**Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer** e-ISSN: 2548-964X
Vol. 5, No. 3, Maret 2021, hlm. 1070-1077 http://j-ptiik.ub.ac.id

# Pemanfaatan Spark untuk Analisis Sentimen Mengenai Netralitas Berita dalam Membahas Pemilu Presiden 2019 Menggunakan Metode *Naïve Bayes Classifier*

**Reza Aprilliana Fauzi[1], Imam Cholissodin[2], Bayu Rahayudi[3]**

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: [1]rfauziii41@gmail.com, [2]imamcs@ub.ac.id, [3]ubay1@ub.ac.id

**Abstrak**

Sebuah berita yang aktual dan netral merupakan harapan dari masyarakat selaku penerima informasi dari media penyampaian berita. Terutama dalam masa Pemilihan Umum di Indonesia, masih ada berita-berita yang disampaikan secara berpihak atau tidak aktual. Hal tersebut pun membuat masyarakat masih berpandangan bahwa banyak berita yang memiliki unsur keberpihakan dalam memberikan informasi. Maka dari itu, penelitian ini melakukan analisis sentimen berita dari berbagai portal berita yang membahas Pemilu tahun 2019 di Indonesia. Data pada penelitian ini diambil dari berbagai portal berita dan masing-masing portal berita mengambil 20 hingga 25 berita, sehingga akan menghasilkan data yang begitu banyak hingga 100 data. Pada penelitian ini memanfaatkan *Resilient Distributed Dataset* (RDD) yang dimiliki oleh Spark sebagai tipe data dalam mengklasifikasi sentimen berita. Metode yang digunakan untuk mengklasifikasi sentimen sebuah data (pada kasus ini adalah teks berita) yaitu *Naïve Bayes Classifier*. Metode *Naïve Bayes* memiliki kemampuan yang baik dalam mengklasifikasi sebuah data besar yang tidak terstruktur, serta memiliki model yang sederhana. Penelitian ini menggunakan tabel *Confusion Matrix* sebagai tabel evaluasi dari hasil klasifikasi sentimen berita, dengan menghitung nilai evaluasi seperti akurasi, presisi, *recall*, dan *F-Measure*. Berdasarkan berbagai pengujian dan skenario yang telah dilakukan, nilai evaluasi terbaik dihasilkan pada pengujian menggunakan *K-Fold Cross Validation* dengan nilai K=10. Pada pecahan (*fold*) ke-8 menghasilkan nilai akurasi sebesar 100%, presisi sebesar 100%, recall sebesar 100%, dan F-Measure sebesar 100%.

**Kata kunci**: analisis sentimen, *Confusion Matrix*, *Naïve Bayes Classifier*, netralitas berita, Pemilu, *Resilient Distributed Dataset*, Spark

**Abstract**

*An actual and neutral news is the hope of the public as the recipient of information to the news delivery media. Especially during the General Election in Indonesia, there are still news that are conveyed in a one side or not actual way. This also makes people still view that a lot of news has an element of partiality in providing information. Therefore, this study analyzes news sentiment from various news portals that discuss the 2019 Election in Indonesia. The data in this study were taken from various news portals and each news portal took 20 to 25 news stories, resulting in a large amount of data up to 100 data. This study using the Resilient Distributed Dataset (RDD) from the Spark platform as a data type in classifying news sentiments. The method used to classify the sentiment of a data (in this case is a news text) is the Naïve Bayes Classifier method. Naïve Bayes method has a good ability in classifying an unstructured big data, and has a simple model. This study uses the Confusion Matrix table as an evaluation table of the results of news sentiment classification, by calculating evaluation values such as accuracy, precision, recall, and F-Measure. Based on the various tests and scenarios that have been carried out, the best evaluation value is generated in the test using K-Fold Cross Validation with a value of K=10. In the 8th fraction (fold), the accuracy value is 100%, precision is 100%, recall is 100%. , and F-Measure of 100%.*

**Keywords**: *sentiment analysis, Confusion Matrix, Naïve Bayes Classifier, neutrality news, General Election, Resilient Distributed Dataset, Spark*

## 1. PENDAHULUAN

Masyarakat berpandangan bahwa banyak berita yang memiliki unsur keberpihakan dalam memberikan informasi. Sebuah *survey* mengenai kepercayaan terhadap berita online menguatkan pandangan-pandangan dari masyarakat, pada hasil *survey* tersebut menyatakan 58,2% responden cenderung tidak percaya (Priambodo, 2016). Keberpihakan berita akan terasa ketika masa Pemilu (Pemi-lihan Umum). Menurut Abraham (2016), Pemilu Presiden merupakan pesta demokrasi yang meningkatkan unsur keberpihakan sebuah berita. Setiap calon akan berlomba-lomba untuk mem-brandingkan dirinya lewat apapun, terutama media massa (Abraham, 2016). Mengambil dari permasalahan tersebut, pada penelitian ini akan melakukan analisis sentimen dari netralitas sebuah berita. Penilaian sentimen yang diguna-kan berupa positif (netral) atau negatif (tidak netral) terhadap seseorang atau sekelompok yang dijadikan materi subjeknya.

