Volume 3 No. 2, Agustus 2019 P-ISSN 1907-1205
Universitas Banten Jaya E-ISSN 2622-6391

PERBANDINGAN ALGORITMA NAÏVE BAYES DAN SUPPORT VECTOR MACHINE (SVM) DALAM KLASIFIKASI SMS *SPAM* BERBAHASA INDONESIA

Widyawati[1], Sutanto[2]

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Banten Jaya
Jl. Syeh Nawawi Albantani, Curug, Serang – Banten
email: widyawati.astrabuwono@gmail.com[1], sutanto@unbaja.ac.id[2]

ABSTRACT

SMS Spam (Short Message Service Spam) is an unwanted message, including advertisements and fraud. The direct impact of the spam is the inconvenience on the receiver side, therefore there is a need to have a spam screening process. One of the possible approach is to filter the spam by classifying the messages. In this paper, we compare classification performance by Naïve Bayes and Support Vector Machine (SVM). The process include preprocessing (tokenizing, stopwords removal, and stemming) from each training and testing dataset before process the information through Naïve Bayes and SVM. The results from the processing data of 1143 records (765 training and 378 testing) showed that Naïve Bayes performance surpassed SVM in term of recall (94%) and Precision (95%).

Keywords: *Classification, Naïve Bayes, SMS Spam, Support Vector Machine (SVM)*

PENDAHULUAN

Menurut Rahmayani (2019) Jumlah penduduk Indonesia yang mencapai 250 juta jiwa merupakan pasar yang besar. Perkembangan yang sangat pesat dalam penggunaan smartphone. Berdasarkan Lembaga riset digital marketing Emarketer memperkirakan pada tahun 2018 jumlah pengguna aktif *smartphone* di Indonesia lebih dari 100 juta orang. Dengan jumlah sebesar itu, Indonesia akan menjadi negara dengan pengguna aktif smartphone terbesar keempat di dunia setelah Cina, India, dan Amerika. Karena jumlah pengguna yang sangat banyak ini menyebabkan maraknya penipuan melalui SMS berupa SMS *spam* (*spamming*).

Menurut Ma (2016), SMS *Spam* (*Short Message Service Spam*) adalah pesan yang tidak diinginkan atau tidak diminta, termasuk iklan, penipuan dan lain sebagainya. Pesan tersebut sangat menggangu dalam kehidupan keseharian, selain itu mengkonsumsi beberapa sumber daya peralatan komunikasi *mobile* yang digunakan serta menambah antrian penggunaan jaringan komunikasi. Hal ini terjadi dikarenakan SMS merupakan salah satu alat komunikasi yang masih banyak digunakan oleh masyarakat dengan proses yang cepat, mudah dan murah, selain itu set data pengguna *smartphone* saat ini mudah didapatkan.

Dampak langsung dari adanya SMS *spam* ini adalah adanya ketidaknyamanan dari sisi pengguna *provider*, oleh karena itu perlu adanya tahapan dalam melakukan penyaringan SMS *spam* ini. Ada salah satu pendekatan yang mungkin bisa dilakukan dalam penyaringan SMS *spam* yaitu dengan cara mengklasifikasikan SMS *spam.*

Ada berbagai macam algoritma yang digunakan dalam melakukan tahapan klasifikasi menurut Pratiwi (2016) yaitu Naïve Bayes, Support Vector Machine (SVM), k-Nearest Neighbor (KNN) dan lainnya. Menurut Chen, Lu dan Huang (2009), Support Vector Machine (SVM) merupakan salah satu metode terbaik yang dapat digunakan dalam masalah klasifikasi pola. Sedangkan menurut Sebastian Raschka (2014), Naïve Bayes merupakan teknik klasifikasi yang didasarkan pada teorema probabilitas bayes yang popular dan menciptakan model yang sederhana serta berkinerja baik. Penelitian lainnya menunjukan S.L. Ting, W.H. Ip dan Albert H.C. Tsang (2011) melakukan perbandingan algoritma Naïve Bayes dengan beberapa algoritma lain Support Vector Machine (SVM), Neural Network (NN), dan Desicion Tree (DT)) dalam melakukan pengklasifikasian dokumen teks. Hasil yang didapatkan yaitu algoritma Naïve Bayes memberikan nilai akurasi tertinggi sebesar 97.0%, Support Vector Machine memberikan nilai akurasi sebesar 96.9%, Neural Network sebesar 93.0% dan algoritma Desicion Tree memberikan nilai akurasi terendah sebesar 91.1%.

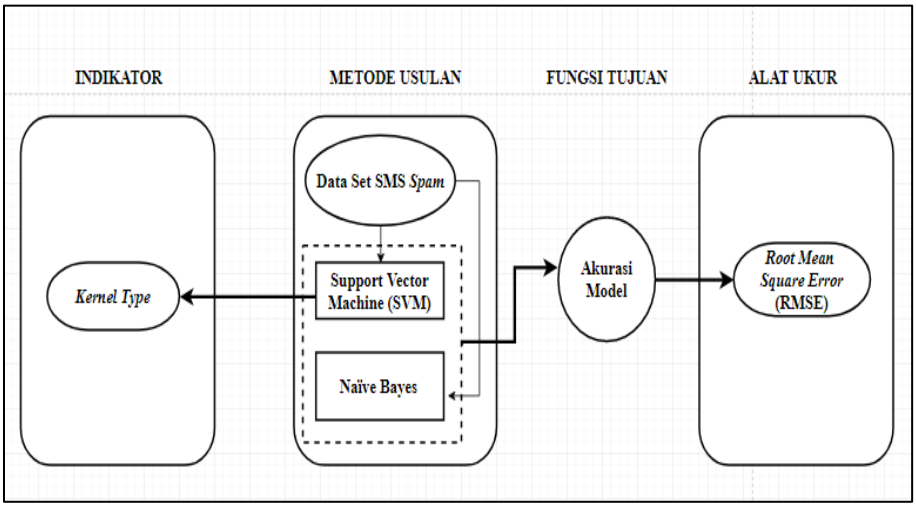
Berdasarkan latar belakang diatas, penulis mengajukan penelitian dengan judul "Perbandingan Algoritma Naïve Bayes dan Support Vector Machine Dalam Klasifikasi SMS *Spam* Berbahasa Indonesia".

