

IMPLEMENTASI NAIVE BAYES DAN POS TAGGING MENGGUNAKAN METODE HIDDEN MARKOV MODEL VITERBI PADA ANALISA SENTIMEN TERHADAP AKUN TWITTER PRESIDEN JOKO WIDODO DI SAAT PANDEMI COVID - 19

Ahmadi Yuli Ananta¹, Annisa Puspa Kirana², Artha Ilma Imanidanantoyo³

^{1,2,3}Program Studi Teknik Informatika, Jurusan Teknologi Informasi, Politeknik Negeri Malang

¹ahmadi@polinema.ac.id, ²puspakirana@polinema.ac.id, ³arthailma12345@gmail.com

Abstrak— *Twitter* adalah tempat yang tepat untuk berbagi ide, bank gagasan, tempat untuk mengumpulkan informasi, untuk menginspirasi pikiran, atau untuk melihat apa yang teman lakukan. Banyak pengguna *Twitter* yang melakukan posting ekspresi dan pendapat mereka terhadap sebuah produk, layanan, isu politik atau hal-hal yang sedang viral. Pemerintahan saat ini yang dipimpin Presiden Joko Widodo menuai berbagai macam komentar, mulai dari pujian, kritik, saran, sindiran bahkan ujaran kebencian. Dengan banyaknya pengguna *Twitter* yang menyampaikan opini-opini tersebut dapat dimanfaatkan untuk mencari sebuah informasi. Analisa sentimen atau *opinion mining* merupakan proses memahami, mengekstrak dan mengolah data tekstual secara otomatis untuk mendapatkan informasi sentimen yang terkandung dalam suatu kalimat opini. Dalam penelitian ini dilakukan analisa sentimen masyarakat terhadap kinerja Presiden Joko Widodo yang diungkapkan melalui jejaring sosial *Twitter* pada saat pandemi *Covid-19* menggunakan algoritma *Naïve Bayes Classifier* dan algoritma *Opinion Detection* (penggabungan antara algoritma *Pos Tagging Hidden Markov Model Viterbi* dan *Rule Opinion*). Hasil dari penelitian ini didapatkan bahwa *Opinion Detection* menggunakan metode *Pos Tagging Hidden Markov Model Viterbi* dan *Rule Opinion* menghasilkan akurasi sebesar 69%, 70% dan 71%; sementara hasil dari klasifikasi menggunakan metode *Naïve Bayes Classifier* menghasilkan akurasi sebesar 80%, 81% dan 83%.

Kata kunci: *analisa sentimen, Pos Tagging, Hidden Markov Model Viterbi, Naïve Bayes Classifier, Rule Opinion.*

I. PENDAHULUAN

Di era digital ini, media sosial merupakan salah satu tempat atau media untuk berekspresi dan berpendapat tentang berbagai macam topik. Sosial media adalah satu set

baru komunikasi dan alat kolaborasi yang memungkinkan banyak jenis interaksi yang sebelumnya tidak tersedia untuk orang biasa. *Twitter* adalah tempat yang tepat untuk berbagi ide, bank gagasan, tempat untuk mengumpulkan informasi, untuk menginspirasi pikiran, atau untuk melihat apa yang teman lakukan [1]. Media sosial ini secara global memiliki 330 juta pengguna pada Januari-Maret 2018, naik 9 juta dibanding kuartal terakhir. Namun, turun sedikit dari tahun lalu. *Twitter* saat ini tak lagi menggunakan metode perhitungan tersebut. Sebagai gantinya, mereka menghitung pengguna harian mencapai 134 juta, naik dari tahun lalu sebanyak 120 juta [2]. Banyak pengguna *twitter* yang melakukan posting ekspresi dan pendapat mereka terhadap sebuah produk, layanan, isu politik atau hal – hal yang sedang viral. Pemerintahan yang sedang berjalan pada saat inipun tidak luput dari komentar publik dalam media sosial *twitter*. Pemerintahan saat ini yang dipimpin Presiden Joko Widodo menuai berbagai macam komentar, mulai dari pujian, kritik, saran, sindiran bahkan ujaran kebencian. Dengan banyaknya pengguna *Twitter* yang menyampaikan opini-opini tersebut dapat dimanfaatkan untuk mencari sebuah informasi. Namun dalam pemanfaatannya membutuhkan analisis yang tepat sehingga informasi yang dihasilkan dapat membantu banyak pihak untuk mendukung suatu keputusan atau pilihan. Salah satu metode yang dapat digunakan untuk menganalisis opini – opini *tweet* adalah analisa sentimen.

