

Klasifikasi Pelayanan Kesehatan Berdasarkan Data Sentimen Pelayanan Kesehatan menggunakan *Multiclass Support Vector Machine*

Moh. Heri Setiawan¹, I Gede Aris Gunadi², Gede Indrawan³

Universitas Pendidikan Ganesha

e-mail: ¹heri.setiawan@undiksha.ac.id, ²igedearisgunadi@undiksha.ac.id, ³gindrawan@undiksha.ac.id

Diajukan: 15 November 2022; Direvisi: 15 Februari 2023; Diterima: 2 April 2023

Abstrak

Penelitian ini bertujuan untuk melakukan klasifikasi pada data sentiment pelayanan Kesehatan menggunakan Multiclass SVM pendekatan One versus One (OvO) dengan fitur unigram dan bigram. Sumber data sentimen berasal dari data laporan survei kepuasan pelayanan puskesmas Denpasar 2021 oleh Center for Public Health Innovation (CPHI) FK UNUD. Ekstraksi fitur menggunakan TF-IDF satu term/unigram dan dua term/bigram lalu kemudian diolah menggunakan Support Vector Machine OvO. KFold Cross Validation digunakan untuk membagi data latih dan data tes sekaligus memvalidasi model. Hasil yang didapatkan pada proses pengklasifikasian data train SVM OvO unigram didapatkan skor akurasi 97,09%, presisi 97,97%, recall 96,90%, dan f1-score 97,40%, sedangkan pada SVM OvO bigram didapatkan skor akurasi 97,91%, presisi 98,56%, recall 37,39%, dan f1-score 37,79%. Pada pengklasifikasian data tes didapatkan SVM OvO unigram mendapatkan skor akurasi 68,77%, presisi 73,13%, recall 61,67%, dan f1-score 64,13%, sedangkan SVM OvO bigram mendapatkan skor akurasi 47,92%, presisi 66,41%, recall 37,39%, dan f1-score 37,79%. Perbedaan skor yang jauh pada data train dan data tes dikarenakan adanya overfitting, sehingga perlu adanya seleksi fitur sebelum fitur digunakan sebagai masukan untuk SVM. Selain itu dapat disimpulkan SVM OvO dengan menggunakan fitur unigram memiliki performa lebih baik dibandingkan dengan SVM OvO dengan menggunakan fitur bigram.

Kata kunci: Klasifikasi, Pelayanan Kesehatan, SVM, OvO, TF-IDF.

Abstract

This study aims to classify the sentiment data of Health services using Multiclass SVM with the One versus One (OvO) approach with unigram and bigram features. The sentiment data source comes from the data on the 2021 Denpasar Denpasar Health Center service satisfaction survey report by the Center for Public Health Innovation (CPHI) FK UNUD. Feature extraction using TF-IDF one term/unigram and two term/bigram then processed using Support Vector Machine OvO. KFold Cross Validation is used to share training data and test data as well as validate the model. The results obtained in the process of classifying the SVM OvO unigram train data obtained an accuracy score of 97.09%, precision 97.97%, recall 96.90%, and f1-score 97.40%, while the SVM OvO bigram obtained an accuracy score of 97, 91%, 98.56% precision, 37.39% recall, and 37.79% f1-score. In the classification of test data, SVM OvO unigram got an accuracy score of 68.77%, precision 73.13%, recall 61.67%, and f1-score 64.13%, while SVM OvO bigram got an accuracy score of 47.92%, precision 66.41%, recall 37.39%, and f1-score 37.79%. The large difference in scores on the train data and test data is due to overfitting, so feature selection is needed before the feature is used as input for SVM. In addition, it can be concluded that SVM OvO using the unigram feature has better performance than SVM OvO using the bigram feature.

Keywords: Classification, Health Service, SVM, OvO, TF-IDF.

