JASIE "Jurnal Aplikasi Sistem Informasi Dan Elektronika"
Vol.2, No. 2, Desember 2020, Hal: 52 - 65
p-ISSN: 2714-612X, e-ISSN: Dalam Proses

KINERJA ALGORITMA *MULTINOMIAL NAÏVE BAYES (MNB), MULTIVARIATE BERNOULLI* DAN *ROCCHIO ALGORITHM* DALAM KLASIFIKASI KONTEN BERITA *HOAX* BERBAHASA INDONESIA DENGAN *JUPYTER NOTEBOOK*

Hamdhan Ashari[1], Deni Arifianto[2], Habibatul Azizah Al Faruq[3]
Program Studi Teknik Informatika, Fakultas Teknik, Universitas Muhammadiyah Jember
Email: hamdhanashari12@gmail.com1, deniarifianto@unmuhjember.ac.id[2],
habibatulazizah@unmuhjember.ac.id[3]

ABSTRAK

Berita *hoax* adalah informasi palsu atau bohong yang disebarkan untuk orang banyak namun diterima sebagai berita yang benar. Penyebaran informasi di era modern saat ini sangat cepat. Salah satu media penyebarannya adalah media sosial. Pada penelitian ini dilakukan klasifikasi dokumen terhadap konten berita *hoax* berbahasa Indonesia pada situs resmi *turnbackhoax.id*. Metode klasifikasi pada penelitian ini adalah membandingkan antara algoritma *Multinomial Naive Bayes (MNB), Multivariate Bernoulli* dan *Rocchio.* Hasil yang didapatkan dari penelitian ini adalah pada algoritma *Multinomial Naive Bayes (MNB)* didapatkan hasil akurasi sebesar 74%, presisi 83,33% dan *recall* 60%. Pada algoritma *Multivariate Bernoulli* mendapatkan hasil akurasi sebesar 70%, presisi 62,50% dan *recall* 100%. Pada algoritma *Rocchio* mendapatkan hasil akurasi sebesar 76%, presisi 88,24% dan *recall* 60%.

Kata Kunci : Klasifikasi dokumen, *Hoax, Multinomial Naive Bayes (MNB), Multivariate Bernoulli, Rocchio.*

ABSTRACT

Hoax news is false or untrue information that is spread to many people but is accepted as true news. The spread of information in today's modern era is very fast. One of the distribution media is social media. In this study, document classification of hoax news content in Indonesian language was carried out on the official website turnbackhoax.id. The classification method in this study is to compare the Multinomial Naive Bayes (MNB), Multivariate Bernoulli and Rocchio algorithms. The results obtained from this study are the Multinomial Naive Bayes (MNB) algorithm, the results obtained are 74% accuracy, 83.33% precision and 60% recall. In the Multivariate Bernoulli algorithm, the results obtained are 70% accuracy, 62.50% precision and 100% recall. The Rocchio algorithm gets 76% accuracy, 88.24% precision and 60% recall.

Keyword: *Document classification, Hoax, Multinomial Naive Bayes (MNB), Multivariate Bernoulli, Rocchio.*

1. PENDAHULUAN

Informasi merupakan sekumpulan data yang sudah diproses dan dikelola sehingga dapat dipahami dan bermanfaat bagi penerimanya. Informasi juga merupakan sarana komunikasi antara sesama manusia. Informasi adalah hasil pengolahan data yang memberikan arti dan manfaat tertentu bagi orang yang menerimanya (Susanto, 2017). Pada jaman dahulu penyebaran informasi sangat dibatasi dengan jarak dan waktu karena teknologi pada saat itu masih terbatas. Di era teknologi modern saat ini kendala jarak dan waktu dalam penyebaran informasi sudah teratasi. Apalagi pada era teknologi modern saat ini informasi dapat disebarkan dengan mudah melalui berbagai macam media. Salah satunya yang paling popular adalah media sosial. Media sosial sendiri merupakan wadah atau sarana yang dapat menampung aspirasi dari setiap manusia di berbagai

belahan dunia. Media sosial juga merupakan media komunikasi yang juga berperan sebagai media penyebaran informasi. Informasi yang dikirimkan melalui media sosial akan sangat cepat diterima dan dikonsumsi oleh setiap orang.

Berbagi informasi bagi setiap orang merupakan hal yang diperlukan, namun tidak semua informasi yang disebarkan melalui media benar adanya. Telah banyak terjadi kasus penyebaran berita yang tidak benar atau biasa disebut dengan berita *hoax*. Berita *hoax* sendiri merupakan informasi berbahaya yang menyesatkan persepsi manusia dengan menyebarkan informasi yang salah namun dianggap sebagai kebenaran (Rasywir & Purwarianti, 2015). Penyebaran berita *hoax* bisa disebabkan oleh individu atau kelompok organisasi yang memiliki tujuan tertentu yang kemudian berita tersebut disebarkan kepada masyarakat luas. Penyebaran berita *hoax* memiliki dampak buruk bagi masyarakat. Telah dilansir oleh situs KEMENKOMINFO (Kementrian Komunikasi & Informatika) pada 13 Desember 2017 yang menyebutkan bahwa ada 800.000 situs di Indonesia yang telah terindikasi sebagai penyebar informasi palsu. Disebutkan juga internet telah salah dimanfaatkan oleh oknum tertentu untuk keuntungan pribadi dan kelompoknya dengan cara menyebarkan konten-konten negatif yang menimbulkan keresahan di masyarakat.

