

RESEARCH ARTICLE

Implementasi Metode *Support Vector Machine* (SVM) pada Model Prediksi Rating Obat Berdasarkan Ulasan Pasien

Wisnu Jayanata, Isman Kurniawan* and Erwin Budi Setiawan

Fakultas Informatika, Universitas Telkom, Bandung, 40257, Jawa Barat, Indonesia

* Corresponding author: ismankrn@telkomuniversity.ac.id

Received on 2 August 2023; accepted on 31 August 2023

Abstrak

Pesatnya perkembangan *internet* dan media sosial mengakibatkan membaca ulasan sebelum membeli suatu produk, terutama produk obat menjadi hal yang lumrah. Namun, jumlah ulasan yang banyak dan tersebar mengakibatkan kesulitan dalam melakukan penilaian kualitas produk obat. Oleh karena itu, sistem yang dapat membantu pelanggan dalam menghadapi kendala ini sangat dibutuhkan. Pada pemodelan sistem, digunakan TF-IDF untuk mereduksi fitur dan *Support Vector Machine* sebagai metode klasifikasi. Dilakukan pula *Hyperparameter Tuning* untuk meningkatkan performa sistem. Pada penelitian ini didapat bahwa *polynomial* merupakan kernel SVM yang paling optimal untuk memprediksi rating obat berdasarkan ulasan pasien dengan akurasi mencapai 75.00% dan *f-1 score* sebesar 74.23%.

Key words: analisis sentimen, *Support Vector Machine*, TF-IDF, ulasan, obat

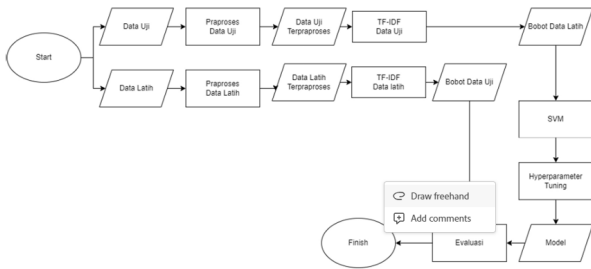
Pendahuluan

Sentiment analysis atau bisa disebut *opinion mining*, adalah bidang studi yang menganalisis opini, sentimen, evaluasi, perilaku dan emosi seseorang terhadap entitas seperti produk, jasa, organisasi, individu, masalah, peristiwa, topik, dan atribut mereka[1]. Perkembangan *Sentiment analysis* ini beriringan dengan berkembangnya teknologi, khususnya internet, yang mengakibatkan melimpahnya *user-generated data* seperti ulasan produk, ulasan layanan/jasa, dan penggunaan media sosial. *Sentiment analysis* sendiri telah diaplikasikan ke berbagai bidang, salah satunya adalah kesehatan. Berdasarkan Organisasi Kesehatan Dunia (WHO)[2], kesehatan dapat dideskripsikan sebagai kondisi kesejahteraan fisik, mental, dan sosial yang lengkap dan bukan sekadar tidak adanya penyakit atau kelemahan. Kondisi kesejahteraan ini sendiri bisa bervariasi antara individu yang mengalami gejala yang sama[3]. Subjektivitas pada kondisi kesehatan membuat kesehatan menjadi topik yang cocok untuk diteliti dengan *Sentiment Analysis*.