Analisa sentimen atau *opinion mining* merupakan proses memahami, mengekstrak dan mengolah data tekstual secara otomatis untuk mendapatkan informasi sentimen yang terkandung dalam suatu kalimat opini [3]. Dalam penelitian ini analisis sentimen dilakukan untuk melihat pendapat atau kecenderungan opini terhadap sebuah masalah atau objek mengandung opini negatif atau positif menggunakan

algoritma *Naïve Bayes Classifier (NBC)* dan algoritma *POS Tagging Hidden Markov Model Viterbi*.

Berdasarkan penjelasan diatas, akan dilakukan penelitian tentang analisa sentimen Twitter untuk mengklasifikasikan opini dari komentar masyarakat pada kinerja Presiden Joko Widodo. Data tersebut akan diolah menggunakan algoritma *Opinion Detection* (penggabungan antara algoritma *Hidden Markov Model Viterbi* dan *Rule Opinion*) untuk menghasilkan Tweet yang dikategorikan menjadi opini dan bukan opini. dan algoritma *Naive Bayes* untuk menghasilkan tweet yang telah diklasifikasikan menjadi positif, dan negatif. Pengujian akan dilakukan berdasarkan hasil klasifikasi. Pengujian bertujuan untuk mengetahui tingkat akurasi dari klasifikasi. Hasil klasifikasi akan memberikan kemudahan bagi pengguna untuk mengategorikan tweet menjadi opini dan bukan opini, opini positif dan opini negatif.

A. Algoritma *Naïve Bayes*

Algoritma *Naive Bayes* merupakan sebuah metode klasifikasi menggunakan metode probabilitas dan statistik yg dikemukakan oleh ilmuwan Inggris Thomas Bayes. Algoritma *Naive Bayes* memprediksi peluang di masa depan berdasarkan pengalaman di masa sebelumnya sehingga dikenal sebagai *Teorema Bayes* [11]. Berikut adalah rumus algoritma *Naive Bayes* :

Persamaan (1) merupakan rumus untuk perhitungan *Naive Bayes*.

$$V_{MAP} = \operatorname{argmax} P(C_j) \prod P(W_i|C_j) \quad (1)$$

Persamaan (2) merupakan rumus untuk perhitungan probabilitas dokumen.

$$P(C_j) = \frac{|category_j|}{|corpus|} \quad (2)$$

Persamaan (3) merupakan rumus untuk perhitungan probabilitas kemunculan kata pada suatu dokumen.

$$P(W_i|C_j) = \frac{1+n_i}{n+|kosakata|} \quad (3)$$

Keterangan:

V_{MAP}	= Label kategori yang paling tinggi probabilitasnya
$P(C_j)$	= Peluang kemunculan dokumen pada kategori j
$category_j$	= Jumlah kata dalam dokumen pada kategori j
$corpus$	= Jumlah dokumen
$P(W_i)$	= Peluang kemunculan sebuah kata
$P(W_i C_j)$	= Peluang sebuah kata i masuk ke dalam kategori j
n_i	= Jumlah munculnya kata i pada tiap dokumen
n	= Jumlah total kata dari dokumen
$kosakata$	= Jumlah total kata dalam dokumen latihan

B. Pembobotan *Laplace Correction*

Dalam proses prediksi untuk menghindari probabilitas 0 (nol) yang dapat menyebabkan *Naive Bayes Classifier*

tidak dapat mengklasifikasi sebuah data inputan dengan baik maka digunakan teknik *Laplace Correction*. Yaitu sebuah teknik yang menambahkan nilai 1 pada setiap kombinasi atribut. Untuk jumlah data yang banyak (hingga ribuan) teknik ini akan sangat akurat karena tidak akan membuat perbedaan yang berarti pada estimasi probabilitas. Dengan persamaan sebagai berikut :

$$P(X_k|C) = \frac{P(X_k|C)+1}{P(C)+|V|} \quad (4)$$

Keterangan :

$P(X_k|C)$ = Probabilitas tiap atribut dari X_k
 $P(C)$ = Total probabilitas dalam X_k

$|V|$ = Jumlah kemungkinan nilai dari X_k .

