Klasifikasi Artikel Berita Bahasa Indonesia Dengan Naive Bayes Classifier

Anthony Setiawan[1], Leo Willyanto Santoso[2], Rudy Adipranata[3]
Program Studi Informatika,
Fakultas Teknologi Industri,
Universitas Kristen Petra
Jl. Siwalankerto 121-131, Surabaya 60236
Telp (031) – 2983455, Fax. (031) - 8417658
m26415096@gmail.com [1], leow@petra.ac.id [2], rudya@petra.ac.id [3]

ABSTRAK

Akses manusia untuk berita terbaru sekarang semakin mudah dan semakin banyak, disebabkan oleh perkembangan teknologi yang sudah maju pada masa ini. Tetapi karena pengisian kategori dari berita masih dilakukan secara manual, maka terkadang terjadi kesalahan pemilihan kategori yang tepat untuk berita yang dimasukan, atau malah terkadang ada pihak yang dengan sengaja memasukan berita tersebut ke kategori yang popular walaupun sebenarnya berita tersebut tidak berhubungan dengan kategori tersebut, dikarenakan kategori yang dipilih sedang popular dan pihak yang curang tersebut ingin beritanya dibaca oleh banyak orang. Oleh karena itu dibuatlah aplikasi berupa website yang dapat mengkategorikan berita secara otomatis sesuai dengan isi artikel.

Aplikasi ini menggunakan fitur N-Gram dan metode Naïve Bayes Classifier untuk mengklasifikasikan isi artikel. Fitur N-Gram merupakan suatu fitur yang digunakan untuk mengkelompokan suatu kumpulan kata sesuai dengan jumlah N yang diinginkan, seperti unigram dan bigram. Naïve Bayes Classifier merupakan suatu metode yang menggunakan teori probabilitas untuk menyelesaikan sebuah masalah.

Menurut hasil pengujian terhadap metode Naïve Bayes Classifier, pada perbandingan dataset training dan test 50 : 50, pada unigram didapatkan akurasi ketepatan sebesar 0.901, sedangkan pada bigram didapat sekitar 0.508. Pada perbandingan dataset sebesar 60 : 40, pada unigram didapatkan akurasi ketepatan rata-rata sebesar 0.904, sedangkan pada bigram didapat sekitar 0.498. Pada perbandingan dataset sebesar 70 : 30, pada unigram didapatkan akurasi ketepatan rata-rata sebesar 0.947, sedangkan pada bigram didapat sekitar 0.519. Pada perbandingan dataset sebesar 80 : 20, pada unigram didapatkan akurasi ketepatan rata-rata sebesar 0.887, sedangkan pada bigram didapat sekitar 0.507. Sehingga bisa diambil kesimpulan bahwa perbandingan dataset training dan test sebesar 70 : 30 memiliki akurasi ketepatan yang paling besar baik pada unigram(0.947) maupun bigram(0.519).

Kata Kunci: *Naïve Bayes Classifier, N-Gram, Klasifikasi Artikel Bahasa Indonesia, Web Scraping.*

ABSTRACT

Human access to latest news now becoming more easier and much more, caused by advanced technological development in latest years. But, the article categorization is still manually inserted by the writer, so sometimes by human error, some mistake can be happening, like inserting wrong category or sometimes the writer purposely insert wrong category just because that category is so popular just to boost his viewer count. That's why there is an application in the form of website to automatically categorizing the article that fit mostly to their its category.

This application is using N-Gram feature and Naïve Bayes Classifier method to classifying news content. N-Gram feature is a feature that group words based on the amount of N, like unigram or bigram. Naïve Bayes Classifier is a method that using probability to solve some problem.

According to the test using Naïve Bayes Classifier, in dataset training and test with ratio of 50 : 50, at unigram section the correct accuracy result are 0.901, and the bigram result are 0.508. In dataset ratio of 60 : 40, at unigram section the correct accuracy result are 0.904, and the bigram result are 0.498. In dataset ratio of 70 : 30, at unigram section the correct accuracy result are 0.947, and the bigram result are 0.519. In dataset ratio of 80 : 20, at unigram section the correct accuracy result are 0.887, and the bigram result are 0.507. So, the conclusion is dataset training and test with ratio of 70 : 30 yield highest accuracy, in unigram (0.947) and also bigram (0.519).