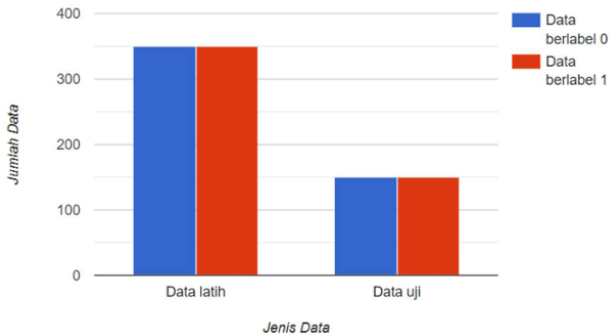
Beberapa penelitian sudah pernah dilakukan terkait analisis sentimen di bidang layanan kesehatan. Ahmet Emre Aladağ dkk melakukan penelitian terhadap pendeteksian niat bunuh diri pada forum[4]. Penelitian ini sendiri dilakukan di forum Reddit, dan sekitar 10.000 kiriman berbahasa Inggris digunakan sebagai dataset. Berbagai metode untuk ekstraksi fitur digunakan dan salah satunya adalah *Term Frequency – Inverse Document Frequency* (TF-IDF). Adapun metode klasifikasi yang digunakan salah satunya adalah *Support Vector Machine* (SVM). Akurasi yang didapat dari sistem yang dibangun dengan SVM lumayan

tinggi dengan bisa mencapai 92%. Penelitian lain dilakukan oleh Abdulaziz dkk[5] untuk mengklasifikasikan kebutuhan pasien dari data yang didapat dari *Electronic Health Records* (EHRs). Salah satu metode klasifikasi yang digunakan SVM dengan kernel yang digunakan adalah *linear*. Sistem yang dibangun mendapat akurasi sebesar 95%.

Namun, analisis sentimen di bidang kesehatan sayangnya masih belum cukup maju. Penelitian yang dilakukan oleh Anastazia Zunic dkk[6] menemukan bahwa *Sentiment Analysis* pada bidang kesehatan masih belum cukup maju dibandingkan dengan bidang lain, dengan penyebab yang masih dipertanyakan. Dengan pesatnya perkembangan internet dan media sosial, ulasan-ulasan terhadap suatu obat bukanlah hal yang asing lagi. Ulasan-ulasan tersebut dapat mempengaruhi keputusan seorang calon pelanggan. Situs ulasan dapat membantu calon pelanggan dalam mengambil keputusan dengan memberikan masukan mengenai obat yang akan mereka beli. Berdasarkan survey yang dilakukan oleh Powers[7], lebih dari 99.9% konsumen mengatakan bahwa membaca ulasan saat melakukan belanja *online*, dan 98% konsumen merasa bahwa ulasan adalah salah satu faktor penting dalam mengambil keputusan berbelanja. Umumnya, ulasan obat terbagi menjadi dua bagian. *Rating* yang merepresentasikan tanggapan konsumen secara numerikal, dan komentar yang memberikan tanggapan konsumen dengan lebih *detail*. Namun, dengan banyaknya ulasan yang bisa dijumpai, tentunya calon pelanggan akan merasa kesulitan memnaca semua ulasan yang ada. Oleh karena itu, dibutuhkan sebuah sistem yang dapat membantu calon pelanggan dalam menghadapi kendala ini.



Gambar 1. Flowchart Pembuatan Program



Gambar 2. Ilustrasi Pembagian Dataset

Penelitian ini bertujuan untuk membuat model prediksi ulasan obat berdasarkan ulasan pelanggan dengan menggunakan *Support Vector Machine*. *Support Vector machine* digunakan dalam pembangunan model dengan menggunakan kernel linear, *RBF* dan *polynomial*. Penelitian ini menggunakan SVM karena SVM memiliki keunggulan dalam mengklasifikasikan data linier dan non-linier melalui penggunaan kernel[8].

Metodologi Penelitian

Dalam penelitian ini, dirancang sistem untuk memprediksi kepuasan pembeli obat berdasarkan ulasan. Alur pembangunan sistem diilustrasikan pada Gambar 1.