C. Algoritma *Viterbi*

Viterbi merupakan proses yang digunakan untuk mendapatkan kata berlabel terprediksi, dimana harus dimasukkan kata, dan model HMM yang terdiri dari inialisasi, transisi, dan emisi. Hasil dari Viterbi berupa kata berlabel terprediksi, digunakan untuk menghitung akurasi, yang didapat dari perbandingan antara label dari data set, dengan label hasil Viterbi.

Model HMM ini terdiri dari tiga proses, yaitu inialisasi, transisi, dan emisi. Inialisasi merupakan proses mendapatkan jumlah label (tags) dari masing-masing kata yang terdapat pada data training.

Dalam melakukan inialisasi, perlu dimasukkan kata berlabel dan label kata. Kata berlabel merupakan kumpulan kalimat yang sudah diberi label. Dan label kata adalah jenis-jenis label yang terdapat dalam data set. Dalam melakukan model ini, terdapat 23 label kata dalam data set.

Transisi adalah suatu proses mencari label (tags) kata setelah label kata tersebut. Transisi didapatkan dengan memasukkan kata berlabel dan label kata pada sistem.

Emisi merupakan proses mencari jumlah kata dari masing – masing label (tags) yang terdapat pada data training. Dalam mendapatkan emisi, perlu dimasukkan kata berlabel, label kata, dan kamus kata yang berasal dari database. Kamus kata merupakan kumpulan kata yang terdapat pada data set. dalam melakukan model ini, terdapat 259.615 kata pada data set.

Model yang didapatkan digunakan sebagai proses untuk melakukan pengujian. Database yang digunakan saat pengujian adalah kata, kata berlabel, dan model HMM [5].

II. METODE PENELITIAN

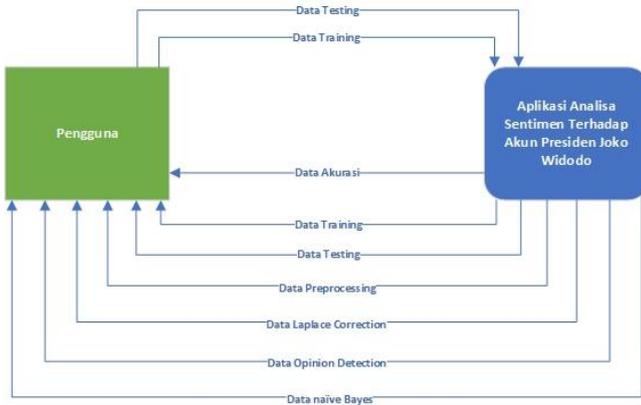
A. Analisis Sistem

Sistem yang akan dibangun diharapkan dapat mengklasifikasikan *tweet* yang telah didapatkan dari proses *Scraping* menggunakan *scrapy python* ke dalam label positif dan negatif pada algoritma *naive bayes*, label opini dan buka opini pada algoritma *opinion detection* (penggabungan algoritma *HMM Viterbi* dan *rule opinion*). Sistem ini akan memiliki beberapa sub proses yang akan dilakukan untuk menghasilkan satu proses yang utuh.

B. Data Flow Diagram (DFD)

- Diagram Konteks
Sistem Aplikasi Analisa Sentimen Terhadap Akun Presiden Joko Widodo memiliki 1 entitas yaitu entitas

pengguna. Entitas pengguna dapat menginputkan data *Testing* dan data *Training* yang didapat dari hasil *scraping*. Entitas pengguna bisa melihat dan memproses data *training*, data *testing*, data *preprocessing*, data *laplace correction*, data *opinion detection* dan data *naïve bayes*.

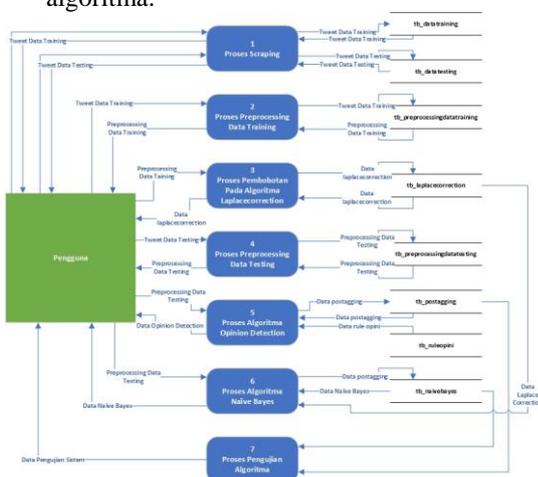


Gambar 1. Diagram Konteks

- *Data Flow Diagram (DFD) Level 1.*

DFD Level 1 merupakan penguraian sistem dari Diagram Konteks. Pada gambar 4.1 menjelaskan tentang proses informasi yang terdapat pada sistem, sistem memiliki tujuh proses yaitu:

1. Pengguna dapat mendapatkan dataset dari proses *scraping*.
2. Pengguna dapat melihat dan memproses *preprocessing* data *training*.
3. Pengguna dapat melihat dan memproses pembobotan pada algoritma *laplace correction*.
4. Pengguna dapat melihat dan memproses *preprocessing* data *testing*.
5. Pengguna dapat melihat dan memproses algoritma *opinion detection*.
6. Pengguna dapat melihat dan memproses algoritma *naïve bayes*.
7. Pengguna dapat melihat dan memproses pengujian algoritma.



Gambar 2. Data Flow Diagram (DFD) Level 1

C. Data

Data testing yang digunakan dalam penelitian ini adalah tweet pada sosial media Twitter yakni sentimen terhadap kinerja presiden joko widodo disaat pandemi *corona*

berlangsung. Data tweet diambil pada saat virus corona mulai masuk indonesia pada tanggal 2 maret sampai 14 juni 2020, jumlah data testing algoritma *naïve bayes* yang digunakan penulis yaitu 90 data tweet dan jumlah data testing *Opinion Detection* yaitu 120.

Data training yang digunakan penulis di bagi menjadi 2 yaitu data training untuk algoritma *opinion detection* dan data training untuk algoritma *naïve bayes*. Data training untuk *opinion detection* yang digunakan penulis di bagi menjadi dua yaitu dataset *POS tagging Viterbi* sebanyak 259.368 kata dan dataset *rule opinion* sebanyak 15 rule. Data training untuk algoritma *naïve bayes* yang digunakan penulis yaitu 300 data tweet.

D. Pengumpulan Data

Dalam penelitian ini data didapatkan dari media sosial *Twitter* dengan metode pengambilan data yaitu metode *scraping*. Pengambilan data dilakukan secara berkala dengan target 8 kata kunci, yaitu bantuan *corona jokowi*, kartu *prakerja jokowi*, psbb *jokowi*, BLT *jokowi*, cicilan *jokowi*, Bansos *jokowi*, *jokowi asimilasi narapidana*, *jokowi new normal*. Pengambilan data secara periodik ini dilakukan dengan metode *scraping* menggunakan *library scrapy* pada bahasa pemrograman *python*, setelah data berhasil di *scraping*, data tersebut akan disimpan di *database*.

E. Pengolahan Data

Tahap analisa data dilakukan dengan melalui beberapa proses diantaranya proses *preprocessing*, pembobotan menggunakan algoritma *Laplace Correction*, *POS Tagging* dan klasifikasi sentimen menggunakan algoritma *Naïve Bayes*. *Preprocessing* adalah pengolahan data yang bertujuan untuk meminimalisir data / kata yang tidak memiliki nilai dalam object penelitian dan data yang tidak konsisten. Tahap *preprocessing* dibagi menjadi beberapa proses yaitu *cleaning*, *casefolding*, *tokenizing*, *normalisasi*, *filtering* dan *stemming*.

- *Case Folding*

Case folding adalah mengubah semua huruf besar atau kapital dalam *Twitter* menjadi huruf kecil.

- *Cleaning*

Data *cleaning* merupakan proses pembersihan kata dengan menghilangkan delimiter tanda baca, *hashtag*, *mention*, angka, *url*. Pembersihan kata bertujuan untuk mengurangi *noise*.

- *Tokenizing*

Tahap *Tokenizing* adalah tahap pemotongan tiap kata dalam kalimat atau parsing dengan menggunakan spasi sebagai *delimiter* yang akan menghasilkan *token* berupa kata. Tokenisasi secara garis besar memecah sekumpulan karakter dalam suatu teks ke dalam satuan kata, bagaimana membedakan karakter-karakter tertentu yang dapat diperlakukan sebagai pemisah kata atau bukan [8]. Pada *tokenizing* terdapat beberapa proses yang harus dilakukan yaitu merubah semua huruf besar menjadi kecil (*text to lowercase*). Proses selanjutnya adalah penguraian, proses penguraian yang dimaksud adalah membagi text menjadi kumpulan kata tanpa memperhatikan keterhubungan antara

kata satu dengan kata lain serta peran dan posisinya pada kalimat [4].

