

Pembuatan Aplikasi Complaint Management System pada Universitas Kristen Petra dengan menggunakan Metode Support Vector Machine Multiclass One vs Rest

Isak Imanuel Leong, Rolly Intan, Leo Willyanto Santoso
Program Studi Informatika Fakultas Teknologi Industri Universitas Kristen Petra
Jl. Siwalankerto 121 – 131 60236
Telp. (031) – 2983455, Fax. (031) – 8417658
Isakleong26@gmail.com, rintan@petra.ac.id, leow@petra.ac.id

ABSTRAK

Salah satu fasilitas di Universitas Kristen Petra untuk menampung aspirasi dari civitas akademika adalah kotak saran. Namun, penggunaan kotak saran dirasa kurang efektif dan objektif karena adanya rasa enggan untuk mengisi dan kurang memiliki bukti yang kuat melalui dokumentasi foto.

Untuk menjawab masalah diatas, dibuat sebuah aplikasi *complaint management system* berbasis Android yang mendukung proses penyampaian aspirasi yang lebih efektif dan objektif melalui penggunaan *mobile device* yang banyak digunakan oleh civitas akademika. Aplikasi juga didukung oleh model klasifikasi yang dibangun dengan menggunakan metode *Support Vector Machine* (SVM) dengan pendekatan berupa *multiclass One vs Rest* untuk mengklasifikasikan biro atau unit terkait untuk mengatasi kesalahan penentuan pihak penerima aspirasi.

Hasil penelitian menunjukkan bahwa penggunaan parameter preprocessing berupa normalisasi, stemming dan stopwords removal mempengaruhi tingkat akurasi model. Jenis kernel terbaik pada SVM untuk klasifikasi teks aspirasi adalah linear dengan nilai $C=1$ yang menghasilkan akurasi sebesar 95.441%. Selain itu, hasil survei terhadap administrator yang mengurus manajemen pengolahan aspirasi menunjukkan bahwa aplikasi yang telah dibuat sudah menjawab kebutuhan dan permasalahan yang ada, dalam hal ini mendukung objektifitas dan efektifitas penyampaian data aspirasi.

Kata Kunci: *Complaint Management System, Support Vector Machine, Aplikasi Android, Klasifikasi Teks, Feature Extraction, TF-IDF.*

ABSTRACT

One of the facilities at Petra Christian University for the accommodation of aspirations from the academic community is the suggestion box. However, the use of suggestion boxes is considered less effective and objective because it has a reluctance to fill out and lacks strong evidence through photo documentation.

To answer the above problem, an Android-based complaint management system was created that supports the aspiration process that is more effective and objective through the use of mobile devices that are more widely used by the academic community. Application supported by the model. Supported by Vector Support Engine (SVM) with the approach of multiclass

One vs. Rest to classify the bureau or related units to overcome errors in determining the recipient of the aspirational recipient.

The results of the research conducted show that preprocessing parameter such as normalization, stemming and stopwords removal affect the accuracy of the model. The best kernel type in SVM for aspiration text classification is linear with value of $C = 1$ which results in an accuracy of 95.441%. In addition, the results of a survey to administrator who manage the management of aspiration shows that the application created already answer the needs and problems, in this case supporting the objectivity and effectiveness of aspiration data delivery.

Keywords: *Complaint Management System, Support Vector Machine, Android Application, Text Classification, Feature Extraction, TF-IDF.*

1. PENDAHULUAN

Penggunaan fasilitas kotak saran di Universitas Kristen Petra untuk menampung aspirasi civitas akademika dirasa kurang efektif dan kurang objektif. Hal ini disebabkan karena adanya rasa enggan untuk mengisi kotak saran. Selain itu, aspirasi cenderung kurang memiliki bukti yang kuat karena hanya berdasarkan tulisan orang tertentu. Masalah tersebut dapat diselesaikan dengan memanfaatkan penggunaan aplikasi untuk mendukung efektifitas dan objektifitas melalui dokumentasi foto aspirasi. Aplikasi tersebut juga dapat mengalihkan aspirasi ke pihak yang tepat untuk menghindari kesalahan penentuan pihak penerima aspirasi.

Penelitian tentang penggunaan *text mining* pada teks *complaint* sudah banyak dilakukan sebelumnya. Salah satunya adalah klasifikasi topik disertai *clustering* pada *complaint tweet* bahasa Indonesia [12]. Hasilnya cukup baik membandingkan kinerja baik pada metode *classification* maupun *clustering*. Namun, masih ada *misclassification* karena ambiguitas dan *error* deteksi kata singkatan (*abbreviations*). Selain itu, ada kesalahan *clustering* pada *event* yang dikelompokkan menjadi *centroid* yang berbeda karena memiliki kata yang berbeda. Penelitian lain adalah menganalisa dan klasifikasi teks laporan berupa opini atau keluhan masyarakat pada *website* LAPOR! [11]. Metode yang digunakan adalah *Support Vector Machine* dan *Self Organizing Maps*. Hasil penelitian menunjukkan akurasi klasifikasi sebesar 66% dan proses *clustering* dapat menemukan topik spesifik dari masing-masing kelas. Penelitian lainnya adalah *Named Entity Recognition* pada teks Nepali. Yang dilakukan adalah *multiclass classification* karena bisa terdapat lebih dari dua *NE tags* dalam *natural language* [1]. Metode yang digunakan adalah *Support*

Vector Machine (SVM) *One vs Rest* yang menghasilkan akurasi untuk jumlah *tokens* terbanyak dengan *precision* 86,85%, *recall* 98,53%, dan *F-score* 92,31%. Dasar masalah pada penelitian ini adalah membangun aplikasi *complaint management system* untuk mendukung objektivitas dan efektifitas penyampaian aspirasi. Metode yang digunakan adalah SVM *multiclass One vs Rest* untuk *multiclass classification* terkait penentuan pihak penindaklanjut aspirasi.