Dataset

Dataset yang digunakan dalam penelitian ini adalah *dataset* ulasan obat yang digunakan untuk Winter 2018 Kaggle University Club Hackathon dan sekarang tersedia untuk khalayak publik[9]. Ulasan ini sendiri berasal dari pasien yang berbahasa Inggris. Pada penelitian, hanya digunakan dua bagian dari ulasan tersebut, yaitu *review* dan *rating*. *Review* ini sendiri berisikan ulasan pasien, sementara *rating* merupakan nilai dari 1-10 dimana nilai 10 menandakan kepuasan tertinggi. *Dataset* sendiri sudah terpisah menjadi data latih dan data uji dari sumber. Namun karena besarnya jumlah data dari sumber maka hanya 1000 data yang diproses untuk model yang akan dibangun. Di penelitian ini juga hanya menggunakan *dataset* dengan nilai *rating* yang *ekstrim* yaitu 10 dan 1. *Dataset* diatur sedemikian rupa sehingga data latih dan data uji berbanding 7:3. Data latih dan data uji berisikan *dataset* dengan pembagian data berlabel 0 dan 1 sama rata. Kemudian "*review*" diganti dengan "label", dimana *rating* 10 diganti dengan nilai 1, dan *rating* 1 diganti dengan nilai 0. Pada akhirnya, terdapat 350 data latih berlabel 0, 350 data latih berlabel 1, 150 data latih berlabel 0, 150 data uji berlabel 1. Pembagian *dataset* telah diilustrasikan melalui Gambar 2.

Table 1. Contoh Hasil *Case Folding*

Sebelum	Sesudah
I have been on Tassigna for just over 3 years now (300mg x 2 times a day) Tassigna worked for me within a few weeks I have been in remission for basically the entire 3 years. As for the side effects, I feel sick to my stomach, tired.	i have been on tassigna for just over 3 years now (300mg x 2 times a day) tassigna worked for me within a few weeks i have been in remission for basically the entire 3 years. as for the side effects, i feel sick to my stomach, tired.

Table 2. Contoh Hasil *Tokenization*

Sebelum	Sesudah
i have been on tassigna for just over 3 years now (300mg x 2 times a day) tassigna worked for me within a few weeks i have been in remission for basically the entire 3 years. as for the side effects, i feel sick to my stomach, tired.	'i', 'have', 'been', 'ton', 'tassigna', 'for', 'just', 'over', 'years', 'now', 'mgt', 'x', 'times', 'day', 'tassigna', 'worked', 'Tor', 'me', 'within', 'few', 'weeks', 'i', 'have', 'been', 'in', 'remission', 'for', 'basically', 'the', 'entire', 'years', 'as', 'for', 'the', 'side', 'effects', 'feel', 'sick', 'to', 'my', 'stomach', 'tired'

PraProses

Untuk meningkatkan kualitas model prediksi, dilakukan praproses pada dataset. Berikut tahap-tahap praproses yang telah dilakukan.

a. Case Folding

Langkah pertama pada proses *preprocessing* ini adalah *case folding*, yaitu membuat semua huruf menjadi *lowercase*. Contoh hasil *case folding* bisa dilihat pada Tabel 1 Contoh Hasil *Case Folding*.

b. Tokenization

Tokenization merupakan proses untuk membagi suatu teks menjadi bagian-bagian kecil yang disebut token. Pada proses ini juga dilakukan penghilangan tanda baca seperti tanda titik (.), koma (,), spasi dan karakter angka yang ada pada suatu teks. Contoh hasil *Tokenization* bisa dilihat pada Tabel 2.

c. Stop Word Removal

Pada tahapan ini, akan dilakukan penghilangan beberapa kata yang ada pada teks. Kata yang dihilangkan adalah *stop-words*. *Stop-words* adalah kata yang tidak memiliki makna yang berarti. Contoh hasil *Stop Word Removal* bisa dilihat pada Tabel 3.

d. Stemming

Stemming merupakan proses reduksi kata menjadi bentuk *stem* atau disebut pula dengan kata dasar. Hal ini dilakukan dengan menghilangkan imbuhan pada kata tersebut. Contoh hasil *Stemming* bisa dilihat pada Tabel 4.

e. Detokenization

Detokenization adalah proses mengembalikan token-token yang dibentuk tadi kembali menjadi kalimat utuh. Contoh hasil *Detokenization* bisa dilihat pada Tabel 5.

