

Membandingkan Pengaruh Feature Selection Terhadap Algoritma Naïve Bayes dan Support Vector Machine

Aldi Rahmansyah, Ovitasari Dewi, Puti Andini, Trisna Hastuti Puspita Ningrum, Muhammad Eka Suryana

Program Studi Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam

Universitas Negeri Jakarta

Jakarta

Abstrak—*Feature selection* menjadi perhatian di banyak bidang penelitian yang menggunakan *machine learning*, karena menyediakan pengklasifikasi untuk mempercepat proses, menghemat biaya dan meningkatkan akurasi. Pada penelitian ini kami menggunakan beberapa algoritma dengan menerapkan Algoritma *Machine learning*. Algoritma yang kami gunakan yaitu *Support Vector Machine* dan *Naïve Bayes* dengan sebelas *features*. Dan untuk mempermudah penerapannya di dunia nyata kami juga melakukan *feature selection* dengan harapan mengurangi *features* yang digunakan. Hasil yang kami dapatkan adalah pengaruh *feature selection* terbaik ada pada algoritma *Naive Bayes* yang mendapat nilai presisi dan *recall* 0.5 dan 0.52 yang sebelumnya adalah 0.455 dan 0.434. Sebaliknya pada *Support Vector Machine*, hasil terbaiknya adalah ketika semua *feature* digunakan tanpa ada yang dikurangi.

Kata kunci—*Machine learning; Support Vector Machine; Naive Bayes; Feature Selection*

I. PENDAHULUAN

Popularitas *Machine learning* di dunia sedang meningkat beberapa tahun belakangan, sedangkan di Indonesia belum banyak orang yang menekuni salah satu di bidang ilmu komputer ini yaitu *Machine learning*. Dalam praktiknya, banyak yang menggunakan *Machine learning* tanpa menerapkan *Feature Selection*.

Terdapat banyak klasifikasi *Machine learning* yang menggunakan terlalu banyak *feature* padahal belum tentu di antara semua *feature* tersebut *relevant* terhadap klasifikasi sehingga menyulitkan dalam penerapannya di dunia nyata dan *feature* yang *irrelevant* tersebut dapat memperburuk akurasi dari Algoritma *Machine learning* yang diterapkan. Kami ingin menguji hipotesis yang kami buat yaitu dalam algoritma *Machine learning*, semakin banyak *features* semakin tinggi juga akurasinya.

Kami melakukan penelitian menggunakan *Feature selection* untuk mengetahui apa pengaruhnya terhadap Algoritma *Machine learning* berbasis *training* dan *non-training*. Dan kami berharap untuk dapat mendeteksi *feature* yang tidak terlalu relevan kemudian menghilangkan *feature* tersebut dari proses klasifikasi sehingga mempermudah penerapannya dalam kehidupan nyata. Kami memilih

dataset yang tidak berfokus pada topik klasifikasi *dataset* tetapi berfokus pada jumlah *features* dan *class*. Kami menggunakan *dataset wine quality* dari UCI *machine repository dataset* [1].

Wine dipandang menjadi suatu barang yang mewah, namun saat ini dapat dinikmati oleh masyarakat luas. Portugal merupakan salah satu dari sepuluh negara pengekspor *wine* terbesar di dunia pada tahun 2005. Vinho Verde [2] merupakan salah satu kedai pengekspor *wine* di Portugal, kedai tersebut mendapatkan penghasilan sebesar 36% dari hasil ekspor *wine* pada tahun 1997 sampai 2007 [3].

Untuk mendukung pertumbuhan anggur yang baik, maka banyak industri *wine* yang berinvestasi dalam teknologi yang baru untuk pembuatan *wine* dan proses penjualannya ketika menjadi *wine*. Sertifikasi dan kualitas dari *wine* merupakan penilaian yang menjadi sebuah kunci untuk memilih *wine* yang baik. Sertifikasi dapat mencegah pemalsuan *wine* secara ilegal dan menjamin kualitas dari tempat penjualan *wine* tersebut. Kualitas menjadi bagian dari proses sertifikasi dan digunakan untuk pembuatan *wine* agar menjadi lebih baik lagi juga untuk memberikan dan mengetahui stratifikasi *wine*, seperti *premium wine* (menetapkan harga *wine*) [3].