1. Pendahuluan

Perkembangan teknologi sangat mempengaruhi pengambilan keputusan dalam berbagai aspek kehidupan baik keputusan pribadi maupun keputusan sebuah lembaga. Pengambilan keputusan diambil berdasarkan pertimbangan faktor internal dan eksternal. Salah satu faktor eksternal yang dapat mempengaruhi yaitu penilaian dari pelanggan atau pengguna. Hal ini juga termasuk lembaga pelayanan kesehatan untuk memperhatikan *feedback* dari pengguna berupa saran dan kritikan untuk pengembangan

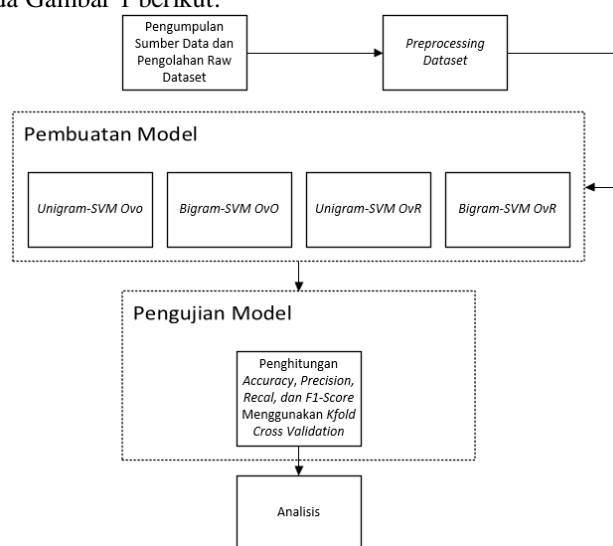
lembaga. Pengolahan data saran dan kritikan diperlukan sebagai bentuk usaha membantu pemerintah dalam mengupayakan penguatan layanan kesehatan [1]. Hasil dari pengolahan data saran dan kritikan berupa klasifikasi pelayanan kesehatan sehingga dapat dipergunakan oleh pengelola lembaga.

Permasalahan klasifikasi berdasarkan data teks dapat diselesaikan salah satunya dengan menggunakan metode *Support Vector Machine* (SVM). Hasil klasifikasi menggunakan SVM cukup memuaskan pada teks, hal ini sesuai dengan salah satu penelitian dari Sujadi pada analisis sentimen pengguna media sosial *twitter* terhadap wabah covid 19 dengan SVM dan *Naïve Bayes Classifier* (NBC) dimana SVM menghasilkan nilai akurasi 81,6% sedangkan NBC 78,3% [2]. Selain itu juga penelitian lain juga menghasilkan nilai akurasi SVM sebesar 70,71%, sedangkan SVM sebesar 80,81% [3]. Kelebihan SVM dibandingkan dengan metode lainnya adalah kemampuan generalisasi, *curse of dimensionality*, landasan teori, dan *feasibility* [4]. Pada penelitian [2] dilakukan pendekatan *One versus One* (OvO) dengan jumlah label sebanyak 3 kelas yaitu negatif, netral, dan positif, sedangkan pada penelitian [3] penggunaan SVM menggunakan klasifikasi biner (OvO) yaitu kelas berlabel positif dan negatif.

Data sentimen pelayanan kesehatan memiliki 6 label berdasarkan bidang pelayanan kesehatannya. Oleh karena itu pendekatan SVM yang digunakan pada penelitian ini akan mencoba melihat performa pendekatan OvO. Selain melihat performa tersebut pada penelitian ini juga akan membandingkan pengaruh fitur *unigram* dan *bigram Term Frequency-Inverse Document Frequency* (TF-IDF) dengan pendekatan OvO.