Begitu maraknya penyebaran berita *hoax* yang telah terjadi pada media pemberitaan terutama media sosial, maka banyak ide gagasan untuk melakukan pencegahan dalam penyebaran berita *hoax*. Telah banyak muncul saran dan tips untuk menghindari berita-berita *hoax* yang banyak tersebar di dalam media sosial. Adapun sebagian dari media sosial yang memberikan layanan tambahan untuk melaporkan konten yang mengandung unsur berita yang tidak benar atau *hoax*. Telah dikembangkan juga beberapa sistem penangkal yang memprediksi konten berita *hoax* dengan berbagai macam metode yang digunakan dalam pengimplementasiannya dengan hasil akurasi yang bermacam-macam. Untuk mengetahui informasi yang tersebar masuk ke dalam berita *hoax* atau benar akan dilakukan proses klasifikasi. Dari hasil klasifikasi akan didapatkan nilai persentase akurasi. Pada penelitian Rahman, Wiranto & Doewes (2017) yang berjudul "*Online News Classification Using Multinomial Naïve Bayes*" membandingkan algoritma *Multinomial Naïve Bayes* dengan seleksi fitur TF-IDF dan *Document Frequency Thresholding* dalam kasus *text mining* klasifikasi dan didapatkan hasil akurasi terbaik yaitu algoritma *Multinomial Naïve Bayes* menggunakan seleksi fitur TF-IDF dengan akurasi sebesar 86,62% sedangkan pada seleksi fitur *Document Frequency Thresholding* mendapatkan akurasi sebesar 85,98%. Sedangkan pada penelitian Wisaksono & Mujiyatna (2017) yang berjudul "Klasifikasi Berita Berkategori Olahraga dengan Algoritma *Multivariate Bernoulli Naïve Bayes* dan *Multinomial Naïve Bayes*" membandingkan antara algoritma *Multinomial Naïve Bayes (MNB)* dan *Multivariate Bernoulli* dalam pengklasifikasian berita olahraga dan didapatkan hasil akurasi yaitu algoritma *Multivariate Bernoulli* dengan akurasi sebesar 97,1% sedangkan *Multinomial Naïve Bayes (MNB)* mendapatkan akurasi sebesar 94%. Dan pada penelitian Pantouw (2017) yang berjudul "Perbandingan Klasifikasi *Rocchio* dan *Multinomial Naïve Bayes* pada Analisis Sentimen Data *Twitter* Bahasa Indonesia" membandingkan antara algoritma *Rocchio* dan *Multinomial Naïve Bayes (MNB)* dalam pengklasifikasian sentiment data *twitter* dan didapatkan hasil akurasi terbaik yaitu algoritma *Rocchio* dengan akurasi sebesar 96,283% sedangkan *Multinomial Naïve Bayes (MNB)* mendapatkan akurasi sebesar 85,399%. Dari beberapa sumber penelitian yang didapat, maka dari itu akan dibandingkan antara

akurasi terbaik dari referensi diatas sehingga didapatkan ketiga algoritma yang akan dibandingkan yaitu algoritma yaitu *Multinomial Naïve Bayess (MNB), Multivariate Bernoulli* dan *Rocchio Algorithm* untuk menemukan akurasi yang terbaik di antara metode tersebut pada kasus konten berita *hoax* berbahasa Indonesia pada media sosial.

2. TINJAUAN PUSTAKA

A. Media Sosial

Media sosial adalah sarana bagi masyarakat untuk saling berkomunikasi dan saling berbagi atau menyebarkan informasi dengan orang lain dalam dunia maya. Media sosial adalah media yang digunakan oleh individu agar menjadi sosial atau menjadi sosial dengan cara berbagi isi, berita, foto dan lain-lain dengan orang lain (Taprial & Kanwar, 2012). Media sosial sangat sering digunakan oleh masyarakat modern saat ini. Adapun berbagai macam media sosial yang sudah diciptakan, seperti Facebook, Twitter, Instagram dan masih banyak lagi. Adapun istilah yang erat kaitannya dengan media sosial yaitu jejaring sosial. Jejaring sosial adalah struktur sosial yang terdiri dari beragam individu ataupun kelompok organisasi yang dihubungkan karena memiliki kesamaan sosialitas, visi, ide dan lain sebagainya (Priansya, 2017). Dalam setiap jejaring sosial terdapat cara masing-masing dalam berkomunikasi dengan anggotanya. Cara-cara tersebut biasa memanfaatkan fitur yang diberikan pada setiap media sosial, misalnya menggunakan fitur mengirim gambar sebagai sarana utama dalam menyebarkan informasi. Ada pula yang hanya menggunakan percakapan *(chatting)* sebagai hal utama dalam menyebarkan informasi.

Salah satu dari jejaring sosial adalah Masyarakat Anti Fitnah Indonesia (MAFINDO). Organisasi tersebut merupakan perkumpulan resmi yang berdiri pada tanggal 19 November 2016 dengan Akta Notaris Nomor 1 tanggal 19 November 2016 yang dibuat oleh Isma Januarti, SH,. M.KN. Organisasi bergerak dengan mengumpulkan konten berita melalui forum jejaring sosial facebook, kemudian setiap konten berita yang didapatkan akan didiskusikan dalam forum yang kemudian hasil forum tersebut akan disebarkan melalui situs resmi milik MAFINDO yaitu *turnbackhoax.id*. Penanggung jawab pada situs *turnbackhoax.id* terdiri dari Muhammad Khairil Haesy M.Hum, Dedy Helsyanto S.IP, Bentang Febrylian S.IKOM dan Syarief Ramaputra S.IKOM.

