ALGORITMA NAÏVE BAYES CLASSIFIER UNTUK PREDIKSI STRESS

Antonius Bun Wijaya[1] , Yulia Wahyuningsih[2]
[1,2]Ilmu Informatika, Fakultas Teknik, Universitas Katolik Darma Cendika, Kota SBY, Jawa Timur 60117
[1]Alamat e-mail antonius.wijaya@student.ukdc.ac.id

Abstrak

Stress adalah perasaan umum yang dapat kita alami ketika dibawah tekanan atau bergumul dengan suatu situasi. *Stress* yang berlebihan dapat berdampak negatif pada suasana hati, kesehatan fisik dan mental kita, serta hubungan kita dengan orang lain, terutama saat dirasa di luar kendali. Dalam penelitian ini kita mencoba untuk membuat suatu prediksi seseorang *Stress* atau tidak melalui teks dengan menggunakan algoritma *Naïve Bayes Classifier* dari dataset yang tersedia publik oleh *kaggle* . *Naïve Bayes Classifier* merupakan salah satu algoritma klasifikasi dengan berfokus pada probabilitas bersyarat. Adapun hasil penelitian ini akurasi terbaik didapatkan menggunakan *Naïve Bayes Classifier* dengan nilai 75% ,*Support Vector Machine* dengan nilai 71% dan *AdaBoost* dengan nilai 67%.

Kata Kunci: *AdaBoost, Naïve Bayes Classifier, Support Vector Machine.*

Abstract

Stress is a common feeling that we can experience when under pressure or struggling with a situation. Excessive stress can negatively affect our mood, physical and mental health, and our relationships with others, especially when it feels out of control. In this study we try to make a prediction of whether someone is Stressed or not through text by using the Naïve Bayes Classifier algorithm from a publicly available dataset by kaggle. Naïve Bayes Classifier is a classification algorithm that focuses on conditional probabilities. As for the results of this study the best accuracy was obtained using the Naïve Bayes Classifier with a value of 0.75, Support Vector Machine with a value of 0.71 and AdaBoost with a value of 0.67.

Keywords: *AdaBoost, Naïve Bayes Classifier, Support Vector Machine.*

.

PENDAHULUAN

Stress adalah keadaan mengganggu keseimbangan pola respon seseorang (Aryani, 2016). Salah satu dampak negatif dari *Stress* dapat berpengaruh terhadap pekerjaan (Putra & Sriathi, 2018). Adapun *Stress* dapat diketahui melalui bersosial media (Budury et al., 2019). Dalam bersosial media biasanya kita menggunakan tulisan sebagai komunikasi. *Reddit* merupakan salah satu *platform* sosial media. Data yang digunakan ialah data yang sudah tersedia secara umum dari *platform kaggle*

Algoritma *Naïve Bayes Classifier* merupakan salah satu algoritma klasifikasi yang menggunakan probabilitas untuk menghasilkan sebuah hipotesis (Sumathi & Esakkirajan, 2007). Algoritma ini sudah banyak digunakan pada penelitian terdahulu yaitu *Naïve Bayes Classifier* dapat menyelesaikan beberapa permasalahan seperti melakukan klasifikasi masyarakat miskin (Annur, 2018), klasifikasi *text mining review* produk kosmetik untuk *teks* bahasa Indonesia (Indrayuni, 2019), klasifikasi berita *hoax* (Mustofa & Mahfudh, 2019), klasifikasi keluhan masyarakat pada pemkot probolinggo (Ariyanti & Iswardani, 2020) dan Analisis setimen data *review twitter* BMKG Nasional (Darwis et al., 2021).

Adapun Tujuan dari penelitian ini ialah uji coba untuk mengukur apakah dengan *Naïve Bayes Classifier* baik dalam memprediksi *Stress*. Pada penelitian ini juga melihat perbandingan akurasi menggunakan algoritma selain *Naïve Bayes Classifier* seperti *Adaboost* dan *Support Vector Machine.*

METODE

Pada penelitian ini melakukan uji coba untuk prediksi stress menggunakan algoritma *Naïve Bayes Classifier.* Adapun tahapan dalam penelitian ini dimulai dari Pengumpulan Data, Pengolahan Data , Implementasi *Naïve Bayes Classifier* dan Pengukuran Performa serta Hasil Analisa. Alur Metode dapat dilihat pada Gambar 1 berikut.

Gambar 1. Alur Metode

Dataset pada penelitian ini menggunakan dataset dari ***kaggle*** yang merupakan hasil dari Analisis Sentimen data *reddit*. Data yang digunakan pada penelitian ini hanya dreaddit-train.csv, data yang digunakan hanya memilih kolom *text* dan *label.*

Pengolahan data dilakukan pada kolom *text* diolah dengan menerapkan *Tokenizing, Case Folding, Filtering* dan *Stemming*. Pada kolom label karena sudah *one hot encoding* 0 dan 1 sehingga hanya memberikan keterangan dengan 0 adalah *No Stress* dan 1 adalah *Stress*.