- *Normalisasi*

Normalisasi bertujuan untuk menormalkan kata “gaul” menjadi kata baku [10]. Tahapan ini bertujuan untuk mengembalikan bentuk penulisan dari masing-masing kata yang sesuai dengan Kamus Besar Bahasa Indonesia (KBBI).

- *Filtering*

Tahap *Filtering* adalah tahap penyaringan kata yang didapat dari *Tokenizing* yang dianggap tidak penting atau tidak memiliki makna dalam proses *Text mining* yang disebut *stopword*. *Stopword* berisi katakata umum yang sering muncul dalam sebuah dokumen dalam jumlah banyak namun tidak memiliki kaitan dengan tema tertentu. Kata yang biasanya dihilangkan adalah kata preposisi, adjektiva, konjungsi dan kata – kata lainnya yang sering muncul yang jika dihilangkan tidak akan mengubah makna dari data tersebut [9]. Contoh stopwords adalah “yang”, “di”, “dan”, dll.

- *Stemming*

Tahap stemming adalah tahap menghilangkan semua imbuhan, dan menghasilkan kata dasar yang sesuai dengan struktur morfologi bahasa Indonesia [6]. Penelitian ini menggunakan stemmer Sastrawi. Sastrawi adalah *library php* sederhana yang menyediakan *stemming* kata bahasa Indonesia. Proses *stemming* oleh Sastrawi sangat bergantung pada kamus kata dasar, kamus kata dasar yang digunakan Sastrawi berasal dari *kataglo.com* dengan sedikit perubahan dan masing-masing mempunyai lisensi Sastrawi dan lisensi *kataglo*. Sastrawi dapat diunduh secara gratis di alamat <http://Sastrawi.github.io/> [12].

Hasil dari *preprocessing* kemudian dilakukan proses penilaian *propabilitas* dengan metode *laplace correction*. *Laplace correction* adalah suatu cara untuk menangani nilai *probabilitas* 0 (nol). Dari sekian banyak data di training set, pada setiap perhitungan datanya ditambah 1 dan tidak akan membuat perbedaan yang berarti pada estimasi *probabilitas* sehingga bisa menghindari kasus nilai *probabilitas* 0 (nol). Setelah proses pembobotan atau proses *laplace correction* berhasil, maka akan di lanjutkan ke proses *opinion detection* (penggabungan algoritma *postagging* dan *rule opini*) untuk memisahkan antara sentiment opini dan sentimen bukan opini. Setelah proses *laplace correction* berhasil, selanjutnya adalah proses *naïve bayes* dimana metode *naïve bayes* berguna untuk memisahkan anantara opini positif dan opini negative. Setelah proses *naïve bayes* berhasil, tahap selanjutnya adalah hitung akurasi, Data yang diuji adalah data yang telah diketahui sentimennya, sehingga setelah dilakukan klasifikasi sentimen maka akan dilakukan perhitungan akurasi atau ketepatan sistem dalam melakukan klasifikasi [7].

III. PENGUJIAN DAN ANALISA

Pada pengujian akurasi sistem dilakukan pengujian terhadap hasil klasifikasi pada proses testing. Untuk mengetahui keakuratan algoritma dalam proses klasifikasi maka akan dibandingkan hasil klasifikasi dengan hasil klasifikasi pelabelan manual oleh user. Perbandingan

tersebut akan dihitung tingkat keakurasiannya menggunakan *accuracy*, *precision* dan *recall*. Semakin tinggi nilai *accuracy*, *precision* dan *recall* pada sebuah algoritma yang digunakan maka menunjukkan bahwa algoritma tersebut berjalan dengan baik dan cocok untuk proses klasifikasi.

A. Pengujian Akurasi Algoritma *Opinion Detection* (*Viterbi* dan *Rule Opinion*)

1. *Pengujian Pertama Algoritma Opinion Detection.*
Pengujian Pertama Algoritma *Opinion Detection* merupakan pengujian algoritma *opinion detection* pada data testing 120 *tweet*, data training *viterbi* 259.615 kata dan data *rule opinion* 10 *rule* yang menghasilkan *accuracy* 69%, *precision* opini 86%, *precision* bukan opini 42%, *recall* opini 70% dan *recall* bukan opini 66%.

Tabel 1 merupakan pengujian data testing 120 *Tweet*, data training *viterbi* 259.615 kata dan data *rule opinion* 10 *rule*.

Tabel 1. Pengujian Pertama Algoritma *Opinion Detection*.

