Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer e-ISSN: 2548-964X
Vol. 7, No. 5, Mei 2023, hlm. 2132-2141 http://j-ptiik.ub.ac.id

Klasifikasi Judul Berita *Online* menggunakan Metode *Support Vector Machine* (SVM) dengan Seleksi Fitur *Chi-square*

Putu Rama Bena Putra[1], Indriati[2], Rizal Setya Perdana[3]

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: [1]ramabenarb@student.ub.ac.id, [2]indriati.tif@ub.ac.id, [3]rizalespe@ub.ac.id

Abstrak

Perkembangan teknologi mempengaruhi berbagai sektor salah satunya sektor berita. Penyebaran berita mulai memanfaatkan teknologi dengan munculnya berita *online*. Berita *online* yang ada memiliki bermacam-macam kategori. Saking banyaknya kategori terkadang suatu berita dapat memiliki kategori yang salah. Berita yang tidak sesuai dengan kategorinya dapat mengecoh pembaca. Dalam menilai suatu berita *online* masuk kategori mana selain melihat dari isinya dapat melalui judulnya, dikarenakan judul berita merupakan representasi utama dari berita. Dari permasalahan yang ada maka dilakukan penelitian klasifikasi judul berita *online*. Penelitian menggunakan *support vector machine* (SVM) sebagai metode klasifikasi. Selain menggunakan SVM dilakukan juga seleksi fitur dengan *chi-square* untuk mengurangi dimensi fitur. Data yang dipakai berjumlah 2400 judul berita *online* dengan 6 kategori. Proses klasifikasi dimulai dari *pre-processing text*, *term weighting* menggunakan TF-IDF, selanjutnya seleksi fitur dengan *chi-square*, dan terakhir klasifikasi dengan SVM. Penelitian dilakukan dengan mencari parameter SVM yang terbaik dan juga nilai terbaik dari *threshold chi-square*. Hasil dari pengujian memberikan hasil terbaik yaitu akurasi 93,06%, presisi 92,11%, *recall* 93,06%, dan *f1-score* 93,04%, terjadi ketika *threshold chi-square* sebesar 80% serta nilai parameter SVM berupa *kernel* menggunakan *polynomial* derajat 2, C=1, λ=1, konstanta γ=0,01, ε=10^{-8}, dan maksimal iterasi sebesar 10.

Kata kunci: *Klasifikasi, Judul Berita, Support Vector Machine, Chi-square*

Abstract

Technological developments affect various sectors, including the news sector. News dissemination has begun to take advantage of technology with the advent of online news. The existing online news has various categories. With so many categories, sometimes news can fall into the wrong category. News that does not fit the category can mislead readers. In assessing which online news falls into which category besides looking at its contents, it can also look at the title due to news title is the main representation of the news. Based on existing problems, research on the classification of online news titles has been conducted. The research used the support vector machine (SVM) as a classification method. In addition to using SVM, feature selection is also carried out with chi-square to reduce feature dimensions. The data used amounted to 2400 online news titles with 6 categories. The classification process starts from text pre-processing, term weighting using TF-IDF, feature selection with chi-square, and finally classification with SVM. The research was conducted by looking for the best SVM parameters and the best value of the chi-square threshold. The results of the test gave the best results, namely accuracy of 93.06%, precision of 92.11%, recall of 93.06%, and f1-score of 93.04%, when the chi-square threshold was 80% and the SVM parameter value was in the form of a kernel using polynomial degree 2, C=1, λ=1, constants γ=0.01, ε=10^{-8}, and a maximum of 10 iterations.

Keywords: *Classification, News Title, Support Vector Machine, Chi-square*

1. PENDAHULUAN

Perkembangan teknologi dan informasi terjadi semakin cepat dan luas seiring dengan perkembangan zaman. Cepatnya perkembangan informasi yang ada salah satunya muncul dari sektor berita. Penyebaran berita pada zaman sekarang mulai memanfaatkan adanya teknologi yaitu ditandai dengan banyak munculnya berita *online*. Isi dari berita *online* beragam seperti

mengenai olahraga, politik, kesehatan, makanan, keuangan dan lain sebagainya. Salah satu cara melihat kategori berita selain dari isinya adalah berdasarkan judul berita. Judul berita merupakan representasi utama dari isi berita serta judul merupakan bagian paling mendapat perhatian pembaca berita (Maisarah, 2021).

Saking banyaknya berita *online* yang dimasukkan ke dalam portal berita, sehingga terkadang berita tersebut tidak tepat dengan kategori seharusnya. Berita yang salah dikategorikan dapat membuat pembaca terkecoh karena berita yang disajikan tidak relevan dengan kategorinya. Kesalahan kategori ini dapat terjadi akibat *human error* saat melakukan klasifikasi. Dalam mengurangi kesalahan klasifikasi berita maka diperlukan sistem yang otomatis untuk melakukan klasifikasi.

Pada proses klasifikasi teks dapat dilakukan dengan menerapkan model pembelajaran mesin yang salah satu metodenya bernama *support vector machine* (SVM). SVM adalah model klasifikasi umum dan dapat diterapkan untuk menyelesaikan masalah yang berada di berbagai domain. Saat ini, pengklasifikasi berbasis SVM dapat juga dilakukan untuk klasifikasi teks (Saigal & Khanna, 2020).