II. TINJAUAN PUSTAKA

Machine learning adalah cabang dari Ilmu Komputer berkaitan dengan *Artificial Intelligent* atau ilmu kecerdasan buatan yang utamanya berfokus pada pembangunan dan studi suatu sistem dengan tujuan agar mampu belajar dari data-data yang diperolehnya. Menurut Arthur Samuel, *Machine learning* adalah bidang studi yang memberikan kemampuan pada program komputer untuk belajar tanpa secara eksplisit diprogram [4].

Hal yang mendasar dan sangat diperlukan untuk menerapkan algoritma *Machine learning* yaitu data. *Machine learning* tidak akan pernah dapat bekerja tanpa adanya data yang tersedia. Dari data tersebut biasanya dibagi menjadi dua kelompok, yaitu data *training* dan data *testing*. Data *training* digunakan untuk pengembangan model dan melatih algoritma, sedangkan data *testing* digunakan sebagai referensi dan untuk mengetahui bagaimana performa dari

algoritma yang sudah sebelumnya dilatih ketika menemukan data baru yang belum pernah diketahui

Feature selection

Feature selection adalah suatu proses menghapus *features* yang berlebihan dan tidak relevan dari *dataset* yang sebenarnya. Sehingga waktu yang digunakan mengeksekusi dari pengklasifikasi yang memproses data berkurang, dan dapat meningkatkan akurasi juga karena *features* yang tidak relevan dapat memperburuk data mempengaruhi akurasi klasifikasi secara negatif [5]. Dengan *feature selection* dapat meningkatkan pemahaman dan biaya penanganan data menjadi lebih kecil [6].

Algoritma *Feature selection* dibagi menjadi tiga kelompok: *filters*, *wrappers*, dan *embedded selectors*. *Filters* mengevaluasi setiap *feature* secara bebas dari pengklasifikasi, memberikan peringkat pada *feature* setelah mengevaluasi dan mengambil yang unggul [7]. *Wrappers* mengambil *subset* dari *feature set*, mengevaluasi kinerja pengklasifikasian pada *subset* ini, dan kemudian *subset* lainnya dievaluasi oleh pengklasifikasi. *Subset* yang memiliki kinerja paling maksimum pada pengklasifikasian yang akan dipilih. Jadi *wrappers* bergantung pada pengklasifikasi yang dipilih. Bahkan *wrappers* lebih dapat diandalkan karena algoritma klasifikasi mempengaruhi tingkat akurasi [8]. Teknik *Embedded* disini lain melakukan *feature selection* selama proses mempelajari data sama seperti yang dilakukan jaringan syaraf tiruan.

Terdapat banyak algoritma *features selection*, yang akan kami gunakan dalam penelitian ini yaitu algoritma *features selection* yang bersifat *univariate* yang disebut *select k best*.

Untuk *Univariate features selection* secara umum bekerja dengan cara memilih *features* terbaik berdasarkan *test statistic univariate*. Hal ini dapat dilihat sebagai langkah *preprocess* sebuah estimator. *Select k best* secara khusus bekerja dengan cara memilih sejumlah *k features* terbaik berdasarkan pengujian statistic [9].

Kami menjalankan penelitian ini dengan menggunakan dua Algoritma *Machine learning* dan dua algoritma *Feature selection*. Dua algoritma *Machine learning* yang kami gunakan adalah *Support Vector Machine* dan *Naïve Bayes*.

A. Support Vector Machine

Support Vector Machine (SVM) adalah suatu Algoritma *Machine learning* yang digunakan untuk pengujian non-probabilistik seperti regresi dan teknik klasifikasi. SVM dapat disebut juga dengan nama lain yaitu *Support Vector Networks* [10]. SVM adalah algoritma Melalui SVM kita menempatkan *data item* berdasarkan ruang sampel dari *n*-dimensi, dimana *n* adalah *feature* yang dipertimbangkan.