Table 3. Contoh Hasil *Stop Word Removal*

Sebelum	Sesudah
'i', 'have', 'been', 'on',	'tasigna', 'years
'tasigna', 'for', 'just', 'over',	'mg', 'x', 'times', 'day',
'years',	'tasigna', 'worked', 'within',
'now', 'mg', 'x', 'times', 'at', 'day',	'weeks',
'tasigna', 'worked' 'for', 'me',	'remission', 'basically',
'within',	'entire', 'years',
'a', 'few', 'weeks', 'have',	'side', 'effects', 'feel',
'been', 'in', 'remission', 'fort',	'sick', 'stomach', 'tired'
'basically',	
'the', 'entire', 'years', 'as', 'for',	
'the', 'side', 'effects', 'feel',	
'sick', 'to', 'my', 'stomach', 'tired'	

Table 4. Contoh Hasil *Stemming*

Sebelum	Sesudah
'tasigna', 'years',	'tasigna', 'year',
'mg', 'x', 'times', 'day',	'mg', 'x', 'time', 'day',
'tasigna', 'worked', 'within',	'tasigna', 'work', 'within',
'weeks',	'week', 'remiss', 'basic',
'remission', 'basically',	'entir', 'year',
'entire', 'years',	'side', 'effect', 'feel',
'side', 'effects', 'feel',	'sick', 'stomach', 'tire'
'sick', 'stomach', 'tired'	

Table 5. Contoh Hasil *Detokenization*

Sebelum	Sesudah
'tasigna', 'year',	tasigna year mg x time day
'mg', 'x', 'time', 'day',	tasigna work within week
'tasigna', 'work', 'within',	remiss basic entir year
'week',	side effect feel sick
'remiss', 'basic',	stomach tire
'entir', 'year',	
'side', 'effect', 'feel',	
'sick', 'stomach', 'tire'	

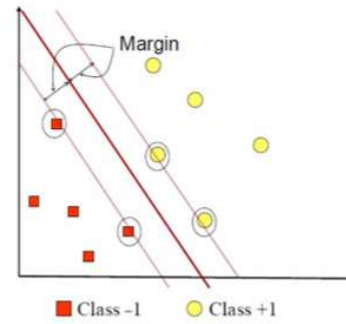
Term Frequency – Inverse Document Frequency (TF-IDF)

Sederhananya TF-IDF adalah perkalian antara *Term Frequency* (TF) dan *Inverse Document Frequency* (IDF), dimana TF adalah jumlah suatu kata dalam satu dokumen dibagi dengan total kata dalam dokumen, dan IDF adalah nilai logaritma dari jumlah dokumen dibagi jumlah kata dalam dokumen itu[10]. Formula TF-IDF adalah:

$$w = df_i \cdot \log \frac{D}{df_i} \tag{1}$$

Dimana df_i adalah banyaknya dokumen yang mengandung kata atau fitur i yang dicari. D adalah banyak dokumen. Berikut contoh perhitungan TF-IDF dengan dua dokumen beserta 6 jenis kata. Perhitungan TF-IDF bisa dilihat dengan detail pada Tabel 6. Pada tahapan ini juga, akan dilakukan feature selection dimana peneliti akan mencari fitur yang memberikan skor maksimum. Fitur sendiri dibatasi antara 100 sampai 1000 fitur.

- Dokumen 1: “tasigna 2 day work”
- Dokumen 2: “feel tasigna sick sick”
- Dokumen 3: “tasigna feel remission”