2. Metode Penelitian

Alur proses penelitian dalam klasifikasi data sentimen pelayanan kesehatan menggunakan SVM dapat dilihat seperti pada Gambar 1 berikut:



Gambar 1. Metode Penelitian

2.1. Pengumpulan Sumber Data dan Pengolahan Raw Dataset

Dataset pada penelitian ini didapatkan dari laporan hasil survei dari lembaga *Center for Public Health Innovation* (CPHI) FK UNUD [5]. Dari laporan tersebut didapatkan sentimen dari pengguna lembaga kesehatan di daerah Denpasar. Saran dan kesimpulan yang didapatkan dari laporan tersebut masih belum dalam format yang dibutuhkan untuk digunakan sebagai inputan model SVM, sehingga perlu perubahan format agar sesuai dengan kebutuhan. Dataset yang dibuat dilakukan pelabelan kelas secara manual oleh ahli. Contoh pelabelan kelas sesuai dengan bidangnya tampak seperti pada Tabel 1 di bawah:

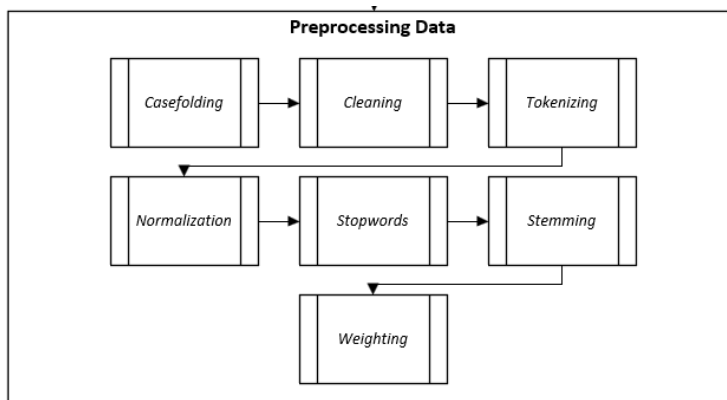
Tabel 1. Pelabelan Kelas

No	Saran/Kritik	Label
1	Pelayanan agar lebih ditingkatkan	Pelayanan
2	Lahan parkir diperluas	Sarana dan Prasarana
3	Loket diperbanyak, ada tempat bermain untuk anak agar tidak bosan	Sarana dan Prasarana

Perubahan format merupakan proses penginputan saran/kritik ke dalam *Microsoft Excel* dan disimpan dengan format “.xlsx”. Berdasarkan hasil perubahan format didapatkan jumlah saran/kritik yaitu 1031 baris kalimat dengan jumlah masing-masing kelas yaitu: Pelayanan 274 baris, Sarana dan Prasarana 240 baris, Sumber Daya Manusia 156 baris, Administrasi dan Manajemen 104, Peralatan 29, Netral 228.

2.2. Preprocessing dataset

Tahapan ini adalah proses pengolahan dataset sebelum digunakan sebagai data latih model SVM dengan tujuan akhir yaitu mengubah data teks menjadi data numerik. Adapun Langkah-langkah dalam prosesnya adalah seperti pada berikut:



Gambar 2. Tahapan *preprocessing dataset*

2.2.1. Casefolding

Tahapan ini bertujuan untuk mengubah huruf kapital menjadi huruf kecil pada kalimat. Contoh hasil *casefolding* menggunakan data Tabel 2 adalah sebagai berikut:

Tabel 2. Hasil Proses *Casefolding*

No	Saran/Kritik
1	pelayanan agar lebih ditingkatkan
2	lahan parkir diperluas
3	loket diperbanyak, ada tempat bermain untuk anak agar tidak bosan
4	dokter spesialis ditingkatkan

2.2.2. Cleaning

Proses *cleaning* merupakan proses penghapusan format teks yang tidak diperlukan antara lain penghapusan tab, *new line*, *back slice*, kode ASCII, angka, tanda baca, spasi berlebih, dan penghapusan 1 (satu) karakter. Contoh proses *cleaning* dengan data pada Tabel 2. Hasil Proses *Casefolding* adalah sebagai berikut:

Tabel 3. Hasil Proses *Cleaning*

No	Saran/Kritik
1	pelayanan agar lebih ditingkatkan
2	lahan parkir diperluas
3	loket diperbanyak ada tempat bermain untuk anak agar tidak bosan
4	dokter spesialis ditingkatkan