2. DASAR TEORI

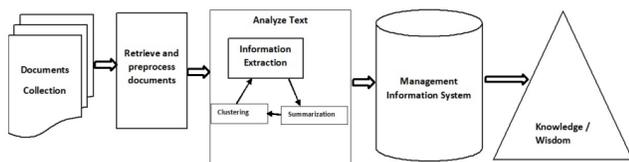
2.1 Complaint Management System

Complaint management system berkaitan erat dengan adanya ketidakpuasan dari pihak tertentu terhadap hal yang dirasakan, dilihat, ataupun digunakan. Ketidakpuasan itu juga berhubungan dengan adanya komplain yang muncul sebagai respon terhadap pelayanan yang tidak memenuhi ekspektasi pemberi komplain [2]. Dalam lingkup universitas, komplain berkaitan dengan respon warga kampus terhadap kualitas pelayanan yang diberikan kampus.

Sebagai suatu sistem, sebuah *complaint management system* didukung adanya interaksi antar *user* baik itu pemberi maupun penerima saran, kritik atau keluhan. Interaksi yang ada membantu meningkatkan layanan pelanggan (*customer service*) yang adalah tujuan utama dari *complaint management* [15]. Melalui tujuan tersebut, adanya *complaint management system* di lingkungan kampus dapat membangun sinergi antara mahasiswa dengan biro / unit yang melayani kebutuhan dari kampus. Sinergi tersebut adalah dalam hal peningkatan kualitas kinerja kampus. Selain itu, kepekaan dan kepedulian mahasiswa terhadap kampusnya juga dapat tumbuh.

2.2 Text Mining

Text mining merupakan usaha yang bertujuan menggali teks untuk mendeteksi informasi yang sebelumnya tidak diketahui [13]. Dilihat dari namanya, *text mining* tentu berbeda dengan *data mining*. Perbedaan mendasar terdapat pada sumber *mining*, dimana *text mining* dilakukan pada teks. Sedangkan, *data mining* melakukan penggalian pola-pola yang menarik dari *database* yang besar [6]. Oleh karena itu, *text mining* lebih fokus ke arah penggalian teks yang bentuknya tidak terstruktur.



Gambar 1. Arsitektur *text mining* [8]

Gambar 1 menunjukkan arsitektur dari proses *text mining* yang dimulai dengan pengumpulan dokumen diikuti *preprocessing* dan ekstraksi informasi. Informasi tersebut nantinya digunakan untuk membangun sebuah *knowledge* baru.

2.3 Preprocessing

Preprocessing menjadi tugas awal sebelum suatu teks dilakukan proses *mining*. Tujuannya untuk menjaga data yang akan diolah bersih dari *noise*, berdimensi lebih kecil, dan lebih terstruktur [7]. Dalam konteks *text mining*, tahapan ini dikenal dengan istilah *text preprocessing*. Umumnya, terdapat beberapa proses

preprocessing seperti *case folding*, *tokenization*, *filtering*, *stopwords removal*, dan *stemming* [14]. *Case folding* mengubah teks menjadi huruf kecil (*lowercase*). *Tokenization* memecah (*splitting*) teks menjadi beberapa bagian (*token*) berupa kata, frasa, angka, dan simbol. *Stopwords removal* mengeliminasi kata yang kurang memiliki makna (misal preposisi dan konjungsi). *Stemming* menghilangkan imbuhan pada kata untuk menemukan bentuk kata dasarnya. *Filtering* melakukan normalisasi seperti *negation handling*, menghapus tanda baca (*punctuation*), angka, *special character*, dan lainnya.

2.4 Feature Extraction

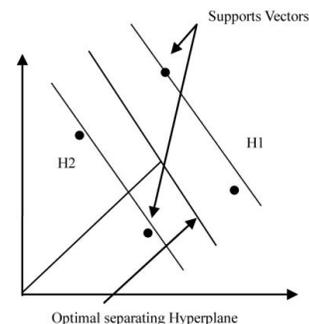
Feature extraction dilakukan setelah tahapan *preprocessing* yang bertujuan mengekstraksi *feature* pada teks. Sebuah data teks dapat terdiri dari istilah (*terms*) yang dikelompokkan ke dalam kalimat dan paragraf [4]. Teknik yang digunakan adalah TF (*Term Frequency*) dan IDF (*Inverse Document Frequency*). Teknik TF menunjukkan bobot banyak kata yang terdapat pada teks. Bobot tersebut tergantung pada distribusi setiap kata dalam suatu dokumen [4]. Sedangkan, teknik IDF menunjukkan seberapa umum atau jarang suatu kata dalam seluruh dokumen. Penggabungan kedua teknik tersebut dilakukan untuk mengambil informasi (*information retrieval*) berkaitan dengan tingkat kepentingan kata atau frasa tertentu pada teks dengan label kelas (*class*) tertentu. Yang dilakukan adalah memberikan bobot (*weight*) pada kata sebagai *term*. Bobot direpresentasikan secara numerik, dimana bobot untuk istilah (*term*) (*i*) adalah sebagai berikut [9]:

$$W_i = \frac{\left(TF_i \times \log \left(\frac{N}{n_i} \right) \right)}{\sqrt{\sum_{i=1}^n \left(TF_i \times \log \left(\frac{N}{n_i} \right) \right)^2}} \quad (2.1)$$

dimana *N* adalah jumlah dokumen dan *n_i* adalah frekuensi dokumen dari *term i*