METODE

Metode penelitian ini dilakukan dengan beberapa tahap seperti studi literatur, kerangka pemikiran, pengumpulan data, pengolahan data, dan analisis hasil.

1. Kerangka Pemikiran

Berikut ini merupakan kerangka pemikiran yang dilakukan dalam penelitian:



Gambar 1. Kerangka Pemikiran

2. Pengumpulan Data

Menurut Rahmi dan Wibisono (2016) sumber data yang digunakan dalam penelitian ini merupakan data sekunder yang diperoleh dari sumber yang sudah ada yaitu data set SMS *spam* Bahasa Indonesia. Berikut ini merupakan contoh data set yang digunakan didalam penelitian:

Tabel 1. Contoh Data SMS *Spam*

No.	Contoh SMS *Spam*
1	Dptkn Free member card Larissa Aesthetic Center, disc 15% high treatment & asuransi jiwa, tiap isi ulang pulsa Rp 100rb di Galeri Indosat Jateng DIY s.d 31Des 13,2
2	Dptkn kesempatan raih hadiah utama mobil Alphard dg cara tukar 500 poin senyum Anda dg 1 kupon Undian s.d tgl 20Jun13,ketik Undian Jmlh Poin ke 7887(Rp25),2

3	Dukung rehabilitasi & pengembangan potensi anak berkebutuhan khusus di YPAC Jakarta. Mari berbagi melalui SMS Donasi dengan mengetik *123*65#,2
4	Flash Promo s.d 9 Agt! Ayo datangi Gallery Smartfren terdekat, DISKON 290rb dg tukar HP Smartfrenmu ke Andromax 4G LTE atau 190 utk tukar dg MiFi 4G LTE,2

Dengan rincian sebagai berikut:

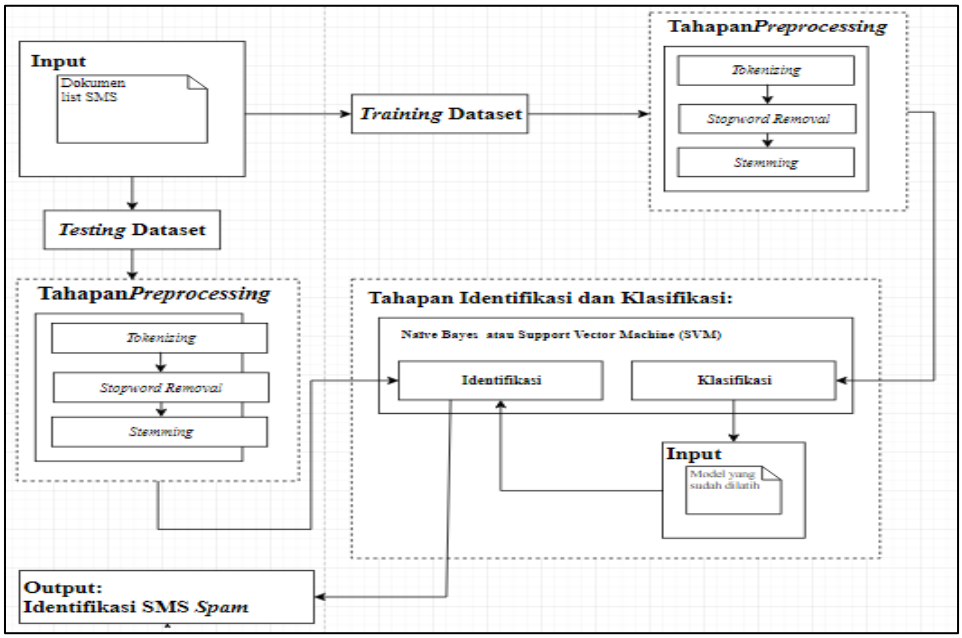
Tabel 2. Rincian dataset SMS *Spam*

Penjelasan Label		Jumlah per kelas:
0	sms normal	569
1	fraud atau penipuan	335
2	promo	239
Jumlah data:		1143

3. Metode Pengolahan Data

Adapun teknik pengolahan data yang dilakukan yaitu dengan menerapkan algoritma Naïve Bayes dan Support Vector Machine (SVM)**.**

Berikut ini merupakan tahapan secara umum dalam melakukan proses pengolahan data:



Gambar 2. Tahapan Pengolahan Data

4. Metode Naïve Bayes

Menurut Santosa (2007), Naïve Bayes merupakan salah satu algoritma *Machine Learning* yang menggunakan konsep probabilitas. Algoritma ini melakukan klasifikasi dengan menghitung nilai probabilitas p(h|x), yaitu probabilitas kelas h jika diketahui suatu b, berdasarkan algoritma NaïveBayes.

Proses klasifikasi dapat dilakukan dengan menentukan nilai suatu kelas $h \in H$ dari suatu dokumen $x \in X$ dengan $H = \{h_1, h_2, h_3, \cdots, h_p\}$ dan $X = \{x_1, x_2, x_3, \cdots, x_q\}$.

Penentuan kelas dalam klasifikasi dokumen tersebut dilakukan dengan cara memilih nilai maximum dari p(h|x), berdasarkan distribusi probabilitas $P = \{p(h|x) \mid h \in H \text{ dan } x \in X\}$.