Keywords: *Naïve Bayes Classifier, N-Gram, Classification of Indonesian News Article, Web Scraping.*

1. PENDAHULUAN

Akses manusia kepada berita terbaru sekarang semakin mudah dan semakin banyak, disebabkan oleh berkembangnya teknologi pada masa ini. Banyak website yang diperuntukan untuk penyaluran berita secara cepat dan bisa diakses oleh siapa saja dengan mudah. Tetapi hingga kini pengkategorian dari berita yang ada masih banyak dilakukan secara manual, dan terkadang ada beberapa kategori yang kurang tepat untuk berita yang ada atau malah disalahgunakan oleh pihak tertentu yang ingin artikel mereka dibaca oleh lebih banyak orang sehingga mereka mencantumkan banyak kategori yang popular yang mungkin tidak ada hubungannya sama sekali pada artikel yang mereka buat. Oleh karena itu dibutuhkan suatu aplikasi yang dapat mengkategorikan berita pada website-website tersebut secara otomatis sesuai dengan isi dalam artikel yang ada pada website tersebut.

Pada penelitian ini akan dibuat aplikasi yang dapat mengkategorikan berita-berita yang berasal dari situs berita berbahasa Indonesia yang berasal dari website tribunnews.com dan vivanews.com, tetapi tidak terbatas pada situs ini saja. Berita yang ada akan dikategorikan menjadi 5 kategori yang dipilih

berdasarkan jumlah data terbanyak yang ada pada situs tersebut. Sebelum dikategorikan, akan dilakukan pengumpulan data terlebih dahulu dengan menggunakan *web scraping*, yaitu suatu cara pengambilan teks tertentu dari situs yang dituju berdasarkan keinginan pengguna. Setelah itu akan dilakukan proses *Text Preprocessing*, yaitu pembersihan data teks yang berasal dari hasil *web scraping*, dan setelah itu akan dipisah-pisah menjadi perkata dengan menggunakan fitur *N-Gram* yang hasilnya akan digunakan untuk melakukan training dengan metode *Naïve Bayes Classifier*. Setelah mendapat hasil dari training, kemudian akan diklasifikasikan berita pada dataset test. Dataset training dan test berasal dari situs berita yang sudah memiliki kategori terlebih dahulu. Kelebihan dari metode *Naïve Bayes Classifier* adalah metode ini bersifat sederhana, mudah dipahami, dan relatif memiliki akurasi yang tinggi.

2. LANDASAN TEORI

2.1. Tinjauan Studi

• Pada penelitian tentang klasifikasi dokumen yang menggunakan metode *Naïve Bayes Classification* menghasilkan hasil akhir yang memiliki akurasi yang terbilang tinggi pada kategori dokumen politik, yaitu sebesar 95.8%, sehingga pada penelitian ini akan menggunakan metode *Naïve Bayes Classifier* karena memiliki akurasi yang terbilang tinggi pada saat akan mengklasifikasikan data menjadi kategori-kategori [8].

• Pada penelitian tentang klasifikasi emosi pada teks bahasa Indonesia dengan menggunakan metode *Multinomial Naïve Bayes Classification* terdapat hasil akhir berupa akurasi sebesar 61.7% dengan perbandingan dataset training dan test sebesar 60 : 40. Oleh karena itu pada penelitian ini akan dilakukan pembandingan akurasi yang dihasilkan oleh dataset yang perbandingannya berbeda [2].

• Pada penelitian mengenai klasifikasi dokumen bahasa Jawa dengan menggunakan *N-Gram* ini mendapat akurasi Unigram sebesar 60%, Bigram sebesar 67%, dan Trigram sebesar 73%. Oleh karena itu pada penelitian ini akan menggunakan fitur *N-Gram* untuk mengelompokan kumpulan kata yang hasilnya akan digunakan untuk melakukan klasifikasi dengan metode *Naïve Bayes Classifier*, dan diuji akurasinya [1].

2.2. Web Scraping

Web Scraping adalah suatu metode yang digunakan untuk membantu *user* mengumpulkan informasi yang terdapat pada *World Wide Web (WWW). Web Scraping* akan melakukan ekstraksi data pada *WWW*, lalu data yang didapat akan disimpan pada *file system* atau *database* yang nantinya bisa diambil kembali atau dianalisis. *Web Scraping* berbeda dengan *Web Crawler*, perbedaannya adalah *Web Crawler* mengunjungi semua situs yang berhubungan dengan situs utama yang dijadikan patokan untuk melakukan *Web Crawler*, sedangkan *Web Scraping* hanya mengunjungi situs yang ingin dituju dan mengambil data tertentu saja sesuai dengan keinginan pengguna. Oleh karena itu pada penelitian ini akan digunakan *Web Scraping* karena *Web Crawler* rawan untuk masuk atau mengambil data yang tidak berhubungan dengan data yang ingin dicari, seperti masuk kedalam situs *advertisement* maupun dihentikan oleh pemilik situs dengan menggunakan *dummy* page seperti robot.txt ataupun terjebak dalam *loop* yang bisa saja sudah disiapkan oleh pemilik situs. Berikut adalah gambar perbedaan antara *Web Crawler* dengan *Web Scraping* [3].

