SUPPORT VECTOR MACHINE UNTUK EKSTRAKSI DOKUMEN KARYA ILMIAH

Jihad[1], Nelly Indriani Widiastuti [2], Kania Evita Dewi[3]

[1,2,3] Universitas Komputer Indonesia
Jl. Dipati Ukur No.112-116, Lebakgede, Coblong, Kota Bandung, Jawa Barat 40132
E-mail : balgohum@gmail.com[1] , nelly.indriani@unikom.ac.id[2],
kania.evita.dewi@email.unikom.ac.id[3]

ABSTRAK

Ekstraksi informasi dokumen karya tulis ilmiah proses ekstraksi karya ilmiah secara otomatis untuk mendapatkan informasi terstruktur. Penelitian bertujuan untuk mengukur performansi *Support Vector Machine* (SVM) dalam mengektrak informasi didalam karya ilmiah. Dokumen karya ilmiah yang digunakan dalam penelitian ini adalah berupa lembar sampul dan abstrak dari skripsi, yang tersimpan dalam bentuk format .pdf. Setiap dokumen diubah terlebih dahulu ke format text. Hasil mengubah format dokumen masuk ke dalam tahapan persiapan, yaitu filtering, segmentasi, tokenizing, pelabelan, ektraksi fitur, dan seleksi fitur. Fitur yang digunakan dalam penelitian ini ada sebanyak 14 fitur. Hasil dari seleksi fitur setiap dokumen masuk kedalam proses klasifikasi untuk menentukan mengklasifikasi 16 kelas dari setiap dokumen tersebut. Di dalam penelitian ini performansi dari model yang dibuat oleh SVM mengunakan akurasi. Berdasarkan pengujian yang telah dilakukan dengan nilai *gamma*=0.5, dihasilkan akurasi tanpa *Information Gain* sebesar 90.68% sementara akurasi dengan *Information Gain* sebesar 90.99%. Untuk nilai *error rate* sebesar 9.32%, nilai *precision* sebesar 93.79%, nilai *recall* sebesar 90.74% dan nilai f-1 *score* sebesar 89.21%. Kesalahan yang paling banyak terjadi, pada pengklasifikasian judul lembar sampul. Kesalahan ini terjadi dikarenakan didalam penelitian ini dokumen yang digunakan, lembar sampul dan abstrak karya ilmiah, berasal dari sebuah dokumen yang sama, sehingga judul lembar sampul dan abstrak berisi hal yang sama, sehingga judul lembar sampul sering terklasifikasi judul abstrak.

Kata kunci : Dokumen Karya Tulis Ilmiah, *Support Vector Machine* (SVM), Ekstraksi Fitur, *Information Gain* (IG).

1. PENDAHULUAN

Karya ilmiah adalah karya tulis yang akan dibuat oleh mahasiswa jika ingin lulus. Karya ilmiah terdahulu sering dibutuhkan oleh peneliti baru, sebagai referensi. Tetapi penyusunan hasil karya ilmiah yang terdahulu tidak selalu serapi jaman sekarang. Membutuhkan waktu yang lama untuk mendata satu per satu karya ilmiah terdahulu dan terkadang menjadi sulit dikarenakan format yang berubah-ubah. Salah satu solusi untuk menjawab untuk masalah ini adalah ektraksi informasi. Ekstraksi Informasi adalah proses ekstraksi secara otomatis untuk memperoleh informasi terstruktur seperti entitas, hubungan antara entitas, dan atribut yang menggambarkan entitas dari sumber yang tidak terstruktur [1]. Spesifikasi dasar dari tugas ekstraksi mencakup hanya jenis struktur yang akan diekstraksi [1].

Penelitian tentang ekstraksi informasi terhadap karya ilmiah sudah banyak dilakukan sebelumnya. Berdasarkan penelitian Dimas dan Nelly [2] tentang ektraksi informasi pada karya ilmiah dengan menggunakan metode berbasis aturan diperoleh akurasi yang baik. Hal ini dikarenakan format lembar sampul dan abstrak yang digunakan untuk menganalisis kata kunci yang dibutuhkan untuk membuat rule base dan pengujian menggunakan format yang sama. Dalam penelitian tersebut, belum teruji jika data uji yang dimasukan berbeda dengan format data yang dianalisis. Berdasarkan penelitian Firdamdam dan Ken [3] dengan menggunakan metode LVQ dan bantuan perbaikan kelas berbasis aturan. Pengujian dilakukan pada dokumen yang memiliki format tahun 2017 dan 2013 diperoleh akurasi diperoleh rata-rata akurasi sebesar 57%. Salah satu masalah dalam penelitian ini fitur Organization, yang sering membuat entitas fakultas dan universitas salah klasifikasi.