Klasifikasi teks pada berita *online* menggunakan SVM pernah dijadikan penelitian oleh (Liliana dkk., 2011). Data yang dipakai adalah berita bahasa Indonesia yang diambil dari www.kompas.com dengan 4 kategori dan 180 berita. Penelitian dilakukan dengan mencari pengaruh SVM terhadap klasifikasi berita *online* Indonesia serta kombinasi parameter terbaik di dalamnya yaitu parameter C dan *gamma*. Dari hasil penelitian ini, ditemukan bahwa akurasi terbaik dicapai ketika parameter *C* bernilai 110 dan *gamma* 1 yaitu akurasi sebesar 91%.

Klasifikasi judul berita pernah dijadikan penelitian oleh (Mukhtar dkk., 2021) tentang klasifikasi berita Pakistan berdasarkan judul berita. Pada penelitian tersebut membandingkan berbagai macam model pembelajaran mesin dan juga *multi layer perceptron*. Hasil yang didapat adalah dengan TF-IDF dan metode SVM menjadi akurasi tertinggi yaitu mendapat akurasi 0,82 dan *f-measure* 0,81.

Penelitian lain dilakukan oleh (Shahi & Pant, 2018) dengan membandingkan 3 metode klasifikasi yaitu Naïve Bayes, SVM, dan *Neural Networks* pada berita Nepal. Hasil dari penelitian ini diperoleh SVM dengan *kernel* RBF menjadi yang tertinggi dengan akurasi 74,65%.

Masalah utama klasifikasi teks adalah tingginya dimensi fitur yang ada. Dimensi fitur yang tinggi disebabkan karena setiap kata akan dijadikan sebagai fitur. Semakin banyak dokumen yang digunakan, semakin banyak juga kata atau fitur yang tercipta. Terlalu banyaknya fitur dapat menurunkan akurasi dan juga dapat memperlambat proses klasifikasi (Chen dkk., 2009). Dari masalah banyaknya fitur yang ada sehingga proses pemilihan fitur atau seleksi fitur menjadi diperlukan.

Penelitian mengenai seleksi fitur pada klasifikasi berita pernah dilakukan oleh (Alshalabi dkk., 2013). Penelitian ini membandingkan seleksi fitur *chi-square* dengan information gain. Hasil dari penelitian ini adalah seleksi fitur *chi-square* dengan k-NN menjadi yang terbaik dengan nilai *f-measure* 96,14%.

Dalam penelitian (Zainuddin & Selamat, 2014) yang berjudul "*Sentiment Analysis Using Support Vector Machine"* dilakukan sentimen analisis menggunakan SVM dan juga mencari pengaruh seleksi fitur *chi-square* terhadap akurasi model. Hasil yang didapat yang sebelumnya tanpa *chi-square* mendapat akurasi 79,83% setelah memakai *chi-square* menjadi 81%.

Berdasarkan paparan latar belakang yang telah dijelaskan penggunaan SVM beserta parameter di dalamnya mampu menghasilkan akurasi yang tinggi terhadap klasifikasi judul berita. Seleksi fitur *chi-square* juga dapat meningkatkan akurasi dibandingkan tanpa seleksi fitur. Penelitian ini kemudian dilakukan dengan mengambil judul "Klasifikasi Judul Berita *Online* Menggunakan Metode *Support Vector Machine* (SVM) dengan Seleksi Fitur *Chi-square"*.

2. DASAR TEORI

2.1. Berita

Pada KBBI berita merupakan cerita atau keterangan mengenai kejadian atau peristiwa yang hangat. Dalam suatu berita terdiri dari berbagai komponen di antaranya judul, isi, dan juga bukti berita baik berupa gambar maupun video. Judul berita merupakan salah satu komponen penting dalam berita karena dalam judul dapat membuat pembaca lebih cepat memahami apa yang akan dibahas dalam berita tersebut (Sumadiria, 2006).

2.2. *Pre-processing Text*

Pre-processing text adalah proses penyiapan

dan pengolahan teks sebelum masuk ke tahap klasifikasi. Pada *Pre-processing text* bertujuan mengubah data yang sebelumnya berbentuk tidak terstruktur menjadi terstruktur (Haddi dkk., 2013). Beberapa tahap *pre-processing text* yaitu *case folding, cleaning, tokenizing, stopword removal,* dan *stemming*. Masing-masing penjelasan tahapan *pre-processing text* adalah sebagai berikut:

1. *Case folding*
   Proses *case folding* yaitu mengubah semua huruf menjadi huruf kecil. *Case folding* bertujuan untuk mencegah ambigu dari 2 kata sama dengan struktur huruf berbeda (Petrović & Stanković, 2019).
2. *Cleaning*
   Proses *cleaning* adalah proses menghilangkan karakter yang dianggap tidak penting untuk proses klasifikasi. Karakter yang dapat dihilangkan berupa angka, tanda baca, dan karakter spasi berlebih.
3. *Tokenizing*
   Tokenizing adalah proses memecah dokumen menjadi *token*. Proses pemisahannya dapat memilih satu atau beberapa karakter sebagai karakter pemisah atau sering disebut *delimiter*. Karakter *delimiter* yang paling sering digunakan adalah spasi sehingga setiap *token* akan berupa kata (Vijayarani & Janani, 2016).
4. *Stopword Removal*
   Stopword removal adalah penghapusan kata yang tidak membawa informasi signifikan atau menunjukkan subjek dari teks yang diproses. Dengan melakukan proses *stopword removal* dapat meningkatkan akurasi dalam klasifikasi teks dikarenakan membuat setiap dokumen hanya terdiri dari kata-kata penting saja (Anoual & Zeroual, 2021).
5. *Stemming*
   Stemming adalah proses mengubah kata menjadi bentuk dasarnya atau disebut *stem*. Proses *stemming* dilakukan dengan memotong bagian yang dianggap awalan dan akhiran pada kata (Kannan dkk., 2014).

