilai terendah diperoleh pada penggunaan fungsi *kernel* Polynomial. Fungsi kernel Linear menghasilkan nilai tertinggi dibandingkan dengan kedua fungsi kernel lainnya menandakan bahwa data abstrak dapat diklasifikasikan secara linear meskipun menghasilkan nilai performansi yang kecil. Pada Gambar 4 ditampilkan grafik batang persentase nilai akurasi testing dari penggunaan *kernel* Linear, Polynomial, dan RBF.

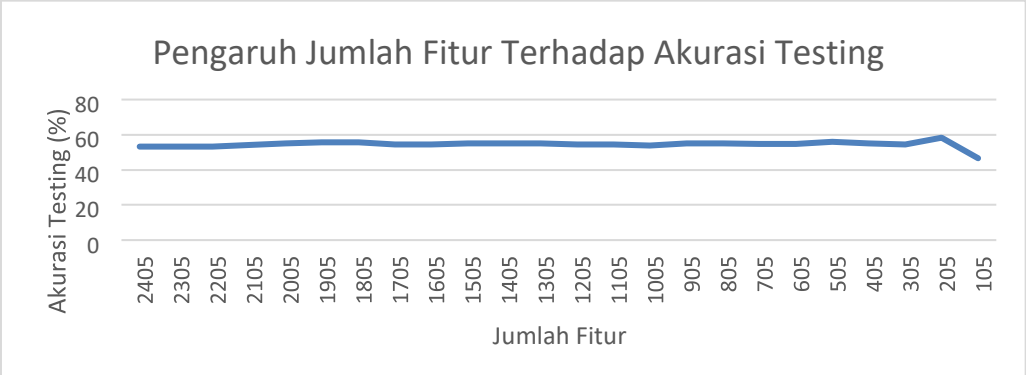


Gambar 4. Grafik Persentase Akurasi *Testing* Hasil Klasifikasi.

Berdasarkan Gambar 4, nilai akurasi tertinggi untuk 5 parameter performansi klasifikasi dicapai pada penggunaan *kernel* Linear. Sedangkan nilai akurasi pada penggunaan *kernel* Polynomial dan RBF menghasilkan nilai yang sama persis. Dari hasil analisis yang dilakukan selama penelitian, terdapat beberapa faktor yang mempengaruhi hasil akurasi model klasifikasi.

1. Jumlah anggota kelas pada *dataset* yang digunakan
   Dasar pengelompokan matriks Dennings yang diterapkan pada *dataset* dalam penelitian ini menghasilkan jumlah anggota antar kelas yang tidak seimbang. Hal ini menyebabkan *classifier* tidak bisa mengenali pola klasifikasi sehingga kebanyakan fitur yang dihasilkan dari proses *text mining* dikenali sebagai ciri dari kelas keilmuan yang jumlah anggotanya paling besar. Akibatnya terjadi kesalahan prediksi pada kelas yang anggotanya sedikit.
2. Jumlah fitur hasil *text mining*

Pada penelitian ini dilakukan percobaan pengurangan jumlah fitur dengan menerapkan interval sebesar 100 fitur. Hasil percobaan pengurangan jumlah fitur yang dilakukan menunjukkan adanya pengaruh jumlah fitur yang dihasilkan dari proses *feature extraction* terhadap nilai akurasi *testing*. Hasil percobaan ini dapat dilihat dalam bentuk grafik pada Gambar 5.



Gambar 5. Grafik Pengaruh Jumlah Fitur terhadap Akurasi Testing

Berdasarkan Gambar 5, akurasi *testing* tertinggi diperoleh pada penggunaan 205 fitur dengan persentase nilai akurasi sebesar 58,3%. Grafik pada Gambar 5 menunjukkan bahwa semakin sedikit atau semakin banyak jumlah fitur yang dihasilkan belum tentu meningkatkan nilai akurasi *testing*.

4. KESIMPULAN

Kesimpulan yang diperoleh berdasarkan penelitian yang telah dilakukan yaitu akurasi *testing* tertinggi dari klasifikasi abstrak Jurnal Komputasi diperoleh pada penggunaan kernel Linear dengan nilai akurasi sebesar 58,3%. Terdapat dua faktor yang mempengaruhi akurasi klasifikasi. Faktor pertama yaitu jumlah anggota antar kelas keilmuan yang tidak seimbang sehingga menyebabkan *classifier* tidak mengalami proses pembelajaran terhadap kelas yang anggotanya sedikit. Faktor kedua yaitu jumlah fitur yang dihasilkan dari tahap praproses abstrak. Akurasi tertinggi diperoleh pada penggunaan 205 fitur, namun sedikit jumlah fitur hasil proses *feature extraction* belum tentu meningkatkan akurasi klasifikasi.

DAFTAR PUSTAKA

[1] Kementrian Pendidikan dan Kebudayaan (Kemdikbud). 2018. Arti kata jurnal - Kamus Besar Bahasa Indonesia (KBBI) Online, https://kbbi.web.id/jurnal, Diakses pada 6 Desember 2018.

[2] Khalid, Rintyarna, B. S., dan Arifin, A. Z. 2015. "Seleksi fitur dua tahap menggunakan information gain dan artificial bee colony untuk kategorisasi teks berbasis support vector machine". *SYSTEMIC*, vol. 1, no. 2, ISSN 2460-8092, pp. 22-26.

[3] Pratama, D. H. 2013. "Implementasi support vector machine (svm) untuk klasifikasi dokumen". *Institut Pertanian Bogor*, pp. 1-19.

[4] Somantri, O. dan Apriliani, D. 2018. "Support vector machine berbasis feature selection untuk sentiment analysis kepuasan pelanggan terhadap pelayanan warung dan restoran kuliner kota tegal", *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 5, no. 5, pp. 537–548.