Penelitian lain tentang ekstraksi informasi adalah penelitian Aditya, dkk [4]. Dalam penelitian ini dilakukan ektraksi informasi terhadap makalah ilmiah. Salah satu yang menghasilkan akurasi terbaik adalah menggunakan fitur local dan tata letak dengan klasifikasi SVM diperoleh akurasi sebesar 100%. Pengunaan metode SVM juga sudah Begitupun pada bidang lainnya seperti penelitian terhadap analisis sentiment review film [5], yang melakukan perbandingan terhadap metode ANN, SVM dan NB dengan tanpa seleksi fitur diperoleh hasil akurasi ANN sebesar 51,80%, SVM sebesar 81,10% dan NB sebesar 74%. Kemudian penelitian terhadap berita *Online* Menggunakan Metode *Support Vector*

Machine dan *K- Nearest Neighbor* [6], menyatakan bahwa metode yang lebih unggul adalah metode *Support Vector Machine* dengan perolehan akurasi sebesar 93,2%. Oleh karena itu, Berdasarkan beberapa penelitian sebelumnya maka diperoleh asumsi bahwa SVM dapat bekerja dengan baik pada proses ekstraksi informasi.

Fitur adalah salah satu hal yang mendukung kinerja *machine learning*. Dapat dilihat pada penelitian yang dilakukan oleh Firdamdam Sasmita [3], beberapa kelas salah terprediksi dikarenakan penggunaan fitur yang membuat ambigu. Penggunaan fitur organization membuat kelas fakultas dan universitas salah terklasifikasi. Pada penelitian Aditya, dkk [4], diperlihatkan untuk mendeteksi kelas yang dimiliki, dicoba kombinasi antara fitur lokal, fitur tata letak dan fitur named entity. Dapat terlihat pada penelitian tersebut kombinasi fitur dan machine learning yang menghasilkan akurasi terbaik untuk suatu kelas. Oleh karena itu, perlu dilakukannya analisis terhadap pengaruh pemilihan fitur yang digunakan dan meminimalisir fitur yang kurang berpengaruh. Dalam penelitian ini untuk melihat pengaruh fitur terhadap kelas klasifikasi menggunakan *information* gain. Nilai gain akan diurutkan dari yang terbesar hingga terkecil, semakin besar nilai gain suatu fitur terhadap suatu kelas, menunjukkan semakin berpengaruh terhadap suatu kelas. Asumsi *Information* gain dapat mengoptimalkan kinerja SVM berdasarkan penelitian yang dilakukan oleh Kiki Prima Wijaya dan Much Aziz Muslim [7]. Dalam penelitian ini dapat dilihat bahwa *Information gain* berhasil menunjukan peningkatan sebesar 0,75% pada akurasi dari 97,75% menjadi 98,5% dalam proses diagnosa penyakit ginjal kronis.

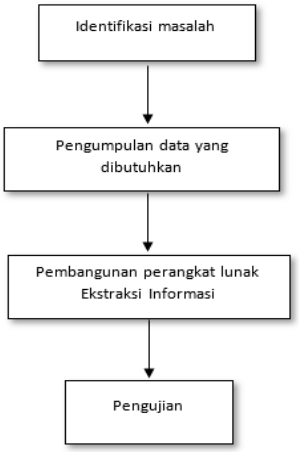
Berdasarkan uraian diatas, maka pada penelitian ini performansi kinerja kombinasi SVM dan *Information Gain* diperlihatkan dalam bentuk akurasi dalam mengekstrak informasi karya ilmiah. Dalam penelitian ini dokumen karya ilmiah yang akan digunakan adalah skripsi mahasiswa. Tidak semua karya ilmiah akan digunakan sebagai data, hanya bagian lembar sampul dan abstrak saja.

2. ISI PENELITIAN

Isi penelitian menjelaskan mengenai metode penelitian, dokumen karya tulis ilmiah skripsi, arsitektur sistem, filtering, segmentasi, tokenisasi, ekstraksi fitur, algoritma SVM, information gain, dan hasil pengujian.

2.1 Metode Penelitian

Pada penelitian ini ada empat tahapan alur kerja penelitian, yaitu identifikasi masalah, pengumpulan data yang dibutuhkan, pembangunan perangkat lunak ekstraksi informasi, serta pengujian. Pada tahapan identifikasi masalah dilakukan analisis kebutuhan penelitian. Tahap berikutnya adalah pengumpulan data, didalam pengumpulan data dilakukan studi literatur yang berkaitan dengan penelitian dan pengumpulan lembar sampul dan abstrak dari skripsi. Tahapan pembangunan perangkat lunak menggunakan model waterfall. Tahapan terakhir adalah tahapan pengujian. Pada tahapan pengujian dilakukan pengukuran perfomansi kinerja SVM-*Information Gain* dalam mengekstrak informasi dari karya ilmiah. Berikut blok diagram alur kerja penelitian, dapat dilihat pada gambar 1.