Pada tahapan ini akan menggunakan algoritma *Naïve Bayes Classifier*. *Naïve Bayes Classifier* adalah algoritma yang mempelajari klasifikasi probabilitas berdasarkan ciri-ciri dari sebuah hipotesis yang akan terjadi. Adapun *Naïve Bayes* yang akan digunakan ialah *Bernoulli* yaitu klasifikasi biner atau 2 kelas. Berikut teorema dari Naïve Bayes.

$$P(A \mid B) = \frac{P(B|A).\ P(A)}{P\ (B)}$$

A,B	Kejadian
P(A|B)	Probabilitas A yang diberikan B adalah benar
P(B|A)	Probabilitas B yang diberikan A adalah benar
P(A), P(B)	Probabilitas independen dari A dan B

Gambar 2. Rumus Naive Bayes

Pengukuran performansi akan menampilkan hasil dari *Confusion Matrix* dan *Classification Report. Confussion Matrix* adalah suatu hasil yang menampilkan antara nilai aktual dan nilai prediksi.

Tabel 1. Confussion Matrix

		Nilai Aktual	
		1 (Positif)	0 (Negatif)
Nilai Prediksi	1 (Positif)	**Positif Benar**	**Positif Salah**
	0 (Negatif)	**Negatif Salah**	**Negatif Benar**

Classification Report adalah suatu informasi yang berisi *precission, recall, dan f1 score* .Perbedaan *precission* dan *recall* ialah *precission* menggunakan Positif Salah sedangkan *recall* menggunakan Negatif Salah. Kemudian *f1 score* adalah hasil rata-rata dari *precission* dan *recall.*

$$Precission = \frac{\text{Positif Benar}}{\text{Positif Benar} + \text{Positif Salah}}$$

$$Recall = \frac{\text{Positif Benar}}{\text{Positif Benar} + \text{Negatif Salah}}$$

$$F1\ Score = \frac{1}{\text{F1}} = \frac{1}{2}\left(\frac{1}{Precission} + \frac{1}{Recall}\right)$$

Gambar 3. Rumus Precission, Recall dan F1 Score

Disini pengukuran performansi juga akan menampilkan hasil dari algoritma lainnya seperti *Support Vector Machine* dan *Adaboost Classifier*. *Support Vector Machine* adalah pembelajaran mesin yang menggunakan fungsi – fungsi linear untuk memberikan hipotesis (Rahman Isnain et al., 2021). Sedangkan *Adaboost Classifier* adalah *ensemble learning* yaitu algoritma yang biasanya dikombinasikan dengan algoritma klasifikasi lainnya (Listiana & Muslim, 2017). Pada uji coba kali

ini kita hanya membandingkan dengan *Support Vector Machine* dan *Adaboost* bawaan tanpa kombinasi algoritma klasifikasi lainnya.

Hasil Analisa merupakan hasil pengamatan dari sebuah penelitian, untuk dapat memberikan informasi dari hasil Prediksi *Stress*. Dari hasil Analisa akan menampilkan sebuah Tabel Hasil.

HASIL DAN PEMBAHASAN

Dataset yang digunakan memiliki 116 kolom dengan jumlah baris sebanyak 2838. Data yang dipilih yaitu text dan label . text untuk mendeskripsikan tulisan dari pengguna reddit sedangkan label memberikan pemberitahuan Stress atau No Stress dari tulisan tersebut. Adapun gambar dari contoh data dapat dilihat pada Gambar 4 berikut.

	text	label
He said he had not felt that way before, sugge...		1
Hey there r/assistance, Not sure if this is th...		0
My mom then hit me with the newspaper and it s...		1
until i met my new boyfriend, he is amazing, h...		1
October is Domestic Violence Awareness Month a...		1

Gambar 4. Contoh Data

Pengolahan data dilakukan dengan *Tokenizing* , *Case Folding, Filtering* dan *Stemming.* Adapun langkahnya sebagai berikut.

Tokenizing

Dilakukan untuk mengubah suatu kalimat menjadi kata tunggal tanpa Spasi , Koma , Titik Dua atau lainnya. Perbedaan sebelum dan sesudah t*okenization* dapat dlihat pada Tabel 2.