2.3. *Term Weighting*

Term weighting merupakan pembobotan kata yang berguna untuk memberi nilai pada setiap kata. *Term weighting* yang dipakai menggunakan metode *Term Frequency-Inverse Document Frequency* (TF-IDF). Proses TF-IDF dapat dipecah menjadi 2 yaitu TF dan IDF. TF mengandung arti seberapa sering suatu kata tersebut muncul dalam suatu dokumen, sedangkan IDF mengandung arti seberapa langka kata tersebut muncul di semua dokumen. Fungsi menghitung nilai TF dapat dilihat pada Persamaan (1), sedangkan untuk IDF pada Persamaan (2). Setelah mendapat nilai TF dan IDF, kedua nilai tersebut dikalikan untuk menjadi nilai bobot kata *t* pada dokumen *d* seperti pada Persamaan (3).

$$tf_{t,d} = \begin{cases} 1 + \log_{10} f_{t,d} \ , jika\ f_{t,d} > 0 \\ 0 \ , jika\ f_{t,d} = 0 \end{cases} \quad (1)$$

$$idf_t = \log_{10}\left(\frac{N}{df_t}\right) \quad (2)$$

$$w_{t,d} = tf_{t,d} \times idf_t \quad (3)$$

Keterangan:

$tf_{t,d}$	=	Nilai TF kata *t* dalam dokumen *d*
$f_{t,d}$	=	Banyaknya kata *t* yang muncul dalam dokumen *d*
idf_t	=	Nilai IDF pada kata *t*
N	=	Banyaknya dokumen yang ada
df_t	=	Banyaknya dokumen yang mengandung kata *t*
$w_{t,d}$	=	Bobot kata *t* terhadap dokumen *d*

2.4. *Support Vector Machine*

Support vector machine (SVM) merupakan salah satu metode klasifikasi dalam pembelajaran mesin. SVM diperkenalkan pada tahun 1992 melalui *Annual Workshop on Computational Learning Theory* oleh Boser, Guyon, dan Vapnik. Cara kerja SVM dengan mencari *hyperplane* atau garis pemisah yang dapat memisahkan dua kelas data, juga memaksimalkan nilai *margin* atau jarak dua kelas data yang berbeda (Nugroho dkk., 2003).

Pada mulanya prinsip kerja dari SVM adalah linear *classifier* yaitu klasifikasi yang diselesaikan dengan garis lurus pemisah. SVM lalu dikembangkan agar dapat menangani masalah non-linear dengan memasukkan data x ke dalam fungsi $\phi(x)$, yang mana fungsi $\phi(x)$ adalah fungsi transformasi data x ke ruang dimensi yang lebih tinggi sehingga dapat dipisahkan secara linear.

Umumnya nilai baru dari transformasi ϕ tidak diketahui dan sangat sulit dipahami. Dari teori Mercer perhitungan *dot product* antara fungsi ϕ dapat diganti dengan *kernel trick* yang mendefinisikan secara langsung dari transformasi ϕ. Berbagai fungsi *kernel* yang ada dapat dilihat melalui Tabel 1.

Tabel 1. Daftar *Kernel*

Nama *Kernel*	Fungsi *Kernel*
Linear	$K(x_i, x_j) = x_i . x_j$
Polynomial	$K(x_i, x_j) = (x_i . x_j)^d$
Gaussian RBF	$K(x_i, x_j) = \exp\left(\frac{-\|x_i - x_j\|^2}{2\sigma^2}\right)$
Sigmoid	$K(x_i, x_j) = \tanh(\sigma(x_i, x_j) + c)$

Proses klasifikasi SVM dilakukan dengan mencari nilai *f(x)* dengan x adalah data yang ingin diklasifikasikan. Rumus dari mencari nilai *f(x)* dapat dilihat pada Persamaan (4).

$$f(x) = \sum_{i=1}^{n} a_i y_i K(x_i, x) + b \quad (4)$$

Keterangan:

$f(x)$ = Fungsi klasifikasi untuk data x
n = Banyak data latih
α_i = *lagrange multiplier* untuk data ke-*i*
y_i = Label data latih ke-*i*
$K(x_i, x)$ = Fungsi *kernel* untuk data latih x_i dan x
b = Nilai bias

Pada persamaan di atas terdapat nilai alpha (α) dan juga bias yang mana nilai ini diperoleh dari proses pelatihan SVM. Salah satu metode pelatihan SVM adalah *sequential training*.

Dalam klasifikasi biner atau klasifikasi yang terdiri dari hanya 2 kelas, hasil dari *f(x)* kemudian dimasukkan dalam fungsi *sign* yang menghasilkan +1 jika hasil *f(x)* positif dan -1 jika negatif. Pada klasifikasi *multi class* diperlukan metode tambahan untuk memprediksi hasil. Salah satu metode untuk menangani *multi class* SVM bernama *one-againts-all*.