Gambar 1. Blok Diagram Alur Kerja Penelitian

2.2 Dokumen Karya Tulis Ilmiah Skripsi

Dokumen yang digunakan dalam penelitian ini menggunakan dokumen karya tulis ilmiah skripsi Program Studi Teknik Informatika, Fakultas Teknik dan Informatika, Universitas Komputer Indonesia. Tidak semua bagian dokumen skripsi akan digunakan sebagai dataset, tetapi hanya Sebagian saja, yaitu bagian lembar sampul dan abstrak. Dalam sebuah dataset terdapat 80 dokumen. Dokumen yang berjumlah 80 dibagi menjadi 2, yang pertama terdiri dari 40 dokumen dari tahun 2012 hingga 2018 untuk menjadi data training. Sisa data sebanyak 40 dokumen dijadikan data testing, 40 data ini pun memiliki dokumen dari tahun 2012 dan 2018, yang berupa 20 lembar sampul dan 20 abstrak. Alasan pemilihan tahun 2012 hingga 2018, karena terdapat perbedaan format penulisan pada bagian lembar sampul, sehingga bisa melihat konsistensi model yang dibangun SVM dengan format yang berbeda-beda. Setiap dokumen yang terdiri lembar sampul dan abstrak akan diberi beberapa kelas/kategori. Berikut kategori pada lembar sampul dan abstrak dapat dilihat pada Tabel 1.

Tabel 1. Kategori Lembar Sampul Dan Abstrak

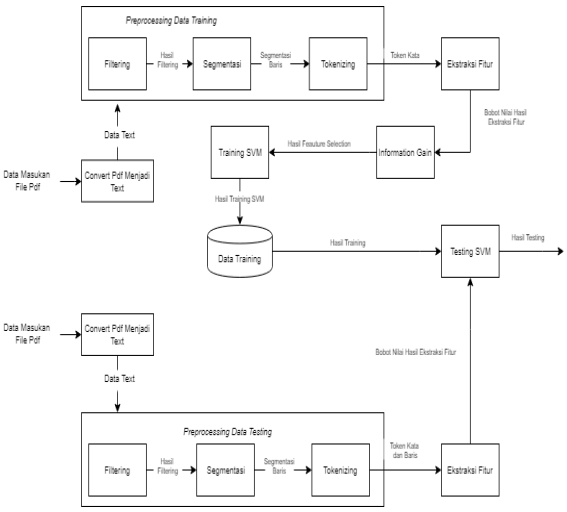
Lembar			
No	**Sampul**	**No**	**Abstrak**
1	Judul Penelitian (Sampul)	**10**	Judul Halaman Abstrak

2	Jenis Penelitian	**11**	Judul Penelitian (Abstrak)
3	Kalimat Pengajuan	**12**	Other
4	Penulis (Sampul)	**13**	Penulis (Abstrak)
5	NIM (Sampul)	**14**	NIM (Abstrak)
6	Program Studi	**15**	Isi Abstrak
7	Fakultas	**16**	Kata Kunci
8	Universitas		
9	Tahun		

Terdapat 16 kategori dalam setiap doken seperti dapat dilihat pada table 1. Sehingga dalam pembuatan model SVM digunakan SVM *multiclass*.

2.3 Arsitektur Sistem

Sistem ekstraksi informasi yang dibangun terdiri dari beberapa proses. Pertama-tama, setiap dokumen yang berformat .pdf diubah kedalam format text. Kemudian baik dokumen untuk training maupun testing masuk kedalam proses prepocessing. Pada penelitian prepocessing terdiri dari proses filtering, segementasi, dan tokenizing. Kemudian diektrak menggunakan 14 fitur. Hasil ekstraksi fitur masuk kedalam proses seleksi fitur yang dalam penelitian ini menggunakan *Information Gain*, proses ini hanya terjadi untuk dokumen training. Hasil proses Information Gain akan digunakan untuk membentuk model dengan menggunakan SVM. Sedangkan dokumen testing untuk fitur hanya mengikuti hasil seleksi fitur pada dokumen training. Model yang terbentuk pada pelatihan dan hasil ekstraksi fitur dokumen testing masuk kedalam proses testing, untuk melihat setiap kelas pada setiap dokumen testing. Blok diagram arsitektur sistem pada penelitian ini dapat dilihat pada gambar 2 berikut.

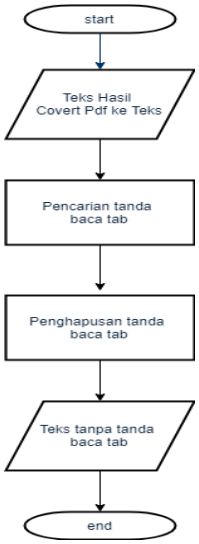


Gambar 2. Blok Diagram Sistem

Seperti yang sudah dijelaskan diawal dan dapat dilihat pada gambar 2 bahwa tahapan pertama pada sistem ekstraksi informasi adalah dengan mengupload dokumen karya tulis ilmiah yang terdiri dari lembar sampul dan abstrak ke system untuk diubah menjadi text. Hasil ini digunakan pada tahap *preprocessing*. Tahapan *preprocessing* dibagi kedalam tiga bagian yaitu *filtering*, *segmentasi*, dan *tokenizing*. Hasil dari *prepocessing* setiap token akan memiliki 2 ID, id kalimat dan id token. Pembobotan akan dilakukan per token. Kemudian hasil dari perhitungan ekstraksi fitur akan diproses dalam *fiture selection*. Pada tahapan ini berguna untuk mengurangi fitur yang kurang berpengaruh. Pada penelitian ini nilai gain yang berasal dari *information gain* yang akan digunakan sebagai tolok ukur suatu fitur pengaruh atau tidak terhadap suatu fitur. Fitur-fitur yang sudah terpilih masuk kedalam proses training dan testing menggunakan metode SVM. Performansi yang digunakan sebagai tolok ukur model yang dibangun sudah baik atau tidak adalah akurasi. Pada penelitian ini membandingkan dengan fitur yang sudah diseleksi oleh *information gain* dan yang tidak menggunakan *information gain*. Hasil yang disampaikan adalah hasil proses yang mendapatkan hasil akurasi yang terbesar.