Gambar 3. *Hyperplane*

Pengembangan Model

Untuk penelitian ini, model prediksi yang dibangun menggunakan metode *Support Vector Machine* (SVM). SVM adalah metode yang diusulkan oleh Vapnik dan rekan lainnya pada 1995 (Boser, Guyon & Vapnik, 1992; Cortes & Vapnik, 1995, dan Vapnik, 1995). SVM memaksimalkan margin antar hyperplane yang memisahkan data. SVM adalah klasifikator linier terhadap parameter, tetapi mudah dikembangkan menjadi nonlinier[11]. *Hyperplane* pemisah terbaik antara kedua kelas dapat ditemukan dengan mengukur margin *hyperplane* tersebut dan mencari titik maksimalnya. Margin adalah jarak antara hyperplane tersebut dengan *pattern* terdekat dari masing-masing kelas. *Pattern* yang paling dekat ini disebut sebagai *support vector*. Pada Gambar 3, didapat satu garis *hyperplane* dengan margin terbaik antar garis dengan data terdekat dari masing-masing kelas. Usaha untuk mencari lokasi *hyperplane* ini merupakan inti dari proses pembelajaran pada SVM[8].

SVM adalah algoritma dengan kelas metode kernel yang berakar pada teori teori statistik. Mesin linier hanya dapat menyelesaikan data linier. Kernel dapat mendorong algoritma SVM untuk menyelesaikan data yang lebih implisit[8]. Fungsi kernel yang digunakan pada penelitian ini adalah sebagai berikut.

$$K_{Linear}(x, y) = x \cdot y \tag{2}$$

$$K_{Polynomial}(x, y) = (x \cdot y + c)^d \tag{3}$$

$$K_{RBF}(x, y) = \exp \left(-\frac{\|x - y\|^2}{2\sigma^2} \right) \tag{4}$$

Pada penelitian ini, dibuat 3 model dimana ketiga model tersebut menggunakan kernel yang berbeda. Tiap model akan dilakukan hyperparameter tuning menggunakan metode *Gridsearch Cross-Validation* (*GridsearchCV*). *GridsearchCV* merupakan salah satu metode untuk melakukan parameter tuning pada tiap model yang telah dibuat dengan mengevaluasi untuk setiap parameter yang dispesifikasikan. Parameter dan nilainilai parameter yang digunakan untuk penelitian ini bisa dilihat pada Tabel 7.

Validasi Model

Pada tahap terakhir, akan dievaluasi kinerja sistem dalam mengklasifikasi data uji. Evaluasi dilakukan dengan membandingkan kelas aktual dan kelas prediksi dengan menggunakan *confusion matrix*. Nilai-nilai yang diperhitungkan dalam *confusion matrix* dapat dilihat pada Tabel 8. Dari nilai yang didapat pada tabel ini, akan dilakukan kalkulasi parameter-parameter berikut.

Table 6. Contoh Pembobotan dengan TF-IDF

Word	TF			df	D/df	IDF	W = TF x IDF		
	D1	D2	D3				D1	D2	D3
Tasigna	1	1	1	3	1	0	0	0	0
2	1	0	0	1	3	0.4771	0.4771	0	0
day	1	0	0	1	3	0.4771	0.4771	0	0
work	1	0	0	1	3	0.4771	0.4771	0	0
feel	0	1	1	2	1.5	0.1760	0	0.1760	0.1760
sick	0	2	0	2	1.5	0.1760	0	0.3520	0
remission	0	0	1	1	3	0.4771	0	0	0.4771

Table 7. Parameter dan Nilai-Nilainya untuk SVM

Parameter	Nilai
C	[0.001,0.01, 0.1, 1, 10, 100, 1000]
Gamma	['scale','auto']
Degree	[2,3,4,5]

Table 8. Confusion Matrix

Aktual	Prediksi	
	Positif	Negatif
Positif	True Positive (TP)	False Negative (FN)
Negatif	False Positive (FP)	True Negative (TN)

a. Akurasi

Akurasi digunakan untuk mengukur kemampuan suatu classifier, dalam kasus ini SVM, dalam mengklasifikasi data dengan tepat. Perhitungan akurasi dilakukan dengan formula berikut[12].

$$Akurasi = \frac{TP + TN}{TP + FN + FP + TN} \times 100\% \quad (5)$$

b. Precision

Precision adalah pengukuran persentase dari data prediksi yang kelasnya sama dengan data aktual. Rumus dari precision adalah sebagai berikut.

