

# Perbandingan Klasifikasi SMS Berbasis Support Vector Machine, Naive Bayes Classifier, Random Forest dan Bagging Classifier

Devi Irawan<sup>[1]\*</sup>, Eza Budi Perkasa<sup>[2]</sup>, Yurindra<sup>[3]</sup>, Delpiah Wahyuningsih<sup>[4]</sup>, Ellya Hel mud<sup>[5]</sup>

Fakultas Teknologi Informasi<sup>[1], [2], [3], [4], [5]</sup>

Institut Sains dan Bisnis Atma Luhur

Pangkalpinang, Indonesia

[deviirawan@atmaluhur.ac.id](mailto:deviirawan@atmaluhur.ac.id)<sup>[1]</sup>, [ezabudiperkasa@atmaluhur.ac.id](mailto:ezabudiperkasa@atmaluhur.ac.id)<sup>[2]</sup>, [yurindramail@gmail.com](mailto:yurindramail@gmail.com)<sup>[3]</sup>, [delphibabel@atmaluhur.ac.id](mailto:delphibabel@atmaluhur.ac.id)<sup>[4]</sup>

[,ellyahel mud@atmaluhur.ac.id](mailto:ellyahel mud@atmaluhur.ac.id)<sup>[5]</sup>

**Abstract**— Short message service (SMS) is one of the important communication media to support the speed of using mobile phones by users. The SMS classification hybrid system is used to detect SMS that are considered junk and correct. In this research, what is needed is to collect SMS datasets, feature selection, preprocessing, vector creation, filtering and updating the system. Two types of SMS classification on mobile phones are currently listed as blacklisted (rejected) and whitelisted (accepted). This study uses several algorithms such as support vector machine, Naive Bayes classifier, Random Forest and Bagging Classifier, which aims to produce an algorithm that has the highest performance score and is accurate in filtering incoming SMS. In this study, it was found that the Bagging classifier algorithm got the highest performance score from other algorithms which can be used as a means to filter incoming SMS into the user's inbox and the Bagging classifier algorithm so as to provide accurate filtration results to filter incoming SMS.

**Keywords** — Data Processing, Tin Ore, Rocking Table, System Monitoring and Optimization

**Abstrak**— Short message service (SMS) adalah salah satu media komunikasi yang penting untuk mendukung kecepatan penggunaan ponsel oleh pengguna. Sistem hibrid klasifikasi SMS digunakan untuk mendeteksi sms yang dianggap sampah dan benar. Dalam penelitian ini yang diperlukan adalah mengumpulkan dataset SMS, pemilihan fitur, prapemrosesan, pembuatan vektor, melakukan penyaringan dan pembaharuan sistem. Dua jenis klasifikasi SMS pada ponsel saat ini ada yang terdaftar sebagai daftar hitam (ditolak) dan daftar putih (diterima). Penelitian ini menggunakan beberapa algoritma seperti support vector machine, Naive Bayes classifier, Random Forest dan Bagging Classifier. Tujuan dari penelitian ini adalah untuk menyelesaikan semua masalah SMS yang teridentifikasi spam yang banyak terjadi pada saat ini sehingga dapat memberikan masukan dalam perbandingan metode yang mampu menyaring dan memisahkan sms spam dan sms non spam. Pada penelitian ini menghasilkan bahwa Bagging classifier algorithm ini mendapatkan ferformance score tertinggi dari algoritma yang lain yang dapat dipergunakan sebagai sarana untuk memfiltrasi SMS yang masuk ke dalam inbox pengguna dan Bagging classifier algorithm dapat memberikan hasil filtrasi yang akurat untuk menyaring SMS yang masuk.