2.4 Filtering

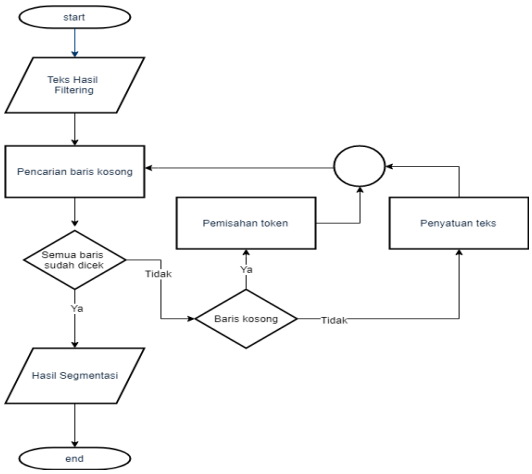
Proses *filtering* yang akan membuang tanda baca yang kecuali titik, ".". Karena didalam penelitian ini dokumen akan dipecah per kalimat. Contohnya, seperti yang sudah dijelaskan sebelumnya setelah dokumen diupload maka dokumen yang berformat .pdf diubah kedalam bentuk text. Hasil pengubahan format didalam dokumen, terdapat beberapa pemisahan dalam satu kata pada hasil convert menggunakan tab dalam program ditandai dengan "\t". Maka tanda baca ini dibuang pada proses *filtering* ini. Flowchart *filtering* dapat dilihat pada Gambar 3.



Gambar 3. Flowchart Filtering

2.5 Segmentasi

Segmentasi yang dilakukan pada penelitian ini adalah mengelompokkan baris yang seharusnya satu kelompok. Tahapan pertama dalam proses ini adalah menghapus baris yang kosong. Kemudian dilakukan beberapa logic untuk mengelompokkan baris teks yang seharusnya menjadi suatu kelompok dan juga dilakukan pemisahan baris jika memang seharusnya berbeda kelompok. Proses segmentasi pada penelitian ini dapat dilihat pada gambar 4 berikut ini.



Gambar 4. Flowchart Segmentasi

2.6 Tokenizing

Proses *tokenizing* pada penelitian ini berfungsi untuk melakukan pemotongan string input tiap barisnya yang sudah dilakukan proses *segmentasi*. Proses tokenzing ini dilakukan per baris hasil segmentasi. Proses tokenizing ini dengan bantuan spasi antar kata. Sehingga hasil dari proses ini adalah kata-kata. Proses *tokenizing* dapat dilhat pada Gambar 6.

Gambar 6. Flowchart Tokenizing

2.7 Ekstraksi Fitur

Ekstraksi fitur adalah proses pengolahan baris menggunakan fitur-fitur sehingga menghasilkan nilai-nilai fitur yang akan digunakan untuk proses selanjutnya. Pembobotan dilakukan per baris yang dihasilkan dalam proses ini bernilai 0 sampai dengan 1. Pembobotan ini mengikuti perhitungan peluang peluang pada teori probabilitas [9], agar mendapatkan nilai peluang 0 sampai dengan 1 terhadap kemunculan setiap fitur pada token kata dalam token baris.

Untuk perhitungan probabilitas dapat dilihat pada persamaan (1).

$$P(E) = \frac{x}{n} \qquad (1)$$

Keterangan :
P : probablitas
E : suatu kejadian atau peristiwa
x : banyak jumlah token yg dicari dalam satu baris
n : jumlah token dalam satu baris

Proses ini dilakukan setelah melalui tahap preprocessing. Fitur-fitur yang digunakan pada penelitian ini berjumlah 14 fitur, untuk 13 fitur merujuk pada penelitian yang dilakukan oleh Firdamdam [3] dan Aditya [4], sedangkan untuk fitur LINE merupakan fitur yang dibuat oleh peneliti. Dalam penjelasan fitur-fitur yang digunakan dalam penelitian ini dapat dilihat pada Tabel 2.

Tabel 2. Keterangan Ekstraksi Fitur

No	Nama Fitur (Fi)	Keterangan
1	**INITCAPS**	Mengenali setiap token yang hurufnya diawali dengan kapital.
2	**ALLCAPS**	Mengenali setiap token yang semua hurufnya kapital.
3	**CONTAINSDIGIT**	Mengenali setiap token yang mengandung angka.
4	**ALLDIGIT**	Mengenali setiap token yang semuanya angka
5	**CONTAINSDOTS**	Mengenali setiap token yang mengandung titik.
6	**LOWERCASE**	Mengenali setiap token yang semuanya huruf kecil.
7	**PUNCTUATION**	Mengenali setiap token yang mengandung tanda tertentu seperti titik, koma, titik dua, titik koma, tanda kurung, dan tanda seru.