$$Precision = \frac{TP}{TP + FP} \times 100\% \quad (6)$$

c. Recall

Recall atau bisa disebut dengan *sensitivity* adalah pengukuran persentase seberapa banyak data aktual yang diprediksi benar oleh sistem. Perhitungan recall dapat dilakukan dengan rumus berikut[12].

$$Recall = \frac{TP}{TP + FN} \times 100\% \quad (7)$$

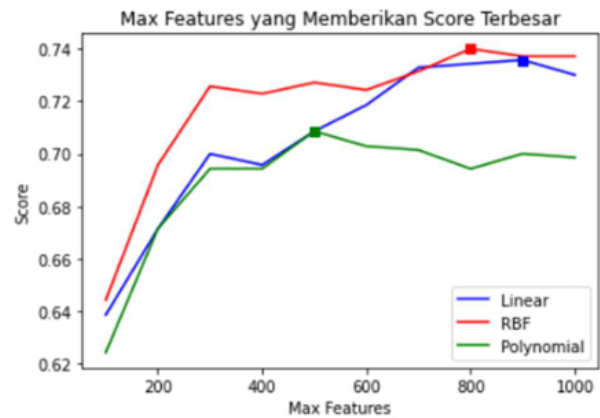
d. F1-Score

Untuk F-1 score dapat didapat dengan menggunakan formula berikut.

$$F-1\ Score = \frac{2(Recall \times Precision)}{Recall + Precision} \times 100\% \quad (8)$$

Table 9. Hasil Seleksi Fitur

Kernel	Max Features	Score
Linear	900	0.7357
RBF	800	0.7400
Polynomial	500	0.7086

**Gambar 4.** Hasil Seleksi Fitur**Hasil dan Pembahasan****Seleksi Fitur**

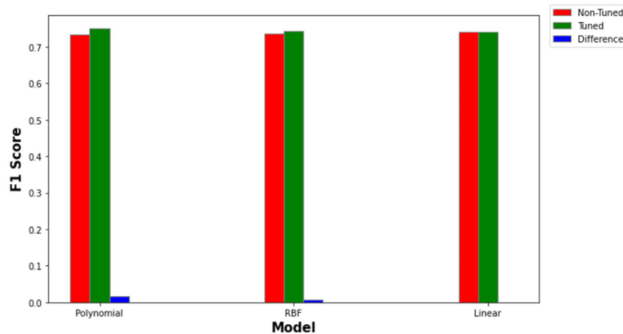
Hasil dari seleksi fitur dapat dilihat pada Gambar 4 dan Tabel 9. Pada proses seleksi fitur, dilakukan pencarian fitur maksimum yang memberikan performa optimal untuk masing-masing model yang telah dibuat. Sebelum dilakukan seleksi fitur, didapat fitur yang didapat dari data latih sebesar 3640 fitur. Untuk model *linear*, didapat model yang menggunakan fitur maksimum 900 memberikan performa paling maksimal. Sementara pada model *polynomial*, model dengan 500 fitur maksimum merupakan model yang paling optimal. Terakhir untuk model *RBF*, model yang memberikan performa paling baik adalah model dengan 800 fitur maksimum.

Hyperparameter Tuning

Pada model *linear* dan *polynomial*, mereka tetap menggunakan nilai *default* untuk parameter *C* yaitu 1 sehingga tidak mengalami perubahan. Sementara untuk model *RBF*, 10 adalah parameter *C* yang paling optimal. Disisi lain, *polynomial* mengalami perubahan parameter *degree* dari nilai defaultnya dari 2 menjadi 3. *Polynomial* dan

Table 10. Hasil Hyperparameter Tuning

Parameter	Linear	Polynomial	RBF
C	1 (1)	1 (1)	10 (1)
Degree	-	2 (3)	-
Gamma	-	Scale (Scale)	Scale (Scale)

**Gambar 5.** Perbandingan Model Sebelum dan Sesudah Dituning

RBF memiliki parameter optimal yang sama untuk gamma yaitu scale, sehingga tidak mengalami perubahan dari nilai *default*. Hasil dari *Hyperparameter Tuning* tertera di Tabel 10.