**Kata Kunci**— Short Message Service (SMS); Support Vector

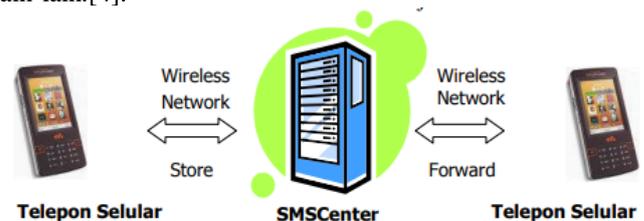
*Machine; Naive Bayes; Random Forest; Bagging Classifier; ham; spam.*

## I. PENDAHULUAN

Short Message Service atau SMS merupakan sebuah media komunikasi berbentuk teks yang memungkinkan pengguna ponsel untuk saling berbagi teks[1]. Meningkatnya komunikasi melalui SMS telah banyak dimanfaatkan oleh pihak-pihak tertentu untuk mengirimkan pesan-pesan yang tidak bermanfaat atau SMS spam [2]. SMS spam adalah pesan yang tidak diinginkan, yang kita tidak ingin pesan tersebut berada di dalam kotak pesan kita [3].

Dalam sistem SMS, mekanisme utama yang dilakukan dalam suatu sistem adalah melakukan pengiriman *short message* dari satu terminal customer ke terminal yang lain. Hal ini dapat dilakukan berkat adanya sebuah entitas dalam sistem SMS yang bernama *Short Message Service Center* (SMSC), disebut juga *Message Center* (MC). Pada saat pesan SMS dikirim dari *handphone* (*mobile orginated*) pesan tersebut tidak langsung dikirim ke *handphone* tujuan (*mobile terminated*), akan tetapi terlebih dahulu ke SMSC, baru kemudian pesan tersebut dikirimkan ke *handphone* tujuan.

SMSC merupakan sebuah perangkat yang melakukan tugas *store and forward trafik short message*. Di dalamnya termasuk penentuan atau pencarian rute tujuan akhir dari *short message*. Sebuah SMSC biasanya didesain untuk dapat menangani *short message* dari berbagai sumber seperti *Voice Mail System* (VMS), *Web-based messaging*, *Email Integration*, *External Short Message Entities* (ESME), dan lain-lain.[4].



Gambar 1. Skema Cara Kerja SMS

II. TINJAUAN PUSTAKA

Kebutuhan informasi merupakan kebutuhan mendasar setiap orang untuk melakukan komunikasi. Salah satu cara untuk penyebaran informasi adalah melalui pengiriman *Short Message Service* (SMS). SMS merupakan data dengan tipe *asynchronous message* yang pengiriman datanya dilakukan dengan mekanisme protokol *store and forward*. Hal ini berarti bahwa pengirim dan penerima SMS tidak perlu berada dalam status berhubungan (*connected/online*) satu sama lain ketika akan saling bertukar pesan SMS. Pengiriman pesan SMS secara *store and forward* berarti pengirim pesan SMS menuliskan pesan dan nomor telepon tujuan dan kemudian mengirimkannya (*store*) ke server SMS atau *SMS-Center*. *SMS-Center* (SMSC) bertanggung jawab untuk menangani sebuah pesan SMS pada jaringan nirkabel (*wireless*) yang kemudian berikutnya bertanggung jawab untuk mengirimkan pesan tersebut (*forward*) ke nomor telepon tujuan [5].

Akhir – akhir ini banyak *spamer* yang muncul sehingga jumlah SMS *spam* menjadi sangat tinggi. Akibatnya, banyak pihak yang dirugikan, misalnya pengguna SMS dan masyarakat umum [6]. Karena rata-rata SMS *spam* berisi pemasaran (promosi), penipuan, dan ancaman maka masyarakat menjadi terganggu saat menerima sms *spam* bahkan ada yang menjadi korban penipuan setelah merespon sms tersebut. Penerima sms *spam* akan melakukan hal yang diminta oleh pengirim sms misalnya mengirimkan pulsa padahal itu adalah penipuan sehingga penerima sms *spam* akan mengalami kerugian finansial setelah mengirimkan pulsa yang diminta.