Suatu dokumen x ke i dapat dipresentasikan sebagai *vector* dan nilai-nilai fitur yang ada pada dokumen tersebut sehingga $x = \{f_{i1}, f_{i2}, f_{i3}, \cdots, f_{in}\}$. Nilai dari elemen tiap vektor merupakan nilai untuk fitur f_j pada himpunan fitur $F = \{f_1, f_2, f_3, \cdots, f_n\}$ dengan f_{ij} merupakan nilai dari fitur ke j pada dokumen x ke i. Berdasarkan algoritma NaïveBayes berikut ini merupakan persamaan perhitungan nilai dari probabilitas p(h|x) :

$$\text{Posterior} = \frac{\text{Likelihood . Prior}}{\text{Evidence}}$$

$$p(h|x) = \frac{p(x|h) \cdot p(h)}{p(x)}$$

Keterangan:

p(h|x) = Nilai posterior atau probabilitas kata h dari kelas x

p(x|h) = Nilai likelihood atau probabilitas kemunculan kelas x untuk kata h

p(h) = Nilai prior atau probabilitas kata h

p(x) = Nilai evidence atau probabilitas kelas x

5. Metode Support Vector Machine (SVM)

Menurut Santosa (2007), Support Vektor Machine (SVM) merupakan suatu teknik yang relatif baru (1995) untuk melakukan prediksi baik dalam kasus klasifikasi maupun

regresi. SVM berada didalam satu kelas dengan Artificial Neural Network (ANN) dalam hal fungsi dan kondisi permasalahan yang bisa diselesaikan. Keduanya termasuk kedalam *supervised learning.* Metode SVM telah banyak diimplementasikan dalam kehidupan sehari-hari diantaranya dalam gane expression, teknik analisis, finansial, cuaca dan bidang kedokteran. Terbukti dari hasil implemantasi tersebut didapatkan bahwa SVM mampu memberikan hasil yang lebih baik dibandingkan dengan ANN. ANN menemukan solusi berupa *local optimum* sedangkan SVM menemukan solusi berupa *global optimum.*

Menurut Feldman dan Sanger (2007) SVM pertama kali diperkenalkan oleh Vapnik pada tahun 1992 sebagai rangkaian harmonis konsep-konsep unggulan dalam bidang *pattern recognition*. Menurut Nugroho dan Witarto (2003) SVM adalah metode *machine learning* yang bekerja atas prinsip *Structural Risk Minimization* (SRM) dengan tujuan menemukan hyperplane terbaik yang memisahkan dua buah class pada input space. Menurut Miner,at al Usaha untuk mencari lokasi hyperplane ini merupakan inti dari proses pembelajaran pada SVM. Pada penelitian ini fungsi yang digunakan untuk mencari hyperplane memenuhi persamaan berikut:

$$\text{f(x)} = \sum_{i=1}^{SV} \alpha_i . K(SV_i, x) + b$$

Dengan keterangan :

αi = fungsi kernel yang digunakan

(SV_i, x) = fungsi kernel yang digunakan

b = error atau bias

DISKUSI

Penelitian yang dilakukan dalam penerapan perbandingan algoritma Naïve Bayes dan Support Vector Machine (SVM) dalam klasifikasi SMS *Spam* Berbahasa Indonesia menggunakan bantuan perangkat lunak Python versi 3.7.

Berikut ini merupakan penjelasan dari setiap tahapan dalam proses pengolahan data:

1. Input Data (Seleksi Data)

Tahap awal dari perancangan penelitian ini adalah tahap seleksi data, yaitu pemilihan atribut yang nantinya dilakukan proses transformasi data dari data mentah kedalam bentuk tabel matriks untuk diolah ke tahap selanjutnya. Dimana data yang diolah adalah data set dari SMS *spam* dengan berbagai jenis seperti 0 untuk SMS normal, 1 untuk SMS penipuan, dan 2 untuk SMS promo.

2. *Training Dataset*

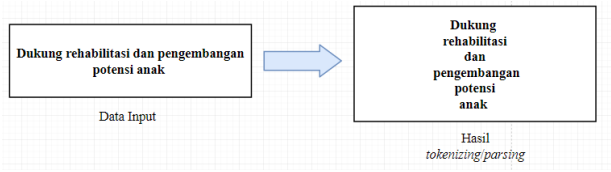
Data yang digunakan dalam proses *training* dataset adalah sebesar 67% dari jumlah data yang digunakan, yaitu 67%x1143 data = 765,81data ≈ 765 dataset *training*.

3. Tahapan *Preprocessing*

Preprocessing merupakan tahapan awal dalam mengolah data input sebelum memasuki proses tahapan utama dari penerapan algoritma Naïve Bayes dan SVM. Tujuannya adalah untuk penyeragaman dan kemudahan dalam proses pembacaan. Terdapat tiga tahapan didalam tahapan *preprocessing* yaitu *tokenizing*, *stopword removal*, dan *stemming*.

Tokenizing*/*Parsing

Menurut Triawati (2009) Tahap tokenizing/parsing adalah tahap pemotongan kata berdasarkan tiap kata yang menyusunnya. Selain itu, spasi digunakan untuk memisahkan antar kata.