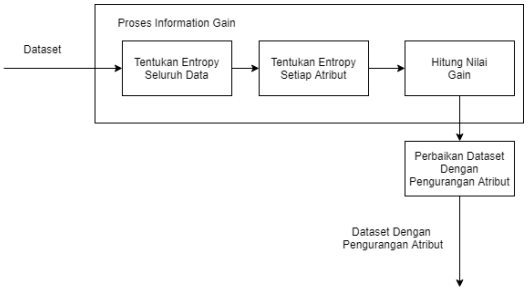
No	Nama Fitur (Fi)	Keterangan
8	**EIGHTDIGIT**	Fitur tambahan pada penelitian ini, fitur ini dikhususkan untuk mengenali token yang memiliki digit dengan panjang 8 digit.
9	**WORD**	Fitur tambahan pada penelitian ini, fitur ini dikhususkan untuk memberikan bobot pada token untuk kelas JENIS_PENELITIAN dan KALIMAT_PENGAJUAN. 10
10	**LINE_START**	Mengenali posisi token pada indeks array awal.
11	**LINE_IN**	Mengenali posisi token pada indeks array tengah.
12	**LINE_END**	Mengenali posisi token pada indeks array akhir
13	**LINE**	Mengenali posisi setiap token
14	**YEAR**	Mengenali ciri token tahun.

2.8 Pelabelan Kelas

Proses pelabelan dilakukan secara manual oleh peneliti. Pemberian label dilakukan per baris. Artinya token yang berada dalam suatu baris akan memiliki kelas yang sama. Pemberian kelas pada setiap baris di dokumen diperlukan untuk pembangunan model pada proses pelatihan. Mengingat metode klasifikasi yang akan digunakan termasuk metode *supervised*.

2.9 Information Gain

Information gain merupakan ekspektasi dari pengurangan entropi yang dihasilkan dari partisi objek dataset berdasarkan fitur tertentu. Blok diagram information gain yang akan digunakan pada penelitian ini dapat dilihat pada Gambar 7



Gambar 7. Blok Diagram Information Gain

Tahapan pertama dilakukannya perhitungan probabilitas dan entropy dari jumlah dataset terhadap 16 kelas pada Tabel 3.21. perhitungan entropy dilakukan dengan persamaan (2).

$$Entropy\ (S) = \sum_{i=1}^{c} -P_i log_2 P_i \quad (2)$$

Untuk perhitungan proporsi pada persamaan (3).

$$P(E) = \frac{x}{n} \quad (3)$$

Setelah mendapatkan nilai entropy seluruh kelas, kemudian hitung entropy dari setiap atribut di dalam kelas dengan dua kondisi nilai. Kondisi pertama nilai lebih besar dari 0 yang menandakan fitur memiliki pengaruh terhadap data atau kondosi kedua, yaitu nilai sama dengan 0 yang menandakan fitur tidak memiliki pengaruh terhadap data. Dilakukannya perhitungan proporsi fitur (Fi) terhadap kelas, didapatkannya jumlah Fi yang memiliki nilai lebih besar dari 0 atau lebih kecil dari 0 pada kelas.

Karena untuk Entropy fitur memiliki 2 kondisi maka dilakukannya perhitungan Entropy setiap fitur berdasarkan kondisi yang dimiliki, dengan persamaan (4).

$$Entropy_{Fi}(S) = \sum_{j=1}^{v} \frac{|n_j|}{|n|} Entropy(Fi(E)) \quad (4)$$

Keterangan :
v : jumlah kondisi
n_j : jumlah data setiap kondisi

Selanjutnya hitung nilai Gain menggunakan persamaan (5).

$$IG(S.A) = Entropy\ (S) - \sum cevalues(A) \frac{|S_v|}{|S|} Entropy(S_v) \quad (5)$$

Lakukan pengurangan $Entropy(S)$ seluruh kelas dan $Entropy_{Fi}(S)$.

2.10 *Support Vector Machine*

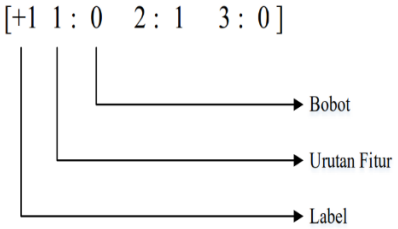
Support Vector Machine (SVM) adalah salah satu metode yang dapat digunakan dalam ekstraksi informasi. *Support Vector Machine* (SVM) dikembangkan oleh Boser, Guyon, dan Vapnik, pertama kali diperkenalkan pada tahun 1992 di Annual Workshop on Computational Learning Theory. Pemahaman dalam cara kerja SVM adalah memisahkan dua buah kelas yang terpisah secara linier dengan membuat sebuah garis pemisah yang disebut hyperplane [10]. Dalam SVM dikenal istilah margin, yaitu jarak antara garis hyperplane dengan data yang paling dekat yang disebut dengan support vector [4]. Usaha untuk mencari lokasi hyperplane ini merupakan inti dari proses pelatihan pada svm [10].