Keterangan: Nilai di dalam kurung merupakan nilai *default*. Kemudian dilakukan perbandingan *F-1 Score* antara model sebelum tuning dan sesudah *tuning*. Hasil perbandingan tersebut dapat dilihat pada Gambar 5. Untuk model *polynomial*, model yang belum dituning memberikan *F-1 Score* sebesar 73.33%, dan setelah dituning didapat *F-1 Score* sebesar 75%. *Polynomial* mengalami kenaikan sebesar 1.66%. Model linear yang belum dituning mendapat *F-1 Score* sebesar 73.67%, dan setelah dituning mendapat 74.33% sebagai *F-1 Score* sehingga ada kenaikan sebesar 0.66%. *RBF* mendapat *F-1 Score* sebesar 74% untuk model yang belum dituning dan tidak mengalami kenaikan.

Kenaikan performa masing-masing model berkaitan dengan perubahan parameter. Model dengan kernel linear tidak mengalami perubahan performa dikarenakan tidak ada perubahan nilai pada kernel yang relevan. Model *RBF* mengalami peningkatan performa dikarenakan perubahan nilai pada parameter *C*. Model dengan kernel *polynomial* disebabkan oleh perubahan nilai pada parameter *Degree*.

Validasi Model

Dari hasil rangkuman validasi yang dapat dilihat pada Tabel 11, dapat dilihat bahwa *RBF* memiliki performa tertinggi dibanding kernel *polynomial* dan kernel *linear* pada data *train* dengan nilai akurasi, presisi, *recall* dan *f1-score* mencapai sebesar 100%. Disisi lain, *linear* memiliki performa terburuk dengan akurasi 93.71%, presisi 94.57%, *recall* 92.98% dan *f1-score* 93.77% ketika dihadapi data *train*. Ketika berhadapan dengan data *test*, kernel *polynomial* memiliki tingkat akurasi, presisi dan *f1-score* lebih unggul dibanding kernel lain dengan nilai akurasi sebesar 75%. Kernel *linear* juga paling unggul dari segi *recall* dibanding kernel *polynomial* dan *RBF* dengan nilai *recall* sebesar 76.87%.

Setelah proses *tuning*, terjadi perubahan pada parameter *C* dan *Degree*. Parameter *C* digunakan untuk menentukan seberapa besar penalti untuk setiap data yang termisklasifikasi. Nilai parameter *C* yang kecil menyebabkan peningkatan misklasifikasi yang bisa ditoleransi, begitu pula sebaliknya. Sementara *Degree* merupakan parameter yang khusus digunakan oleh kernel *polynomial*. Parameter ini mempengaruhi bentuk akhir dari *Hyperplane* yang dibuat. Setelah dituning, kernel

RBF memiliki peningkatan nilai parameter *C*, hal ini mengakibatkan model prediksi menerima *error* yang lebih sedikit dibanding parameter *default*. Akibatnya, *decision boundary* yang dibuat mejadi lebih sempit. *Decision boundary* yang sempit dapat menyebabkan model prediksi kesulitan untuk memprediksi data yang asing, dengan kata lain, *overfit*. Hal ini dapat diamati dari penurunan akurasi kernel *RBF* dari 100% menjadi 73.17%. Dapat disimpulkan bahwa peningkatan parameter *C* menyebabkan model prediksi dengan kernel *RBF* mengalami *overfit*. Disisi lain, model dengan kernel *polynomial* mengalami perubahan parameter *Degree* menjadi 2 yang menyebabkan bentuk *Hyperplane* menyerupai parabola. Karena akurasi yang didapat oleh *polynomial* lebih besar dari *linear*, bisa disimpulkan bahwa *Hyperplane* yang menyerupai parabola lebih cocok membagi data *rating* obat dibanding *Hyperplane* yang menyerupai garis lurus.