2.5. *Sequential Training*

Sequential training merupakan teknik pelatihan SVM yang dikembangkan oleh (Vijayakumar & Wu, 1999). Teknik *sequential training* SVM mampu menghasilkan solusi yang optimal dan dapat mempercepat proses iterasi pelatihan. Langkah-langkah *sequential training* SVM sebagai berikut:

1. Inisialisasi parameter yang dipakai, antara lain lambda (λ), konstanta gamma (γ), *complexity* (C), epsilon (ε), dan maksimal iterasi. Inisialisasi juga variabel α_i untuk *i* dari 1 sampai dengan banyak data dengan nilai 0.
2. Buat matriks *hessian* dengan mencari nilai *hessian* untuk setiap pasangan data menggunakan Persamaan (5).

$$D_{i,j} = y_i y_j \left(K(x_i, x_j) + \lambda^2\right) \quad (5)$$

3. Mulai data ke-i dari 1 sampai ke-N hitung nilai *error* data ke-i (E_i) menggunakan Persamaan (6), lalu hitung besar perubahan alpha ke-i ($\delta\alpha_i$) dengan Persamaan (7), terakhir perbarui nilai α_i menggunakan Persamaan (8).

$$E_i = \sum_{j=1}^{N} \alpha_i D_{i,j} \quad (6)$$

$$\delta\alpha_i = min\{max[\gamma(1 - E_i), -\alpha_i], C - a_i\} \quad (7)$$

$$\alpha_i = \alpha_i(lama) + \delta\alpha_i \quad (8)$$

Dalam Persamaan (7) nilai γ (gamma) didapat dari Persamaan (9).

$$\gamma = \frac{konstanta\ \gamma}{\max(D_{i,i})} \quad (9)$$

4. Ulangi langkah ke-3 sampai nilai α mencapai konvergen yaitu ketika nilai terbesar perubahan *alpha* kurang dari parameter epsilon (ε) yang lebih rincinya dengan memenuhi Persamaan (10) atau sampai jumlah iterasi mencapai nilai maksimum iterasi.

$$\text{Max}(|\delta\alpha|) < \varepsilon \quad (10)$$

5. Dari hasil proses di atas didapat nilai α untuk setiap data. Berikutnya dihitung nilai bias dengan Persamaan (11).

$$b = -\frac{1}{2}\left(\sum_{i=1}^{N} \alpha_i y_i K(x_i, x^+) + \sum_{i=1}^{N} \alpha_i y_i K(x_i, x^-)\right) \quad (11)$$

Keterangan:

x^+ = Data kelas positif dengan α tertinggi
x^- = Data kelas negatif dengan α tertinggi

2.6. *One-Againts-All*

Dalam menangani permasalahan klasifikasi *multi class* diperlukan teknik tambahan pada SVM. Solusi paling awal dan salah satu yang paling banyak digunakan adalah metode *one-against-all*. Pada metode *one-against-all* memiliki cara membangun klasifikasi SVM sebanyak kelas yang ada, dengan SVM ke-*i* memisahkan kelas ke-*i* dari kelas lain yang tersisa (Liu & Zheng, 2005).

Langkah-langkah dari *one-against-all* yaitu untuk setiap kelas yang ada lakukan iterasi kelas. Label diubah menjadi 1 jika sama dengan kelas

yang diuji sekarang, sedangkan menjadi -1 jika tidak. Selanjutnya dilakukan pelatihan dan pengujian SVM untuk mendapatkan nilai $f(x)$. Proses prediksi kemudian dicari melalui Persamaan (12).

$$class\ of\ x = \arg\max_{i=1,\dots,m} \left(f_i(x)\right) \quad (12)$$

Keterangan:

arg max = Fungsi mencari indeks dari nilai tertinggi
m = Banyaknya kelas yang ada
$f_i(x)$ = Fungsi klasifikasi untuk kelas ke-i, $f_i(x)$ didapat melalui Persamaan (4).

2.7. *Chi-square*

Chi-square adalah metode seleksi fitur dengan cara menghitung tingkat ketergantungan fitur. Dalam pemrosesan teks, diukur tingkat ketergantungan sebuah kata atau fitur dengan kelas tertentu (Ling dkk., 2014). Semakin tinggi nilai *chi-square* pada suatu fitur maka semakin tinggi juga ketergantungan fitur tersebut.

Pada pemrosesannya umumnya dicari nilai ketergantungan suatu kata *t* terhadap kelas *c* (Yang & Pedersen, 1997). Persamaan dari *chi-square* dapat dilihat pada Persamaan (13).

$$x^2(t,c) = \frac{N(A.D-C.B)^2}{(A+C)(B+D)(A+B)(C+D)} \quad (13)$$

Keterangan:

N = Banyaknya kelas yang ada
A = Jumlah dokumen kelas c yang terdapat kata t
B = Jumlah dokumen yang bukan kelas c yang terdapat kata t
C = Jumlah dokumen kelas c yang tidak terdapat kata t
D = Jumlah dokumen yang bukan kelas c yang tidak terdapat kata t

Hasil dari *chi-square* supaya dapat dipakai untuk seleksi fitur diperlukan nilai *chi-square* tunggal dari setiap kata *t.* Cara yang dilakukan untuk mendapatkan nilai *chi-square* tunggal dengan cara mencari nilai maksimal nilai *chi-square* kata *t* terhadap semua kelas yang ada (Yang & Pedersen, 1997). Persamaan *chi-square* tunggal untuk setiap kata *t* dapat dilihat pada Persamaan (14) dengan *m* adalah banyaknya kelas yang ada.

$$X^2{}_{max}(t) = \max_{\mathrm{i=1,..,m}} \{x^2(t,c_i)\} \quad (14)$$

Setelah didapat nilai *chi-square* setiap kata, berikutnya dilakukan pengurutan kata dari nilai *chi-square* tertinggi ke terendah atau memiliki arti dari kata yang terpenting ke yang tidak penting.