Klasifikasi SMS dapat menentukan kumpulan data set untuk satu set kelas yang telah ditetapkan. Teknik klasifikasi akan disusun dalam hirarki kategori pesan SMS *spam* yang harus disaring, dan berita yang akan disampaikan ke pengguna [6]. Dalam klasifikasi *Support Vektor Machine*, *Naive Bayes Classifier*, *Random Forest* dan *Bagging Classifier*. semua kata di sebuah SMS yang diberikan saling independen. Ini adalah bentuk paling sederhana dari jaringan semua metode yang digunakan yang dapat diartikan sebagai bersyarat independen. Dalam pengklasifikasian SMS yang telah dimasukkan kata-kata secara efektif yang digunakan untuk menghasilkan *short message service* dianggap benar ataupun di anggap tidak benar dari keseluruhan metode yang digunakan.

Berdasarkan latar belakang yang telah diuraikan, maka permasalahan yang didapat adalah sebagai berikut : Bagaimana memilih algoritma terbaik diantaranya *support vector machine*, *naive bayes classifier*, *random forest* dan *bagging classifier* untuk dapat mengidentifikasi *short message service* yang dianggap benar dan tidak benar ?

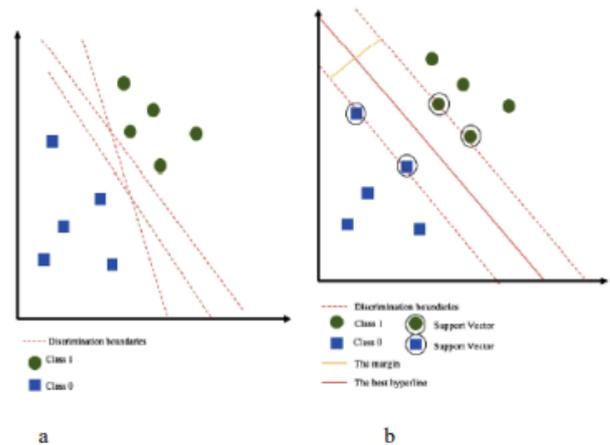
Tujuan dari penelitian ini adalah untuk menyelesaikan semua masalah SMS yang teridentifikasi *spam* yang banyak terjadi pada saat ini sehingga dapat memberikan masukan dalam perbandingan metode yang mampu menyaring dan memisahkan sms *spam* dan sms *non spam*.

Adapun dalam penelitian ini, menggunakan 4 algoritma, sebagai berikut:

1. *Support Vector Machine*

*Support Vector Machine* pertama kali diperkenalkan oleh Vapnik pada tahun 1992 sebagai rangkaian harmonis konsep-konsep unggulan dalam bidang *pattern recognition* [7]. SVM adalah algoritma *machine learning* yang bekerja atas prinsip *Structural Risk Minimization* (SRM) dengan tujuan menemukan *hyperplane* terbaik yang memisahkan dua buah *Class* pada *input space*.

Pengertian *Support Vector Machine* (SVM) yaitu sistem pembelajaran yang menggunakan ruang hipotesis berupa fungsi – fungsi linier dalam sebuah fitur yang berdimensi tinggi dan dilatih dengan menggunakan algoritma pembelajaran yang didasarkan pada teori optimasi. SVM pertama kali diperkenalkan pada tahun 1992 oleh Vapnik sebagai rangkaian dari beberapa konsep – konsep unggulan dalam bidang *pattern recognition* [8].