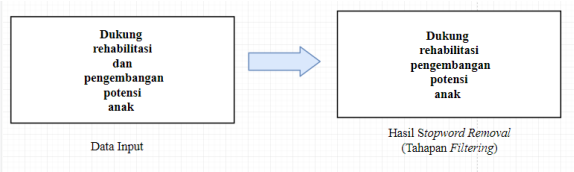


Gambar 3. Tahapan *Tokenizing*/*Parsing*

***Stopword Removal* (Tahapan *Filtering*)**

Menurut Triawati (2009) *Stoplist* /*stopword* adalah kata-kata yang tidak deskriptif yang dapat dibuang dalam pendekatan *bag-of-words.* Contoh *stopword* adalah "yang", "dan", "di", "dari" dan lain–lain. Dimana tahapan stopword ini bisa digunakan sebagai tahap *filtering*. *Filtering* merupakan tahapan mengambil kata-kata penting saja dari hasil

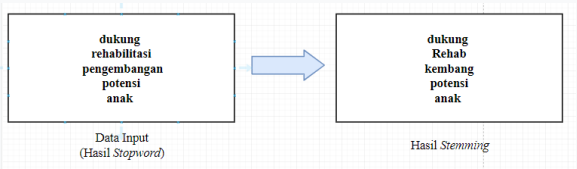
tokenizing, dengan menggunakan algoritma *stoplist* (membuang kata yang kurang penting) atau *wordlist* (menyimpan kata penting).



Gambar 4. Tahapan *Stopword Removal Filtering*)

Stemming

Menurut Agastya (2018) *Stemming* adalah proses untuk mendapatkan basis atau akar kata dengan menghilangkan imbuhan dan sufiks.



Gambar 5. Tahapan *Stemming*

4. Testing *Dataset*

Data yang digunakan dalam proses *testing* dataset adalah sebesar 33% dari jumlah data yang digunakan, yaitu 30%x1143 data = 378 dataset *training*.

5. Klasifikasi

Tahapan klasifikasi merupakan tahapan yang menggunakan data training yang sebelumnya telah dilakukan *preprocessing* seperti *tokenizing, stopword removal,* dan *stemming.* Dimana data tersebut sebagai masukan kedalam tahapan selanjutnya yaitu tahapan klasifikasi. Didalam tahapan klasifikasi terdapat proses k-*fold cross validation*, tahapan ini membagi data training tersebut kedalam 10 *fold* (k=10) yang bertujuan agar hasil klasifikasi yang dilakukan memiliki hasil yang lebih akurat. Dimana setiap *fold* tersebut dilakukan proses pengolahan data menggunakan metode Naïve Bayes atau *Support Vector Machine* (SVM). Data tersebut akan dijadikan sebagai data seleksi, yang akan dijadikan sebagai data pembanding dalam tahapan identifikasi data *testing*.

6. Identifikasi

Identifikasi merupakan tahapan dimana data *testing* yang telah dilakukan tahapan *preprocessing* sebelumnya akan ditesting dan diidentifikasi terhadap data *training* yang sudah ada apakah data tersebut apakah termasuk kedalam kategori SMS *spam*.

7. Output

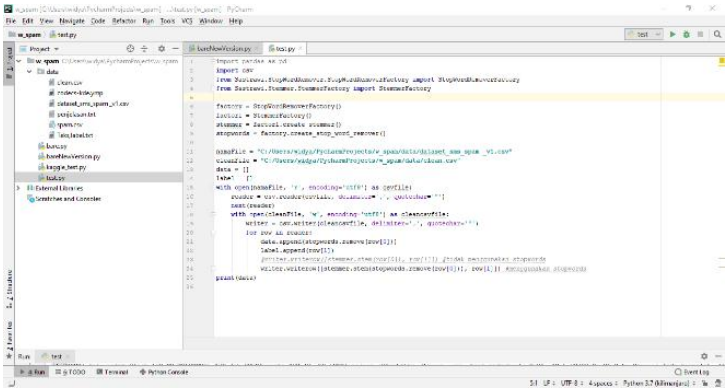
Output merupakan hasil yang menunjukan apakah data testing tersebut termasuk kedalam kategori SMS Spam dan termasuk kategori SMS spam 0, 1 atau 2 (0 adalah normal, 1 adalah penipuan, dan 2 adalah promo).

8. Proses Pengolahan Data

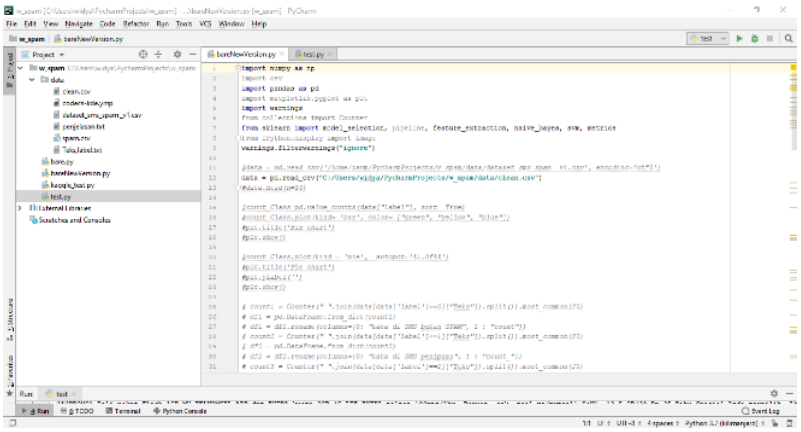
Dataset *training* dan *testing* yang digunakan berjumlah 1143 (765 dataset *training* dan 378 dataset *testing*) diolah menggunakan aplikasi Python versi 3.7. Dalam proses *preprocessing* data set baik *training* maupun *testing* akan dilakukan perbandingan bagaimana hasil menggunakan tahapan *stopword removal* ataupun tidak. Berikut hasil dari pengolahan data yang dilakukan menggunakan aplikasi python:

Stopword Removal

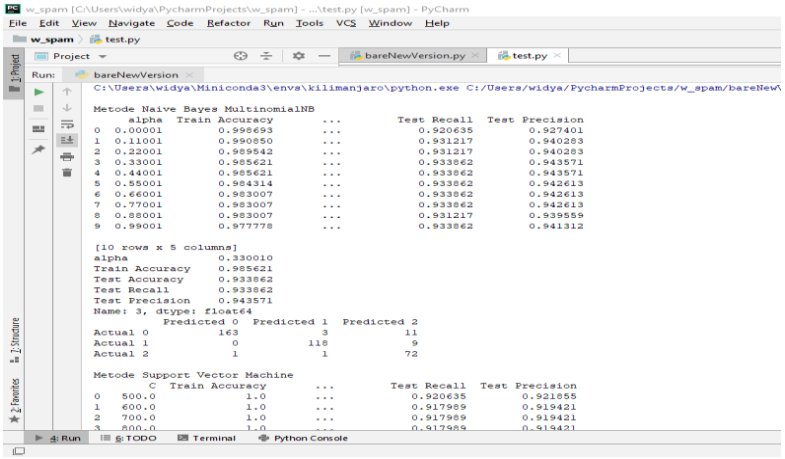
Berikut ini merupakan proses dalam pengolahan data menggunakan tahapan *preprocessing* dengan menambahkan tahapan *stopword removal.*



Gambar 6. Proses Pengolahan Data Naïve Bayes Menggunakan *Stopword Removal*

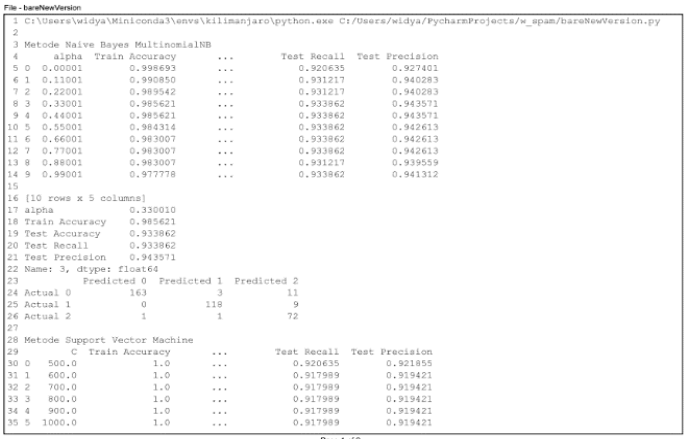


Gambar 7. Proses Pengolahan Data SVM Menggunakan *Stopword Removal*



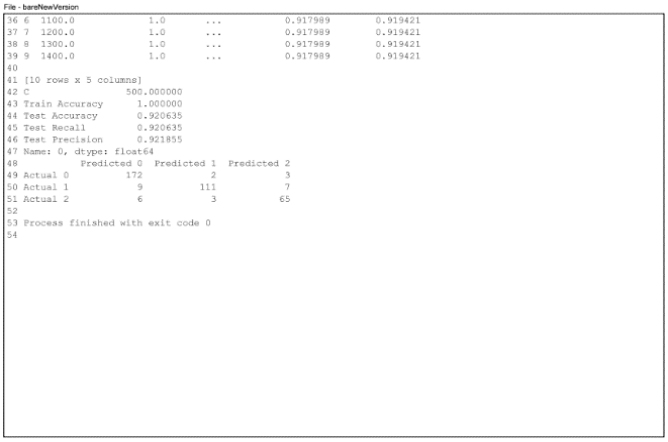
Gambar 8. Tampilan Keluaran Menggunakan *Stopword Removal*

Berikut ini merupakan hasil keluaran (*output*) dari proses menjalankan program klasifikasi di aplikasi Python dalam pengolahan data menggunakan tahapan *preprocessing* dengan menambahkan tahapan *stopword removal*.



```
File - bareNewVersion
C:\Users\widya\Miniconda3\envs\killimanjaro\python.exe C:/Users/widya/PycharmProjects/w_spam/bareNewVersion.py

Metode Naive Bayes MultinomialNB
      alpha  Train Accuracy       ...        Test Recall  Test Precision
0  0.00001        0.998693       ...           0.920635        0.927401
1  0.11001        0.990850       ...           0.931217        0.940283
2  0.22001        0.989542       ...           0.931217        0.940283
3  0.33001        0.985621       ...           0.933862        0.943571
4  0.44001        0.985621       ...           0.933862        0.943571
5  0.55001        0.984314       ...           0.933862        0.942613
6  0.66001        0.983007       ...           0.933862        0.942613
7  0.77001        0.983007       ...           0.933862        0.942613
8  0.88001        0.983007       ...           0.931217        0.939559
9  0.99001        0.977778       ...           0.933862        0.941312

[10 rows x 5 columns]
alpha             0.330010
Train Accuracy    0.985621
Test Accuracy     0.933862
Test Recall       0.933862
Test Precision    0.943571
Name: 3, dtype: float64
           Predicted 0  Predicted 1  Predicted 2
Actual 0           163            3           11
Actual 1             0          118            9
Actual 2             1            1           72

Metode Support Vector Machine
         C  Train Accuracy       ...        Test Recall  Test Precision
0   500.0             1.0       ...           0.920635        0.921855
1   600.0             1.0       ...           0.917989        0.919421
2   700.0             1.0       ...           0.917989        0.919421
3   800.0             1.0       ...           0.917989        0.919421
4   900.0             1.0       ...           0.917989        0.919421
5  1000.0             1.0       ...           0.917989        0.919421
```