2.3. DOM Parser

DOM Parser adalah sebuah *application programming interface* (*API*) yang digunakan untuk mengakses dan memanipulasi dokumen, khususnya dokumen *HTML* dan *XML*. Biasanya digunakan untuk mengambil suatu informasi yang ada pada suatu dokumen atau website sesuai dengan keinginan penggunanya. Dalam penelitian ini akan digunakan untuk mengambil isi artikel dan link *HTML* yang ada pada website berita yang digunakan. *DOM Parser* ini baik digunakan ketika pengguna mengerti struktur dari website atau dokumen yang dituju, karena *DOM Parser* akan mencari terlebih dahulu elemen khusus yang menyimpan informasi yang dicari pengguna. Setelah itu informasi tersebut bisa disimpan pada suatu *file system* ataupun *database* agar nantinya dapat diakses lagi oleh pengguna sesuai dengan kebutuhannya. Sebagai contoh pada penelitian ini akan mencari tag *HTML* yang berisikan isi berita pada website yang dituju, misalnya mencari elemen class article untuk mengambil isi teks pada class itu saja yang memuat isi dari artikel [4].

2.4. Text Preprocessing

Text Preprocessing adalah tahapan dimana pengguna melakukan seleksi data yang akan diolah menjadi lebih terstruktur. Tidak ada aturan pasti tentang setiap tahapan didalam proses *Text Preprocessing*, semua tergantung dengan keinginan pengguna atau jenis data yang akan diolah agar lebih terstruktur. Pada penelitian ini akan menggunakan proses sebagai berikut, yaitu *Case Folding, Tokenization, Stopword Removal*, dan *Stemming*. *Case Folding* adalah proses mengubah semua huruf yang ada pada data yang kita inginkan menjadi huruf kecil semua. Pada proses *Tokenization* akan memecah kalimat yang ada menjadi sebuah kata, satu persatu. Setelah itu akan dilakukan proses *Stopword Removal* yang membuang kata yang dianggap sering muncul, tidak relevan, dan juga tidak bermakna, seperti kata "dan", "saya", dan juga lain sebagainya yang ada pada *Stop List Library* yang digunakan. Setelah itu akan dilakukan proses *Stemming* yang akan mengubah kata menjadi bentuk asalnya (bentuk dasar atau bentuk bakunya). Sebagai contoh kata "Berolahraga" akan diubah menjadi bentuk bakunya yaitu "olahraga". *Library* yang akan digunakan adalah *Python* Sastrawi [9].

2.5. N-Gram

N-Gram adalah sebuah *probabilistic language model* yang digunakan untuk mengelompokan kata sesuai dengan jumlah N kata yang akan dikelompokan. Biasa digunakan untuk teori komunikasi, komputasi linguistik, dan juga kompresi data. *N-Gram* yang digunakan pada penelitian ini adalah Unigram dan Bigram. Sebagai contoh adalah kata "olahraga hidup sehat" akan diuraikan menjadi sebagai berikut:

Unigram : olahraga, hidup, sehat

Bigram : olahraga hidup, hidup sehat, sehat olahraga

Setiap karakter akan dihitung berapa jumlah kemunculan pada suatu string atau kalimat. Misalkan jika muncul *N-Gram* untuk "hidup sehat" sebanyak 3 kali pada suatu kalimat, maka jumlah counter *N-gram* untuk "hidup sehat" akan bertambah sebanyak 3. Semua *N-Gram* akan dihitung berapa kali kemunculannya pada tiap artikel pada penelitian ini, lalu hasilnya akan digunakan untuk mengklasifikasikan kategori dengan menggunakan metode *Naïve Bayes Classifier* [7].