2.5 Support Vector Machine

Support Vector Machine (SVM) merupakan salah satu metode dengan pendekatan *supervised learning* yang banyak digunakan untuk tugas klasifikasi. Pembelajaran diperoleh dari korelasi antara berbagai data pada *training set* untuk memprediksi kelas dari data baru [3]. Data baru tersebut disebut juga dengan *testing set* yang belum diberi label kelas. Metode SVM dapat juga dikatakan sebagai *binary classifier*. Hal ini sejalan dengan *hyperplane* yang dibangun SVM untuk membagi data menjadi dua kelas (*class*) tertentu [1]. *Hyperplane* diperoleh dengan memaksimalkan *margin* pada masing-masing *class*.



Gambar 1. Arsitektur *support vector machine* [1]

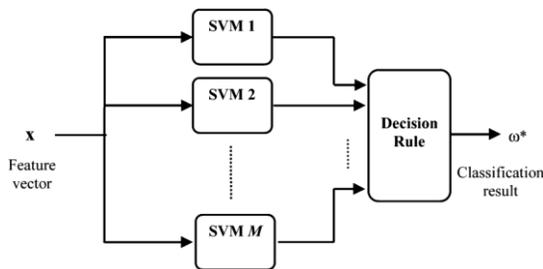
Terdapat sekumpulan titik atau data yang disebut juga dengan *support vector* sedang dipisahkan oleh garis yang disebut dengan *hyperplane*. *Support vector* tersebut merupakan data yang letaknya paling dekat dengan *hyperplane* yang ada. Sebuah *hyperplane* optimal memiliki *margin* terbesar. Yang dimaksud disini adalah mencari titik maksimal dari perhitungan jarak (*margin*) dari *hyperplane* ke *support vector* pada masing-masing *class*. Metode SVM melakukan tugas klasifikasi berdasarkan fungsi berikut:

$$f(x) = \text{sign}(\sum_i a_i y_i \vec{x}_i^T \vec{x} + b) \quad (2.2)$$

2.6 Multiclass One vs Rest Classification

Multiclass classification merupakan klasifikasi dimana setiap data pada *training set* digolongkan ke dalam N kelas yang berbeda. Setiap data tentunya merupakan anggota satu kelas dari total jumlah kelas yang ada. Misalkan untuk kelas A, B, dan C, setiap data yang diklasifikasikan hanya akan memiliki salah satu label kelas diantara ketiga kelas yang ada. Pendekatan ini dibutuhkan ketika melakukan klasifikasi pada lebih dari satu kelas.

One vs Rest dikenal juga dengan *One vs All* merupakan salah satu pendekatan untuk *multiclass classification*. Pendekatan lainnya adalah *One vs One*. Berbeda dengan *One vs One*, pendekatan *One vs Rest* hanya akan membangun *classifier* sesuai jumlah kelas yang akan diklasifikasikan.



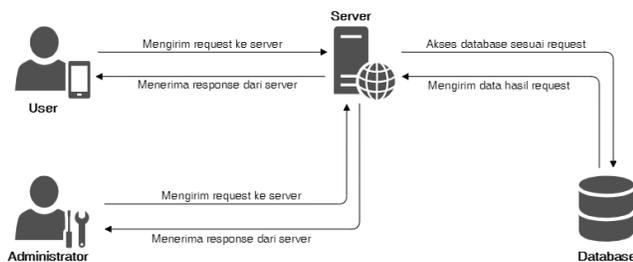
Gambar 2. Arsitektur one vs rest classification [5]

Gambar 2 menunjukkan arsitektur dari pendekatan *One vs Rest classification*. Terdapat satu *classifier* untuk setiap kelas yang berarti ada N *classifier* untuk masalah klasifikasi kelas N [5]. Dalam hal penggunaan SVM sebagai *binary classifier*, setiap *classifier* akan dilatih untuk membedakan satu kelas dari kelas N-1 yang terisa [10].

3. DESAIN SISTEM

3.1 Desain Arsitektur Sistem

Bagian ini akan menjelaskan arsitektur pada sistem *complaint management system* yang akan dibangun. Desain arsitektur sistem dapat dilihat pada Gambar 3.

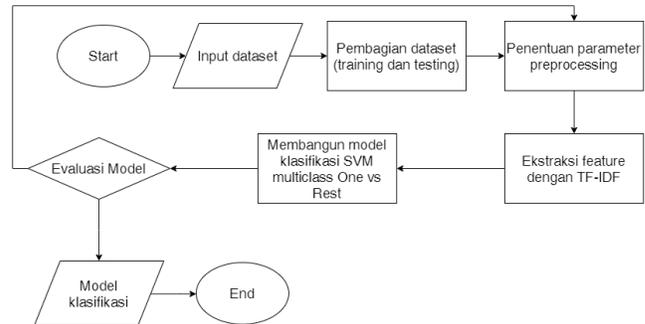


Gambar 3. Desain arsitektur sistem

3.2 Desain Sistem Klasifikasi

Bagian ini akan menjelaskan desain sistem klasifikasi teks aspirasi yang akan dibuat. Data akan terlebih dahulu melalui proses *feature extraction* dengan TF-IDF, dilanjutkan *training* dengan SVM *Multiclass One vs Rest*.