Penelitian yang dilakukan oleh Etaiwi, Biltawi, dan Naymat (2017) dalam meng-klasifikasi *behavior* dari nasabah bank dengan menggunakan metode *Naïve Bayes* dan *Support Vector Machine* (SVM). Pada proses pengolahan datanya menggunakan *open source framework* untuk mengatasi ukuran data yang besar, yaitu Apache Spark. Manfaat dari penggunaan Spark adalah mampu mendistribusikan data pada banyak mesin. Hasil dari penelitian ini menun-jukkan *Naïve Bayes* lebih efisien berdasarkan nilai presisi, *recall*, dan *F-Measure* yang ada. Etaiwi sendiri mengatakan bahwa metode *Naïve Bayes* menciptakan model yang sederhana dan berkinerja dengan baik (Etaiwi, et al., 2017).

Metode *Naïve Bayes* memiliki kemampuan yang baik dalam mengklasifikasi sebuah data besar yang tidak terstruktur, serta memiliki model yang sederhana. Sehingga penelitian ini akan menggunakan metode *Naïve Bayes* dalam pengklasifikasiannya. Selain itu pengolahan datanya akan dibantu dengan penerapan Spark. Pada penerapan Spark ini akan diketahui juga dampaknya ketika diterapkan dengan metode *Naïve Bayes*. Kemampuan dari metode *Naïve Bayes* pun akan dilihat hasil klasifikasinya dalam memberikan klasifikasi sentiment pada pada sebuah berita, dengan melakukan pengujian metode berupa akurasi, presisi, *recall*, dan *F-Measure*.

## 2. TINJAUAN PUSTAKA

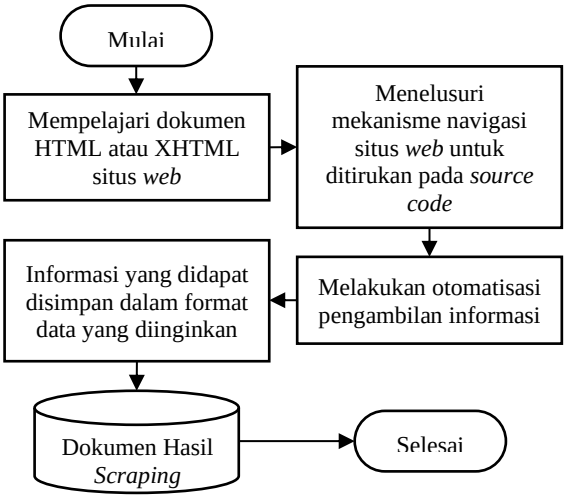
### 2.1 Spark

Apache Spark merupakan proyek yang dikembangkan oleh Matei Zaharia pada 2009 di UC Berkeley AMPLab. Ada beberapa fitur Spark yang menjadi alasan kenapa Spark baik untuk digunakan, yaitu (Jonnalagadda, et al., 2016):

1) *Easy to use*, Spark mudah untuk di-*install*.
2) *Speed*, Spark memiliki kecepatan luar biasa dibandingkan dengan Hadoop *MapReduce*.
3) *It run everywhere*, Spark dapat berjalan di Hadoop, *cloud*, bahkan secara *standalone*.
4) *MapReduces*, Spark memiliki keunggulan yang dapat menyederhanakan operasi *map* dan *reduce*.

Spark diciptakan menggunakan sebuah konsep bernama *Resilient Distributed Dataset* (RDD) (Cholissodin, et al., 2020). RDD dapat diartikan sebagai kumpulan objek terdistribusi yang bersifat *immutable*. Arti *immutable* adalah ketika terjadi proses transformasi pada sebuah RDD, maka RDD yang sebelumnya tidak akan berubah. RDD memiliki 2 jenis operasi, yang pertama adalah *transformation*, yaitu operasi yang akan merubah struktur data RDD. Kemudian yang kedua adalah *action*, yaitu operasi yang melakukan evaluasi fungsi pada RDD dan mengembalikan sebuah nilai sebagai hasil dari operasi (Ryanto, et al., 2018).