3. METODOLOGI PENELITIAN

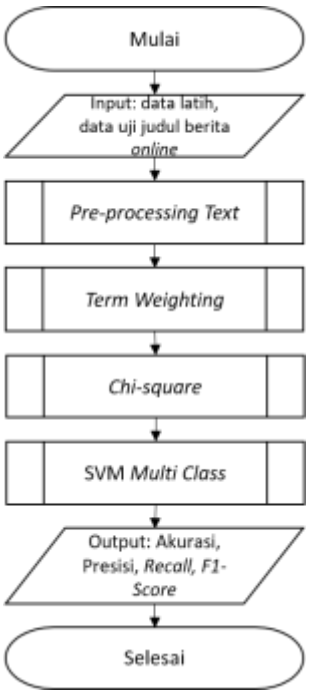
3.1. *Dataset*

Data diambil dari situs portal berita *online* https://www.kompas.com. Judul berita yang dipakai menggunakan berita yang dipublikasikan pada tanggal 22 Februari 2023 sampai 7 April 2023. Label berjumlah 6 yaitu olahraga, kesehatan, berita nasional, kuliner, keuangan, dan teknologi dengan masing-masing label diambil 400 data sehingga total data yang terkumpul adalah 2400 data.

Dari data yang telah terkumpul akan dibagi menjadi 2 bagian, yaitu data latih dan data uji. Pembagian data latih dan data uji memakai perbandingan 70%:30% maka jumlah data latih adalah 1680 dan data uji berjumlah 720 data. Proses pembagian data dilakukan dengan memperhatikan pemerataan label untuk mencegah hasil yang bias, sehingga 70% bagian tiap label sebagai data latih dan 30% sisanya sebagai data uji.

3.2. Alur Proses Sistem

Proses dari sistem yang dilakukan dapat dilihat melalui diagram alir pada Gambar 1.



Gambar 1. Diagram Alir Proses Sistem

Alur proses sistem secara umum membutuhkan data berupa data latih dan data uji judul berita *online*. Selanjutnya data tersebut akan masuk ke dalam 4 tahap utama yaitu:

1. *Pre-processing text*

Dalam *pre-processing text* dilakukan *case folding*, lalu *cleaning* dengan menghilangkan angka, tanda baca dan spasi berlebih, berikutnya *tokenizing*, *stopword removal*, dan terakhir *stemming*.

2. *Term weighting*
   Term weighting yang dilakukan menggunakan TF-IDF.

3. *Chi-square*
   Proses *chi-square* dimulai dari perhitungan $X^2(t,c)$ yaitu *chi-square* kata *t* terhadap kelas *c*, berikutnya $X^2_{max}(t)$ yaitu *chi-square* tunggal kata *t*. Setelah didapat $X^2_{max}(t)$ untuk setiap kata, selanjutnya fitur diurutkan dari nilai *chi-square* tertinggi ke terendah. Proses pemilihan fitur dilakukan dengan mengambil sekian persen dari fitur-fitur terbaik. Banyaknya persen fitur yang diambil atau disebut *threshold* akan menjadi bahan pengujian.
4. SVM *multi class*
   Proses SVM pada *multi class* menggunakan metode *one-againts-all*. Pada *one-againts-all* akan dilakukan iterasi untuk setiap kelas yang ada. Proses diawali dengan mengubah label menjadi 1 jika sama dengan iterasi kelas sekarang dan menjadi -1 jika tidak. Selanjutnya dilakukan pelatihan SVM menggunakan *sequential training* dan pengujian SVM dengan mencari nilai *f(x)*. Suatu data uji akan masuk ke dalam kelas yang memiliki nilai *f(x)* tertinggi. Proses akhir dari SVM adalah melakukan evaluasi.

4. PENGUJIAN DAN ANALISIS

4.1. Pengujian *Kernel* SVM

Pengujian *kernel* SVM menggunakan 4 jenis *kernel* yaitu linear, *polynomial*, RBF, dan *sigmoid*. Pengujian dilakukan dengan mencoba 10 kemungkinan *kernel* SVM dengan berbagai nilai parameter di dalamnya. Parameter SVM lainnya yang digunakan adalah nilai bawaan yaitu bernilai C=1, λ=1, konstanta γ=0,05, $\varepsilon=10^{-10}$, maksimal iterasi=20, dan *threshold chi-square*=100%. Hasil dari pengujian *kernel* SVM dapat dilihat melalui Tabel 2.

Tabel 2. Hasil Pengujian *Kernel*

Kernel	Akurasi	Presisi	*Recall*	*F1-score*
Linear	72,64%	86,26%	72,64%	74,48%
Polynomial (d=2)	92,08%	92,17%	92,08%	92,09%
Polynomial (d=3)	91,11%	91,13%	91,11%	91,03%
RBF(σ=0,5)	20,83%	64,01%	20,83%	12,43%
RBF(σ=1)	52,92%	69,85%	52,92%	49,67%
RBF(σ=2)	35,14%	70,89%	35,14%	30,49%
Sigmoid(σ=0,001)	62,08%	84,35%	62,08%	64,84%
Sigmoid(σ=0,01)	65%	85,35%	65%	67,80%
Sigmoid(σ=0,1)	62,08%	83,49%	62,08%	64,08%
Sigmoid(σ=1)	52,22%	75,68%	52,22%	53,86%

Kernel terbaik dengan *polynomial* derajat 2 yaitu akurasi 92,08%, presisi 92,17%, *recall* 92,08%, dan *f1-score* 92,09%. *Kernel* RBF menjadi *kernel* yang memiliki hasil paling kecil dengan akurasi maksimal adalah 52,92%. *Kernel polynomial* menjadi *kernel* yang paling cocok untuk menangani masalah klasifikasi teks pada penelitian ini dengan semua hasil dari *kernel polynomial* mendapat akurasi tertinggi.