Kemudian klasifikasi dilakukan dengan memanfaatkan *hyperplane* untuk memisahkan *class*. Setiap *hyperplane* dicirikan oleh arah (*w*), (*b*) adalah posisi yang tepat dalam ruang atau nilai ambang batas, (*x_i*) adalah vektor *input* dari dimensi *N* atau teks konten dan menunjukkan *class*. Salah satu contoh satu *set* berlabel *training* sebagai berikut:

$$(x_1, y_1), (x_2, y_2), \dots, (x_k, y_k), X \in \mathbb{R}^d \quad (1)$$

dimana *d* adalah *dimensionality* dari vektor; $y_i \in \{-1, +1\}$; $i = 1, 2, \dots, k$

Kami mempertimbangkan fungsi keputusan dengan bentuk $f(x, w, b) = \text{sgn}((w \cdot x_i) + b)$, $w \in \mathbb{R}^d$, $b \in \mathbb{R}$.

Kemudian, wilayah antara *hyperplane* jika dan hanya jika memisahkan dua *class*, disebut dengan *margin*.

Lebar dari *margin* sama dengan $\frac{1}{2} \|w\|$ dan akan didapatkan nilai *margin* maksimum yang mungkin mendasari ide dari Algoritma SVM. Memaksimalkan *margin* membutuhkan minimalisasi, yaitu:

$$f(w, b) = \frac{1}{2} \|w\|^2 \quad (2)$$

dengan:

$$\begin{aligned} wx_i + b &>> 1, \text{ if } y_i = 1 \\ wx_i + b &<< 1, \text{ if } y_i = -1 \end{aligned}$$

K adalah tetapan yang ditentukan pengguna dan ϵ adalah kesalahan *margin*. Kesalahan *margin* terjadi jika data termasuk dalam satu *class* dan berada di sisi yang salah dari *hyperplane*. Meminimalkan biaya adalah permasalahan antara *margin* yang besar dan sejumlah kecil kesalahan *margin*. Penyelesaian dari masalah pengoptimalan ini diperoleh sebagai berikut:

$$w = \sum_{i=1}^N \lambda_i \gamma_i \chi_i \quad (3)$$

Formula diatas adalah rata-rata tertimbang dari *training features*. Dimana λ_i adalah pengganda *Lagrange* dari tugas optimasi dan γ_i adalah label *class*. Nilai λ_i tidak nol untuk semua nilai yang ada di dalam *margin* ketika berada pada sisi yang benar dari pengklasifikasi [11].

B. Naïve Bayes Classifier

Naïve Bayes Classifier (NBC) merupakan algoritma sederhana berdasarkan Teori *Bayes* yang mengasumsikan bahwa atribut - atribut klasifikasi bersifat ketidakbergantungan (*independent*) dan tidak terdapat korelasi di antara mereka [12]. Kelebihan dari penggunaan NBC ini adalah hanya membutuhkan *training data* yang tidak besar untuk dapat menentukan perkiraan parameter yang diperlukan dalam proses pengklasifikasian. Metode ini bisa digunakan dalam pengujian probabilistik seperti segregasi berdasarkan relevansi pendidikan, mempermudah memilih kandidat untuk proses perekrutan tenaga kerja berdasarkan tingkat pendidikannya, dan lain-lain.

Secara garis besar berikut adalah Teorema *Bayes* dimana $F_1 \dots F_n$ adalah sampel *dataset* dengan *class* yang belum diketahui, *C* adalah hipotesis yang dibuat dan merupakan *class* secara spesifik, $p(C | F_1, \dots, F_n)$ adalah probabilitas hipotesis *C* berdasarkan kondisi F_1, \dots, F_n atau bisa disebut *posterior probability*, $p(C)$ adalah probabilitas dari hipotesis *C* atau disebut *prior probability*, $p(F_1, \dots, F_n | C)$ adalah probabilitas berdasarkan kondisi F_1, \dots, F_n pada hipotesis *C*

atau disebut *likelihood* dan $p(F_1, \dots, F_n)$ adalah probabilitas dari seluruh dataset.

$$p(C | F_1, \dots, F_n) = \frac{p(C)p(F_1, \dots, F_n | C)}{p(F_1, \dots, F_n)} \quad (4)$$

Teknik *Naïve Bayes* berdasarkan pada probabilitas bersyarat dan kemungkinan kejadian maksimum, seperti pada formula berikut:

$$\text{Posterior} = \frac{\text{Prior} * \text{Likelihood}}{\text{Evidence}} \quad (5)$$