Dalam penilitian ini *Support Vector Machine* (SVM) adalah metode yang dipilih untuk klasifikasi. Sebelum dilakukannya klasifikasi, fitur-fitur yang sudah terpilih oleh *Information Gain* disusun dalam bentuk vektor, representasi data. Dalam penelitian ini kelas yang terlibat ada 16 kelas. Model SVM

digunakan dalam penelitian ini adalah *Multiclass one vs all.*

2.11 Representasi Data

Dalam penelitian ini untuk representasi data menggunakan format *sparse* data *representation* dengan vektor sebagai inputnya. Format data input untuk klasifikasi SVM dalam penelitian ini adalah [label row-id:bobot(row-id) ..... row-id-ke-n:bobot(row-id-ke-n)]. Dengan label masukan pertama +1 atau -1 menyatakan dua kelas. Angka kedua menyatakan dimensi (row_id) dan angka ketiga setelah tanda ":" menyatakan bobot, setiap bobot didalam dokumen dipisahkan dengan spasi. Contoh hasil format data input seperti pada gambar 3.14.

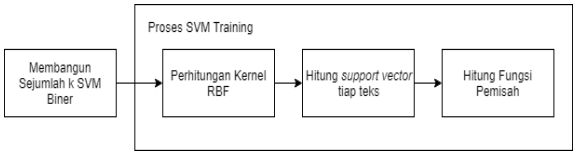


Gambar 8. Contoh Format Data Input Vektor

Data latih dan data uji akan diubah menjadi data vektor. Data latih adalah data yang sudah memiliki kelas. Sedangkan data uji adalah data yang belum memiliki kelas. Data latih digunakan untuk melakukan proses pembelajaran terhadap sistem. Proses pembelajaran ini akan menghasilkan model baru yang akan digunakan pada klasifikasi.

2.12 Training *Support Vector Machine*

Pada proses pelatihan SVM bertujuan untuk mencari hyperplane. Hyperplane adalah kurva atau bidang pemisah antara dua kelas. Fungsi pemisah umumnya berupa fungsi polinom, sehingga tugas utama adalah menemukan vektor α dan konstanta b, dengan margin yang maksimal. Margin adalah jarak antara support vector dengan hyperplane. Vektor dengan α > 0 dinamakan *support vector* dan menyatakan data training yang diperlukan untuk mewakili fungsi keputusan yang optimal[1]. Konstanta b menentukan lokasi fungsi pemisah *relative* terhadap titik asal (*origin*)[1]. Berikut blok diagram proses training *Support Vector Machine*.



Gambar 9. Blok Diagram Training SVM

Pada penelitian ini, data ditranformasi mengunakan fungsi *kernel* RBF yang didefinisikan sebagai $K(x,y) = \exp(-\gamma\|x-y\|^2)$, $\gamma > 0$
Formulasi yang digunakan adalah dualitas *Lagrange multiplier* pada persamaan (6).

$$L\,p = \frac{1}{2}||w||^2 - \sum_{i=1}^{n} \alpha_i y_i (w^\tau x_i + b) - 1 \quad (6)$$

sehingga untuk w harus menggunakan persamaan (7).

$$\frac{\partial Lp}{\partial w} = 0 \rightarrow w = \sum_{i=1}^{N} \alpha_i y_i x_i \quad (7)$$

Persamaan (7) yang sudah dimodifikasi untuk x dengan fungsi *kernel*, menjadi $W = \sum_{i=1}^{N} a_i y_i \phi x_i$

Selanjutnya harus dilakukan kernelisasi pada set data dari fitur dimensi lama sehingga mendapatkan set data dengan fitur baru dimensi tinggi. Dengan *kernel* $K(X,X_i) = \exp(-\gamma\|X-X_i\|^2)$ dan set data berdimensi Nx1 maka akan didapatkan dimensi baru NxN, dimana N adalah banyaknya data. Kemudian hasil dari matriks K diatas setiap elemenya merupakan hasil exp $(-\gamma\|x - x_i\|^2)$ yang akan berkorelasi dengan $a_i a_j$ dalam *Lagrange Multiplier* persamaan (8).

$$Ld = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \boldsymbol{x_i}.\boldsymbol{x_j} \quad (8)$$

Dengan menggunakan kernel K sebagai pengganti *dot-product* $x_i x_j$ dalam persamaan dualitas *Lagrange multiplier*, didapatkan:

$$Ld = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j K(\boldsymbol{x_i}, \boldsymbol{x_j}) \quad (9)$$

Dengan syarat pada persamaan 10 dan 11.

Syarat 1 :

$$a_1 y_1 + a_2 y_2 + a_3 y_3 + a_4 y_4 + a_5 y_5 = 0 \quad (10)$$

Syarat 2 :

$$a_1, a_2, a_3, a_4, a_5 \geq 0 \quad (11)$$

Dalam fungsi tujuan, suku kedua sudah dikalikan dengan $y_i y_j$. Persamaan tersebut memenuhi standar *Quadratic programming* sehingga bisa dibantu penyelesainnya dengan solver komersial untuk *Quadratic Programming*(QP) untuk mendapatkan semua nilai *a* dan *b*. Setelah menemukan semua nilai a dan b, maka model SVM sudah siap digunakan untuk prediksi.