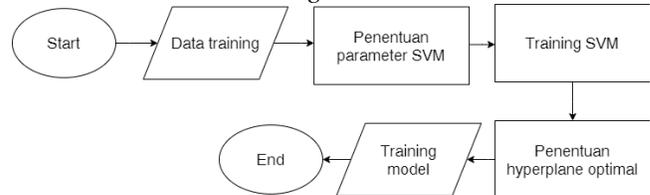
3.2.1 Flowchart Sistem Klasifikasi



Gambar 4. Flowchart sistem klasifikasi

Gambar 4 menunjukkan alur kerja sistem klasifikasi dimulai dengan pengumpulan *dataset* teks aspirasi. Setelah itu, dilakukan penentuan parameter untuk *preprocessing* yang dilanjutkan dengan proses *feature extraction* dengan TF-IDF. Proses selanjutnya adalah membangun model klasifikasi SVM dengan pendekatan *multiclass One vs Rest*. Proses selanjutnya adalah *feature extraction* untuk ekstraksi *feature* yang berguna dari setiap kata pada teks aspirasi. Proses tersebut juga membantu mengubah teks ke angka. Daftar kata yang digunakan pada proses tersebut adalah *dataset* teks *complaint* hasil proses *preprocessing*. Perhitungan bobot TF-IDF dimulai dengan menghitung nilai *Term Frequency* (TF) yang diikuti dengan perhitungan nilai *Inverse Document Frequency* (IDF). Nilai TF menunjukkan frekuensi kemunculan kata yang dihitung tanpa memperhatikan tingkat kepentingan. Sedangkan, perhitungan nilai IDF menunjukkan tingkat kepentingan kata melalui tingkat kemunculan tidak hanya untuk sebuah teks, tetapi seluruh *dataset* teks.

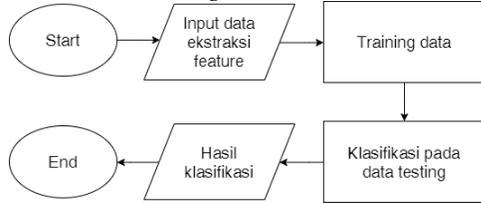
3.2.2 Flowchart Training SVM



Gambar 5. Flowchart training SVM

Gambar 5 menunjukkan *flowchart* proses pelatihan (*training*) pada SVM. Proses *training* diawali dengan penentuan parameter pada metode SVM. Parameter tersebut antara lain pemilihan *kernel*, nilai *cost* dan *gamma* dalam menentukan *hyperplane* yang optimal, serta parameter proses *feature extraction*. Setelah parameter yang optimal diperoleh, proses selanjutnya adalah proses *training* itu sendiri. Tujuannya adalah menemukan *hyperplane* yang optimal untuk membagi data ke dalam kategori kelas. *Output* dari proses *training* SVM adalah *training model* yang digunakan untuk pengujian klasifikasi pada data *testing*.

3.2.3 Flowchart Testing SVM



Gambar 6. Flowchart testing SVM

Gambar 6 menunjukkan *flowchart* dari proses pengujian (*testing*) pada SVM. Proses pengujian menggunakan *training model* untuk membandingkan data baru pada data *testing*. Hasil pembagian data *testing* ke dalam kelas oleh *hyperplane* merupakan *output* dari proses pengujian klasifikasi pada data *testing*.

4. PENGUJIAN SISTEM

Pengujian yang dilakukan terbagi menjadi pengujian fitur pada *complaint management system* dan aplikasi Android, serta model klasifikasi.

4.1 Pengujian Model Klasifikasi

Dataset yang digunakan pada pengujian model klasifikasi adalah hasil survey aspirasi civitas akademik. Jumlah data yang diperoleh sebesar 3360 data yang dibagi menjadi *training set* sebesar 70% dan *testing set* sebesar 30%. Untuk proses *tuning parameter*, *training set* akan dibagi menjadi *validation set* dengan nilai *split* sebesar 5 untuk menghindari terjadinya *overfitting*. Nilai tersebut menghasilkan *training set* sebanyak 4/5 data *training*, dan *validation set* sebanyak 1/5 data *training*. Pengujian model dimulai dengan proses penentuan parameter *preprocessing* dilanjutkan dengan proses *tuning parameter* yang menggunakan *class GridSearchCV*.

4.1.1 Pengujian Preprocessing

Parameter yang diuji adalah berupa normalisasi teks, *stemming*, dan *stopwords removal*. Pengujian dilakukan untuk *kernel linear*, *radial basis function* (RBF) dan *polynomial*.