Gambar 9. Keluaran Pengolahan Data Naïve Bayes Menggunakan *Stopword Removal*

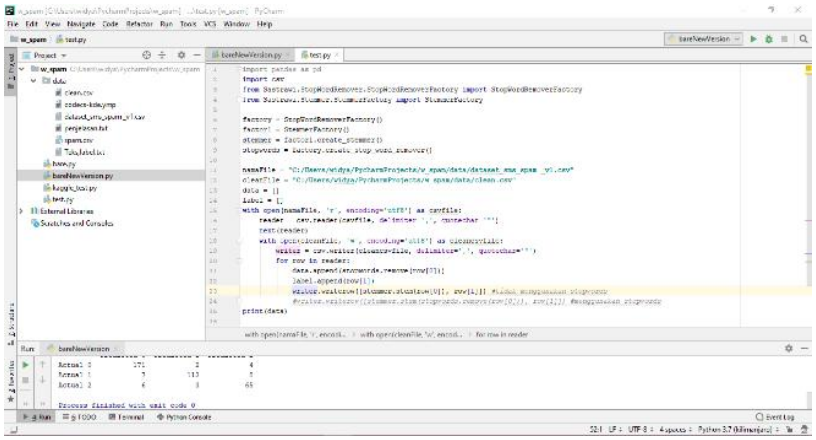


```
File - bareNewVersion
6  1100.0             1.0       ...           0.917989        0.919421
7  1200.0             1.0       ...           0.917989        0.919421
8  1300.0             1.0       ...           0.917989        0.919421
9  1400.0             1.0       ...           0.917989        0.919421

[10 rows x 5 columns]
C                 500.000000
Train Accuracy      1.000000
Test Accuracy       0.920635
Test Recall         0.920635
Test Precision      0.921855
Name: 0, dtype: float64
           Predicted 0  Predicted 1  Predicted 2
Actual 0           172            2            3
Actual 1             9          111            7
Actual 2             6            3           65

Process finished with exit code 0
```

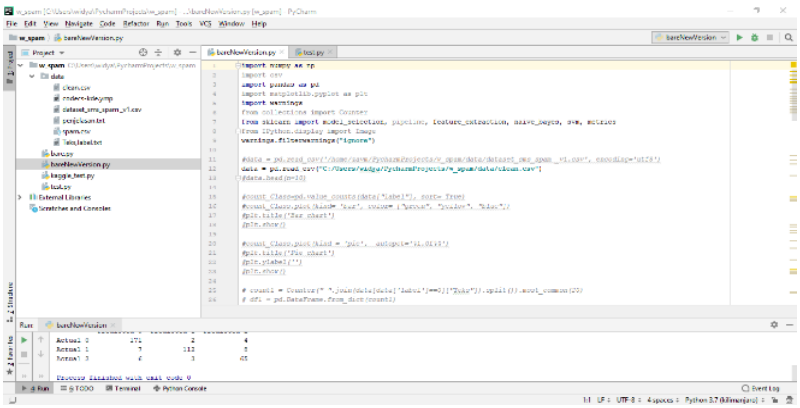
Gambar 10. Keluaran Pengolahan Data SVM Menggunakan *Stopword Removal*

Tanpa *Stopword Removal*

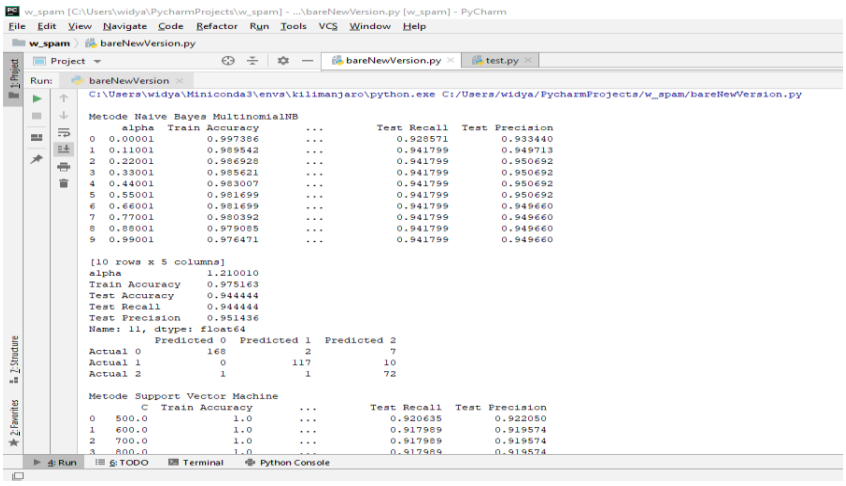
Berikut ini merupakan proses dalam pengolahan data menggunakan tahapan *preprocessing* dengan **tanpa** menambahkan tahapan *stopword removal*



Gambar 11. Proses Pengolahan Data Naïve Bayes Tanpa Menggunakan *Stopword Removal*



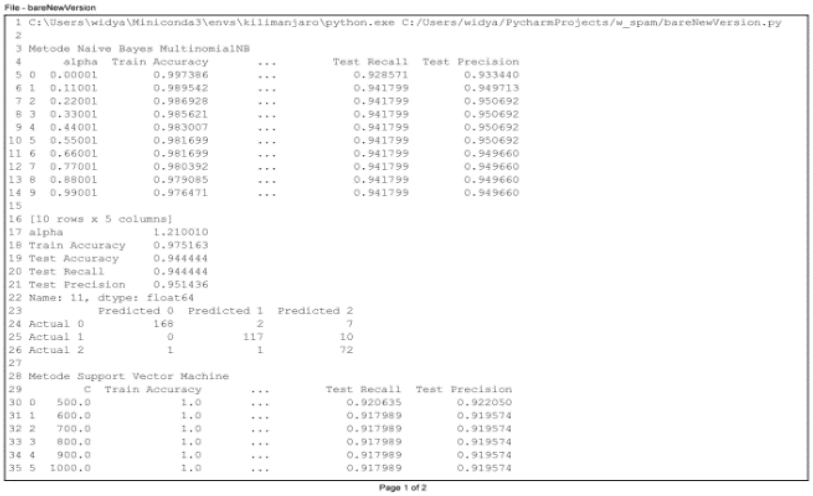
Gambar 12. Proses Pengolahan Data SVM Tanpa Menggunakan *Stopword Removal*