2.2.3. Tokenizing

Pada tahapan ini bertujuan untuk memenggal sebuah kalimat menjadi sebuah kata per kata. Pada proses ini memanfaatkan *library* dari *Natural Language Toolkit* (NLTK) [6] untuk memproses dataset menjadi sebuah token. Hasil proses *tokenizing* dengan data pada Tabel 34 adalah sebagai berikut:

Tabel 4. Hasil Proses *Tokenizing*

No	Saran/Kritik
1	[pelayanan, agar, lebih, ditingkatkan]
2	[lahan, parkir, diperluas]
3	[loket, diperbanyak, ada, tempat, bermain, untuk, anak, agar, tidak, bosan]
4	[dokter, spesialis, ditingkatkan]

2.2.4. Normalization

Tahap *normalization* merupakan tahapan perubahan kata-kata singkatan, kata tidak baku, dan akronim menjadi kata baku sesuai dengan singkatan, kata baku, dan kepanjangan dari kata akronimnya. Pada tahapan ini menggunakan daftar kata yang sering digunakan dalam kalimat pesan singkat. Daftar kata berisikan 1029 kata yang sudah diberikan kata padanannya sesuai dengan kata baku berbahasa Indonesia [7]. Hasil proses *normalization* pada Tabel 45 adalah sebagai berikut:

Tabel 5. Hasil Proses *Normalization*

No	Saran/Kritik
1	[pelayanan, agar, lebih, ditingkatkan]
2	[lahan, parkir, diperluas]
3	[loket, diperbanyak, ada, tempat, bermain, untuk, anak, agar, tidak, bosan]
4	[dokter, spesialis, ditingkatkan]

2.2.5. Stopwords

Stopwords merupakan proses untuk menghapus kata-kata yang tidak digunakan misalnya kata di, nggak, tadi, datang, dsb. Penghapusan menggunakan *library* dari NLTK [6] dan sumber *corpus* menggunakan hasil penelitian dari Tala [8] serta ditambah dengan beberapa kata secara manual. Hasil proses *stopwords* pada Tabel 56 adalah sebagai berikut:

Tabel 6. Hasil proses *Stopwords*

No	Saran/Kritik
1	[pelayanan, ditingkatkan]
2	[lahan, parkir, diperluas]
3	[loket, diperbanyak, bermain, anak, bosan]
4	[dokter, spesialis, ditingkatkan]

2.2.6. Stemming

Pada tahapan ini dilakukan proses perubahan kata yang memiliki imbuhan diubah menjadi kata dasar. Proses ini menggunakan *library PySastrawi* [9] dalam pemrosesannya. Selain *library* tersebut digunakan juga *library swifter* [10] yang berfungsi untuk membantu mempercepat proses *stemming* [11]. Contoh hasil tahapan dari *casefolding* hingga *stemming* dapat dilihat pada Tabel 77 berikut:

Tabel 7. Hasil proses *casefolding* hingga *stemming*

No	Saran/Kritik
1	[layan, tingkat]
2	[lahan, parkir, luas]
3	[loket, banyak, main, anak, bosan]

2.2.7. Weighting

Pembobotan atau *weighting* merupakan proses perubahan data token teks menjadi data numerik. Metode yang digunakan untuk perubahan data teks menjadi numerik adalah metode TF-IDF. Persamaan TF-IDF yang digunakan adalah sebagai berikut [12]–[15]:

$$W_{i,j} = tf_{i,j} \times idf_i \tag{1}$$

Dimana $W_{i,j}$ = Bobot TF-IDF, idf_i = *Inverse Document Frequency*, dan $tf_{i,j}$ = *Term Frequency*. Berdasarkan hasil proses ini didapatkan fitur untuk *unigram* sebesar 645 fitur dan untuk *bigram* sebesar 1902 fitur, sehingga ukuran data inputan untuk model SVM adalah (1031, 645) untuk fitur *unigram* dan (1031, 1902) untuk fitur *bigram*. Contoh perhitungan bobot *unigram* menggunakan TF-IDF pada Tabel 78 adalah sebagai berikut:

Tabel 8. Hasil Proses *Weighting*

Term (t)	Nilai TF				df	d	idf _i	W _{ij}			
	t ₁	t ₂	t ₃	t ₄				W(t ₁)	W(t ₂)	W(t ₃)	W(t ₄)
layan	1	0	0	0	1	4	0,602	0,602	0	0	0
tingkat	1	0	0	1	2	4	0,301	0,602	0	0	0,602
lahan	0	1	0	0	1	4	0,602	0	0,602	0	0
parkir	0	1	0	0	1	4	0,602	0	0,602	0	0
luas	0	1	0	0	1	4	0,602	0	0,602	0	0
loket	0	0	1	0	1	4	0,602	0	0	0,602	0
banyak	0	0	1	0	1	4	0,602	0	0	0,602	0
main	0	0	1	0	1	4	0,602	0	0	0,602	0
anak	0	0	1	0	1	4	0,602	0	0	0,602	0
bosan	0	0	1	0	1	4	0,602	0	0	0,602	0
dokter	0	0	0	1	1	4	0,602	0	0	0	0,602
spesialis	0	0	0	1	1	4	0,602	0	0	0	0,602

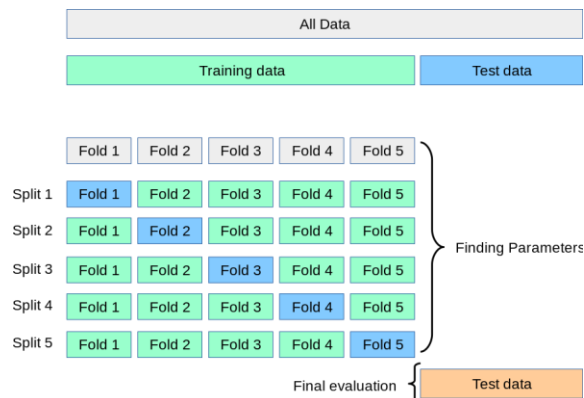
2.3. Pembuatan Model

Pada tahapan ini akan dibuat model SVM menggunakan pendekatan OvO *unigram* dan *bigram*, serta model SVM menggunakan pendekatan OvR *unigram* dan *bigram*. Pembuatan model menggunakan *library scikit-learn* [16] dengan modul yang digunakan adalah SVM untuk model SVM OvO dan *LinearSVC* untuk model SVM OvR. Pengaturan parameter pada SVM OvO antara lain parameter $C=10$, $kernel='linear'$, $decision_function_shape='ovo'$, $max_iter=10000$. Pada model SVM OvR pengaturan parameternya antara lain parameter $C=10$, $multi_class='ovr'$, dan $max_iter = 10000$.

Perbedaan pendekatan antara SVM OvO dengan SVM OvR terlepas pada *library* yang digunakan adalah pada bagian penentuan keanggotaan kelas. Jika pada SVM OvO penentuan keanggotaan kelas didasarkan pada *vote strategy* (strategi voting) dan jika terdapat jumlah *vote* yang sama maka hasil klasifikasi ditentukan pada jumlah *vote* terbanyak dengan *index* terkecil [17], sedangkan pada SVM OvR penentuan keanggotaan ditentukan berdasarkan nilai tertinggi keanggotaan, jika terdapat nilai yang sama maka akan ditentukan berdasarkan index terkecil dari nilai yang sama tersebut [18].