### 2.2 *Web Scraping*




*Gambar 1. Proses Web Scraping*

*Web scraping* adalah proses pengambilan atau pengikisan (*scraping*) sebuah dokumen/ data dari situs *web* yang dituju. Umumnya laman *web* selalu dibangun dengan Bahasa *markup* yaitu HTML atau XHTML, dari *markup* tersebut akan diambil datanya sesuai kepentingan yang melakukan *scraping* (Johnson & Gupta, 2012). Proses *web scraping* untuk penelitian ini dapat dilihat pada Gambar 1.

### 2.3 *Text Preprocessing*

*Text preprocessing* berfungsi sebagai transformasi data teks tidak terstruktur (seperti dokumen) menjadi data terstruktur (Langgeni, et al., 2010). Tahap yang dilakukan *tokenizing*, *filtering*, *stemming*, dan *term weighting* (Vijayarani, et al., 2015).

*Tokenizing* adalah proses memecah kalimat-kalimat menjadi kumpulan kata. Tahapan ini juga menghilangkan karakter seperti angka, tanda baca, dan karakter lainnya yang bukan huruf alfabet. Hal tersebut dikarenakan karakter selain huruf alfabet dianggap sebagai pemisah kata (*delimeter*) (Melita, et al., 2018).

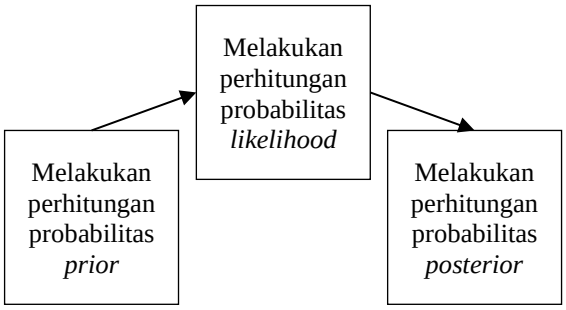
*Filtering* adalah proses penghilangan kata-kata yang tidak penting dalam pemrosesan teks melalui pengecekan setiap katanya (Melita, et al., 2018). Kata-kata yang tidak penting atau tidak berpengaruh pada proses analisis sentiment adalah kata seperti "dan, yang, atau, adalah" dan kata-kata sejenisnya. Kata-kata tersebut dapat dikatakan sebuah *stop word*, sehingga proses *filtering* bisa disebut juga sebagai proses *stop word removal* (Nata & Yudiastra, 2017).

*Stemming* adalah proses transformasi kata menjadi kata dasar, hasil *filtering* yang sudah dilakukan akan di-*check* setiap katanya dan dicari *root* (kata dasar) dari kata tersebut (Langgeni, et al., 2010). Tujuan dari proses *stemming* adalah untuk menyeragamkan kata secara akurat (Vijayarani, et al., 2015).

*Term weighting* (pembobotan kata) adalah proses pemberian nilai/bobot kata dari seluruh dokumen. Proses ini dilakukan ketika sudah melakukan *stemming*, sehingga dapat dikatakan hasil akhir dari preprocessing disebut sebagai *term* (kata). Metode yang paling umum digunakan untuk menghitung *term weighting* adalah metode TF-IDF (Melita, et al., 2018).

### 2.4 *Naïve Bayes Classifier*

Klasifikasi *Naïve Bayes* adalah klasifikasi linier yang dikenal sederhana namun sangat efisien. Proses Naïve Bayes dinyatakan pada Gambar 2, dan setiap istilah-istilah pada fungsi metode Naïve Bayes yang terdapat pada Gambar 2 memiliki perhitungannya sendiri. Persamaan dari perhitungan dari *prior* dan *posterior* adalah sebagai berikut (Aggarwal, 2015).




*Gambar 2. Proses Metode Naïve Bayes*

Mencari nilai *prior* untuk tiap-tiap kelas klasifikasi dapat dicari dengan menghitung rata-rata tiap kelas dengan Persamaan 1 berikut (Aggarwal, 2015).

$$P(Y) = \frac{N_Y}{N} \quad (1)$$

Keterangan:
$P(Y)$ = nilai *prior* dari kelas Y
$N_Y$ = jumlah data dari kelas Y
$N$ = jumlah seluruh data

Mencari nilai *posterior* dari tiap kelas klasifikasi yang ada dengan menggunakan Persamaan 2 berikut (Aggarwal, 2015).

$$P(X|Y) = P(Y) \times P(Y|X) \quad (2)$$

Keterangan:
$P(Y)$ = nilai *prior* dari kelas Y
$P(Y|X)$ = nilai *likelihood* dari kelas Y dengan karakteristik X
$P(X|Y)$ = nilai *posterior* dengan karakteristik X untuk kelas Y

### 2.5 *Multinomial Naive Bayes Classifier*

Klasifikasi *Multinomial Naïve Bayes* merupakan pengembangan dari metode *Naïve Bayes* yang bertujuan untuk mengklasifikasi data berupa teks atau dokumen. *Multinomial Naïve Bayes* merubah per-samaan dalam menghitung *likelihood* dari sebuah data yang akan diklasifikasinya.