Gambar 2. Model *support vector machine*

SVM merupakan algoritma klasifikasi yang menggunakan ruang hipotesis berupa fungsi-fungsi linear dalam sebuah ruang berdimensi fitur tinggi. SVM memiliki tujuan menemukan fungsi pemisah terbaik antara kelas. Fungsi pemisah paling baik apabila fungsi pemisah yang menghasilkan nilai margin paling besar antara dua vektor dari dua kelas yang berbeda dan berada ditengah-tengah kedua vektor tersebut. Dalam penelitian ini, fungsi pemisah yang dicari adalah fungsi linear. Fungsi linear dapat dicari dengan persamaan berikut [9].

$$g(x) = \omega \cdot x + w_0$$

Dimana :

- $\omega$  : vektor untuk *hyperplane*
- $x$  : nilai vektor masukan

$W_0$  : bias yang merepresentasikan posisi relatif terhadap pusat koordinat

Pada langkah awal dalam proses SVM akan dilakukan *input* data dari *database*. Setelah itu data akan dihitung menggunakan *kernel polynomial* untuk diteruskan ke proses *training* dengan menggunakan proses *Sequential Training SVM*. *Output* yang akan dihasilkan adalah berupa hasil klasifikasi, yang berupa perbandingan dari hasil klasifikasi system dengan hasil dari pakar [9].

SVM memiliki kelebihan yaitu mampu mengidentifikasi *hyperplane* terpisah yang memaksimalkan margin antara dua kelas yang berbeda. Namun *Support Vector Machine* memiliki kekurangan terhadap masalah pemilihan parameter atau fitur yang sesuai. Pemilihan fitur sekaligus penyetingan parameter di SVM secara signifikan mempengaruhi hasil akurasi klasifikasi. Berikut ini adalah perhitungan secara matematis klasifikasi pada algoritma SVM [10].

a. Vektor Dalam SVM, vektor adalah hal yang penting untuk melakukan proses klasifikasi. vektor adalah objek yang memiliki besaran dan arah [10]. Besaran (*The Magnitude Of Vector*) Besaran atau panjangnya vektor  $x$  ditulis dengan  $|x|$  atau disebut dengan norm  $x$ .

$$|x| = \sqrt{x_1^2 + x_2^2 + \dots + x_n^2}$$

Arah (*The Direction Of Vector*) Arah vektor  $u = (u_1, u_2, \dots, u_n)$  adalah vektor  $w = (|u_1|, |u_2|, \dots, |u_n|)$

b. *The Dot Product* Secara geometris, ini adalah perkalian dari besaran Euclidian dari dua vektor dan kosinus sudut diantara vektornya. Yang berarti jika kita memiliki dua vektor  $x$  dan  $y$  dan ada sudut  $\theta$  di antara vektor, perkalian titiknya adalah :  $x \cdot y = |x||y| \cos(\theta)$

c. *Hyperplane SVM* bertujuan untuk menemukan *hyperplane* pemisah yang optimal yang memaksimalkan margin data latih (*training data*). Dalam SVM, *hyperplane* terbaik adalah *hyperplane* berada pada posisi di tengah-tengah antara dua set obyek dari dua kelas. Mencari *hyperplane* terbaik ekuivalen dengan memaksimalkan margin, yaitu jarak tegak lurus antara *hyperplane* dengan obyek terdekat. Obyek terdekat ini dinamakan *support vectors*. Beberapa referensi menyebut margin adalah jarak tegak lurus antara *support vector* dari dua kelas. Dikatakan *hyperplane* terbaik karena memberikan nilai margin terbesar [10]. Fungsi pemisahannya adalah fungsi linier yang didefinisikan sebagai:  $(x) = \text{sign}(f(x))$

2. *Naïve Bayes Classifier*

*Naïves Bayes* “Bayesian klasifikasi adalah pengklasifikasi statistik yang dapat digunakan untuk memprediksi probabilitas keanggotaan suatu *class*. *Bayesian classification* didasarkan pada teorema bayes yang memiliki kemampuan klasifikasi serupa dengan *Decision Tree* dan *Neural Network*. *Bayesian classification* terbukti memiliki akurasi dan kecepatan yang tinggi saat diaplikasikan kedalam *database* dengan data yang besar.”[11].