Tabel 1. Skenario pertama pengujian parameter preprocessing

<i>Kernel</i>	Akurasi <i>Training</i>	Akurasi <i>Testing</i>	<i>Cross Validation</i>
<i>Linear</i>	0.9947	0.93954	0.94944
RBF	0.9996	0.94350	0.94643
<i>Polynomial</i>	1	0.88602	0.89186

Pengujian skenario pertama menunjukkan *kernel linear* memiliki nilai tertinggi untuk skor *cross validation*. Sedangkan, *kernel* RBF memiliki akurasi *testing* tertinggi. Pada pengujian ini, belum ditambahkan parameter *preprocessing* apapun.

Tabel 2. Skenario kedua pengujian parameter preprocessing

<i>Kernel</i>	Akurasi <i>Training</i>	Akurasi <i>Testing</i>	<i>Cross Validation</i>	Perbandingan
<i>Linear</i>	0.99849	0.9468	0.95241	(+) 0.0727
RBF	0.9996	0.95044	0.95113	(+) 0.00693
<i>Polynomial</i>	1	0.90188	0.90303	(+) 0.01585

Pengujian skenario kedua menunjukkan penambahan parameter normalisasi dapat meningkatkan akurasi model yang akan dibangun.

Tabel 3. Skenario ketiga pengujian parameter preprocessing

<i>Kernel</i>	Akurasi <i>Training</i>	Akurasi <i>Testing</i>	<i>Cross Validation</i>	Perbandingan
<i>Linear</i>	0.99404	0.95341	0.95625	(+) 0.00660
RBF	0.9991	0.95341	0.95540	(+) 0.00297
<i>Polynomial</i>	1	0.91873	0.92092	(+) 0.01684

Pengujian skenario ketiga menunjukkan penambahan parameter *stemming* dapat meningkatkan akurasi model yang akan dibangun.

Tabel 4. Skenario keempat pengujian parameter preprocessing

<i>Kernel</i>	Akurasi <i>Training</i>	Akurasi <i>Testing</i>	<i>Cross Validation</i>	Perbandingan
<i>Linear</i>	0.992304	0.5441	0.95577	(+) 0.009
RBF	0.9991	0.95639	0.95749	(+) 0.00297
<i>Polynomial</i>	1	0.93260	0.93407	(+) 0.013875

Pengujian skenario keempat menunjukkan penambahan parameter *stopwords removal* dapat meningkatkan akurasi model yang akan dibangun.

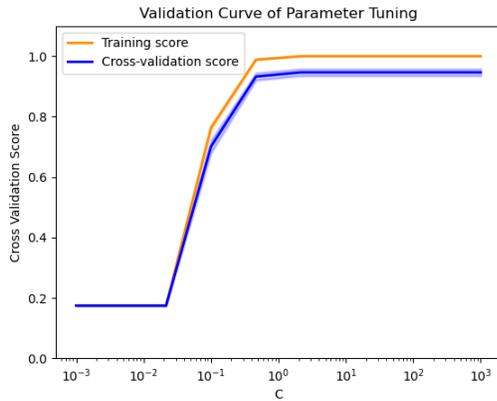
4.1.2 Pengujian Kernel Linear

Parameter yang digunakan untuk *kernel linear* adalah C (*cost*) yang menggambarkan *misclassification*. Range nilai C untuk *kernel linear* yang digunakan adalah (0.0001, 0.001, 0.01, 0.1, 1, 10, dan 100). Sedangkan, parameter pada *class TfidfVectorizer* yang digunakan adalah *ngram_range* ((1, 1), (1, 2), (1,3), (2, 2), (2, 3), (3, 3)), *norm* (l1 dan l2) untuk normalisasi dan *max_df* (0.25, 0.5, 0.75, 1) untuk mengabaikan *features* yang kurang penting.

Tabel 5. Hasil akhir pengujian kernel linear

<i>smooth_idf</i>	<i>sublinear_tf</i>	C	Skor <i>Cross Validation</i>
TRUE	TRUE	100	0.956633
TRUE	FALSE	100	0.957908
FALSE	TRUE	100	0.957483
FALSE	FALSE	100	0.957908
TRUE	TRUE	10	0.956633
TRUE	FALSE	10	0.957908
FALSE	TRUE	10	0.957483
FALSE	FALSE	10	0.957908
TRUE	TRUE	1	0.959184
TRUE	FALSE	1	0.958759
FALSE	TRUE	1	0.960034
FALSE	FALSE	1	0.959184

Tabel 5 menunjukkan hasil akhir dari pengujian parameter terbaik pada *kernel linear*. Hasil pada tabel tersebut menambahkan 2 parameter baru berupa indikator *smooth_idf* dan *sublinear_tf*. Penambahan parameter ini dilakukan setelah menguji penggunaan parameter *ngram_range*, *norm*, dan *max_df* pada pengujian sebelumnya. Hasilnya menunjukkan parameter terbaik untuk nilai C adalah 1. Sedangkan, parameter *feature extraction* terbaik adalah *ngram_range* (1, 2), *norm* (l2), dan *max_df* (0.25), indikator *smooth_idf* (*False*), dan *sublinear_tf* (*True*). Parameter tersebut akan digunakan untuk pengujian *kernel* selanjutnya.



Gambar 7. Validation curve parameter cost

Gambar 7 menunjukkan *validation curve* untuk parameter *cost* pada *kernel linear*. Dilihat dari akurasi *training* dan *cross validation*, nilai *C* yang terlalu rendah membuat model mengalami *underfit*.