Pada penelitian ini, perhitungan nilai jumlah seluruh kata dihitung dari hasil *term weighting* dengan metode TF-IDF. Sehingga, persamaan *multinomial* dengan menggunakan pembobotan kata TF-IDF adalah sesuai pada Persamaan 3 (Rahman, et al., 2017).

$$P(t_n|Y) = \frac{W_{yt}+1}{(\sum W' \in VW'_{yt})+B'} \quad (3)$$

Keterangan:

$W_{yt}$ = nilai pembobotan TF-IDF dari kata t di kelas Y

$B'$ = nilai TF kata unik (nilai IDF tidak dikali dengan TF) pada seluruh dokumen

$\sum W' \in VW'_{yt}$ = nilai total pembobotan TF-IDF dari di kelas Y

## 2.6 Evaluasi

Klasifikasi *biner* seperti penelitian ini dapat menggunakan model evaluasi *Confusion Matrix*. Evaluasi dengan *Confusion Matrix* dapat membantu proses analisis hasil pengujian dengan memberikan visualisasi dalam bentuk *matrix*, dikatakan sebagai *confusion* (kebingungan) karena pada visualisasi *matrix* akan menentukan apakah metode kebingungan atau tidak dalam mengklasifikasi sebuah kelas (Gad, 2020). Bentuk visualisasi *matrix* dari klasifikasi tersebut dapat dilihat pada Tabel 1.

*Tabel 1. Confusion Matrix Klasifikasi Biner*

| | **Prediksi Kelas Positif** | **Prediksi Kelas Negatif** |
|---|---|---|
| **Aktual Kelas Positif** | *True Positive* (TP) | *False Negative* (FN) |
| **Aktual Kelas Negatif** | *False Positive* (FP) | *True Negative* (TN) |

Ketika seluruh data uji sudah dimasukkan ke dalam *matrix* yang ada, perlu melakukan perhitungan lebih lanjut untuk mendapatkan informasi bagaimana performa dari metode *Naïve Bayes* dalam mengklasifikasi sentimen berita. Berikut perhitungan yang dilakukan dengan mencari nilai akurasi, presisi, *recall*, dan *F-Measure* dari data yang diuji menggunakan metode *Naïve Bayes*.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (4)$$

$$Precision = \frac{TP}{TP+FP} \quad (5)$$

$$Recall = \frac{TP}{TP+FN} \quad (6)$$

$$F-Measure = \frac{2\times R\times P}{R+P} \quad (7)$$

# 3. METODOLOGI PENELITIAN

## 3.1 Data Penelitian

Data yang digunakan dalam penelitian ini diambil dengan cara *scraping* situs *web* portal berita. Berita yang diambil merupakan berita yang membahas Pemilu Presiden 2019 di Indonesia yang terbit sebelum tanggal 17 April 2019. Pengambilan berita dilakukan dengan pencarian menggunakan *keyword* yang mengandung nama calon presiden masing-masing kandidat pada Pemilu 2019.

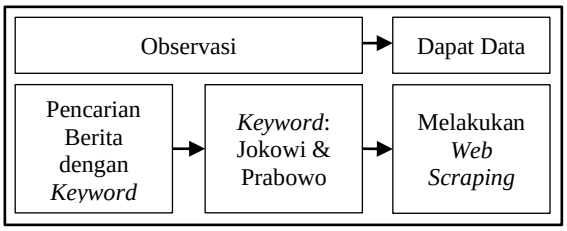
## 3.2 Perancangan

Perancangan yang dilakukan pada penelitian ini yaitu perancangan algoritme berupa diagram alir (*flowchart*) dari masing-masing tahapannya. Selain itu juga melakukan perhitungan manual yang berisi perhitungan langkah-langkah penyelesaian klasifikasi sentimen berita yang membahas Pemilu Presiden 2019 menggunakan metode Naïve Bayes. Perhitungan manual akan dilakukan tanpa proses *MapReduce* dan dengan proses *MapReduce*. Perancangan lainnya pada penelitian ini adalah membuat senario pengujian untuk menentukan bentuk-bentuk pengujian yang dilakukan pada klasifikasi sentimen berita yang membahas Pemilu Presiden 2019 menggunakan metode Naïve Bayes.