Teorema Bayes memiliki bentuk umum sebagai berikut :

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)}$$

Yang mana :

X = data dengan class yang belum diketahui

H = hipotesis data X merupakan satu *class* spesifik

P(H|X) = probabilitas hipotesis H berdasar kondisi X (*posteriori probability*)

P(H) = probabilitas hipotesis H (*prior probability*)

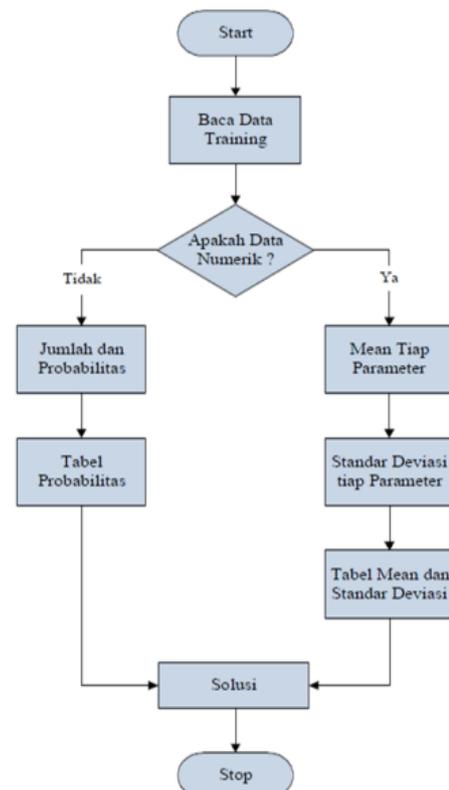
P(X|H) = probabilitas hipotesis X berdasar kondisi pada hipotesis H

P(X) = probabilitas dari X

Algoritma *Naïve Bayes* merupakan sebuah metode klasifikasi yang menerapkan teorema Bayes yang berdasarkan nilai probabilitas [14]. Jika terdapat dua buah kejadian A dan kejadian B dalam suatu kondisi maka nilai probabilitas kejadian B terhadap kejadian A dapat di rumuskan dengan persamaan berikut:

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

Alur dari metode *Naïve Bayes* sebagai berikut [14]:



Gambar 3. Alur dari metode *Naïve Bayes*

Dimana  $p(A|B)$  adalah conditional probability kejadian A yang mempengaruhi kejadian B dan  $p(B|A)$  adalah *conditional probability* kejadian B yang mempengaruhi kejadian A. Sedangkan  $p(A)$  dan  $p(B)$  adalah probabilitas kejadian A dan B. Jika A adalah kategori dan B adalah data berdasarkan algoritme Naïve Bayes dalam mengklasifikasikan sebuah data dapat ditulis :

$$p(\text{kategori} | \text{dokumen}) = \frac{p(\text{dokumen} | \text{kategori}) \times p(\text{kategori})}{p(\text{Dokumen})}$$

Kategori direpresentasikan sebagai  $c_j$ , dimana  $c_j$  merupakan teks yang akan diklasifikasikan dan data direpresentasikan sebagai  $d$ . Sehingga  $p(\text{kategori})$  yang merupakan probailitas dari kategori teks dapat ditulis  $p(c_j)$  dan  $p(\text{data})$  yang merupakan probabilitas data ditulis menjadi  $p(d)$ . Maka persamaan 11 akan menjadi seperti berikut:

$$p(c_j | d) = \frac{p(d | c_j) \times p(c_j)}{p(d)}$$

### 3. Random forest

*Random forest* adalah algoritma klasifikasi dan regresi yang menjadi bagian dari kelompok *ensemble learning* [13]. Metode *random forest* merupakan pengembangan dari *decision tree* dimana setiap *decision tree* telah dilakukan proses pelatihan dengan menggunakan sampel individu. *Random forest* yang dihasilkan memiliki banyak *tree* dan setiap *tree* ditanam dengan cara yang sama. Seiring dengan bertambahnya dataset, maka *tree* juga ikut berkembang [13].