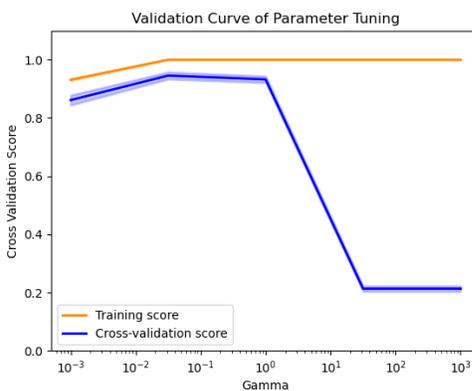
4.1.3 Pengujian Kernel Radial Basis Function (RBF)

Pada *kernel RBF*, terdapat penambahan parameter pada SVM berupa *gamma* sebagai representasi penentuan *decision boundary* pemisah data ke dalam label kelas. *Range* nilai *gamma* yang digunakan sama seperti *range* nilai pada parameter *C*.

Tabel 6. Pengujian akhir nilai gamma untuk kernel rbf

C	Gamma	Skor Cross Validation
100	0.01	0.958759
10	0.1	0.958333
1	1	0.954932
0.1	1	0.948129
0.1	0.0001	0.948129
0.01	1	0.94898
0.001	1	0.94898
0.0001	0.0001	0.948554

Pengujian akhir nilai *gamma* pada *kernel RBF* menunjukkan bahwa skor tertinggi didapat oleh kombinasi nilai *gamma* sebesar 0.01 dan *C* sebesar 100. Hasil tersebut diperoleh dengan cara membandingkan skor tertinggi untuk setiap nilai *C*.



Gambar 8. Validation curve parameter gamma

Gambar 8 menunjukkan *validation curve* untuk parameter *gamma* pada *kernel rbf*. Dapat dilihat bahwa, semakin tinggi nilai *gamma*, maka semakin rendah skor *cross validation*. Hal ini menunjukkan nilai parameter *gamma* yang terlalu tinggi dapat menimbulkan potensi *overfit*.

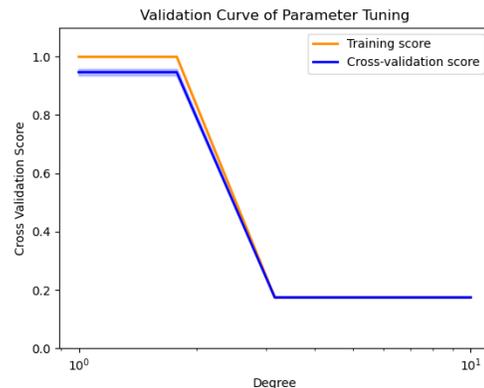
4.1.4 Pengujian Kernel Polynomial

Pada *kernel polynomial*, terdapat penambahan parameter pada SVM berupa *degree* yang mengatur tingkat fleksibilitas *decision boundary* yang dibentuk.

Tabel 7. Pengujian nilai degree untuk kernel polynomial

Degree	Skor Cross Validation
2	0.933248
3	0.914541
4	0.869473
5	0.833759
6	0.776361
7	0.770833
8	0.543793
9	0.221939
10	0.174320

Tabel 7 menunjukkan nilai parameter *degree* terbaik untuk *kernel polynomial* adalah 2. Nilai tersebut dikombinasikan dengan nilai *C* dan *gamma* yang telah diperoleh dari pengujian sebelumnya.



Gambar 9. Validation curve parameter degree

Gambar 9 menunjukkan *validation curve* untuk parameter *degree* pada *kernel polynomial*. Dari gambar tersebut, dapat dilihat adanya potensi *overfit* pada model untuk nilai parameter *degree* yang terlalu tinggi.

4.1.5 Evaluasi Pemilihan Kernel

Tabel 8. Hasil akurasi pada testing set

Kernel	Akurasi
Linear	0.954410307
<i>RBF</i>	0.953419227
<i>Polynomial</i>	0.921704658

Tabel 8 menunjukkan akurasi pada *testing set*. Hasil menunjukkan bahwa *kernel linear* memiliki nilai akurasi tertinggi sehingga akan digunakan untuk membangun model klasifikasi.

Untuk menguji kembali hasil proses *tuning parameter* pada ketiga *kernel* yang ada, dilakukan pengujian menggunakan parameter selain parameter terbaik yang diperoleh dari proses *tuning parameter*. Pengujian tersebut dilakukan untuk 4 skenario yang berbeda. Untuk skenario pengujian pertama sampai ketiga akan menentukan pengaruh penggunaan parameter *feature extraction*. Sedangkan, skenario terakhir akan melihat penggunaan parameter pada *kernel* terbaik.

Tabel 9. Skenario pertama pengujian hasil *tuning parameter*

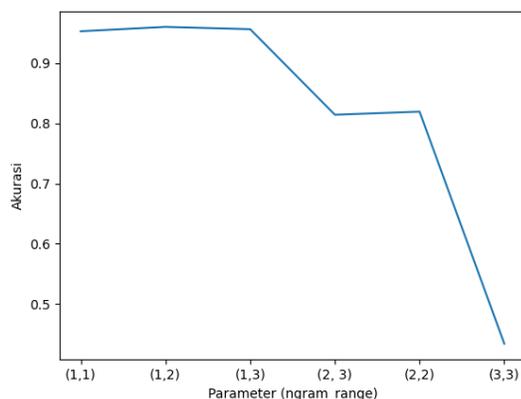
Kernel	Akurasi
Linear	0.325074331
RBF	0.292368681
Polynomial	0.324083250

Tabel 9 menunjukkan penggunaan parameter *ngram_range* (3, 3) dan nilai *max_df* sebesar 1 yang artinya model tidak mengabaikan *features* apapun. Hasilnya menunjukkan adanya penurunan akurasi dibandingkan penggunaan parameter hasil *tuning*. Akurasi tertinggi didapat oleh *kernel linear*.