B. Text Mining

Text mining dapat diartikan sebagai suatu alur proses dimana pengguna berinteraksi dengan kumpulan dokumen menggunakan aplikasi analisis. *Text mining* merupakan suatu proses untuk mendapatkan informasi yang berguna dari sumber data dengan mengidentifikasi dan mengeksplorasi pola tertentu. Menurut Harlian (2006) *text mining* adalah menambang data yang berupa teks dimana sumber data biasanya didapatkan dari dokumen dan tujuannya adalah mencari kata-kata yang dapat mewakili isi dari dokumen sehingga dapat dilakukan analisa keterhubungan antar dokumen. Tujuan dari *text mining* adalah untuk memproses *knowledge discovery* data informasi teks yang tidak terstruktur yang kemudian akan dianalisis untuk menemukan pola yang menarik seperti mengekstrak indeks *numeric* pada teks yang kemudian informasi yang didapat dalam teks dapat diolah menggunakan algoritma *data mining* (Feldman & Sanger, 2007). Pengolahan *text*

pada *text mining* untuk mendapatkan informasi tersebut biasa dikenal dengan nama *text processing*.

C. Text Procesing

Text processing merupakan tahapan dalam mengelola suatu teks yang tidak teratur atau tidak terstruktur agar menjadi lebih terstruktur sehingga teks tersebut dapat diolah kembali sehingga mendapatkan informasi dan pola teks pada suatu dokumen tertentu. Menurut Feldman & Sanger (2007) *text preprocessing* adalah proses pengubahan bentuk menjadi data yang terstruktur sesuai kebutuhannya untuk proses dalam data mining yang biasanya akan menjadi nilai-nilai numerik. Menurut Triawati & Candra (2009), adapun tahapan-tahapan pada saat melakukan *text processing* yaitu:

1. *Text Normalization*
   Proses *text normalization* merupakan proses untuk mengubah teks pada sebuah dokumen dari kata yang tidak tepat atau kata singkatan bahasa gaul menjadi kata yang memiliki arti. Contohnya seperti "yg" menjadi "yang", "tdk" menjadi "tidak" dan lain sebagainya.
2. *Case Folding*
   Case folding adalah proses pada *text processing* yang berguna untuk menghilangkan tanda baca, *double* spasi dan angka yang kemudian setiap kata yang sudah diolah tersebut diubah menjadi huruf kecil.
3. *Tokenization*
   Pada proses *tokenizing* berfungsi untuk mengubah suatu kalimat menjadi setiap kata yang menyusun kalimat tersebut. Misalnya pada kalimat "saya akan pergi berlibur" maka akan menjadi "saya", "akan", "pergi", "berlibur" dengan menggunakan metode *tokenizing*.
4. *Filtering*
   Proses *filtering* merupakan proses untuk menghilangkan kata-kata yang kurang deskriptif atau kata penghubung yang kurang memiliki makna pada setiap dokumen. Proses *filtering* ini biasa disebut juga dengan *stopword removal*. Proses *filtering* ini bertujuan untuk mengurangi jumlah kata pada dokumen yang dianggap kurang penting. Contoh dari kata yang kurang penting adalah "yang", "dan", "dari" dan lain lain.
5. *Stemming*
   Stemming adalah proses pengolahan teks yang berfungsi untuk memotong imbuhan dari setiap kata dan membuat setiap kata tersebut menjadi kata dasar.

D. Pembobotan *TF-IDF*

TF-IDF merupakan kepanjangan dari *Term Frequency Inverse Document Frequency*. TF-IDF adalah algoritma yang berguna untuk menghitung nilai bobot (*weight)* pada setiap kata yang ada pada dokumen. Nilai bobot tersebut merupakan nilai pentingnya sebuah kata yang ada dalam dokumen, semakin besar bobotnya maka peran kata tersebut sangat penting dalam dokumen. Menurut Manning, Raghavan, & Schutze (2009) TF-IDF adalah penggabungan dua konsep untuk perhitungan bobot yaitu *Term Frequency (TF)* dan *Invers Document Frequency (IDF)*. *Term Frequency* adalah banyak jumlah kata atau *term* yang ada pada suatu

dokumen, sedangkan *Invers Document Frequency* adalah frekuensi munculnya kata atau *term* tertentu yang ada pada suatu dokumen.

Pendekatan TF-IDF menyajikan teks dengan ruang vektor yang di setiap fitur dalam teks sesuai dengan satu kata (Zhang, Gong, & Wang, 2005). TF (*Term Frequency*) akan menghitung frekuensi kemunculan sebuah kata dan dibandingkan jumlah seluruh kata yang ada di dalam dokumen (Saadah, Atmagi, Rahayu, & Arifin, 2013). Menurut Havrlant & Kreinovich (2014) menambahkan bahwa rumus pada TF-IDF ditunjukkan pada persamaan 1 adalah:

$$\boldsymbol{W_{dt} = tf_d \times idf_t = tf_d \times \log\left(\frac{N}{df_t}\right)} \qquad \textbf{(1)}$$

Keterangan :