# 4. PERANCANGAN & IMPLEMENTASI

## 4.1 Perolehan Data

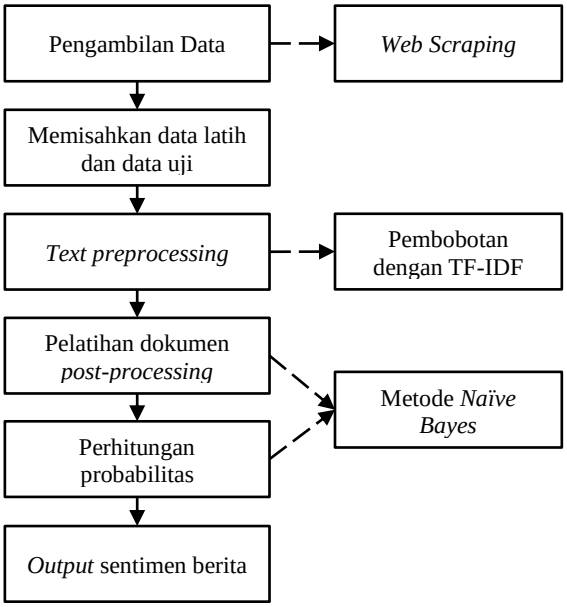
Situs *web* yang digunakan untuk men-*scraping* berita yang membahas Pemilu Presiden 2019 sebanyak 5 portal berita, yaitu portal berita *online* Detik.com, Liputan6.com, Inipasti.com, Cnnindonesia.com, dan Antaranews.com. Masing-masing dari portal berita tersebut akan diambil artikel (laman) sebanyak 20 hingga 25 laman berita, kemudian keseluruhan artikel yang sudah diambil akan dipilih secara acak. Pemilihan secara acak tersebut akan diambil sebanyak 100 laman berita, sehingga jumlah data dari dokumen hasil *scraping* pada pengambilan data ini akan menyimpan sebanyak 100 dokumen. Tahapan pengambilan laman berita untuk dijadikan dokumen hasil *scraping* dapat dilihat dalam diagram blok pada Gambar 3.




Gambar 3. Diagram Blok Pengumpulan Data

### 4.2 Implementasi

Implementasi program sistem dalam pemanfaatan Spark untuk klasifikasi sentimen berita yang membahas Pemilu Presiden 2019 menggunakan metode *Naïve Bayes* ini dilakukan dengan mengacu pada perancangan sistem pada Gambar 4 dan menggunakan bahasa pemrograman Python




*Gambar 4. Proses Metode Naïve Bayes*

Pada Implementasi ini meliputi hal-hal berikut:

1. Pengambilan data dari berbagai portal berita *online*, menggunakan teknik *web scraping*.
2. Membagi 2 dokumen hasil scraping menjadi data latih dan data uji, dengan persentase pembagian yang sudah ditentukan.
3. Melakukan *text preprocessing* sebelum proses analisis sentimen, pembobotan dilakukan dengan menggunakan TF-IDF.
4. Pengolahan data dengan melakukan pelatihan dokumen *post-processing* berdasarkan metode *Naïve Bayes*.
5. Perhitungan probabilitas setiap klasifikasi dengan metode *Naïve Bayes*.
6. Output berupa klasifikasi sentimen berita, yaitu antara positif (netral) dan negatif (tidak netral).

## 5. PENGUJIAN DAN ANALISIS

### 5.1 *Supply Training Test*

*Supply Training Test* merupakan skenario uji dengan membagi porsi dari jumlah data latih dan data uji, pembagian porsi akan dibagi dengan nilai persentase. Skenario uji ini mencari porsi data latih dan data uji yang menghasilkan nilai pengujian (akurasi, presisi, *recall*, dan *F-Measure*) yang terbaik. Nilai pengujian terbaik dilihat dari nilai yang terbesar dari masing-masing skenario uji. Pengujian *Suppy Training Test* dilakukan sebanyak 4 variasi skenario, yaitu dengan perbandingan 60%:40%, 70%:30%, 80%:20%, 90%:10%. Hasil yang diperoleh dari pengujian *Suppy Training Test* dapat dilihat pada Tabel 2.