Tabel 10. Skenario kedua pengujian hasil *tuning parameter*

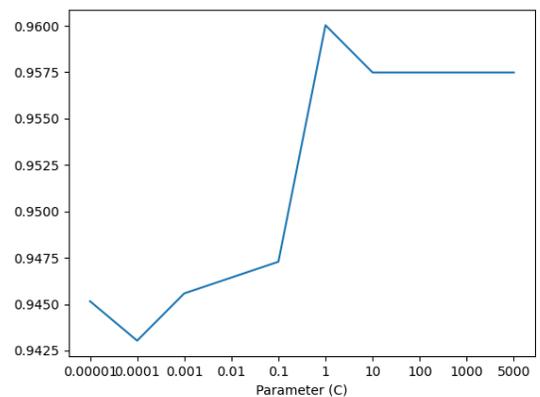
Kernel	Akurasi
Linear	0.447968285
RBF	0.423191278
Polynomial	0.444003964

Tabel 10 menunjukkan peningkatan akurasi untuk penggunaan parameter *max_df* sebesar 0.25, 0.75, dan 0.5 dengan hasil akurasi yang sama. Oleh karena itu, salah satu nilai parameter tersebut digunakan kembali untuk pengujian skenario selanjutnya. Sama seperti pengujian skenario sebelumnya, akurasi tertinggi tetap didapat oleh *kernel linear*.



Gambar 10 Skenario ketiga pengujian hasil *tuning parameter*

Gambar 10 menunjukkan pengujian untuk dapat melihat pengaruh penggunaan parameter *ngram_range*. Pada pengujian ini, *kernel* yang digunakan adalah *kernel linear* karena merupakan *kernel* terbaik yang diperoleh dari pengujian pada skenario sebelumnya. Pengujian yang dilakukan tersebut menggunakan kombinasi nilai parameter *max_df* sebesar 0.25 yang juga diperoleh dari pengujian pada skenario sebelumnya. Hasilnya menunjukkan tingkat akurasi mulai menurun ketika nilai N pada parameter *ngram_range* antara 2 dan 3 (kombinasi *bigram* dan *trigram*, *bigram* saja dan *trigram* saja). Akurasi terendah diperoleh ketika menggunakan nilai N sebesar 3 pada parameter *ngram_range*. Hal ini menunjukkan penggunaan *feature trigram* tidak cocok untuk model.



Gambar 11. Skenario keempat pengujian hasil *tuning parameter*

Gambar 11 menunjukkan skenario terakhir untuk pengujian hasil *tuning parameter*. Pada skenario ini, akan dilihat pengaruh nilai C terhadap tingkat akurasi model. Hasilnya menunjukkan bahwa tingkat akurasi mulai naik seiring dengan meningkatnya nilai C. Akurasi maksimal diperoleh saat nilai parameter C mencapai 1.

4.1.6 Evaluasi Hasil Kuesioner Survei

Kuesioner survei dibuat untuk mengetahui sejauh mana sistem yang dibuat dapat menjawab kebutuhan dan permasalahan yang ada. Kuesioner survei tersebut diberikan kepada salah satu staf *administrator* yang mengurus manajemen pengolahan aspirasi.

Tabel 10. Hasil kuesioner survei

Indikator	Nilai (Sangat Buruk – Sangat Baik)				
	1	2	3	4	5
Kemudahan menggunakan aplikasi					X
Kejelasan tampilan (<i>interface design</i>)					X
Kejelasan penggunaan fitur aplikasi				X	
Kejelasan informasi atau data					X
Menjawab kebutuhan yang ada				X	
Menjawab permasalahan yang ada					X

Tabel 10 menunjukkan hasil penilaian angket pada kuesioner survei terhadap *administrator*. Dari tabel tersebut, dapat dilihat bahwa sistem *complaint management system* yang telah dibuat sudah menjawab kebutuhan dan permasalahan yang ada. Selain itu, tingkat kemudahan penggunaan fitur dan kejelasan data atau informasi yang ditampilkan pada sistem juga sudah baik.

5. KESIMPULAN

Dari hasil pembuatan sistem beserta pengujiannya, dapat diambil kesimpulan, antara lain:

- Penggunaan fitur pada aplikasi *complaint management system* berbasis Android dan *website administrator* berjalan dengan baik.