$\boldsymbol{W_{dt}}$ = Nilai bobot *term* ke-t pada dokumen d
$\boldsymbol{tf_d}$ = Jumlah munculnya *term* t pada dokumen d
N = Jumlah dokumen secara keseluruhan
$\boldsymbol{df_t}$ = Jumlah dokumen yang mengandung *term* t

E. Multinomial Naïve Bayes (MNB)

Multinomial Naïve Bayes adalah model yang dikembangkan dari algoritma *Bayes* yang cocok dalam hal pengklasifikasian teks atau dokumen. Model dari *multinomial* memperhitungkan frekuensi dari setiap kata yang muncul pada dokumen tertentu. Maksud dari multinomial adalah suatu keadaan dimana *value* fitur memiliki lebih dari dua kejadian (McCallum & Nigam, 1998). Perbedaan *Multinomial Naïve Bayes* dengan model *Gaussian Naïve Bayes* adalah pada pemilihan datanya. Untuk *Multinomial Naïve Bayes* cocok pada data yang diskrit sedangkan *Gaussian Naïve Bayes* cocok pada data continue. Kelas dokumen tidak hanya ditentukan oleh kata yang muncul saja tapi juga ditentukan oleh jumlah kemunculannya. Pada model *Multinomial Naïve Bayes* menggunakan rumus yang ditunjukkan pada persamaan 2. (Rahman, Wiranto, & Doewes, 2017):

$$\mathbf{P(c|term\ dok\ d} = \mathbf{P(c)} \times \mathbf{P}(\boldsymbol{t_1}|\mathbf{c}) \times \ldots \times \mathbf{P}(\boldsymbol{t_n}|\mathbf{c}) \qquad \textbf{(2)}$$

Keterangan:

P(c|term dok d) = Probabilitas suatu dokumen dalam kelas c
P(c) = Probabilitas *prior* dari kelas c
P($\boldsymbol{t_n}$|c) = Probabilitas kata ke-n pada kelas c
$\boldsymbol{t_n}$ = kata ke n pada dokumen

Untuk menentukan probabilitas *prior* dari kelas c dihitung dengan menggunakan rumus yang ditunjukkan pada persamaan 3.

$$\boldsymbol{P(c) = \frac{Nc}{N}} \qquad \textbf{(3)}$$

Keterangan:

Nc = Banyak kelas c pada semua dokumen
N = Banyak seluruh dokumen

Probabilitas kata ke-n pada kelas c yang digunakan dengan bobot kata TF-IDF dihitung dengan menggunakan rumus pada persamaan 4.

$$\mathbf{P(tn|c)} = \frac{\boldsymbol{Wct+1}}{(\sum \boldsymbol{W'€VW'ct)+B'}} \quad \textbf{(4)}$$

Keterangan:
Wct = Nilai pembobotan TF-IDF atau nilai W dari *term* t yang ada pada kategori c
$\boldsymbol{\sum W'€VW'ct}$ = Jumlah total W dari seluruh *term* yang berada di kategori c
$\boldsymbol{B'}$ = Jumlah W dari kata unik atau nilai idf yang tidak di kali tf pada seluruh dokumen

F. Multivariate Bernoulli

Multivariate Bernoulli adalah salah satu model algoritma klasifikasi yang dikembangkan dari algoritma *Naïve Bayes Classifier* yang cocok dalam hal pengklasifikasian teks atau dokumen (McCallum & Nigam, 1998). Model dari *Multivariate Bernoulli* memperhitungkan jumlah data yang mengandung kata *term*, bukan frekuensi kemunculan kata. Pada model *Multivariate Bernoulli* menggunakan persamaan pada persamaan 5. (Karunia, Saptono, & Anggrainingsih, 2017):

$$\boldsymbol{P(c\,|d) = P\,(c) x \prod_{i=1}^{N} P(fk|c)\; x\; \prod_{i=1}^{M}(1 - P(fk|c))} \quad \textbf{(5)}$$

Keterangan:
P(c | d) = Probabilitas suatu dokumen dalam kelas c
P(c) = Probabilitas *prior* dari kelas c
P(fk|c) = Probabilitas setiap kata

Untuk menentukan probabilitas *prior* dari kelas c dihitung dengan menggunakan rumus yang ditunjukkan pada persamaan 6.

$$\boldsymbol{P(c) = \frac{Nc}{N}} \quad \textbf{(6)}$$

Keterangan;
Nc = Banyak kelas c pada semua dokumen
N = Banyak seluruh dokumen

Probabilitas kata ke-n pada kelas c yang digunakan dengan bobot kata TF-IDF dihitung dengan menggunakan rumus yang ditunjukkan pada persamaan 7.

$$\boldsymbol{P(fk|c) = \frac{T_{ct}+1}{T_c + \sum c}} \quad \textbf{(7)}$$

Keterangan:
$\boldsymbol{T_{ct}}$ = Banyak dokumen yang mengandung *term t* pada kelas c
$\boldsymbol{T_c}$ = Jumlah data latih pada setiap kelas c
$\boldsymbol{\sum c}$ = Jumlah kelas atau banyak kategori

G. Rocchio

Rocchio merupakan salah satu dari algoritma klasifikasi dengan bentuk linear yang menerapkan prinsp dasar *contiguity hypothesis* yang berarti tidak akan terjadi *overlap* antara kelas yang sama dengan kelas yang berbeda. Nilai *centroid* dihitung dengan persamaan 8. (Manning, Raghavan & Schutze, 2009):

$$\overrightarrow{\boldsymbol{u}}(\boldsymbol{c}) = \frac{\mathbf{1}}{\boldsymbol{Dc}} \sum \boldsymbol{d} \in \boldsymbol{Dc}\overrightarrow{\boldsymbol{v}}(\boldsymbol{d}) \quad \textbf{(8)}$$