4.2. Pengujian Parameter C

Parameter C merupakan parameter yang mengatur tingkat toleransi kesalahan pada klasifikasi SVM. Dalam *sequential training* nilai parameter C akan menjadi nilai maksimal yang dapat diperoleh dari alpha setiap data latih. Pengujian terhadap parameter C dilakukan dengan mencoba 9 kemungkinan nilai C dari paling kecil bernilai 10^{-10} sampai yang tertinggi bernilai 100. Hasil dari pengujian parameter C terdapat pada Tabel 3.

Tabel 3. Hasil Pengujian Parameter C

C	Akurasi	Presisi	*Recall*	*F1-score*
10^{-8}	57,08%	83,20%	57,08%	55,52%
10^{-6}	57,08%	83,20%	57,08%	55,52%
0,0001	92,08%	92,17%	92,08%	92,09%
0,01	92,08%	92,17%	92,08%	92,09%
1	92,08%	92,17%	92,08%	92,09%
10	92,08%	92,17%	92,08%	92,09%

Akurasi yang didapat ketika parameter C bernilai kurang dari atau sama dengan 10^{-6} bernilai 57,08%. Akurasi 57,08% terjadi karena nilai parameter C yang terlalu kecil mengakibatkan nilai maksimal alpha yang diperoleh juga kecil dan belum mencapai nilai optimalnya. Nilai parameter C terbaik adalah mulai dari 0,001 dan lebih tinggi mendapatkan

akurasi 92,08%, presisi 92,17%, *recall* 92,08%, dan *f1-score* 92,09%.

4.3. Pengujian Parameter *Lambda* (λ)

Parameter λ adalah parameter koefisien pada perhitungan *kernel* dalam pembuatan matriks *hessian*. Penggunaan parameter λ dapat membantu meningkatkan nilai dari perhitungan *kernel* sehingga nilai alpha dan bias yang didapat tidak terlalu kecil. Penggunaan nilai parameter λ yang terlalu tinggi dapat menghilangkan informasi dari perhitungan *kernel* sehingga dapat menurunkan akurasi. Hasil pengujian parameter λ dapat dilihat melalui Tabel 4.

Tabel 4. Hasil Pengujian Lambda

λ	Akurasi	Presisi	*Recall*	*F1-score*
0,01	91,94%	92,02%	91,94%	91,96%
0,1	91,94%	92,02%	91,94%	91,96%
0	91,94%	92,02%	91,94%	91,96%
1	92,08%	92,17%	92,08%	92,09%
5	91,53%	91,62%	91,53%	91,41%
10	90,83%	91,16%	90,83%	90,67%
100	47,36%	82,98%	47,36%	44,85%

Hasil tertinggi yang didapat dari pengujian parameter λ adalah ketika nilai λ=1 yaitu akurasi 92,08%, presisi 92,17%, *recall* 92,08%, dan *f1-score* 92,09%. Ketika parameter λ bernilai 0 dapat berarti hasil perhitungan *kernel* tidak ditambahkan berapun, sedangkan ketika λ bernilai kurang dari 0 berarti penambahan hasil perhitungan *kernel* ditambah dengan nilai yang sangat kecil. Saat parameter λ bernilai 100 akurasi yang didapat jauh menurun menjadi 47,36%, hal ini disebabkan karena nilai λ terlalu besar sehingga informasi asli dari perhitungan *kernel* menjadi tidak berpengaruh.

4.4. Pengujian Parameter Konstanta *Gamma* (γ)

Parameter konstanta γ merupakan parameter yang mengatur laju pembelajaran (*learning rate*) dalam mencari nilai alpha. Nilai konstanta γ yang semakin tinggi akan mempercepat perubahan alpha dan membuat lebih cepat juga nilai alpha menjadi nilai maksimalnya yaitu parameter C, namun terlalu tinggi konstanta γ membuat nilai alpha yang didapat bukanlah nilai yang paling optimal. Nilai konstanta γ yang terlalu kecil membuat perubahan alpha yang terjadi juga sangat kecil, perubahan yang kecil ini dapat memperlambat dalam mencari nilai alpha paling optimal dan membutuhkan iterasi yang banyak juga. Tabel 5 menunjukkan hasil pengujian parameter konstanta γ.

Tabel 5. Hasil Pengujian Konstanta Gamma

Konstanta γ	Akurasi	Presisi	*Recall*	*F1-score*
0,001	92,22%	92,26%	92,22%	92,22%
0,01	92,22%	92,26%	92,22%	92,22%
0,05	92,08%	92,17%	92,08%	92,09%
0,1	92,08%	92,17%	92,08%	92,09%
0,5	90,42%	90,91%	90,42%	90,39%
1	79,31%	85,33%	79,31%	79,08%

Dari hasil pengujian konstanta γ nilai tertinggi diperoleh adalah akurasi 92,22%, presisi 92,26%, *recall* 92,22%, dan *f1-score* 92,22% yaitu ketika nilai konstanta γ sama dengan 0.01 atau lebih kecil. Akurasi menjadi turun drastis ketika nilai konstanta γ bernilai 1, yang mana disebabkan karena sudah dianggap terlalu tingginya nilai konstanta γ sehingga besarnya perubahan alpha juga besar dan tidak dapat menuju nilai optimal.