Tabel 2. Contoh *Tokenization*

Sebelum *Tokenization*	Sesudah *Tokenization*
"Also the headaches"	*"Also" + "the" + "headaches"*

Case Folding

Dilakukan untuk menyamaratakan kata menjadi huruf kecil. Sebagai contoh *"TRIGGER AHEAD IF YOUI'RE A HYPOCONDRIAC LIKE ME"* menjadi *"trigger ahead if youi're a hypocondriac like me".*

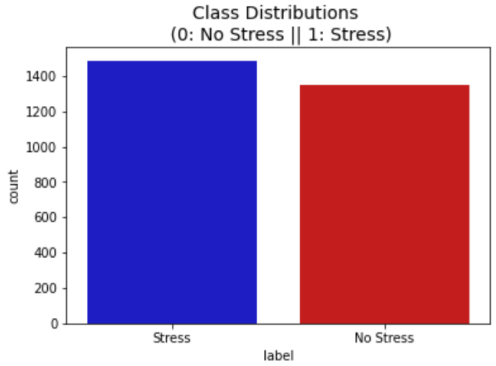
Filtering

Dilakukan untuk melakukan pembuangan kata - kata yang tidak penting . Dengan menggunakan *stopwords* dapat menghilangkan kata *"a" , "the"* atau lainnya.

Stemming

Dilakukan untuk membuat suatu kata kembali ke kata dasar dengan menghilangkan suatu imbuhan. Hal ini dilakukan karena dapat berpengaruh tergadap hasil prediksi.

Langkah selanjutnya setelah data dibersihkan, dilakukan pemberian label dimana 1 menandakan *"Stress"* dan 0 menandakan *"No Stress"* . Dapat dilihat pada Gambar 5 grafik banyaknya data yang berlabel. Setelahnya dilakukan pemisahan data latih dan tes yang dapat dilihat pada Gambar 6.



Gambar 5. Gambaran banyaknya data yang diberi label

```
# train test memisahkan dataset
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split

x = np.array(data["text"])
y = np.array(data["label"])

validation_size = 0.20

cv = CountVectorizer()
X = cv.fit_transform(x)
xtrain, xtest, ytrain, ytest = train_test_split(X, y,
                                                test_size=validation_size,
                                                random_state=42)
```

Gambar 6. Source code memisahkan data *train* dan *test*

Implementasi *Naïve Bayes Classifier*

```
# Melatih model menggunakan BernoulliNB
from sklearn.naive_bayes import BernoulliNB
modelNB = BernoulliNB()
modelNB.fit(xtrain, ytrain)# Menampilkan prediksi dari data test
y_pred_nb = modelNB.predict(xtest)# Menampilkan prediksi dari data test
# Melihat Akurasi Model
from sklearn import metrics
# Model Akurasi
print("Akurasi Naive Bayes dengan BernoulliNB : ",
      metrics.accuracy_score(ytest,y_pred_nb))
```

```
Akurasi Naive Bayes dengan BernoulliNB :  0.7588028169014085
```

Gambar 7. Source Code menerapkan Naive Bayes Classifier

Pengukuran Performansi

```
# Confussion Matrix Naive Bayes
from sklearn.metrics import confusion_matrix
confusion_matrix(ytest, y_pred_nb)
```

```
array([[174,  89],
       [ 52, 253]])
```

Gambar 8. Source code Confussion Matrix Naive Bayes Classifier

```
# Confussion Matrix AdaBoost
from sklearn.metrics import confusion_matrix
confusion_matrix(ytest, y_pred_ab)
```

```
array([[176,  87],
       [103, 202]])
```

Gambar 9. *Source code Confussion Matrix AdaBoost Classifier*

```
# Confussion Matrix Support Vector Machine
from sklearn.metrics import confusion_matrix
confusion_matrix(ytest, y_pred_svm)
```

```
array([[178,  85],
       [ 78, 227]])
```

Gambar 10. *Source code Confussion Matrix Support Vector Machine*

```
# Menampilkan klasifikasi laporan naive bayes
from sklearn.metrics import classification_report
print(classification_report(ytest, y_pred_nb))
```

```
              precision    recall  f1-score   support

   No Stress       0.77      0.66      0.71       263
      Stress       0.74      0.83      0.78       305

    accuracy                           0.75       568
   macro avg       0.75      0.75      0.75       568
weighted avg       0.75      0.75      0.75       568
```

Gambar 11. *Source code* laporan klasifikasi *Naive Bayes Classifier*

```
# Menampilkan klasifikasi laporan AdaBoost
from sklearn.metrics import classification_report
print(classification_report(ytest, y_pred_ab))
```

```
              precision    recall  f1-score   support

           0       0.63      0.67      0.65       263
           1       0.70      0.66      0.68       305

    accuracy                           0.67       568
   macro avg       0.66      0.67      0.66       568
weighted avg       0.67      0.67      0.67       568
```

Gambar 12. *Source code* laporan klasifikasi *AdaBoost Classifier*

```
# Menampilkan klasifikasi laporan Support Vector Machine
from sklearn.metrics import classification_report
print(classification_report(ytest, y_pred_svm))
```

```
              precision    recall  f1-score   support

           0       0.70      0.68      0.69       263
           1       0.73      0.74      0.74       305

    accuracy                           0.71       568
   macro avg       0.71      0.71      0.71       568
weighted avg       0.71      0.71      0.71       568
```

Gambar 13. *Source code* laporan klasifikasi *Support Vector Machine*

Kemudian dilakukan hasil analisa yang ditunjukkan pada Tabel 3.