2.13 Testing *Support Vector Machine*

Setelah mendapatkan model klasifikasi dan mendapatkan nilai *hyperplane*, selanjutnya dapat menentukan data masuk ke dalam kelas positif atau negatif menggunakan nilai w dan *hyperplane* dari data *training*. Jika nilai hasil uji lebih besar dari nilai *hyperplane* maka data tersebut masuk dalam kelas

positif, jika lebih kecil dari nilai *hyperplane* maka data tersebut masuk dalam kelas negative

Semua fungsi pemisah yang memisahkan satu kelas dengan 2 kelas lainnya. $K(x_{training}, x_{testing})$ merupakan fungsi *kernel* RBF exp $(-\gamma\|x - x_i\|^2)$, $\gamma > 0$ dengan nilai a dan b diperoleh dari proses SVM *training* dan y merupakan hasil perhitungan pada fungsi model SVM *training*. Dengan menggunakan fungsi hyperplane dengan kernel $K(x_{training}, x_{testing})$ sebagai pengganti *dot-product* $\boldsymbol{x_i}.\boldsymbol{z}$ pada persamaan (12).

$$\text{hyperplane}(kelas) = \left(\left(\sum_{i=1}^{N} a_i y_i K(x_{trainingi}, x_{testing})\right) + b\right) \quad (12)$$

Dengan menggunakan metode *one-againts-all*, dibangun k buah model SVM biner (k adalah jumlah kelas). Setiap model klasifikasi ke-i dilatih dengan menggunakan keseluruhan data, untuk mencari solusi permasalahan [8]. Contohnya, terdapat permasalahan klasifikasi dengan 4 buah kelas. Maka banyak hyperplane yang harus dibentuk sebanyak 4, seperti yang dapat dilihat pada table 3

Tabel 3. Hyperplane untuk *one vs all*

Yi = +1	Yi = -1	Hipotesis
Kelas 1	Bukan Kelas 1	$f^1(x) = (w^1)x + b^1$
Kelas 2	Bukan Kelas 2	$f^2(x) = (w^2)x + b^2$
Kelas 3	Bukan Kelas 3	$f^3(x) = (w^3)x + b^3$
Kelas 4	Bukan Kelas 4	$f^4(x) = (w^4)x + b^4$

2.14 Hasil Pengujian

Pengujian dilakukan dengan menyamakan hasil klasifikasi sistem dengan kelas token yang sebenarnya. Nilai akurasi dan kesalahan, tebakan sistem terhadap kelas yang sebenarnya adalah rata-rata nilai masing-masing per dokumen.

2.15 Pengujian

Dilakukan pengujian terhadap 40 dokumen yang diantaranya 20 dokumen sampul dan 20 dokumen abstrak dengan menggunakan *Confusion Matrix*. Setelah dilakukan preprocessing didapatkan 322 token baris data yang akan dilakukan klasifikasi. Setiap token baris akan dilakukan pelabelan kelas sebenarnya sebelum melewati proses klasifikasi. Kemudian menghasilkan kelas yang sesuai dan kelas yang tidak sesuai. Berikut hasil kelas klasifikasi dengan kelas sebenarnya.

Tabel 4. Hasil Pengujian Kelas Klasifikasi Sesuai Dengan Kelas Klasifikasi Tidak Sesuai

Jumlah Data	Kelas Klasifikasi Sesuai	Kelas Klasifikasi Tidak Sesuai
20	2	18
20	20	0
20	20	0
20	18	2
19	18	1
21	20	1
21	20	1
20	20	0
20	20	0
20	20	0
20	20	0
21	15	6
19	19	0
21	20	1
20	20	0
20	20	0
Total	**292**	**30**

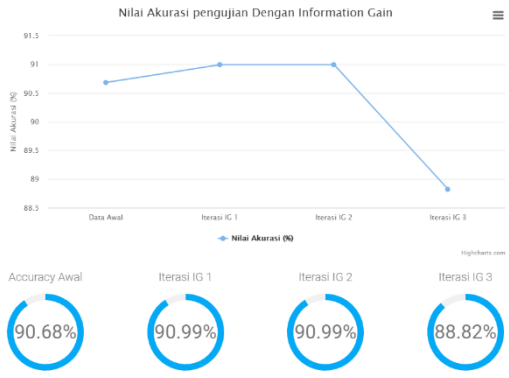
Kemudian hitung tingkat akurasi dengan hasil klasifikasi yang sesuai telah didapatkan sebanyak 292 data. Tingkat akurasi keseluruhan dihitung dengan rumus sebagai berikut.

$$Accuracy = \frac{292}{322} * 100\% = 90.68\%$$

Sedangkan selanjutnya akan dihitung *error rate* dengan hasil klasifikasi yang tidak sesuai telah didapatankan sebanyak 30 data. Tingkat akurasi keseluruhan dihitung dengan rumus sebagai berikut.

$$Error\ Rate = \frac{30}{322} * 100\% = 9.31\%$$

Dan dengan penggunaan seleksi fitur information gain berhasil memberikan peningkatan akurasi sebesar 0.31% menjadi 90.99% pada iterasi ke2 dan mengalami penurunan pada iterasi ke3. Grafik perubahan akurasi dengan information gain dapat dilihat pada Gambar 10.