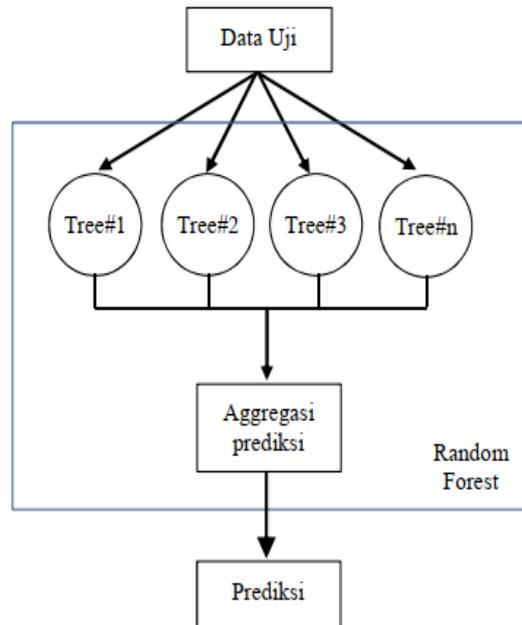
Dalam *random forest*, pemilihan atribut pada setiap kali sebuah *node* akan dipecah akan diambil secara acak. Setiap *tree* diberi sampel data pelatihan dengan menggunakan metode *bagging* dan tiap *tree* dibangun menggunakan metode yang sama untuk membangun CART (*classification and regression tree*).

*Random forest* yang dihasilkan memiliki banyak *tree* dan setiap *tree* akan tumbuh dengan cara yang sama. *Tree* dengan variabel  $x$  akan ditempatkan pada jarak yang jauh dengan *tree* dengan variabel  $y$ . Sejalan dengan bertambahnya dataset maka *tree* pun ikut berkembang. Penempatan *tree* yang saling berjauhan akan memudahkan dalam deteksi jenis *tree*. *Tree* yang berada di sekitar *tree x* maka *tree* tersebut merupakan perkembangan dari *tree x* sedangkan *tree* yang berada disekitar *tree y* maka *tree* tersebut merupakan perkembangan dari *tree y*. Pembangunan *tree* akan berhenti ketika data sudah homogen atau jika batas jumlah data minimum sudah terlewati.

*Random forest* yang dihasilkan memiliki banyak *tree* dan setiap *tree* akan tumbuh dengan cara yang sama. *Tree* dengan variabel  $x$  akan ditempatkan pada jarak yang jauh dengan *tree* dengan variabel  $y$ . Sejalan dengan bertambahnya dataset maka *tree* pun ikut berkembang. Penempatan *tree* yang saling berjauhan akan memudahkan dalam deteksi jenis *tree*. *Tree* yang berada di sekitar *tree x* maka *tree* tersebut merupakan perkembangan dari *tree x* sedangkan *tree* yang berada disekitar *tree y* maka *tree* tersebut merupakan perkembangan dari *tree y*. Pembangunan *tree* akan berhenti ketika data sudah

homogen atau jika batas jumlah data minimum sudah terlewati.

Proses prediksi *random forest* seperti ditunjukkan pada gambar dibawah ini :



Gambar 4. Proses prediksi *random forest*

### 4. Bagging Classifier

*Bagging* adalah singkatan dari *bootstrap aggregating*, menggunakan sub dataset (*bootstrap*) untuk menghasilkan set pelatihan  $L$  (*learning*),  $L$  melatih dasar belajar menggunakan prosedur pembelajaran yang tidak stabil, dan kemudian, selama pengujian, mengambil rata-rata [14].

*Bagging* baik digunakan untuk klasifikasi dan regresi. Dalam kasus regresi, untuk menjadi lebih kuat, seseorang dapat mengambil rata-rata ketika menggabungkan prediksi. *Bagging* adalah sebuah algoritma pembelajaran yang stabil pada perubahan kecil dalam training set menyebabkan perbedaan besar dalam peserta didik yang dihasilkan, yaitu algoritma belajar pada data yang memiliki varians tinggi (*noise*) [14].