2.4. Pengujian Model

Pengujian model menggunakan metode *KFold Cross Validation* dimana dataset akan dipecah menjadi 2 bagian yaitu data latih dan data tes/validasi. Pada data latih akan dipecah kembali menjadi 5 lima *fold*. Ilustrasi metode *KFold Cross Validation* seperti pada Gambar 3 berikut [16]:



Gambar 3. Ilustrasi metode *KFold Cross Validation*

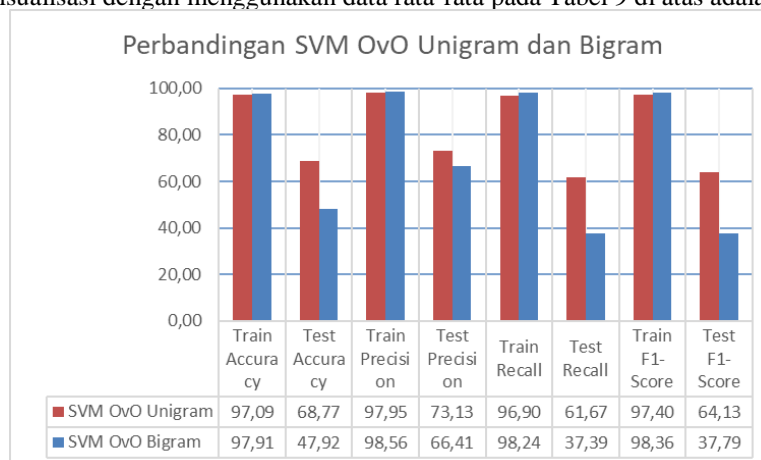
3. Hasil dan Pembahasan

Hasil model dan pengujian menggunakan *KFold Cross Validation* didapatkan beberapa nilai antara lain *accuracy*, *precision*, *recall*, dan *f1-score*. Adapun hasil dari model SVM OvO baik *unigram* maupun *bigram* dapat dilihat pada Tabel 99 dan pada Gambar 4 berikut:

Tabel 9. Hasil Pengujian Model SVM OvO *Unigram* dan *Bigram*

Model	KFold	Train Acc	Test Acc	Train Prec	Test Prec	Train Recall	Test Recall	Train F1	Test F1
SVM OvO <i>Unigram</i>	ke-1	98,18	70,05	98,78	71,17	97,71	63,00	98,23	65,24
	ke-2	97,21	78,16	98,07	82,95	96,83	70,68	97,42	73,78
	ke-3	96,85	72,33	97,73	73,91	96,63	64,21	97,16	66,59
	ke-4	96,48	74,27	97,45	81,19	96,38	68,56	96,90	71,89
	ke-5	96,73	49,03	97,71	56,44	96,97	41,92	97,32	43,15
Rata-Rata		97,09	68,77	97,95	73,13	96,90	61,67	97,40	64,13
SVM OvO <i>Bigram</i>	ke-1	97,94	47,34	98,57	53,12	98,32	34,47	98,41	32,43
	ke-2	97,70	51,94	98,41	87,86	98,11	42,63	98,22	45,89
	ke-3	97,94	51,94	98,57	69,29	98,25	41,09	98,38	43,38
	ke-4	98,18	53,88	98,74	87,10	98,44	43,70	98,55	46,56
	ke-5	97,82	34,47	98,50	34,67	98,09	25,07	98,24	20,68
Rata-rata		97,91	47,92	98,56	66,41	98,24	37,39	98,36	37,79

Hasil visualisasi dengan menggunakan data rata-rata pada Tabel 9 di atas adalah sebagai berikut:



Gambar 4. Visualiasi hasil pengujian model SVM OvO *Unigram* dan *Bigram*

Berdasarkan visualisasi pada Gambar 4 terlihat bahwa pada keseluruhan data train didapatkan skor diatas 90%, tetapi hal ini tidak diikuti dengan skor data tesnya yang hanya berkisar diantara 37,39% - 73,13%. Hal ini disebut dengan *overfitting* [19]–[21], dimana didapatkan skor yang baik hanya pada data train saja, sedangkan pada data tes mengalami penurunan yang signifikan. Permasalahan ini dapat diteliti kembali dengan penambahan seleksi fitur pada proses preprocessing data, sehingga fitur yang didapatkan setelah proses weighting menggunakan TF-IDF dapat dipilih berdasarkan pengaruh fitur tertinggi yang hanya digunakan sebagai masukan ke dalam algoritma SVM [22]–[25].