Keterangan:
$\overrightarrow{\boldsymbol{u}}(\boldsymbol{c})$ = Nilai *centroid* dari masing-masing kelas
$\boldsymbol{Dc}$ = Gugus dokumen
$\sum \boldsymbol{d} \in \boldsymbol{Dc}\overrightarrow{\boldsymbol{v}}(\boldsymbol{d})$ = Jumlah vektor kata dalam kelas c

Untuk melakukan klasifikasi dalam algoritma *rocchio* dilakukan perhitungan *cosine similarity* antara titik d1 dan d2 dengan persamaan yang ditunjukkan pada persamaan 9. (Afriza & Adisantoso, 2017):

$$\boldsymbol{sim}(\boldsymbol{d1}, \boldsymbol{d2}) = \frac{\overrightarrow{\boldsymbol{v}}(\boldsymbol{d1}).\overrightarrow{\boldsymbol{v}}(\boldsymbol{d2})}{||\overrightarrow{\boldsymbol{v}}(\boldsymbol{d1})||.||\overrightarrow{\boldsymbol{v}}(\boldsymbol{d2})||} \quad \textbf{(9)}$$

Keterangan:
$\boldsymbol{sim}(\boldsymbol{d1}, \boldsymbol{d2})$ = Jarak kecocokan antara uji terhadap kelas
$\overrightarrow{\boldsymbol{v}}(\boldsymbol{d1})$ = Nilai vektor *centroid* setiap kelas
$\overrightarrow{\boldsymbol{v}}(\boldsymbol{d2})$ = Nilai vektor data uji
$\left||\overrightarrow{\boldsymbol{v}}(\boldsymbol{d1})|\right|$ = Nilai panjang vektor *centroid*
$\left||\overrightarrow{\boldsymbol{v}}(\boldsymbol{d1})|\right|$ = Nilai panjang vektor data uji

3. HASIL DAN PEMBAHASAN

Data yang akan digunakan berjumlah 200 data berita terdiri dari 110 data berita *hoax* dan 90 data berita benar. Dari 200 data tersebut selanjutnya akan dilakukan *cleaning* agar bersih dari *noise* atau kata yang tidak diperlukan seperti kata penghubung, bahasa gaul, dll. Setelah data terkumpul selanjutnya dilakukan proses *text normalization* yang berfungsi untuk mengubah kata singkatan atau bahasa gaul. Setelah dilakukan *text normalization* masuk ke dalam tahap *text processing.* Pada proses ini terdiri dari beberapa bagian yaitu *case folding*, *tokenizing, filtering* dan *stemming*. Selanjutnya dilakukan pembagian data atau partisi data dengan menggunakan *K Fold Cross Validation* untuk mendapatkan model terbaik yang kemudian akan diujikan ke dalam data uji baru. Penggunaan *K Fold* pada penelitian ini adalah 2, 4, 5, 8 dan 10. Setelah dilakukan pembagian data maka masuk kedalam tahap klasifikasi menggunakan algoritma *Multinomial Naïve Bayes (MNB), Multivariate Bernoulli* dan *Rocchio*.

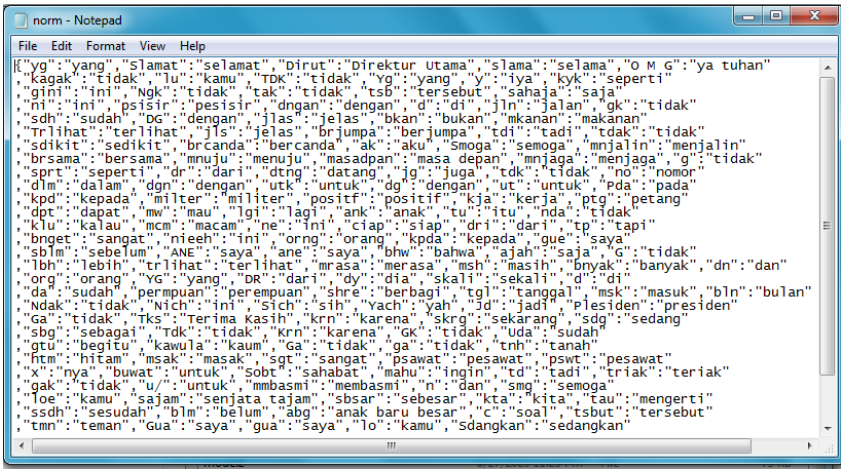
A. *Import Data*

```
data = DataFrame(pd.read_excel('data.xlsx'))
data
```

Gambar 1. *Import Data*

Gambar 1 menunjukkan perintah yang digunakan untuk *import* data yang akan digunakan dalam penelitian tersebut. Data yang digunakan menggunakan *type file* .xlsx yang kemudian akan ditampilkan dalam bentuk frame data yang di *import* dari *library* yang *pandas* yang ada pada *jupyter notebook*.