*Tabel 2. Hasil Pengujian Supply Training Test*

| **Data Latih : Data Uji** | **60% : 40%** | **70% : 30%** | **80% : 20%** | **90% : 10%** |
|---|---|---|---|---|
| **Akurasi** | 75% | 86,67% | 80% | 90% |
| **Presisi** | 77,78% | 92,31% | 100% | 83,33% |
| ***Recall*** | 70% | 80% | 60% | 100% |
| ***F-Measure*** | 73,68% | 85,71% | 75% | 90,91% |

Hasil pengujian menunjukkan bahwa nilai akurasi, presisi, *recall*, dan *F-Measure* terbaik didapatkan ketika jumlah data latih yang digunakan sangat banyak. Semakin banyak data latih yang digunakan maka akan menghasilkan nilai pengujian yang semakin baik juga. Pada skenario ke-4 menghasilkan nilai akurasi, presisi, *recall*, dan *F-Measure* lebih dari 80%. Pada skenario ke-3 memiliki nilai yang di bawah dari skenario ke-2, tetapi nilai *recall* pada skenario ke-3 memiliki nilai *recall* terbaik

### 5.2 *K-Fold Cross Validation*

Selain pengaruh jumlah data latih yang digunakan dalam hasil klasifikasi sentimen data uji, variasi data latih yang digunakan juga dapat mempengaruhi hasil klasifikasi sentimen dari data uji. *K-Fold Cross Validation* adalah proses pengujian dengan melipat (*fold*) data menjadi bagian set data dengan jumlah yang sama sebanyak K lipatan/pecahan. *Pengujian K-Fold Cross Validation* menggunakan nilai 2, 5, dan 10 sebagai nilai K.

Nilai K pertama yang digunakan pada pengujian *K-Fold Cross Validation* kali ini yaitu 2, sehingga terdapat 2 lipatan/pecahan dari dataset yang dimiliki, dan terdapat 2 skenario uji. Hasil yang diperoleh dari pengujian *K-Fold Cross Validation* dapat dilihat pada Tabel 3.

*Tabel 3. Hasil Pengujian 2-Fold Cross Validation*

| Pecahan Data Uji | Akurasi | Presisi | *Recall* | *F-Measure* |
|---|---|---|---|---|
| **1** | 68% | 65,52% | 76% | 70,37% |
| **2** | 68% | 73,68% | 56% | 63,64% |
| **Rata-rata** | 68% | 69,60% | 66% | 67% |

Nilai K kedua yang digunakan pada pengujian *K-Fold Cross Validation* kali ini yaitu 5, sehingga terdapat 5 lipatan/pecahan dari dataset yang dimiliki, dan terdapat 5 skenario uji. Hasil yang diperoleh dari pengujian *K-Fold Cross Validation* dapat dilihat pada Tabel 4.

*Tabel 4. Hasil Pengujian 5-Fold Cross Validation*

| Pecahan Data Uji | Akurasi | Presisi | *Recall* | *F-Measure* |
|---|---|---|---|---|
| **1** | 70% | 64,29% | 90% | 75% |
| **2** | 80% | 100% | 60% | 75% |
| **3** | 75% | 77,78% | 70% | 73,68% |
| **4** | 85% | 88,89% | 80% | 84,21% |
| **5** | 80% | 100% | 60% | 75% |
| **Rata-rata** | 78% | 86,19% | 72% | 76,58% |

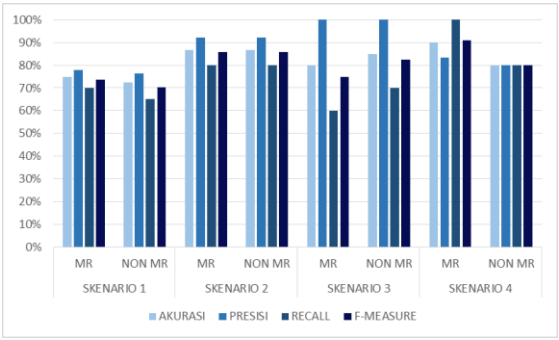
Nilai K ketiga yang digunakan pada pengujian *K-Fold Cross Validation* kali ini yaitu 10, sehingga terdapat 10 lipatan/pecahan dari dataset yang dimiliki, dan terdapat 10 skenario uji. Hasil yang diperoleh dari pengujian *K-Fold Cross Validation* dapat dilihat pada Tabel 5.

*Tabel 5. Hasil Pengujian 10-Fold Cross Validation*

| Pecahan Data Uji | Akurasi | Presisi | *Recall* | *F-Measure* |
|---|---|---|---|---|
| **1** | 70% | 62,50% | 100% | 76,92% |
| **2** | 80% | 80% | 80% | 80% |
| **3** | 80% | 100% | 60% | 75% |
| **4** | 90% | 100% | 80% | 88,89% |
| **5** | 60% | 66,67% | 40% | 50% |
| **6** | 90% | 83,33% | 100% | 90,90% |
| **7** | 70% | 75% | 60% | 66,67% |
| **8** | 100% | 100% | 100% | 100% |
| **9** | 70% | 100% | 40% | 57,14% |
| **10** | 90% | 83,33% | 100% | 90,90% |
| **Rata-rata** | 80% | 85,08% | 76% | 77,64% |