4. Kesimpulan

Penggunaan metode *Multiclass SVM* dapat menyelesaikan permasalahan klasifikasi dengan dataset berupa teks. Sebelum digunakan data teks harus melalui proses *praprocessing data* sehingga dapat digunakan sebagai inputan untuk model SVM OvO. Pada penelitian ini dataset sebagai inputan diberikan 2 fitur yaitu *unigram* dan *bigram* yang kemudian akan digunakan sebagai masukan untuk model SVM OvO. Berdasarkan hasil pengujian menggunakan lima *fold cross validation* didapatkan beberapa kesimpulan yaitu: model SVM OvO unigram memiliki performa lebih baik dengan skor akurasi 68,77%, presisi 73,13%, recall 61,67%, dan f1-score 64,13%, sedangkan pada model SVM OvO bigram didapatkan skor akurasi 47,92%, presisi 66,41%, recall 37,39%, dan f1-score 37,79%.

5. Saran

Berdasarkan hasil skor data train didapatkan bahwa model SVM OvO sudah memiliki performa yang sangat baik, tetapi pada skor data tes model SVM OvO masih kurang baik. Hal ini dikarenakan terjadinya *overfitting*, sehingga perlu dilakukan penelitian selanjutnya dengan menambahkan proses seleksi fitur setelah didapatkan vektor dari proses TF-IDF.

Daftar Pustaka

- [1] Rokom, “Kuatkan Layanan Kesehatan, Pemerintah Lakukan Lima Upaya Secara Simultan,” Nov. 04, 2016. <https://sehatnegeriku.kemkes.go.id/baca/umum/20161104/2918732/kuatkan-layanan-kesehatan-pemerintah-lakukan-lima-upaya-secara-simultan/> (accessed Aug. 30, 2022).
- [2] H. Sujadi, S. Fajar, and C. Roni, “Analisis Sentimen Pengguna Media Sosial Twitter Terhadap Wabah Covid-19 Dengan Metode Naïve Bayes Classifier Dan Support Vector Machine,” *INFOTECH journal*, vol. 3, no. 1, pp. 1689–1699, 2021, [Online]. Available: <http://journal.unilak.ac.id/index.php/JIEB/article/view/3845%0Ahttp://dSPACE.uc.ac.id/handle/123456789/1288>
- [3] D. A. Pangestu, “Analisis Sentimen Terhadap Opini Publik Tentang Kesehatan Mental Selama Pandemi Covid-19 di Media Sosial Twitter Menggunakan Naive Bayes Classifier Dan Support Vector Machine,” Universitas Islam Indonesia, 2020. [Online]. Available: https://dSPACE.uin.ac.id/bitstream/handle/123456789/28594/15611151_Deinda_Afiya_Pangestu.pdf?sequence=1&isAllowed=y
- [4] A. S. Nugroho, A. B. Witarto, and D. Handoko, “Support Vector Machine Teori dan Aplikasinya dalam Bioinformatika,” *Kuliah Umum IlmuKomputer. Com*, 2003.
- [5] Center for Public Health Innovation (CPHI) FK UNUD, “Laporan Survei Kepuasan Puskesmas Denpasar,” Denpasar, 2021.
- [6] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*. O’Reilly Media, Inc., 2009.
- [7] ramaprakoso, “analisis-sentimen.” <https://github.com/ramaprakoso/analisis-sentimen/blob/master/kamus/kbba.txt>, 2017. Accessed: Nov. 04, 2022. [Online]. Available: <https://github.com/ramaprakoso/analisis-sentimen/blob/master/kamus/kbba.txt>
- [8] F. Z. Tala, “A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia,” 2003.
- [9] H. Amal Robbani, “PySastrawi.” <https://github.com/har07/PySastrawi>, 2018. Accessed: Nov. 03, 2022. [Online]. Available: <https://github.com/har07/PySastrawi>
- [10] J. Carpenter, “Swifter.” <https://github.com/jmcarpenter2/swifter>, Aug. 17, 2022.
- [11] B. E. Howard *et al.*, “SWIFT-Review: a text-mining workbench for systematic review,” *Syst Rev*, vol. 5, no. 1, p. 87, Dec. 2016, doi: 10.1186/s13643-016-0263-z.
- [12] K. Sparck Jones, “A Statistical Interpretation of Term Specificity and Its Application In Retrieval,” *Journal of Documentation*, vol. 28, no. 1, pp. 11–21, Jan. 1972, doi: 10.1108/eb026526.
- [13] S. Robertson, “Understanding inverse document frequency: on theoretical arguments for IDF,” *Journal of Documentation*, vol. 60, no. 5, pp. 503–520, Oct. 2004, doi: 10.1108/00220410410560582.