B. *Text Normalization*



Gampbar 2. Library *Text Normalization*

Sebelum masuk ke *text normalizaton* diperlukan *library* kata yang akan digunakan sebagai penyimpanan kata yang akan dilakukan *normalization*. Proses normalization ini berfungsi untuk mengubah kata singkatan, kata gaul, dll. Misalnya seperti kata “yg” menjadi “yang”. Library tersebut dibuat dengan nama norm.txt.

```
file = open("norm.txt")
dataNorm = file.read()
```

Gambar 3. Perintah Membaca Data *.txt*

Selanjutnya buka *file library* tersebut pada *jupyter notebook* dengan menggunakan fungsi *open*. Lalu file tersebut akan di baca dengan fungsi *read()* yang disimpan ke dalam variabel dataNorm.

```
norm = json.loads(dataNorm)
```

Gambar 4. Perintah *json.loads*

Langkah selanjutnya mengubah dataNorm ke dalam tipe *json* dengan perintah *json.loads()* yang kemudian akan disimpan ke dalam variabel norm.

```
def textNorm(text, dicNorm):
    for key, value in dicNorm.items():
        text = re.sub(r"\b%s\b" % key, value, text)
    return text
```

Gambar 5. Fungsi *Text Normalization*

Setelah pembuatan *library* untuk *text normalization* selesai, selanjutnya membuat fungsi yang akan digunakan untuk melakukan proses *text normalization*. Fungsi tersebut diberi nama dengan textNorm yang memiliki dua parameter yaitu *text* untuk memasukkan data asli dan dicNorm untuk memasukkan data *library normalization*.

```
data["hasil_norm"] = data.apply(lambda row: textNorm(row.narasi,norm), axis=1)
data
```

Gambar 6. Menjalankan Fungsi *Text Normalization*

Selanjutnya menjalankan fungsi textNorm yang kemudian akan disimpan dalam kolom hasil_norm pada variabel data.

C. *Text Processing*

```
def prosesTextMining(narasi_input):

    #Case Folding
    teks_narasi = narasi_input.lower()

    #cleaning data
    re_char = re.sub(r"\W", " ", teks_narasi) #hapus character tanda baca
    re_number = re.sub(r"\d", "", re_char) #hapus angka
    hasil_ds = re.sub(r"\s+"," ", re_number, flags = re.I) #hapus double spasi
    hasil_spasi = re.sub(r"^\s+", "", hasil_ds) #hapus spasi di awal dan akhir
    hasil_clean = re.sub(r"\s+[a-zA-Z]\s+", " ", hasil_spasi) #hapus single char

    #Tokenizing
    hasil_tokens = nltk.tokenize.word_tokenize(hasil_clean)

    #Filtering
    listStopword =  stopwords.words('indonesian')
    stop = stoplisttext
    datastop = listStopword+stop

    filtering = [w for w in hasil_tokens if w not in datastop]
    hasil_filter = " ".join(filtering)

    #Stemming
    factory = StemmerFactory()
    stemmer = factory.create_stemmer()
    steming = stemmer.stem(hasil_filter)

    return steming
```

Gambar 7. Fungsi *Text Processing*

Setelah melakukan *text normalization* masuk ke dalam tahap *text processing* yang memiliki beberapa tahapan yaitu *case folding*, *tokenizing*, *filtering*, *stemming*. Proses di atas merupakan pembuatan fungsi *text processing* dengan nama fungsinya yaitu prosesTextMining dengan satu parameter yaitu data yang akan dilakukan proses *text processing.*

```
data["hasil_textProcess"] = data.apply(lambda row: prosesTextMining(row.hasil_norm), axis=1)
```

Gambar 8. Menjalankan Fungsi *Text Processing*

Selanjutnya menjalankan proses *text processing* dengan memanggil fungsi prosesTextMining yang kemudian akan disimpan ke dalam variabel data pada kolom hasil_textProcess.

D. *Pembagian K-Fold Cross Validation*

```
x = data['hasil_textProcess']
y = data['hasil_label']

kf=KFold(n_splits=2)
for trainingIndex, testingIndex in kf.split(x, y):
```

Gambar 9. Perintah K Fold Cross Validation

K Fold digunakan untuk membagi data antara data training dan data testing sesuai dengan k yang digunakan. Pada penelitian ini k yang digunakan adalah k = 2, 4, 5, 8, dan 10. Variabel x merupakan data text yang akan di klasifikasi dan variabel y adalah label outputnya, n_splits berfungsi untuk menentukan k yang akan digunakan pada k fold, perulangan for digunakan untuk membagi data antara training dan testing sesuai dengan k fold yang digunakan.

E. Klasifikasi

```
x = data['hasil_textProcess']
y = data['hasil_label']

i = 1

kf=KFold(n_splits=2, shuffle=True, random_state=25)
for trainingIndex, testingIndex in kf.split(x, y):
    X_train = x[trainingIndex]
    y_train = y[trainingIndex]
    X_test = x[testingIndex]
    y_test = y[testingIndex]

    Tfidf_vect = TfidfVectorizer(max_features=5000)
    Tfidf_vect.fit(data['hasil_textProcess'])
    Train_X_Tfidf = Tfidf_vect.transform(X_train)
    Test_X_Tfidf = Tfidf_vect.transform(X_test)

    model_MNB = naive_bayes.MultinomialNB()
    model_MNB.fit(Train_X_Tfidf,y_train)
    predictions_MNB = model_MNB.predict(Test_X_Tfidf)

    print('HASIL ',i)
    print()
    tes = classification_report(y_test, predictions_MNB, output_dict=True)
    print(confusion_matrix(y_test, predictions_MNB))
    print ()
    print('accuracy \t:',"{:.2f}".format((tes['accuracy'])*100),'%')
    print('presisi \t:',"{:.2f}".format((tes['0']['precision'])*100),'%')
    print('Recall \t\t:',"{:.2f}".format((tes['0']['recall'])*100),'%')
    print ()
    i+=1
```