III. EKSPERIMEN DAN HASIL

Pada penelitian ini, kami menggunakan *dataset* yang terdiri dari 4989 *instances* dan memiliki 11 *features* dengan class yang bernilai dari 0 s.d. 10 yang berupa angka penilaian dari data *wine* tersebut. *Features* yang terdapat pada *dataset* tersebut, yaitu:

1. Keasaman Tetap
2. Keasaman *Volatile*
3. Asam Sitrat
4. Gula Residu
5. Kandungan Klorida
6. Belerang Dioksida Bebas
7. Belerang Dioksida Total
8. Densitas
9. pH
10. Kandungan Sulfat
11. Kandungan Alkohol

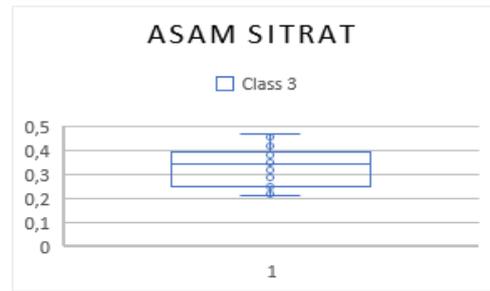
Algoritma *Machine learning* yang kami gunakan, yaitu *Support Vector Machine* (SVM) dan *Naïve Bayes* untuk membandingkan perubahan yang terjadi ketika dilakukan *Feature selection*.

Dalam penelitian ini kami melakukan eksperimen sebagai berikut :

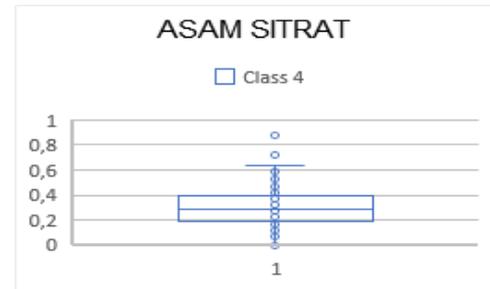
A. Uji *Boxplot*

Analisis pertama yang kami lakukan dengan menggunakan Uji *Boxplot* pada seluruh *features*. Dari *features* yang diuji, kami hanya mengambil beberapa *features* yang signifikan untuk dibandingkan yaitu asam sitrat dan alkohol.

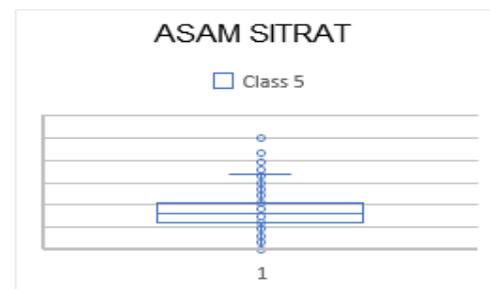
Kami mengambil dua *features* yaitu asam sitrat (*Feature 3*) dan kandungan alkohol (*Feature 11*). Karena dapat dilihat bahwa, rentang *boxplot* antar *class* pada asam sitrat seperti pada Gambar 1 s.d. 7 tidak dapat terlihat adanya perbedaan distribusi data. Sedangkan, rentang *boxplot* antar *class* pada kandungan alkohol seperti pada Gambar 8 s.d. 14 dapat terlihat adanya perbedaan distribusi data. Maka dari itu, kami simpulkan hasil dari algoritma *feature selection* bahwa kandungan alkohol menjadi salah satu kandidat yang baik *features* yang dapat dipilih. Sedangkan, asam sitrat bukan kandidat yang baik untuk dipilih pada *feature selection*.