4.5. Pengujian Parameter *Epsilon* (ε)

Parameter ε merupakan parameter yang menentukan kapan berhentinya suatu iterasi dalam pencarian nilai alpha. Ketika seluruh besar perubahan alpha kurang dari nilai ε maka iterasi pembelajaran berhenti. Nilai ε yang terlalu kecil membuat iterasi semakin sering dilakukan dan banyaknya iterasi dapat mencapai nilai maksimal iterasi. Nilai ε yang terlalu besar dapat membuat iterasi berhenti pada saat pertama kali dijalankan. Pengujian dari parameter ε ditunjukkan pada Tabel 6.

Tabel 6. Hasil Pengujian Parameter Epsilon

ε	Akurasi	Presisi	*Recall*	*F1-score*
0	92,22%	92,26%	92,22%	92,22%
10^{-10}	92,22%	92,26%	92,22%	92,22%
10^{-8}	92,22%	92,26%	92,22%	92,22%
10^{-6}	57,08%	83,20%	57,08%	55,52%
0,0001	57,08%	83,20%	57,08%	55,52%
0,01	57,08%	83,20%	57,08%	55,52%

Nilai ε yang paling kecil diuji adalah 0 yang dapat berarti iterasi akan berhenti ketika besarnya perubahan alpha sama dengan 0 atau tidak ada perubahan. Hasil terbaik yang didapat adalah ketika nilai ε bernilai 10^{-8} atau kurang, dengan akurasi 92,22%, presisi 92,26%, *recall* 92,22%, dan *f1-score* 92,22%. Ketika parameter ε bernilai 10^{-6} atau lebih akurasinya menurun drastis menjadi 57,08%, yang mana disebabkan iterasi pencarian alpha sudah berhenti pada iterasi pertama karena besarnya perubahan alpha

sudah kurang dari ε. Berhentinya pencarian alpha pada iterasi pertama membuat proses pembelajaran SVM tidak terlalu mempelajari data atau sering disebut *underfitting* sehingga akurasi yang didapat jauh menurun.

4.6. Pengujian Parameter Maksimal Iterasi

Parameter maksimal iterasi adalah parameter yang menentukan berapa maksimal iterasi yang dipakai dalam pencarian nilai alpha. Ketika besar perubahan alpha tidak pernah kurang dari ε maka iterasi dapat berhenti saat jumlah iterasi sudah mencapai nilai maksimal. Terlalu kecil dari nilai maksimal iterasi dapat membuat pencarian nilai alpha belum mencapai dan jauh dari titik optimal, sedangkan nilai iterasi yang terlalu besar membuat terlalu lamanya waktu pembelajaran yang dilakukan serta dapat mengakibatkan terjadinya *overfitting*. Tabel 7 menunjukkan hasil pengujian parameter maksimal iterasi.

Tabel 7. Hasil Pengujian Maksimal Iterasi

Maksimal Iterasi	Akurasi	Presisi	*Recall*	*F1-score*
10	92,22%	92,26%	92,22%	92,22%
50	92,22%	92,26%	92,22%	92,22%
100	92,08%	92,26%	92,22%	92,22%
500	91,53%	92,17%	92,08%	92,09%
1000	90,28%	91,78%	91,53%	91,54%

Hasil dari pengujian parameter maksimal iterasi didapat nilai terbaik yaitu akurasi 92,22%, presisi 92,26%, *recall* 92,22%, dan *f1-score* 92,22% ketika maksimal iterasi bernilai 10 sampai 50 iterasi. Ketika maksimal iterasi bernilai 100 atau lebih, akurasi yang didapat terus menurun. Akurasi yang terus menurun dikarenakan semakin sering iterasi yang dijalankan maka semakin dipelajari dari data latih sehingga menyebabkan hasil dari pembelajaran bukanlah hasil yang umum lagi melainkan sudah sangat khusus pada data latih. Terlalu mempelajari dari data latih disebut sebagai *overfitting* yang menyebabkan hasil klasifikasi tidaklah hasil yang umum lagi.

4.7. Pengujian *Threshold Chi-square*

Parameter *threshold chi-square* merupakan parameter dari seleksi fitur *chi-square*. Nilai *threshold* menentukan berapa persen fitur yang akan dipakai dalam klasifikasi menggunakan SVM. Penggunaan seleksi fitur *chi-square* memungkinkan fitur yang dipakai hanyalah fitur-fitur yang penting saja sehingga akurasi yang didapat menjadi lebih tinggi. Terlalu kecil nilai *threshold* yang digunakan membuat banyaknya kehilangan informasi dari data sehingga hasil yang didapat menjadi menurun. Hasil dari pengujian *threshold chi-square* diperlihatkan melalui Tabel 8.

Tabel 8. Hasil Pengujian *Threshold Chi-square*

Threshold Chi-square	Akurasi	Presisi	*Recall*	*F1-score*
10%	71,81%	85,46%	71,81%	73,55%
20%	80,69%	87,70%	80,69%	81,96%
30%	83,47%	88,44%	83,47%	84,43%
40%	87,36%	89,41%	87,36%	87,73%
50%	91,67%	91,79%	91,67%	91,68%
60%	91,94%	92,02%	91,94%	91,93%
70%	92,08%	92,16%	92,08%	92,09%
80%	93,06%	93,11%	93,06%	93,04%
90%	92,78%	92,81%	92,78%	92,76%
100%	92,22%	92,26%	92,22%	92,22%

Hasil menjadi lebih tinggi ketika menggunakan *chi-square* pada *threshold* 80% dan 90% dibandingkan tanpa menggunakan *chi-square* atau *threshold* 100%. Hasil tertinggi yang bisa didapat yaitu pada *threshold* 80% dengan akurasi 92,22%, presisi 92,26%, *recall* 92,22%, dan *f1-score* 92,22%. Penambahan akurasi terjadi ketika menggunakan *chi-square* dengan *threshold* 80%-90% dikarenakan fitur-fitur yang dipakai hanya fitur-fitur yang penting saja sehingga kebisingan data (*data noise*) menjadi berkurang. Ketika menggunakan *threshold* dari 70% sampai 10% hasilnya terus menurun yang mana disebabkan karena terlalu banyak fitur yang dihilangkan sehingga informasi dari data malah hilang.