Gambar 10. Hasil Akurasi Dengan Information Gain

2.15 Analisis Hasil Pengujian

Hasil kelas klasifikasi yang diperoleh memiliki beberapa penyebab. Analisis yang dilakukan terhadap hasil pengujian dengan bentuk pengujian *Confusion Matrix* menyimpulkan bahwa yang pertama adalah karena data masukan yang digunakan oleh sistem adalah file teks, maka terdapat keterbatasan pada fitur yang digunakan. Kedua, berdasarkan pengamatan yang dilakukan, dampak yang mempengaruhi pada kelas hasil klasifikasi tidak sesuai disebabkan oleh fitur kelas Judul Penelitian (Sampul) dan Judul Halaman Abstrak yang memiliki pembobotan fitur yang serupa dikarenakan judul lembar sampul dan abstrak memang sama, karena berasal dari draf yang sama. Ketiga, tidak adanya penggunaan ekstraksi fitur yang dapat membedakan kedua dokumen secara spesifik. Sedangkan, berdasarkan kinerja sistem, nilai akurasi yang diperoleh paling besar adalah 90.99%.

3. PENUTUP

Berdasarkan pengujian fungsionalitas sistem dan pengukuran akurasi yang telah dilakukan, sistem ekstraksi informasi menggunakan algoritma *Support Vector Machine* (SVM) telah berhasil dibangun dengan perolehan akurasi ebesar 90.68% tanpa information gain dan 90.99% dengan information gain pada iterasi ke 2. Perolehan akurasi kelas hasil klasifikasi yang tidak sesuai paling banyak disebabkan oleh fitur kelas Judul Penelitian (Sampul) dan Judul Halaman Abstrak yang memiliki beberapa data dengan pembobotan fitur yang serupa, sehingga algoritma SVM tidak dapat melakukan klasifikasi secara benar.

Saran untuk pengembangan selanjutnya Perlu adanya penambahan fitur, terutama untuk menentukan kategori Judul Penelitian (Sampul) dan Judul Halaman Abstrak karena dari hasil ekstraksi fitur saat ini, ekstraksi fitur dari kedua kategori tersebut memiliki sedikit perbedaan.

DAFTAR PUSTAKA

[1] S. Sarawagi, Information Extraction, India: Foundations and TrendsR ! in Databases Vol. 1, No. 3 (2007) 261–377 , 2008.

[2] D. Mustaqwa dan N. Indriani, "Implementasi Ekstraksi Informasi Pada Dokumen Teks Skripsi Menggunakan Ruled Based," *Skripsi, Universitas Komputer Indonesia*, p. 8, Indonesia, Bandung, Jawa Barat, 2018.

[3] F. Sasmita and K. K. Purnamasari, "Ekstraksi Informasi Dokumen Karya Tulis Ilmiah Menggunakan Algoritma Learning Vector Quantization," *Skripsi, Universitas Komputer Indonesia*, p. 8, Indonesia, Bandung, Jawa Barat, 2018.

[4] A. I. Riaddy, Y. Sibaroni dan A. Aditsania, "Ekstraksi Informasi pada Makalah Ilmiah dengan Pendekatan Supervised Learning," *e-Proceeding of Engineering*, p. 1184, April 2016.

[5] V. Chandani, R. S. Wahono and P. , "Komparasi Algoritma Klasifikasi Machine Learning Dan Feature Selection pada Analisis Sentimen Review Film," *Journal of Intelligent Systems, Vol. 1, No. 1*, February 2015.

[6] S. N. Asiyah dan K. Fithriasari, "Klasifikasi Berita Online Menggunakan Metode Support Vector Machine dan K- Nearest Neighbor," *JURNAL SAINS DAN SENI ITS Vol. 5 No. 2* , 2016.

[7] K. P. Wijaya dan M. A. Muslim, "Peningkatan Akurasi pada Algoritma Support Vector Machine dengan Penerapan Information Gain untuk Mendiagnosa Chronic Kidney Disease," *Seminar Nasional Ilmu Komputer (SNIK 2016)*, Semarang, 10 Oktober 2016.

[8] N. Indriani, E. Rainarli dan K. E. Dewi, "Peringkasan dan Support Vector Machine pada Klasifikasi Dokumen," *JURNAL INFOTEL Informatika - Telekomunikasi - Elektronika*, 09 November 2017.

[9] A. Setiawan, Pengantar Teori Probabilitas, Salatiga: Tisara Grafika, Juni 2015.

[10] E. Alpaydın, Introduction to Machine Learning Second Edition, Cambridge, Massachusetts London, England: The MIT Press, 2010.