Kesimpulan

Berdasarkan hasil analisis dari model prediksi yang telah dibangun, dapat disimpulkan beberapa poin. Feature selection dengan menggunakan TF-IDF telah berhasil dilakukan dengan fitur direduksi dari 3640 fitur menjadi 900 untuk model linear, 800 untuk model *RBF* dan 500 untuk model *polynomial*. Kemudian *Hyperparameter Tuning* juga berhasil meningkatkan performa model yang dibangun dengan model dengan kernel *polynomial* mengalami peningkatan performa sebesar 1.66%, model dengan kernel *linear* mengalami peningkatan sebesar 0.66%, dan model *RBF* tidak mengalami peningkatan. Implementasi model prediksi *rating* obat berdasarkan ulasan pasien juga telah berhasil didapat dengan performa terbaik didapat oleh kernel *polynomial*. Hal ini dapat dilihat dengan melihat nilai akurasi dan *f1-score* yang didapat dari kernel *polynomial*. Kernel *polynomial* mendapat akurasi sebesar 75.00% dan *f1-score* sebesar 74.23% yang lebih besar dibanding kernel lainnya ketika menggunakan data *test*.

Daftar Pustaka

- Zhao Y, Cen Y. Data Mining Applications with R. Amsterdam: Academic Press; 2014.
- World Health Organization. Constitution of the World Health Organization; 2020.
- Huber M, AnotherAuthor Y, AnotherAuthor Z. How should we define health? BMJ. 2011.
- Aladağ AE, Muderrisoglu S, Akbas NB, Zahmacioglu O, Bingol HO. Detecting Suicidal Ideation on Forums: Proof-of-Concept Study. J Med Internet Res. 2018;20(6).
- Bako AT, AnotherAuthor J, AnotherAuthor K. Using Natural Language Processing to Classify Social Work Interventions. Am J Manag Care. 2021;27(1).
- Zunic A, Corcoran P, Spasic I. Sentiment Analysis in Health and Well-Being: Systematic Review. JMIR Med Inf. 2020;8(1).
- PowerReviews. Survey: The Ever-Growing Power of Reviews; 2021. [Accessed: 18-May-2022]. Available from: <https://www.powerreviews.com/insights/power-of-reviews-survey-2021/>.
- Drajana ICR. Metode Support Vector Machine dan Forward Selection Prediksi Pembayaran Pembelian Bahan Baku Kopra. Ilk J Ilm. 2017;(2):9.
- UCI ML Drug Review dataset — Kaggle;. [Accessed: 21-Feb-2022]. Available from: <https://www.kaggle.com/datasets/jessicali9530/kuc-hackathon-winter-2018>.
- Al-Talib GA, Hassan HS. A Study on Analysis of SMS Classification Using TF-IDF. Int J Comput Networks Commun Secur. 2013;1(5):189–194.

Table 11. Rangkuman Hasil Validasi

Model	Kernel	TP	FP	TN	FN	Accuracy	Precision	Recall	F1-Score
Train									
1	Linear	331	19	325	25	93.71%	94.57%	92.98%	93.77%
2	Polynomial	349	1	348	2	99.57%	99.71%	99.43%	99.57%
3	RBF	350	0	350	0	100.00%	100.00%	100.00%	100.00%
Test									
1	Linear	103	47	119	31	74.00%	68.67%	76.87%	72.54%
2	Polynomial	108	42	117	33	75.00%	72.00%	76.60%	74.23%
3	RBF	105	45	118	32	74.33%	70.00%	76.64%	73.17%

11. Amari S, Wu S. Improving Support Vector Machine Classifiers by Modifying Kernel Functions. *Neural Networks*. 1999;12(6):783–789.
12. Han J, Kamber M, Pei J. *Data Mining: Concepts and Techniques* (The Morgan Kaufmann Series in Data Management Systems). 3rd

ed. Elsevier: Morgan Kauffman; 2011.