-
- [14] J. Leskovec, A. Rajaraman, and J. D. Ullman, "Mining of Massive Datasets."
- [15] W. Uther *et al.*, *Encyclopedia of Machine Learning*. Boston, MA: Springer US, 2010. doi: 10.1007/978-0-387-30164-8.
- [16] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011, [Online]. Available: <http://scikit-learn.sourceforge.net>.
- [17] Chih-Wei Hsu and Chih-Jen Lin, "A comparison of methods for multiclass support vector machines," *IEEE Trans Neural Netw*, vol. 13, no. 2, pp. 415–425, Mar. 2002, doi: 10.1109/72.991427.
- [18] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer: Microsoft Research Ltd, 2006. Accessed: Oct. 16, 2022. [Online]. Available: <https://www.microsoft.com/en-us/research/uploads/prod/2006/01/Bishop-Pattern-Recognition-and-Machine-Learning-2006.pdf>
- [19] J. W. G. Putra, *Pengenalan Konsep Pembelajaran Mesin dan Deep Learning*, 1.4. 2020.
- [20] B. Raharjo, *Ilmu Big Data dan Mesin Cerdas*. Semarang: Yayasan Prima AgusTeknik, 2022.
- [21] I. D. Id, *Machine Learning: Teori, Studi Kasus dan Implementasi Menggunakan Python*, 1st ed. Riau: UR PRESS, 2021.
- [22] H. N. Firqiani, A. Kustiyo, and P. Giri, "Seleksi Fitur Menggunakan Fast Correlation Based Filter Pada Algoritma Voting Feature Intervals 5." Accessed: Oct. 28, 2022. [Online]. Available: <https://media.neliti.com/media/publications/245184-seleksi-fitur-menggunakan-fast-correlati-dfef2c7f.pdf>
- [23] I. M. B. Adnyana, "Penerapan Feature Selection untuk Prediksi Lama Studi Mahasiswa," *Jurnal Sistem dan Informatika*, vol. 13, pp. 72–76, May 2019.
- [24] I. M. D. Maysanjaya, M. S. Wibawa, and I. M. A. Wirahadi Putra, "Pengaruh Seleksi Fitur Terhadap Hasil Klasifikasi Fase Plasmodium vivax Pada Citra Mikroskopis Digital Sediaan Darah Tipis," Bali: Prosiding Seminar Nasional Pendidikan Teknik Informatika, Sep. 2017, pp. 96–101. [Online]. Available: <http://pti.undiksha.ac.id/senapati>
- [25] N. Noviati, S. Fauziati, and I. Hidayah, "Pengaruh Penghapusan Outlier Terhadap Kinerja Metode Seleksi Fitur Pada Klasifikasi Konsentrasi Sperma Berdasarkan Faktor Lingkungan, Kesehatan, dan Gaya Hidup," Universitas Gadjah Mada, Yogyakarta, 2015.