Data Testing 120 <i>Tweet</i> , data training <i>viterbi</i> 259.615 kata dan data <i>rule opinion</i> 10 <i>rule</i>				
<i>Accuracy</i>	<i>Precision Opini</i>	<i>Precision Bukan opini</i>	<i>Recall Opini</i>	<i>Recall Bukan Opin</i>
69%	86%	42%	70%	66%

2. *Pengujian Kedua Algoritma Opinion Detection.*
Pengujian Kedua Algoritma *Opinion Detection* merupakan pengujian algoritma *opinion detection* pada data testing 120 *tweet*, data training *viterbi* 259.615 kata dan data *rule opinion* 12 *rule* yang menghasilkan *accuracy* 70%, *precision* opini 86%, *precision* bukan opini 44%, *recall* opini 72% dan *recall* bukan opini 66%.

Tabel 2 merupakan pengujian data testing 120 *Tweet*, data training *viterbi* 259.615 kata dan data *rule opinion* 12 *rule*.

Tabel 2. Pengujian Kedua Algoritma *Opinion Detection*.

Data Testing 120 <i>Tweet</i> , data training <i>viterbi</i> 259.615 kata dan data <i>rule opinion</i> 12 <i>rule</i>				
<i>Accuracy</i>	<i>Precision Opini</i>	<i>Precision Bukan opini</i>	<i>Recall Opini</i>	<i>Recall Bukan Opin</i>
70%	86%	44%	72%	66%

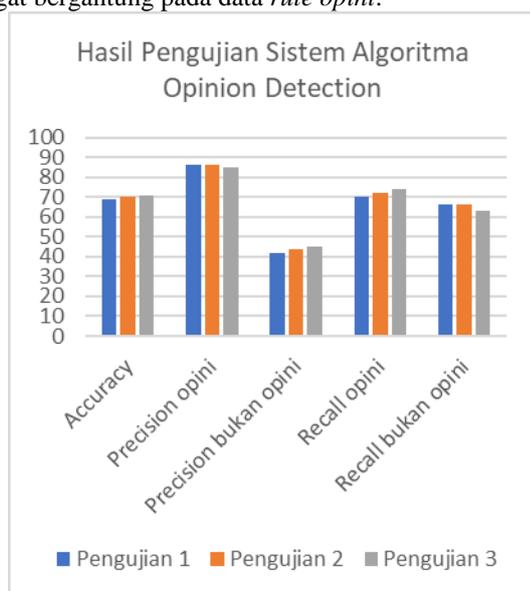
3. *Pengujian Ketiga Algoritma Opinion Detection.*
Pengujian Ketiga Algoritma *Opinion Detection* merupakan pengujian algoritma *opinion detection* pada data testing 120 *tweet*, data training *viterbi* 259615 kata dan data *rule opinion* 15 *rule* yang menghasilkan *accuracy* 71%, *precision* opini 85%, *precision* bukan opini 45%, *recall* opini 74% dan *recall* bukan opini 63%.

Tabel 3 merupakan pengujian data testing 120 *Tweet*, data training *viterbi* 259.615 kata dan data *rule opinion* 14 *rule*.

Tabel 3. Pengujian Ketiga Algoritma *Opinion Detection*.

Data Testing 120 Tweet, data training <i>viterbi</i> 259.615 kata dan data <i>rule opinion</i> 14 rule				
Accuracy	Precision Opini	Precision Bukan opini	Recall Opini	Recall Bukan Opin
71%	85%	45%	74%	63%

Dari hasil pengujian diatas nilai algoritma *Opinion Detection* (Penggabungan Algoritma *POS Tagging Viterbi* dan *Rule Opinion*) berturut-turut didapatkan sebesar 69%, 70% dan 71%. Nilai akurasi terendah adalah 69% dan tertinggi adalah 71%. Setiap terjadi penambahan komposisi data *rule opini*, maka nilai dari *accuracy*, *precision* dan *recall* juga mengalami peningkatan. Hal ini dikarenakan algoritma *Opinion Detection* merupakan algoritma yang sangat bergantung pada data *rule opini*.



Gambar 3 Hasil Algoritma *Opinion Detection*

B. Pengujian Akurasi Algoritma *Naïve Bayes*

1. Pengujian Pertama Algoritma *Naïve Bayes*.

Pengujian pertama algoritma *Naïve Bayes* merupakan pengujian algoritma *naïve bayes* pada 90 *tweet* data testing dan 120 *tweet* data training *naïve bayes* yang menghasilkan *accuracy* 80%, *precision* + 78%, *precision* - 82%, *recall* + 84% dan *recall* - 75%.