5. KESIMPULAN

Berdasarkan penelitian yang dilakukan serta hasil dan analisisnya, maka dapat disimpulkan pengaruh dari parameter metode *support vector machine* dalam klasifikasi judul berita *online* menghasilkan hasil tertinggi ketika menggunakan parameter *kernel polynomial* derajat 2, C=1, λ=1, konstanta γ=0,01, $\varepsilon=10^{-8}$, dan maksimal iterasi sebesar 10 yaitu akurasi 92,22%, presisi 92,26%, *recall* 92,22%, dan *f1-score* 92,22%.

Penggunaan seleksi fitur *chi-square* berpengaruh terhadap peningkatan akurasi klasifikasi judul berita *online* menggunakan metode *support vector machine*. Hasil tertinggi didapat pada *threshold chi-square* 80% yaitu akurasi 92,22%, presisi 92,26%, *recall* 92,22%, dan *f1-score* 92,22%. Hasil meningkat ketika menggunakan seleksi fitur *chi-square* terjadi

dikarenakan fitur-fitur yang dipakai menjadi hanya fitur-fitur yang penting saja sehingga kebisingan data dapat dikurangi.

6. DAFTAR PUSTAKA

Alshalabi, H., Tiun, S., Omar, N., & Albared, M. (2013). Experiments on the use of feature selection and machine learning methods in automatic malay text categorization. *Procedia Technology*, *11*, 748–754.

Anoual, E. kah, & Zeroual, I. (2021). The effects of Pre-Processing Techniques on Arabic Text Classification. *International Journal of Advanced Trends in Computer Science and Engineering*, *10*, 41–48. https://doi.org/10.30534/ijatcse/2021/061012021

Chen, J., Huang, H., Tian, S., & Qu, Y. (2009). Feature selection for text classification with Naïve Bayes. *Expert Systems with Applications*, *36*(3), 5432–5435.

Haddi, E., Liu, X., & Shi, Y. (2013). The role of text pre-processing in sentiment analysis. *Procedia Computer Science*, *17*, 26–32. https://doi.org/10.1016/j.procs.2013.05.005

Kannan, S., Gurusamy, V., Vijayarani, S., Ilamathi, J., Nithya, M., Kannan, S., & Gurusamy, V. (2014). Preprocessing techniques for text mining. *International Journal of Computer Science & Communication Networks*, *5*(1), 7–16.

KBBI. (t.t.). *Berita*. Diambil 19 September 2022, dari https://kbbi.kemdikbud.go.id/entri/berita

Liliana, D. Y., Hardianto, A., & Ridok, M. (2011). Indonesian news classification using support vector machine. *World Academy of Science, Engineering and Technology*, *57*, 767–770.

Ling, J., Kencana, I., & Oka, T. B. (2014). Analisis Sentimen Menggunakan Metode Naïve Bayes Classifier Dengan Seleksi Fitur Chi Square. *E-Jurnal Matematika*, *3*(3), 92–99.

Liu, Y., & Zheng, Y. F. (2005). One-against-all multi-class SVM classification using reliability measures. *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, *2*, 849–854.

Maisarah, W. (2021). Framing Advokasi Perkuliahan Tatap Muka di Masa Normal Baru dalam Pemberitaan Kedaulatan Rakyat. *Jurnal Kajian Jurnalisme*, *4*(2), 192–207.

Mukhtar, R., Iqbal, M. J., & Faheem, Z. bin. (2021). Pakistani News Classification Based on Headlines. *Pakistan Journal of Engineering and Technology*, *4*(4), 79–85.

Nugroho, A. S., Witarto, A. B., & Handoko, D. (2003). Support vector machine. *Proceeding Indones. Sci. Meeiting Cent. Japan*.

Petrović, Đ., & Stanković, M. (2019). The Influence of Text Preprocessing Methods and Tools on Calculating Text Similarity. *Facta Universitatis, Series: Mathematics and Informatics*, 973. https://doi.org/10.22190/fumi1905973d

Saigal, P., & Khanna, V. (2020). Multi-category news classification using Support Vector Machine based classifiers. *SN Applied Sciences*, *2*(3), 458.

Shahi, T. B., & Pant, A. K. (2018). Nepali news classification using Naive Bayes, support vector machines and neural networks. *2018 International Conference on Communication Information and Computing Technology (ICCICT)*, 1–5.

Sumadiria, H. (2006). *Jurnalistik Indonesia menulis berita dan feature: Panduan praktis jurnalis profesional*. Simbiosa Rekatama Media.

Vijayakumar, S., & Wu, S. (1999). Sequential Support Vector Classifiers and Regression. *IIA/SOCO*.

Vijayarani, S., & Janani, R. (2016). Text Mining: open Source Tokenization Tools – An Analysis. *Advanced Computational Intelligence: An International Journal (ACII)*, *3*(1), 37–47. https://doi.org/10.5121/acii.2016.3104

Yang, Y., & Pedersen, J. O. (1997). A comparative study on feature selection in text categorization. *Icml*, *97*(412–420), 35.

Zainuddin, N., & Selamat, A. (2014). Sentiment analysis using support vector machine.

2014 international conference on computer, communications, and control technology (I4CT), 333–337.