Gambar 10. Klasifikasi *Multinomial Naïve Bayes (MNB)*

Pada gambar 10 menunjukkan proses klasifikasi dengan menggunakan algoritma *Multinomial Naïve Bayes (MNB)* dengan mengimport library algoritma *Multinomial Naïve Bayes (MNB)* yang disediakan oleh sklearn.

```
x = data['hasil_textProcess']
y = data['hasil_label']

i = 1

kf=KFold(n_splits=2, shuffle=True, random_state=25)
for trainingIndex, testingIndex in kf.split(x, y):
    X_train = x[trainingIndex]
    y_train = y[trainingIndex]
    X_test = x[testingIndex]
    y_test = y[testingIndex]

    Tfidf_vect = TfidfVectorizer(max_features=5000)
    Tfidf_vect.fit(data['hasil_textProcess'])
    Train_X_Tfidf = Tfidf_vect.transform(X_train)
    Test_X_Tfidf = Tfidf_vect.transform(X_test)

    model_Bernoulli = BernoulliNB()
    model_Bernoulli.fit(Train_X_Tfidf,y_train)
    predictions_Bernoulli = model_Bernoulli.predict(Test_X_Tfidf)

    print('HASIL ',i)
    print()
    tes1 = classification_report(y_test, predictions_Bernoulli, output_dict=True)
    print(confusion_matrix(y_test, predictions_Bernoulli))
    print()
    print('accuracy \t:',"{:.2f}".format((tes1['accuracy'])*100),'%')
    print('presisi \t:',"{:.2f}".format((tes1['0']['precision'])*100),'%')
    print('Recall \t\t:',"{:.2f}".format((tes1['0']['recall'])*100),'%')
    print ()
    i+=1
```

Gambar 11. Klasifikasi *Multivariate Bernoulli*

Pada gambar 11 merupakan proses klasifikasi dengan menggunakan algoritma *Multivariate Bernoulli* dengan mengimport library algoritma *Multivariate Bernoulli* yang disediakan oleh sklearn.

```
x = data['hasil_textProcess']
y = data['hasil_label']

i = 1

kf=KFold(n_splits=2, shuffle=True, random_state=25)
for trainingIndex, testingIndex in kf.split(x, y):
    X_train = x[trainingIndex]
    y_train = y[trainingIndex]
    X_test = x[testingIndex]
    y_test = y[testingIndex]

    Tfidf_vect = TfidfVectorizer(max_features=5000)
    Tfidf_vect.fit(data['hasil_textProcess'])
    Train_X_Tfidf = Tfidf_vect.transform(X_train)
    Test_X_Tfidf = Tfidf_vect.transform(X_test)

    model_Rocchio = NearestCentroid()
    model_Rocchio.fit(Train_X_Tfidf,y_train)
    predictions_Rocchio = model_Rocchio.predict(Test_X_Tfidf)

    print('HASIL ',i)
    print()
    tes2 = classification_report(y_test, predictions_Rocchio, output_dict=True)
    print(confusion_matrix(y_test, predictions_Rocchio))
    print()
    print('accuracy \t:',"{:.2f}".format((tes2['accuracy'])*100),'%')
    print('presisi \t:',"{:.2f}".format((tes2['0']['precision'])*100),'%')
    print('Recall \t\t:',"{:.2f}".format((tes2['0']['recall'])*100),'%')
    print ()
    i+=1
```

Gambar 12. Klasifikasi *Rocchio Algorithm*

Pada gambar 12 menunjukkan proses klasifikasi dengan menggunakan algoritma *Rocchio* dengan mengimport library algoritma *Rocchio* yang disediakan oleh sklearn.

Hasil terbaik yang didapatkan dari hasil klasifikasi algoritma *Multinomial Naïve Bayes (MNB)* ditampilkan Gambar 13.

K-FOLD	AKURASI	PRESISI	*RECALL*
2	78 %	76,47 %	89,65 %
4	80%	76,47 %	92,86 %
5	82,50 %	76 %	95 %
8	84 %	77,78 %	100 %
10	85 %	81,81 %	90 %

Gambar 13. Hasil Klasifikasi *MNB*

Hasil terbaik yang didapatkan dari hasil klasifikasi algoritma *Multivariate Bernoulli* ditampilkan Gambar 14.

K-FOLD	AKURASI	PRESISI	*RECALL*
2	75 %	70,37 %	98,28 %
4	80%	74,36 %	100 %
5	85 %	81,25 %	100 %
8	84 %	78,94 %	100 %
10	85 %	81,25 %	100 %

Gambar 14. Hasil Klasifikasi *Bernoulli*

Hasil terbaik yang didapatkan dari hasil klasifikasi algoritma *Rocchio* ditampilkan Gambar 15.