- Parameter *preprocessing* berupa normalisasi, *stemming*, dan *stopwords* removal mempengaruhi performa akurasi model klasifikasi
- Parameter terbaik metode TF-IDF adalah *ngram_range* (1, 2) atau *unigram* dan *bigram*, nilai *max_df* sebesar 0.25, indikator *smooth_idf* (*false*), dan *sublinear_tf* (*true*).
- Penggunaan *kernel* terbaik untuk metode *Support Vector Machine* (SVM) dalam membangun model klasifikasi adalah *kernel linear* dengan parameter terbaik berupa nilai C (*cost*) sebesar 1.
- Penggunaan parameter *ngram_range* mempengaruhi tingkat akurasi, dimana akurasi menurun untuk nilai N antara 2 dan 3. Akurasi terendah diperoleh ketika menggunakan parameter *ngram_range* (3, 3).
- Penggunaan parameter nilai C pada *kernel* terbaik yaitu *linear* mempengaruhi tingkat akurasi, dimana akurasi mulai naik seiring dengan meningkatnya nilai C. Akurasi maksimal diperoleh saat nilai C mencapai 1.
- Akurasi sistem klasifikasi yang dibangun dengan menggunakan parameter terbaik pada metode SVM sebesar 95.441%.
- Berdasarkan survei terhadap *administrator*, sistem yang dibuat sudah menjawab kebutuhan dan permasalahan yang ada. Tingkat kejelasan informasi dan penggunaan fitur juga sudah baik.

Untuk saran pengembangan selanjutnya, aplikasi diharapkan dapat menerima publikasi data aspirasi dalam bentuk *video* atau file lainnya serta memiliki penambahan fitur lainnya seperti *edit profile*. Aplikasi yang dibuat juga diharapkan dapat berbasis iOS. Selain itu, metode SVM yang telah dibangun juga dapat melakukan proses *retraining* untuk mengatasi kesalahan klasifikasi. Model SVM tersebut juga diharapkan dapat melakukan klasifikasi *multilabel* untuk inputan aspirasi lebih dari satu biro atau unit terkait. .

DAFTAR PUSTAKA

- [1] Bam, S. B., & Shahi, T. B. 2014. Named Entity Recognition for Nepali Text Using Support Vector Machines. *Intelligent Information Management*, 06, 21-29.
- [2] Evantio, Y. B., Rokhmawati, R. I., & Saputra, C. M. 2017. Pengembangan Sistem Informasi E-Complaint Management (Studi Kasus: Batching Plant Produksi Beton P.T. Holcim Indonesia Regional Jawa Timur). *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 2(7), 2801-2810.
- [3] Fabris, F., Magalhaes, J. P., & Freitas, A. A. 2017. A review of supervised machine learning applied to ageing research. *Biogerontology*, 18(2), 171-188. DOI=<https://doi.org/10.1007/s10522-017-9683-y>.
- [4] Fatima, S., & Srinivasu, B. 2017. Text Document categorization using support vector machine. *International Research Journal of Engineering and Technology*, 4(2), 141-147.
- [5] Goswami, A. K., Joshi, H., & Mishra, S. P. 2016. Neural Network Approach for Automatic Landuse Classification of Satellite Images: One-Against-Rest and Multi-Class Classifiers. *International Journal of Computer Applications*, 134(11), 35-42. DOI=<https://doi.org/10.5120/ijca2016908148>.
- [6] Gupta, V., & Lehal, G. 2009. A Survey of Text Mining Techniques and Applications. *Journal of Emerging Technologies in Web Intelligence*, 60-76.
- [7] Jumeilah, F. S. 2017. Penerapan Support Vector Machine (SVM) untuk Pengkategorian Penelitian. *Jurnal Rekayasa Sistem dan Teknologi Informasi*, 1(1), 19-25. DOI=<https://doi.org/10.29207/resti.v1i1.11>.
- [8] Liao, S., Chu, P., & Hsiao, P. 2012. Data mining techniques and applications - A decade review from 2000 to 2011. *Expert Syst. Appl.*, 39, 11303-11311. DOI=<https://doi.org/10.1016/j.eswa.2012.02.063>.
- [9] Mandal, A. K., & Sen, R. 2014. Supervised learning Methods for Bangla Web Document Categorization. *International Journal of Artificial Intelligence & Applications (IJAIA)*, 5(5), 93-105.
- [10] Mehra, N., & Gupta, S. 2013. Survey on Multiclass Classification Methods. *International Journal of Computer Science and Information Technologies (IJCSIT)*, 4(4), 572-576.
- [11] Prajitno, I. S., Megawati, C., Dhini, A., & Hardaya, I. S. 2016. Application of text mining for classification of textual reports: A study of Indonesia's national complaint handling system. *6th International Conference on Industrial Engineering and Operations Management in Kuala Lumpur, IEOM 2016* (pp. 1147-1156). Kuala Lumpur: IEOM Society.
- [12] Pratama, T., & Purwarianti, A. 2017. Topic classification and clustering on Indonesian complaint tweets for bandung government using supervised and unsupervised learning. *2017 International Conference on Advanced Informatics, Concepts, Theory, and Applications (ICAICTA)*, 1-6. DOI=<https://doi.org/10.1109/ICAICTA.2017.8090981>.
- [13] Salloum, S. A., Alhamad, A. Q., Al-Emran, M., & Shaalan, K. 2018. A Survey of Arabic Text Mining. In K. Shaalan, A. E. Hassanien, & F. Tolba (Eds.), *Intelligent Natural Language Processing: Trends and Applications* (pp. 417-431). Springer International Publishing AG.
- [14] Text Preprocessing dengan Python NLTK. 2019. Angri Yulio P. Retrieved April 17, 2020, from <https://devtrik.com/python/text-preprocessing-dengan-python-nltk/>
- [15] What is a Complaint Management System. 2019. Andra Picincu. Retrieved April 17, 2020, from <https://bizfluent.com/facts-6401656-complaint-management-system-.html>