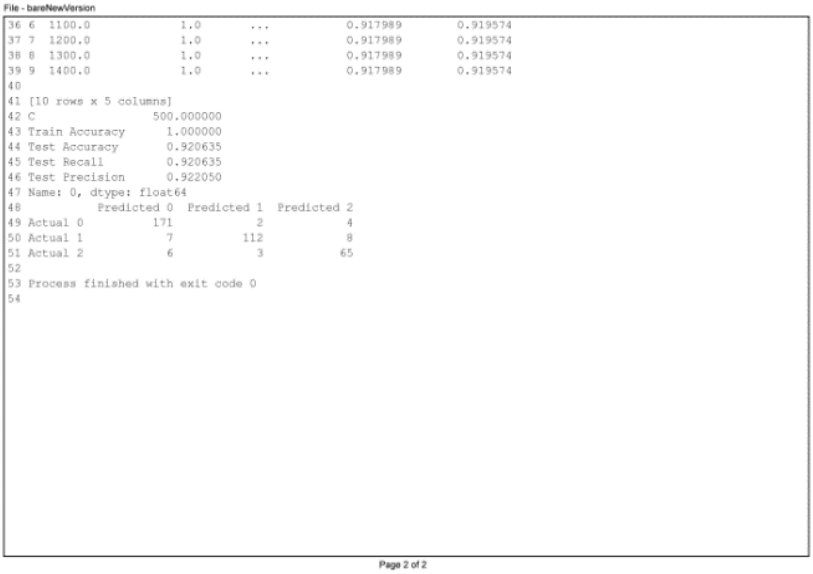
```
C:\Users\widya\Miniconda3\envs\kilimanjaro\python.exe C:/Users/widya/PycharmProjects/w_spam/bareNewVersion.py

Metode Naive Bayes MultinomialNB
      alpha  Train Accuracy       ...        Test Recall  Test Precision
0   0.00001         0.997386       ...           0.928571        0.933440
1   0.11001         0.989542       ...           0.941799        0.949713
2   0.22001         0.986928       ...           0.941799        0.950692
3   0.33001         0.985621       ...           0.941799        0.950692
4   0.44001         0.983007       ...           0.941799        0.950692
5   0.55001         0.981699       ...           0.941799        0.950692
6   0.66001         0.981699       ...           0.941799        0.949660
7   0.77001         0.980392       ...           0.941799        0.949660
8   0.88001         0.979085       ...           0.941799        0.949660
9   0.99001         0.976471       ...           0.941799        0.949660

[10 rows x 5 columns]
alpha             1.210010
Train Accuracy    0.975163
Test Accuracy     0.944444
Test Recall       0.944444
Test Precision    0.951436
Name: 11, dtype: float64
           Predicted 0  Predicted 1  Predicted 2
Actual 0           168            2            7
Actual 1             0          117           10
Actual 2             1            1           72

Metode Support Vector Machine
      C  Train Accuracy       ...        Test Recall  Test Precision
0   500.0             1.0       ...           0.920635        0.922050
1   600.0             1.0       ...           0.917989        0.919574
2   700.0             1.0       ...           0.917989        0.919574
3   800.0             1.0       ...           0.917989        0.919574
```

Gambar 13. Tampilan Keluaran Tanpa Menggunakan *Stopword Removal*

Berikut ini merupakan hasil keluaran (*output*) dari proses menjalankan program klasifikasi di aplikasi Python dalam pengolahan data menggunakan tahapan *preprocessing* **tanpa** menambahkan tahapan *stopword removal.*



Gambar 14. Keluaran Pengolahan Data Naïve Bayes Tanpa Menggunakan *Stopword Removal*



Gambar 15. Keluaran Pengolahan Data SVM Tanpa Menggunakan *Stopword Removal*

9. Analisa

Berikut ini merupakan tabel mengenai rincian keluaran hasil pengolahan klasifikasi dataset SMS *spam*, yang dioleh menggunakan algoritma Naïve Bayes dan SVM. Hasil tersebut menunjukan perbedaan dengan atau tanpa tahapan *stopword removal* pada tahapan *preprocessing.*

Tabel 3. Rincian dataset SMS *Spam*

		Stopword	**Tanpa** *Stopword*
Naïve Bayes	*Test Recall*	0,933862	0,944444
	Test Precision	0,943571	0,951436
SVM	*Test Recall*	0,920635	0,920635
	Test Precision	0,921855	0,922050

Tabel 4. Rincian dataset SMS *Spam*

			Predicted 0	*Predicted 1*	*Predicted 2*
Stopword	**Naïve Bayes**	*Actual 0*	163	3	11
		Actual 1	0	118	9
		Actual 2	1	1	72
	SVM	*Actual 0*	172	2	3
		Actual 1	9	111	7
		Actual 2	6	3	65
Tanpa *Stopword*	**Naïve Bayes**	*Actual 0*	168	2	7
		Actual 1	0	117	10
		Actual 2	1	1	72
	SVM	*Actual 0*	171	2	4

	Actual 1	7	112	8
	Actual 2	6	3	65

Berdasarkan Tabel 3 nilai dari *test recall* sebesar 93,3862 % , *test precision* sebesar 94,3571% untuk pengolahan data klasifikasi SMS *spam* menggunakan Naïve Bayes melalui tahapan *stopworld,* sedangkan nilai *test recall* sebesar 94,4444 % , *test precision* sebesar 95,1436 % untuk pengolahan data klasifikasi SMS *spam* menggunakan Naïve Bayes **tanpa** melalui tahapan *stopworld*. Dimana nilai keluaran dari proses yang melalui tahapan *stopword* memiliki nilai lebih rendah dibandingkan tanpa melalui tahapan *stopword.* Untuk proses pengolahan data menggunakan algortima SVM melalui tahapan *stopword* memiliki nilai *test recall* sebesar 92,0635% dan *test Precision* sebesar 92,1855% sedangkan **tanpa** melalui tahapan *stopword* nilai *test recall* sebesar 92,0635 % dan *test precision* sebesar 92,2050%.