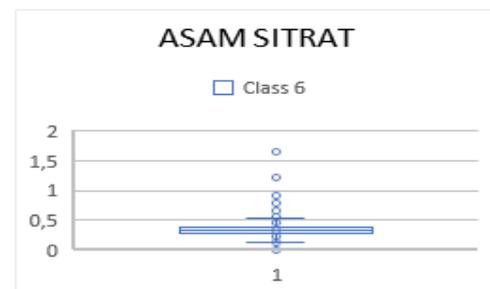
Gambar 1. *Boxplot* terkait dengan class 3 dari asam sitrat (Feature 3)



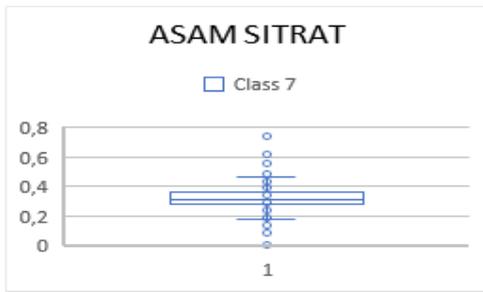
Gambar 2. *Boxplot* terkait dengan class 4 dari asam sitrat (Feature 3)



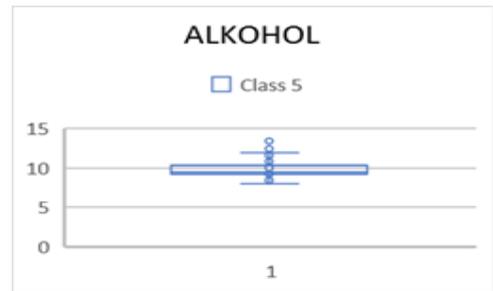
Gambar 3. *Boxplot* terkait dengan class 5 dari asam sitrat (Feature 3)



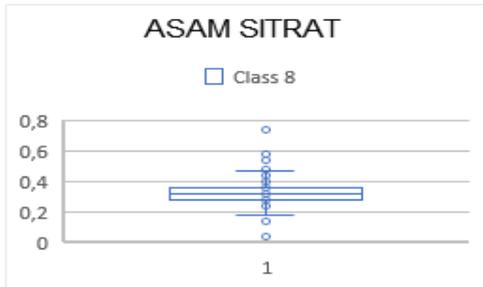
Gambar 4. *Boxplot* terkait dengan class 6 dari asam sitrat (Feature 3)



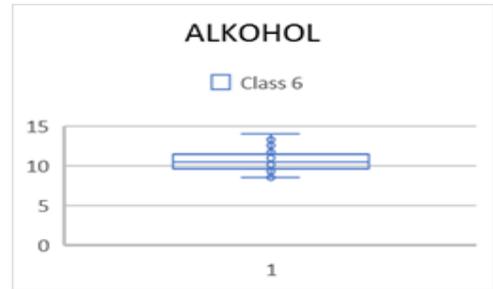
Gambar 5. Boxplot terkait dengan class 7 dari asam sitrat (Feature 3)



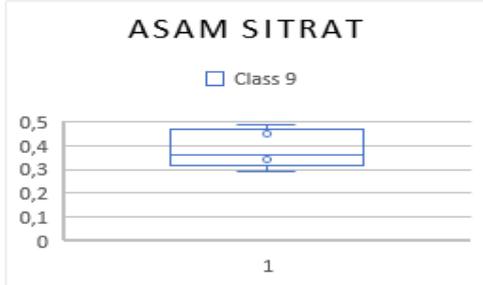
Gambar 10. Boxplot terkait dengan class 5 dari kandungan alkohol (Feature 11)



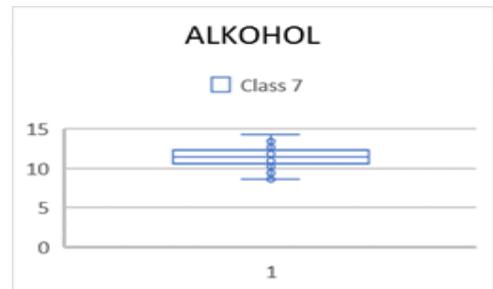
Gambar 6. Boxplot terkait dengan class 8 dari asam sitrat (Feature 3)



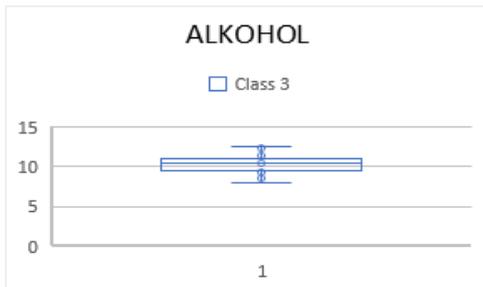
Gambar 11. Boxplot terkait dengan class 6 dari kandungan alkohol (Feature 11)