Tabel 3. Hasil Akurasi

Algoritma	Akurasi (%)
AdaBoost Classifier	67
Naïve Bayes Classifier	75
Support Vector Machine	71

Hasil Akurasi menunjukkan *Naïve Bayes Classifier* memiliki angka tertinggi. Dari hal ini menunjukkan bahwa Algoritma *Naïve Bayes Classifier* terbaik untuk uji coba prediksi stress melalui text. Adapun selisih dengan *Support Vector Machine* tanpa *Kernel Linear* sebesar - 4% sedangkan selisih dengan *AdaBoost* tanpa penggabungan Algoritma Klasifikasi sebesar -8%.

SIMPULAN

Berdasarkan hasil analisa yang didapatkan dari tabel hasil akurasi. *Naïve Bayes Classifier* Mendapatkan akurasi dengan nilai tertinggi yaitu 75% . Kemudian disusul dengan *Support Vector Machine* dengan nilai yaitu 71% dan *Adaboost Classifier* dengan nilai yaitu 67%. Hal ini menunjukkan uji coba dengan *Naïve Bayes Classifier* baik dalam melakukan prediksi terhadap dataset *Stress*. Adapun saran untuk penelitian selanjutnya bisa menggunakan dataset berbahasa indonesia mengingat datasat yang digunakan pada saat ini masih menggunakan bahasa inggris.

DAFTAR PUSTAKA

Annur, H. (2018). Klasifikasi Masyarakat Miskin Menggunakan Metode Naive Bayes. *ILKOM Jurnal Ilmiah*, *10*(2), 160–165. https://doi.org/10.33096/ilkom.v10i2.303.160-165

Ariyanti, D., & Iswardani, K. (2020). Teks Mining untuk Klasifikasi Keluhan Masyarakat Pada Pemkot Probolinggo Menggunakan Algoritma Naïve Bayes. *Jurnal IKRA-ITH Informatika*, *4*(3), 125–132.

Aryani, F. (2016). *Stres Belajar Suatu Pendekatan dan Intervensi Konseling*. http://eprints.unm.ac.id/2478/1/Buku - Stres Belajar.pdf

Budury, S., Fitriasari, A., & -, K. (2019). Penggunaan Media Sosial Terhadap Kejadian Depresi, Kecemasan Dan Stres Pada Mahasiswa. *Bali Medika Jurnal*, *6*(2), 205–208. https://doi.org/10.36376/bmj.v6i2.87

Darwis, D., Siskawati, N., & Abidin, Z. (2021). Penerapan Algoritma Naive Bayes Untuk Analisis Sentimen Review Data Twitter Bmkg Nasional. *Jurnal Tekno Kompak*, *15*(1), 131. https://doi.org/10.33365/jtk.v15i1.744

Indrayuni, E. (2019). Klasifikasi Text Mining Review Produk Kosmetik Untuk Teks Bahasa Indonesia Menggunakan Algoritma Naive Bayes. *Jurnal Khatulistiwa Informatika*, *7*(1), 29–36. https://doi.org/10.31294/jki.v7i1.1

Listiana, E., & Muslim, M. A. (2017). Penerapan Adaboost Untuk Klasifikasi Support Vector Machine Guna Meningkatkan Akurasi Pada Diagnosa Chronic Kidney Disease. *Prosiding SNATIF*, *2015*, 875–881.

Mustofa, H., & Mahfudh, A. A. (2019). Klasifikasi Berita Hoax Dengan Menggunakan Metode Naive Bayes. *Walisongo Journal of Information Technology*, *1*(1), 1. https://doi.org/10.21580/wjit.2019.1.1.3915

Putra, I. W. S., & Sriathi, A. A. A. (2018). Pengaruh Lingkungan Kerja, Stres Kerja Dan Kompensasi Terhadap Loyalitas Karyawan. *E-Jurnal Manajemen Universitas Udayana*, *8*(2), 786. https://doi.org/10.24843/ejmunud.2019.v08.i02.p08

Rahman Isnain, A., Indra Sakti, A., Alita, D., & Satya Marga, N. (2021). Sentimen Analisis Publik Terhadap Kebijakan Lockdown Pemerintah Jakarta Menggunakan Algoritma Svm. *Jdmsi*, *2*(1), 31–37. https://t.co/NfhnfMjtXw

Sumathi, S., & Esakkirajan, S. (2007). Data mining and data warehousing. In *Studies in Computational Intelligence* (Vol. 47). https://doi.org/10.1007/978-3-540-48399-1_10