Tabel 4 merupakan pengujian data training 120 dan data testing 90.

Tabel 4. Pengujian Pertama Algoritma *Naïve Bayes*.

Data Training 120 dan Data Testing 90				
Accuracy	Precision +	Precision -	Recall +	Recall -
80%	78%	82%	84%	75%

2. Pengujian Kedua Algoritma *Naïve Bayes*.

Pengujian kedua algoritma *Naïve Bayes* merupakan pengujian algoritma *naïve bayes* pada 90 *tweet* data testing dan 200 *tweet* data training *naïve bayes*

yang menghasilkan *accuracy* 81%, *precision* + 78%, *precision* - 84%, *recall* + 86% dan *recall* - 75%.

Tabel 5 merupakan pengujian data training 200 dan data testing 90.

Tabel 5. Pengujian Pertama Algoritma *Naïve Bayes*.

Data Training 200 dan Data Testing 90				
Accuracy	Precision +	Precision -	Recall +	Recall -
81%	78%	84%	86%	75%

3. Pengujian Ketiga Algoritma *Naïve Bayes*.

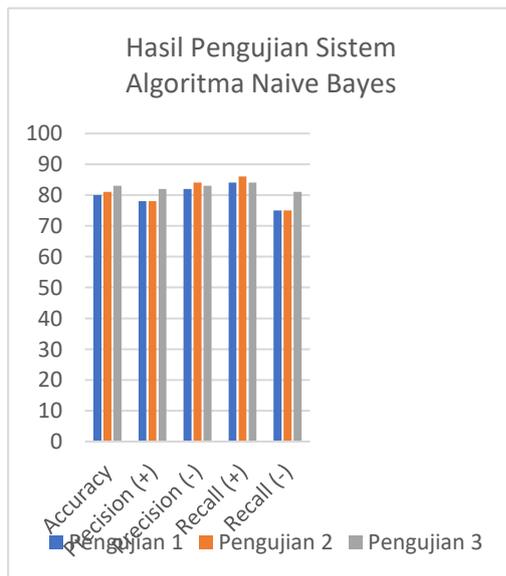
Pengujian Ketiga Algoritma *Naïve Bayes* merupakan pengujian algoritma *naïve bayes* pada 90 *tweet* data testing dan 300 *tweet* data training *naïve bayes* yang menghasilkan *accuracy* 83%, *precision* + 82%, *precision* - 83%, *recall* + 84% dan *recall* - 81%.

Tabel 6 merupakan pengujian data training 300 dan data testing 90.

Tabel 6. Pengujian Pertama Algoritma *Naïve Bayes*.

Data Training 300 dan Data Testing 90				
Accuracy	Precision +	Precision -	Recall +	Recall -
83%	82%	83%	84%	81%

Dari hasil pengujian diatas nilai algoritma *Naïve Bayes* berturut-turut didapatkan sebesar 80%, 81% dan 83%. Nilai akurasi terendah adalah 80% dan tertinggi adalah 83%. Setiap terjadi penambahan komposisi data *training*, maka nilai dari *accuracy*, *precision* dan *recall* juga mengalami peningkatan. Hal ini dikarenakan algoritma *Naïve Bayes* merupakan algoritma yang sangat bergantung pada data training, kemungkinan *accuracy* dapat ditingkatkan lagi dengan menambahkan data training yang lebih banyak lagi. Seperti hasil dari penelitian ini pada gambar setiap terjadi penambahan data training maka akurasi cenderung mengalami peningkatan.



Gambar 4 Hasil Algoritma Naive Bayes

IV. KESIMPULAN

Berdasarkan hasil penelitian dan pengujian yang telah dilakukan dapat ditarik kesimpulan sebagai berikut:

1. Algoritma *Opinion Detection* (Penggabungan Algoritma *POS Tagging Viterbi* dan *Rule Opinion*) dan Algoritma *Naive Bayes* berhasil diterapkan di aplikasi penulis dengan skenario algoritma *Opinion Detection* berfungsi sebagai pemisah antara opini dan bukan opini, Algoritma *Naive Bayes* berfungsi sebagai pemisah antara opini positif dan opini negatif.
2. Pengujian algoritma *Opinion Detection* (Penggabungan Algoritma *POS Tagging Viterbi* dan *Rule Opinion*) berturut-turut didapatkan sebesar 69%, 70% dan 71%. Nilai akurasi terendah adalah 69% dan tertinggi adalah 71%. Setiap terjadi penambahan komposisi data *rule opinion*, maka nilai dari *accuracy*, *precision* dan *recall* juga mengalami peningkatan. Hal ini dikarenakan algoritma *Opinion Detection* merupakan algoritma yang sangat bergantung pada data *rule opinion*.
3. Pengujian algoritma *Naive Bayes* berturut-turut didapatkan sebesar 80%, 81% dan 83%. Nilai akurasi terendah adalah 80% dan tertinggi adalah 83%. Setiap terjadi penambahan komposisi data training, maka nilai dari *accuracy*, *precision* dan *recall* juga mengalami peningkatan. Hal ini dikarenakan algoritma *Naive Bayes* merupakan algoritma yang sangat bergantung pada data training, kemungkinan akurasi dapat ditingkatkan lagi dengan menambahkan data training yang lebih banyak lagi.

V. SARAN

Saran yang dapat diberikan dari hasil penelitian untuk pengembangan sistem ini kedepan sebagai berikut:

1. Sistem dapat dikembangkan dengan menambahkan *preprocessing* yang lebih banyak sehingga mendapatkan *noise* yang sedikit.
2. Sistem dapat dikembangkan dengan crawling data menggunakan *Twitter API* secara *realtime*.
Sistem dapat dikembangkan dengan menambahkan data training yang lebih banyak sehingga mendapatkan akurasi yang lebih tinggi.

DAFTAR PUSTAKA

- [1] Brogan, "Social Media 101 Tactic and Tips to Develop Your Business Online", Wiley, Manhattan, 2010.
- [2] Dado, "Pengguna Capai 330 Juta, Keuntungan Twitter Naik 3 Kali Lipat", Website: <https://www.cnnindonesia.com/teknologi/20190424001600-185-389042/pengguna-capai-330-juta-keuntungan-twitter-naik-3-kali-lipat>, 24 April 2019.
- [3] Imam Fahrur Rozi, Sholeh Hadi Pramono dan Erfan Achmad Dahlan, "Implementasi Opinion Mining (Analisis Sentimen) untuk Ekstraksi Data Opini Publik pada Perguruan Tinggi", Malang: Politeknik Negeri Malang, 2012.
- [4] Faisal Rahutomo, Annisa Taufika Firdausi, Nur Rochman, "Pengembangan Sistem Analisa Keberpikahan Media Online Berdasarkan Trend Waktu menggunakan Naive Bayes Classifier", Malang: Politeknik Negeri Malang, 2019.
- [5] Nitin Sabloak, Beбето Agung Hardono, Derry Alamsyah, "Part-of-Speech (POS) Tagging Bahasa Indonesia Menggunakan Algoritma Viterbi", STMIK GI MDP Palembang, 2019.
- [6] B. Nazief and M. Adriani, "Stemming Indonesian: A confix-stripping approach", *ACM Transactions on Asian Language Information Processing*, 2007.
- [7] Yonathan Sari Mahardhika, Eri Zuliarso, "Anlisis Sentimen Terhadap Pemerintahan Joko Widodo Pada Media Sosial Twitter Menggunakan Algoritma Naive Bayes Classifier", Universitas Stikubank, 2018.
- [8] INFORMATOLOGI, "Text Preprocessing", Website: <https://informatikalogi.com/text-preprocessing/>, diupload tahun 2016.
- [9] Indriani Mentaruk, "Analisis Sentimen Twitter Transportasi Online Berbasis Ontologi (Studi Kasus : Go-Jek)", Universitas Telkom, 2019.
- [10] G.A Buntoro, N. Y, "Analysis Sentimen Calon Gubernur Jawa Timur 2018 di Twitter", *Scientico : Computer Science and Informatics journal*, vol. 1, no. 2, hlm. 37–45, 2018.
- [11] Fajar Ratnawati, "Implementasi Algoritma Naive Bayes Terhadap Analisis Sentimen Opini Film Pada Twitter", *Jurnal Onovtek POLBENG*, 2018.
- [12] Nurirwan Saputra, Teguh Bharata Adji, Adhistya Erna

Permanasari, "Analisis Sentimen Data Presiden Jokowi dengan Preprocessing Normalisasi dan Stemming Menggunakan Metode Naïve Bayes dan SVM", Jurnal Dinamika Informatika, 2015.