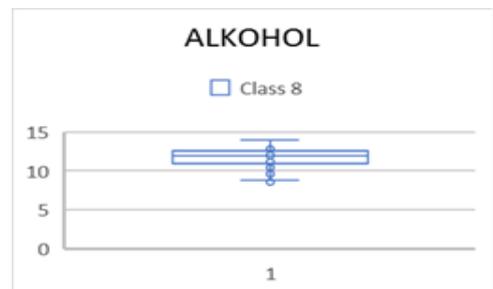
Gambar 7. Boxplot terkait dengan class 9 dari asam sitrat (Feature 3)



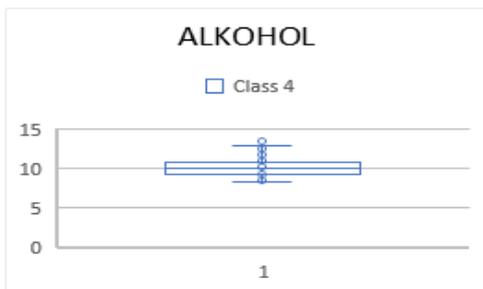
Gambar 12. Boxplot terkait dengan class 7 dari kandungan alkohol (Feature 11)



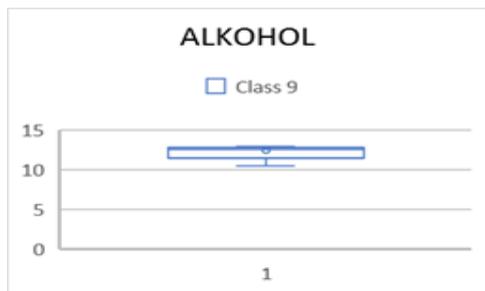
Gambar 8. Boxplot terkait dengan class 3 dari kandungan alkohol (Feature 11)



Gambar 13. Boxplot terkait dengan class 8 dari kandungan alkohol (Feature 11)



Gambar 9. Boxplot terkait dengan class 4 dari kandungan alkohol (Feature 11)



Gambar 14. Boxplot terkait dengan class 9 dari kandungan alkohol (Feature 11)

B. Tuning parameter

Pada tahap ini kami melakukan *tuning parameter SVM* untuk menemukan parameter terbaik untuk data.

Dari hasil yang didapat pada Tabel I bahwa parameter terbaik untuk model yang digunakan adalah dengan *Kernel RBF*, *C* sebesar 1000, dan *Gamma* 0.5.

TABEL I. MENGUJI PARAMETER DENGAN *K-FOLD* = 10

Kernel	C	Gamma	F-Measure
rbf	1	auto	0.42
sigmoid	1	auto	0.39
linear	1	auto	0.44
poly	1	auto	0.27
rbf	0.01	auto	0.27
rbf	0.1	auto	0.27
rbf	1	auto	0.42
rbf	10	auto	0.43
rbf	100	auto	0.46
rbf	1000	auto	0.48
linear	0.01	auto	0.27
linear	0.1	auto	0.4
linear	1	auto	0.43
linear	10	auto	0.43
linear	100	auto	0.43
rbf	1000	0.0001	0.42
rbf	1000	0.001	0.43
rbf	1000	0.01	0.43
rbf	1000	0.1	0.48
rbf	1000	1	0.49
rbf	1000	10	0.43
rbf	1000	100	0.45
rbf	1000	1000	0.29
rbf	1000	0.5	0.49

C. Uji Stabilitas Menggunakan *K-Fold*

Tahap selanjutnya yang kami lakukan adalah pengecekan stabilitas algoritma dengan menggunakan *K* yang berbeda-beda pada *K-Fold*.

Dari hasil pada Tabel II dan Tabel III, didapat bahwa model SVM dan *Naïve Bayes* sudah cukup stabil hingga *K* = 10

D. Select *K-Best Feature Selection*

Selanjutnya setelah didapatkan parameter terbaik dan kestabilan yang baik, kami menerapkan *Feature Selection K-Best* untuk mencari nilai *K* terbaik sebagai jumlah *feature* pilihan yang terbaik.