### 5.3 *MapReduce* & *Non MapReduce*

Pengujian ini akan membandingkan performa algoritme *Naïve Bayes* yang menggunakan *MapReduce* dan tanpa menggunakan *MapReduce*. Pengujian ini tetap menggunakan nilai akurasi, presisi, *recall*, dan *F-Measure* sebagai nilai pembanding dari kedua algoritme. Skenario pada pengujian ini akan menggunakan skenario yang sama pada pengujian *Supply Training Test*. Hasil nilai pengujian yang diperoleh dari masing-masing algoritme dapat dilihat dalam bentuk *chart* pada Gambar 5.



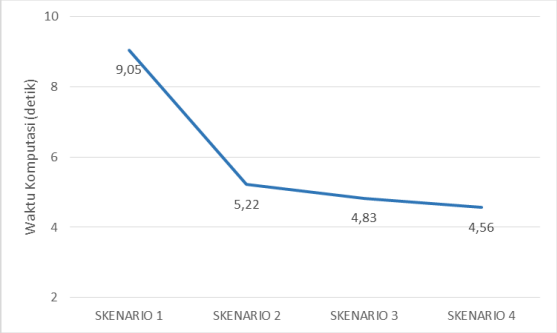

*Gambar 5. Chart Pengujian Algoritme MapReduce Dan Non MapReduce*

Setelah itu pengujian yang dilakukan adalah membandingkan nilai *run time* (waktu eksekusi program) yang didapat. Tujuan dari membandingkan nilai run time adalah untuk mengetahui keunggulan dari algoritme *Naïve Bayes MapReduce* dengan menggunakan Spark. Nilai *run time* yang diambil berdasarkan dari pengujian yang sudah dilakukan sebelumnya. Hasil nilai *run time* yang diperoleh dari masing-masing algoritme dapat dilihat pada Tabel 6.

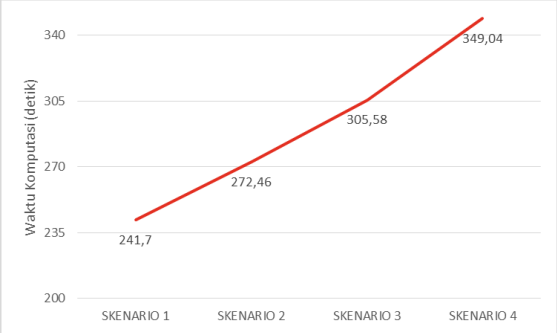
*Tabel 6. Hasil Run Time Algoritme MapReduce Dan Non MapReduce*

| Skenario | 1 | 2 | 3 | 4 | Rata-rata |
|---|---|---|---|---|---|
| ***Run Time* MR (detik)** | 9,05 | 5,22 | 4,83 | 4,56 | 5,915 |
| ***Run Time Non* MR (detik)** | 241,70 | 272,46 | 305,58 | 349,04 | 292,195 |

Algoritme *Naïve Bayes MapReduce* memiliki *run time* kurang dari 10 detik, sedangkan algoritme *Naïve Bayes Non MapReduce* memiliki *run time* lebih dari 240 detik. Gambar 6 dan 7 akan memperlihatkan *chart* waktu yang dibutuhkan setiap skenario yang dijalankan.




*Gambar 6. Run Time Naïve Bayes MapReduce*




*Gambar 7. Run Time Naïve Bayes Non MapReduce*

## 6. PENUTUP

### 6.1 Kesimpulan

1. Penerapan Spark pada penelitian ini digunakan ketika proses pengolahan data dari tahap *text preprocessing* hingga klasifikasi sentimen berita. Keunggulan dari penerapan Spark dapat dilihat pada perbandingan *run time* yang dibutuhkan. Proses pengolahan data dari seluruh skenario dengan *Map-Reduce* menghasilkan *run time* dengan nilai rata-rata 5,915 detik. Proses pengolahan data dari seluruh skenario tanpa *MapReduce* menghasilkan *run time* dengan nilai rata-rata 292,195 detik. Nilai evaluasi yang diberikan dengan menerapkan *MapReduce* pada Spark pun menghasilkan nilai yang lebih baik daripada tanpa *MapReduce*.
2. Nilai persentase terbaik dari akurasi, presisi, *recall*, dan *F-Measure* dari klasifikasi senti-men berita dengan menggunakan metode *Naïve Bayes* adalah pada skenario 10-*Fold Cross Validation*. Skenario pada pecahan (fold) ke-8 menghasilkan nilai akurasi sebesar 100%, presisi sebesar 100%, *recall* sebesar 100%, dan *F-Measure* sebesar 100%.