K-FOLD	AKURASI	PRESISI	*RECALL*
2	76 %	78,33 %	81,03 %
4	82%	85,19 %	82,14 %
5	85 %	81,81 %	90 %
8	88 %	86,67 %	92,86 %
10	95 %	90 %	100 %

Gambar 15. Hasil Klasifikasi *Rocchio*

Dari hasil klasifikasi yang didapatkan, akan diambil model terbaiknya. Untuk *MNB, Bernoulli* dan *Rocchio* akan diambil model pada *fold k=10* karena memiliki hasil yang terbaik dari semua *fold k* yang digunakan. Selanjutnya model akan digunakan sebagai data training untuk pengujian terhadap data uji baru. Data yang digunakan sebagai pengujian berjumlah 50 data yang terdiri dari 25 data berita *hoax* dan 25 data berita benar. Hasil dari pengujian data baru terhadap model terbaik ditampilkan pada Gambar 5.

ALGORITMA	AKURASI	PRESISI	*RECALL*
MNB	74 %	83,33 %	60 %
Bernoulli	70 %	62,50 %	100 %
Rocchio	76 %	88,24 %	60 %

Gambar 5. Hasil Prediksi Data Baru

Berdasarkan Hasil klasifikasi dengan menggunakan data baru didapatkan hasil akurasi, presisi dan *recall* pada algoritma *Multinomial Naïve Bayes (MNB)* sebesar 74% untuk akurasi, 83,33% untuk presisi dan 60% untuk *recall*. Sedangkan pada algoritma *Multivariate Bernoulli* mendapatkan hasil sebesar 70% untuk akurasi, 62,50% untuk presisi dan 100% untuk *recall*. Dan pada algoritma *Rocchio* mendapatkan hasil sebesar 76% untuk akurasi, 88,24% untuk presisi dan 60% untuk *recall*. Hasil yang didapatkan dari validasi data mengalami penurunan, hal ini disebabkan karena adanya karakteristik baru yang ada pada data validasi yang tidak dimiliki pada data model yang menyebabkan penurunan hasil dari segi akurasi, presisi dan *recall*.

4. DAFTAR PUSTAKA

1. Feldman, R & Sanger, J., (2007). *Text Mining Handbook: Advanced Approaches in Analyzing Unstructered Data*. New York: Cambridge University Press.
2. Han, J., Kamber, M, & Pei, J. 2011. *Data Mining: Concept and Techniques, Third Edition*. New York: Elsevier.
3. Harlian, M. 2006. *Machine Learning Text Categorization*. University of Texas, Austin.
4. Hermawati, F.A. 2013. *Data Mining*. Surabaya: Andi.
5. Havrlant, L., & Kreinovich, V. 2014. "A Simple Probabilistic of Term Frequency-Invers Document Frequency (TF-IDF)". *International Journal of General System*. University of Texas.
6. Karunia, S.A., Saptono, R., & Anggrainingsih, R. 2017. "Online News Classification Using Naïve Bayes Classifier with Mutual Information for Feature Selection". *ITSMART Vol. 6 No.1*. Universitas Sebelas Maret.
7. Kuhn, M., & Johnson, K. 2013. *Applied Predictive Modeling*. New York: Springer.
8. Larose, D.T. 2005. *Discovering Knowledge in Data: An Introduction to Data Mining*. New Jersey: John Willey & Sons, Inc.
9. Manning, C., Raghavan, P., & Schutze, H. 2015. *Introduction to Information Retrieval*. New York: Cambridge University Press.
10. McCallum, A., & Nigam, K. 1998. *A Comparison of Event Models for Naïve Bayes Text Classification*. Pittsburgh: Carnegie Mellon University.
11. Pantouw, J.C.W. 2017. Perbandingan Klasifikasi Rocchio dan Multinomial Naïve Bayes pada Analisis Sentimen Data Twitter Bahasa Indonesia. Institut Pertanian Bogor, Bogor.
12. Priansya, S. 2017. Social Media Text Normalization Using Word2vec, Levenshtein Distance, & Jaro-Winkler Distance. Institut Teknologi Sepuluh Nopember Surabaya, Final Project-KS 141501.

13. Rahman, A., Wiranto, & Doewes, A. 2017. "Online News Classification Using Multinomial Naive Bayes". *ITSMART Vol. 6 No.1*. Universitas Sebelas Maret.
14. Rasywir, E., & Purwarianti, A. 2015. "Eksperimen pada Sistem Klasifikasi Berita Hoax Berbahasa Indonesia Berbasis Pembelajaran Mesin". *Jurnal Cybermatika Vol. 3 No.2*. Institut Teknologi Bandung.
15. Taprial, V., & Kanwar, P. 2012. *Understanding Social Media*. New York: Bookboon.
16. Triawati., & Chandra. 2009. Metode Pembobotan Statistical Concept Based untuk Klastering dan Kategorisasi Dokumen Berbahasa Indonesia. Institut Teknologi Telkom, Bandung.
17. Turban, E., Aronson, J.E., & Liang, T.P. 2005. *Decision Support System and Intelligent System.* Yogyakarta: Andi Offset.
18. Wisaksono, A., & Mujiyatna, I.G. 2017. Klasifikasi Berita Berkategori Olahraga dengan Algoritma Multivariate Bernoulli Naïve Bayes dan Multinomial Naïve Bayes. Gadjah Mada University Press, Yogyakarta.
19. Zhang, Y., Gong, L., & Wang, Y. 2005. "An improved TF-IDF approach for text classification*". Journal of Zhejiang University SCIENCE Vol. 6 No.1*. Tersedia di https://doi.org/10.1631/jzus.2005.A0049.
20. Manning, C., Raghavan, P., & Schutze, H. 2015. *Introduction to Information Retrieval*. New York: Cambridge University Press.