Dari hasil pada Tabel IV, didapat *features* pilihan dengan *F-Measure* terbaik yaitu *feature* 2 dan 11 dengan *F-Measure* sebesar 0.486.

Dari hasil Tabel V, dapat dilihat selalu terjadi penurunan ketika dilakukan pengurangan *feature*, sehingga *features* dari SVM tidak berubah.

E. Membandingkan Hasil Terbaik Setelah dijalankan *feature selection*

Setelah didapatkan *feature-feature* pilihan terbaik, kami menjalankan kembali kedua algoritma menggunakan *feature* seluruhnya dan *feature* pilihan untuk membandingkan perubahan yang terjadi pada kedua algoritma tersebut.

Berdasarkan hasil analisis perbandingan dimana pada Tabel VI menggunakan *Naïve Bayes* dan didapatkan hasil terbaik sebelum diterapkan *select K-Best* pada *F-Measure* yaitu 0.423. Setelah menggunakan *select K-Best* seperti pada Tabel VII dengan menggunakan algoritma *Naïve Bayes* didapatkan *F-Measure* terbaik yaitu 0.486. Kemudian dengan menggunakan *select K-Best* seperti pada Tabel VIII dengan menggunakan algoritma SVM didapatkan hasil yang sama, karena *feature* yang tidak berubah yaitu mendapatkan nilai *F-Measure* terbaik 0.498.

TABEL II. MENGUJI *FEATURES* MENGGUNAKAN *K-FOLD SVM*
C = 1000 GAMMA = 0.5

K	F Measure
10	0.498
7	0.494
5	0.48
2	0.44

TABEL III. MENGUJI *FEATURES* MENGGUNAKAN *NAÏVE BAYES*

K	F Measure
10	0.42
7	0.422
5	0.41
2	0.394

TABEL IV. MENGUJI FEATURES MENGGUNAKAN NAÏVE BAYES, 10 FOLDS, DENGAN STATISTIC FUNCTION CHI²

K	Selected Feature	F-Measure
1	11	0.451
2	2,11	0.486
3	2,8,11	0.464
4	2,5,8,11	0.461
5	2,4,5,8,11	0.444
6	2,4,5,7,8,11	0.436
7	2,4,5,6,7,8,11	0.429
8	2,4,5,6,7,8,9,11	0.429
9	1,2,4,5,6,7,8,9,11	0.427
10	1,2,4,5,6,7,8,9,10,11	0.428
11	1,2,3,4,5,6,7,8,9,10,11	0.420

TABEL V. MENGUJI FEATURES MENGGUNAKAN SVM RBF, C=1000, GAMMA = 0.5

K	Selected Features	F-Measure
1	11	0.412
2	2,11	0.455
3	2,8,11	0.457
4	2,5,8,11	0.464
5	2,4,5,8,11	0.466
6	2,4,5,7,8,11	0.473
7	2,4,5,6,7,8,11	0.482
8	2,4,5,6,7,8,9,11	0.490
9	1,2,4,5,6,7,8,9,11	0.497
10	1,2,4,5,6,7,8,9,10,11	0.494
11	1,2,3,4,5,6,7,8,9,10,11	0.498

TABEL VI. MENGUJI SELURUH FEATURES MENGGUNAKAN NAÏVE BAYES

Naïve Bayes			
K-Fold	Precision	Recall	F-Measure
10	0.455	0.434	0.420
7	0.459	0.436	0.423
5	0.451	0.430	0.418
2	0.431	0.409	0.394

TABEL VII. MENGUJI DUA FEATURES PILIHAN MENGGUNAKAN SELECT K BEST

Naïve Bayes dengan Dua Features			
K-Fold	Precision	Recall	F-Measure
10	0.501	0.520	0.486
7	0.501	0.519	0.485
5	0.488	0.508	0.473
2	0.494	0.486	0.450

TABEL VIII. MENGUJI SELURUH FEATURES PILIHAN MENGGUNAKAN SELECT K BEST

SVM			
K-Fold	Precision	Recall	F-Measure
10	0.515	0.532	0.498
7	0.506	0.527	0.494
5	0.506	0.522	0.488
2	0.473	0.471	0.441