### 6.2 Saran

1. Penelitian ini menerapkan Spark dan membandingkan run time yang dibutuhkan dengan tanpa Spark. Perlu perbandingan tingkat lanjut seperti antara *MapReduce* Spark dengan *MapReduce* Hadoop (untuk membuktikan bahwa Spark 100 kali lebih cepat), atau antara Spark *single node* dengan Spark *multi node*.
2. Pembobotan kata (*term weighting*) yang digunakan pada penelitian ini adalah TF-IDF. Walaupun sudah menghasilkan nilai evaluasi yang baik, tetapi perlu menerapkan pembobotan kata lain seperti *Term Frequency-Relevance Frequency* (TF-RF) atau *Weighted Inverse Document Frequency* (WIDF) untuk membandingkan dan mencari nilai evaluasi yang terbaik dalam melakukan klasifikasi sentimen berita dengan metode *Naïve Bayes*.

## 7. DAFTAR PUSTAKA

Abraham, D., 2016. *Mempersoalkan Keberpihakan Media, Sama Saja Bertanya Kapan Kiamat Tiba!*. [Online] Available at: https://www.kompasiana.com/diazab/57fb6d58c5afbda4222d1799/mempersoalkan-keberpihakan-media-sama-saja-bertanya-kapan-kiamat-tiba?page=all [Accessed 10 Maret 2020].

Aggarwal, C. C., 2015. *Data Classification Algorithms and Applications.* Florida: CRC Presss.

Cholissodin, I. et al., 2020. Smart Development of Big Data App for Determining the Modelling of Covid-19 Medicinal Compounds Using Deep AI Core Engine System. *Journal of Physics: Conference Series*, pp. 1-9.

Etaiwi, W., Biltawi, M. & Naymat, G., 2017. Evaluation of Classification Algorithms for Banking Customer's Behavior under Apache Spark Data Processing System. *Elseiver B. V.*, pp. 559-564.

Gad, A. F., 2020. *Evaluating Deep Learning Models: The Confusion Matrix,*

*Accuracy, Precision, and Recall*. [Online]
Available at: https://blog.paperspace.com/deep-learning-metrics-precision-recall-accuracy/
[Accessed 22 December 2020].

Johnson, F. & Gupta, S. K., 2012. Web Content Mining Techniques: A Survey. *International Journal of Computer Applications*, XLVII(11), pp. 44-50.

Jonnalagadda, V. S., Srikanth, P., Thumati, K. & Nallamala, S. H., 2016. A Review Study of Apache Spark in Big Data Processing. *International Journal of Computer Science Trends and Technology (IJCST)*, IV(3), pp. 93-98.

Langgeni, D. P., Baizal, Z. A. & Wibowo, Y. F. A., 2010. *Clustering Artikel Berita Berbahasa Indonesia Menggunakan Unsupervised Feature Selection*, Yogyakarta: Seminar Nasional Informatika.

Melita, R., Amrizal, V., Suseno, H. B. & Dirjam, T., 2018. Penerapan Metode Term Frequency Inverse Document Frequency (TF-IDF) dan Cosine Similarity Pada Sistem Temu Kembali Informasi untuk Mengetahui Syarah Hadits Berbasis Web (Studi Kasus: Syarah Umdatil Ahkam). *Jurnal Teknik Informatika*, XI(2), pp. 149-164.

Nata, G. N. M. & Yudiastra, P. P., 2017. *Preprocessing Text Mining Pada Email Box Berbahasa Indonesia*. Bali, STMIK STIKOM.

Priambodo, B., 2016. *Hasil Survei Kepercayaan terhadap Berita Online*. [Online]
Available at: https://medium.com/@bobbypriambodo/hasil-survei-kepercayaan-terhadap-berita-online-d09afb702219
[Accessed 10 Maret 2020].

Rahman, A., Wiranto & Doewes, A., 2017. Online News Classification Using Multinomial Naive Bayes. *Jurnal Ilmiah Teknologi dan Informasi*, Vi(1), pp. 32-38.

Ryanto, A. M., Ilham, A. A. & Niswar, M., 2018. *Analisis Kinerja Framework Big Data Pada Cluster Tervirtualisasi: Hadoop Mapreduce dan Apache Spark*, Makassar: Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin.

Vijayarani, D. S., Ilamathi, M. J. & Nithya, M., 2015. Preprocessing Techniques for Text Mining - An Overview. *International Journal of Computer Science & Communication Networks*, V(1), pp. 7-16.