## III. METODOLOGI PENELITIAN

Dalam penelitian ini penulis mengambil data dari sekumpulan 425 pesan spam SMS diekstraksi secara manual dari situs *Web Grumbletext*. Ini adalah forum Inggris di mana pengguna ponsel membuat klaim publik tentang pesan spam SMS, kebanyakan dari mereka tanpa melaporkan pesan spam yang diterima. Identifikasi teks pesan spam dalam klaim adalah tugas yang sangat sulit dan memakan waktu, dan melibatkan pemindaian ratusan halaman web dengan cermat.

Metode pengumpulan data pada penelitian ini dikumpulkan Subset dari 3.375 SMS pesan ham yang dipilih secara acak dari NUS SMS Corpus (NSC), yang merupakan kumpulan data sekitar 10.000 pesan yang sah dikumpulkan untuk penelitian di Departemen Ilmu Komputer di National University of Singapore. Pesan-pesan sebagian besar berasal dari Singapura dan sebagian besar dari siswa yang menghadiri Universitas. Pesan-pesan ini dikumpulkan dari sukarelawan yang dibuat sadar bahwa kontribusi mereka akan tersedia untuk umum.

Analisis dilakukan dengan menggunakan metode data mining, khususnya adalah text mining dengan menggunakan algoritma Naive Bayes Classifier, Support Vector Machine, Random Forrest, dan Bagging Classifier. Preprocessing yang digunakan untuk mewakili setiap dokumen sebagai vektor fitur, yaitu untuk memisahkan teks menjadi kata-kata individual adalah menggunakan Count Vectorization.

Pada penelitian ini kami menggunakan empat perbandingan algoritma diantaranya support vector machine, naive bayes classifier, random forest dan bagging classifier. Keempat algoritma ini akan dibandingkan dan akan digunakan algoritma terbaik untuk menghasilkan short message service dianggap benar ataupun di anggap tidak. Sehingga dapat di implementasikan pada ponsel masing - masing pengguna.

IV. HASIL DAN PEMBAHASAN

Pada penelitian ini semua proses pengolahan data SMS menggunakan pengkodean dengan menggunakan tool jupyter notebook.

Dari keempat hasil perhitungan dapat terlihat memiliki perbedaan disemua aspek. Setiap aspek tersebut nantinya akan dibandingkan dan dianalisis dengan menggunakan tipe data dari SMS yaitu Ham dan Spam yang mempunyai data sama. Jadi dengan data yang sama diharapkan mempunyai perbedaan atau kesamaan dan performance score nya.

Pada penelitian ini dengan mengambil dataset dari UCI Machine Learning Repository dan di test dengan model Naive Bayes Classifier, Support Vector Machine, Random Forrest, dan Bagging Classifier, sehingga pada penelitian ini mendapatkan hasil sebagai berikut ini :

Tabel 1. Performance Score hasil penelitian

Data Type of SMS		Model	Performance Score
Ham	Spam		
4824	747	Support Vektor Machine	86.5%
4824	747	Naive Bayes	96.4%
4824	747	Random Forrest	96.8%
4824	747	Bagging Classifier	97.4%

V. KESIMPULAN DAN SARAN

Adapun kesimpulan yang dapat diambil dari peneltian ini adalah sebagai berikut:

- 1) Dalam penelitian ini menghasilkan bahwa Bagging classifier algorithm mendapat fermormance score tertinggi dari algoritma yang lain.
- 2) Bagging classifier algorithm dapat dipergunakan sebagai sarana untuk memfiltrasi SMS yang masuk ke dalam inbox pengguna.
- 3) Bagging classifier algorithm dapat memberikan hasil filtrasi yang akurat untuk menyaring SMS yang masuk.