IV. KESIMPULAN DAN SARAN

Dari penelitian yang sudah dilakukan, didapat kesimpulan bahwa untuk algoritma SVM, ketika menurunkan jumlah *feature* akan menyebabkan penurunan akurasi algoritma. Karena sekecil apapun pengaruh suatu *feature*, untuk algoritma non-probabilistik cukup berpengaruh untuk meningkatkan akurasi total suatu algoritma. Selama *feature* tersebut memang memiliki relevansi terhadap hasil klasifikasi. Sedangkan untuk Algoritma *Naive Bayes*, ketika menurunkan jumlah *feature* terdapat kemungkinan terjadi kenaikan akurasi dengan menghilangkan *feature* yang memiliki relevansi kecil.

Dalam penelitian dan pembuatan makalah yang telah kami lakukan, kami harap dapat bermanfaat kedepannya untuk peneliti dan peneliti selanjutnya. Maka dari itu, kami berharap kepada peneliti selanjutnya agar dapat mengembangkan bahkan memperbaiki hasil penelitian kami di kemudian hari. Kami menyarankan kepada peneliti agar lebih fokus dalam melakukan analisis baik analisis pada *boxplot* maupun *feature selection*. Karena kedua analisis tersebut harus memiliki akurasi yang sama untuk mendapatkan *feature* yang paling baik.

UCAPAN TERIMA KASIH

Kami ucapkan terima kasih kepada Allah SWT, orangtua, dosen dan semua pihak yang terkait dalam penelitian dan penyusunan makalah ini.

REFERENSI

- [1] A. Asuncion dan D. Newmann, UCI Machine learning Repository, <http://www.ics.uci.edu/mllearn/MLRepository.html>, University of California, 2017.

- [2] Portuguese Wine-Vinho Verde, “Comissão de Viticultura da Região dos Vinhos Verdes (CVRVV)”, <http://www.vinhoverde.pt>, diakses pada Januari 2018.
- [3] Cortez, Paulo; Cerdeira, Antonio; Almeida, Fernando; Matos, Telmo; Reis, Jose, “Modeling Wine Preferences by Data Mining From Physicochemical Properties,” *Decision Support Systems, Elsevier, Vol. 47, Issue 4*, 547-553, 2009.
- [4] Fikriya, Zulfa Afiq; Irawan, Mohammad Isa; Soetrisno, “Implementasi *Extreme Learning Machine* untuk Pengenalan Objek Citra Digital,” *Jurnal Sains dan Seni ITS*, Vol. 6, No. 1. 2337-3520, 2017.
- [5] S. Doraisami, dan S. Golzari, “A Study on *Feature selection* and Classification Techniques for Automatic Genre Classification of Traditional Malay Music, Content-Based Retrieval, Categorization and Similarity,” 2008.
- [6] A. Arauzo-Azofra, J. L. Aznarte, and J. M. Benítez, “Empirical study of *feature selection* methods based on individual feature evaluation for classification problems,” *Expert Systems with Applications*, Vol, 38. 8170-8177, 2011.
- [7] Guyon Isabelle dan A. Elisseeff, “An introduction to variable and *Feature Selection*,” *Journal of Machine learning Research*, Vol. 3, Edisi 7-8, 1157-1182, .
- [8] J. Novakovic, “The Impact of *Feature selection* on the Accuracy of Naive Bayes Classifier,” 18th Telecommunications forum TELFOR, 2010.
- [9] Pedregosa, dkk., “*Scikit-learn: Machine Learning in Python*” *JMLR* 12, pp. 2825-2830, 2011.
- [10] Cortes, C dan V. Vaprik, “Support-vector networks.” *Machine learning Vol. 20*, Edisi 3. 273-279, 1995.
- [11] Elssied, Nadir Omer Fadl, dkk, “A Novel *Feature Selection* Based on One-Way ANOVA F-Test for E-mail Spam Classification,” *Research Journal of Applied Sciences, Engineering and Technology* 7(3): 625-638, 2014.
- [12] Karthika, S dan Sairam, N. “A Naïve Bayesian Classifier for Educational Qualification,” *Indian Journal of Science and Technology*, Vol 8(16), DOI: 10.17485/ijst/2015/v8i16/62055, July 2015.