Berdasarkan tabel 4 didapatkan bahwa data aktual untuk SMS *spam* dataset *testing* dengan jenis 0 atau SMS normal adalah sebesar 177, 1 atau SMS penipuan sebanyak 127, dan 2 atau SMS promo sebanyak 74 pesan. Dari tabel juga didapatkan bahwa adanya kesalahan prediksi sebanyak 25 pesan setelah dilakukan klasifikasi menggunakan algoritma Naive Bayes dengan tahapan *stopword*, 30 pesan setalah dilakukan klasifikasi menggunakan algoritma SVM dengan tahapan *stopword*, 21 pesan setelah dilakukan tahapan klasifikasi menggunakan algoritma Naive Bayes **tanpa** melalui tahapan *stopword*, namun pada proses klasifikasi dengan SVM tanpa melalui tahapan *stopword* memang memiliki nilai yang sama sebanyak 30 pesan seperti dengan tahapan *stopword* namun jumlah sebarannya berbeda

KESIMPULAN

Adapun kesimpulan yang didapatkan berdasarkan pengolahan data *testing* untuk penerapan algoritma Naïve Bayes dan Support Vector Machine (SVM) pada klasifikasi SMS adanya perbedaan nilai pada *test recall* dan *test precision.* Dimana nilai *test recall* dan

test precision juga menunjukan nilai yang berbeda berdasarkan *treatment* yang diberikan (penggunaan *stopword removal* atau **tidak** pada tahapan *preprocessing*).

Recall dan *precision* adalah dua perhitungan yang banyak digunakan untuk mengukur kinerja dari sistem/metode yang digunakan. *Recall* adalah tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi**.** Sedangkan *precision* adalah seberapa besar tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem. Dari hasil Analisa didapatkan bahwa nilai *test recall* dan *test precision* terbesar dan terbaik adalah jika data training dan testing diklasifikasikan menggunakan algoritma Naïve Bayes **tanpa** melalui *stopword removal,* hal ini dikarenakan jika melalui tahapan *stopword* maka akan adanya kemungkinan merubah isi konteks yang pada pada dasarnya akan memiliki arti berbeda bahkan berlawanan arah. Sebagai contoh: menghilangkan kata tidak pada konteks "syarat dan ketentuan tidak berlaku" ketika menggunakan tahapan *stopword* maka konteksnya akan berubah menjadi "syarat dan ketentuan berlaku", hal ini yang menjadikan bahwa nilai *test recall* dan *test precision* tanpa tahapan *stopword* menunjukan nilai terbaik.

. Berdasarkan hasil Analisa juga didapatkan kesimpulan bahwa kesalahan klasifikasi awal dari data aktual paling sedikit dilakukan oleh Naïve Bayes baik menggunakan tahapan *stopword* maupun **tanpa** *stopword*.

DAFTAR PUSTAKA

Agastya, M. (2018). Pengaruh Stemmer Bahasa Indonesia Terhadap Performa Analisis Sentimen Terjemahan Ulasan FIlm. *Jurnal TEKNOKOMPAK, Vol. 12, No. 1, 2018, 18-23. ISSN 1412-9663 (print)*, 18-23.

Juang, D. (2016). Analisis Spam dengan menggunakan Naive Bayes . *Jurnal Teknovasi Volume 03, Nomor 2, 2016, 51 – 57 ISSN : 2355-701X* , 51-57.

Ma, J., Zhang, Y., Liu, J., & Yu, K. (2016). Intelligent SMS Spam Filtering Using Topic Model. *2016 International Conference on Intelligent Networking and Collaborative Systems*, 380-383.

Pratiwi, S., & Ulama , B. (2016). Klasifikasi Email Spam dengan Menggunakan Metode Support Vector Machine dan k-Nearest Neighbor. *JURNAL SAINS DAN SENI ITS Vol. 5 No. 2 (2016) 2337-3520 (2301-928X Print)* , D-344 - D-349.

Rahmayani, I. (2019, Juli Sabtu). *https://kominfo.go.id/content/detail/6095*. Retrieved from https://kominfo.go.id: https://kominfo.go.id/content/detail/6095/indonesia-raksasa-teknologi-digital-asia/0/sorotan_media

Rahmi, F., & Wibisono, Y. (2016, Juli Sabtu). *Aplikasi SMS Spam Filtering pada Android menggunakan Naive Bayes, Unpublished manuscript*. Retrieved from http://nlp.yuliadi.pro: http://nlp.yuliadi.pro/dataset

Raschka, S. (2014, Juli Sabtu). *https://sebastianraschka.com/Articles*. Retrieved from https://sebastianraschka.com: https://sebastianraschka.com/Articles/2014_naive_bayes_1.html

Santosa, B. (2007). *Data Mining Teknik Pemanfaatan Data Untuk Keperluan Bisnis.* Yogyakarta: Graha Ilmu.

Ting, S., Ip, W., & Tsang , A. (3, July, 2011). Is Naïve Bayes a Good Classifier for Document Classification? *International Journal of Software Engineering and Its Applications Vol. 5, No.* , 37-46