Adapaun saran untuk penelitian selanjutnya adalah:

- 1) Untuk penelitian selanjutnya diperlukan perbandingan dapat dikembangkan kembali dengan Algoritma dan data lain.
- 2) Perlunya penelitian lebih lanjut untuk mengetahui tingkat efektifitas dari masingmasing algoritma tersebut.
- 3) Data yang digunakan lebih banyak lagi.
- 4) Peneliti memberi saran untuk melakukan tahap preprocessing lanjutan yang dapat memperbaiki kata-kata yang tidak memiliki arti dalam bahasa Inggris, dan juga untuk menghapus kata-kata yang berisi nama merk ataupun produk barang tertentu. Untuk menghapus merk, penulis menyarankan untuk menambahkan khusus merk ataupun produk tertentu ke dalam list stopwords.

REFERENCES

- [1] Dea Delvia Arifin: Shaufiah: Moch. Arif Bijaksana (2016). Enchanging Spam Detection On Mobile Phone Short Message Service (SMS). Performance Using Fp-Growth and Naive Bayes Classifier. IEEE Asia Psific Cobference On Wireles And Mobile (APWIMOB)
- [2] Dewi, I. N., & Supriyanto, C. (2013). Klasifikasi Teks Pesan Spam Menggunakan Algoritma Naive Bayes. Semantik, 3(1).
- [3] Sethi, G., & Bhootna, V. (2014). SMS spam filtering application using Android. Int. J. Comput. Sci. Inf. Technol, 5(3), 4624–4626.
- [4] Asoke K. Talukder, Moblie Computing, 2005.
- [5] Dan, B., Untuk, C., Sms, K. dan Sari, R. (2017) “Komparasi Algoritma Support Vector Machine , Naive,” 2(2), hal. 7–13.
- [6] Informasi, J. T., Munitasri, I., Santosa, S., Supriyanto, C., Tenik, P., Universitas, I., Nuswantoro, D. dan Semarang, P. N. (2018) “ALGORITMA NAIVE BAYES DENGAN SELEKSI,” 14.
- [7] Feldman, R, Sanger, J, (2007), “The Text Mining Handbook : Advanced Approaches in Analyzing Unstructured Data”. New York: Cambridge University Press
- [8] Elly, S., Mira, K. S. & Alfian, A. G., 2015. Implementasi Metode Support Vector Machine untuk Melakukan Klasifikasi Kemacetan Lalu Lintas Pada Twitter. Makassar, S1 Universitas Telkom.
- [9] Perdana, A. dan Furqon, M. T. (2018) “Penerapan Algoritma Support Vector Machine ( SVM ) Pada

Pengklasifikasian Penyakit Kejiwaan Skizofrenia ( Studi Kasus : RSJ . Radjiman Wediodiningrat , Lawang ),” 2(9), hal. 3162–3167.

- [10] J. Chen, Z. Dai, J. Duan, H. Matzinger, and I. Popescu, “Naive Bayes with Correlation Factor for Text Classification Problem,” in 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA), 2019, pp. 1051–1056, doi: 10.1109/ICMLA.2019.00177.
- [11] Prasetio, R. T. (2015) “PENERAPAN TEKNIK BAGGING,” II(2), hal. 395–403.
- [12] Putra, D. S., Wibawa, A. D. dan Purnomo, M. H. (2016) “BERJALAN MENGGUNAKAN RANDOM FOREST,” 1(1), hal. 51–56.
- [13] Saleh, A. (2015) “Implementasi Metode Klasifikasi Naïve Bayes Dalam Memprediksi Besarnya Penggunaan Listrik Rumah Tangga,” 2(3), hal. 207–217.
- [14] Zuviyanto, E., Adji, T. B. dan Setiawan, N. A. (2018) “PERBANDINGAN ALGORITME-ALGORITME PEMBELAJARAN,” hal. 20–26.