2.6. Naïve Bayes Classifier

Naïve Bayes Classifier adalah teknik yang berlandaskan pada Teorema Bayes dan cocok untuk jumlah data yang banyak dan merupakan klasifikasi dengan model statistik untuk menghitung peluang dari suatu kelas yang memiliki masing-masing kelompok atribut yang ada, dan menentukan kelas mana yang paling optimal. Pada metode ini semua atribut akan memberikan kontribusinya dalam pengambilan keputusan, dengan bobot atribut yang sama penting dan setiap atribut saling bebas satu sama lain. Dasar dari teorema *Naïve Bayes Classifier* yang dipakai dalam pemrograman adalah rumus Bayes (*Naïve Bayes Classifier*, 2018) sebagai berikut:

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

Persamaan 1. Rumus Probabilitas *Naïve Bayes Classifier*

Dimana :

P(A | B) = Probabilitas A yang terjadi jika B

P(B | A) = Probabilitas B yang terjadi jika A

P(A) = Probabilitas A

P(B) = Probabilitas B

Sebagai contoh adalah jika terdapat P(Renang) dan P(Olahraga), maka pada P(Renang | Olahraga) adalah seberapa sering ada kata Renang pada saat terdapat kata Olahraga.

Naive Bayes memiliki beberapa kelebihan, yaitu :

- Lebih cepat dalam penghitungan.
- Menangani kuantitatif dan data diskrit.
- Relatif mudah untuk dipahami.
- Memerlukan pengkodean yang relatif sederhana [6].

2.7. Laplace Smoothing

Laplace Smoothing disebutkan dalam statistika, adalah teknik yang digunakan untuk menyelesaikan masalah probabilitas yang bernilai 0. Sebelumnya pada penghitungan probabilitas menggunakan *Naïve Bayes Classifier* bisa gagal disebabkan jika terdapat probabilitas kata yang bernilai 0. Sebagai contoh jika terdapat kalimat "pertandingan berjalan sengit" dan ingin dikategorikan, maka setiap kata pada kalimat tersebut dihitung probabilitasnya dan jumlahnya dikalikan. Lalu akan dilihat probabilitas tertinggi pada kategori apa. Jika pada kategori "News" kata "sengit" tidak muncul, maka probabilitas yang dikeluarkan akan menjadi 0 karena setiap kata pada kalimat tersebut akan dikalikan semua probabilitasnya, sehingga menimbulkan masalah. Oleh karena itu pada penelitian ini akan digunakan *Laplace Smoothing* untuk menyelesaikan masalah ini [5].

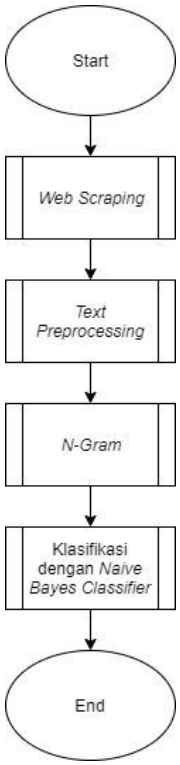
2.8. Dataset

Dataset yang dipakai dalam penelitian ini berasal dari hasil *web scraping* ke situs berita Tribunnews.com dan vivanews.com. Jumlah data yang digunakan berjumlah 3809 yang dipisah menjadi 5 kategori, yaitu News (1111), Superskor (780), Seleb (736), Sport (656), dan Lifestyle (526).

3. ANALISIS DAN DESAIN SISTEM

3.1. Perencanaan Sistem

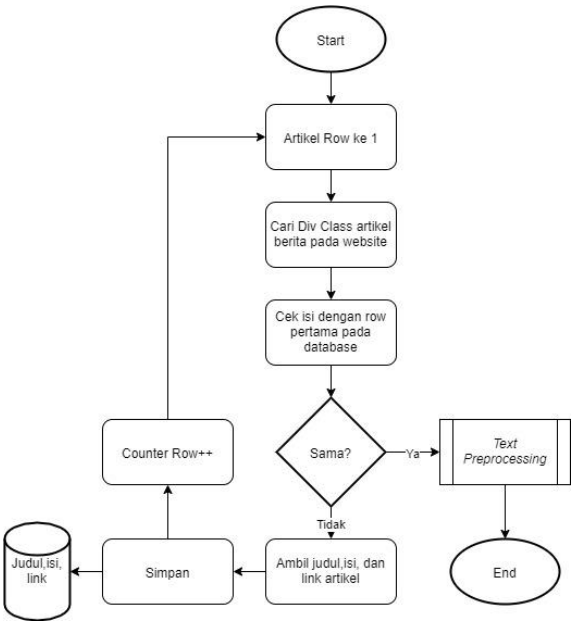
Pada penelitian ini memiliki langkah – langkah secara garis besar seperti pada Gambar 1.



Gambar 1. *Flowchart* Langkah Kerja Secara Keseluruhan

3.2. Proses Web Scraping

Pada proses *Web Scraping* akan memiliki langkah seperti pada Gambar 2.

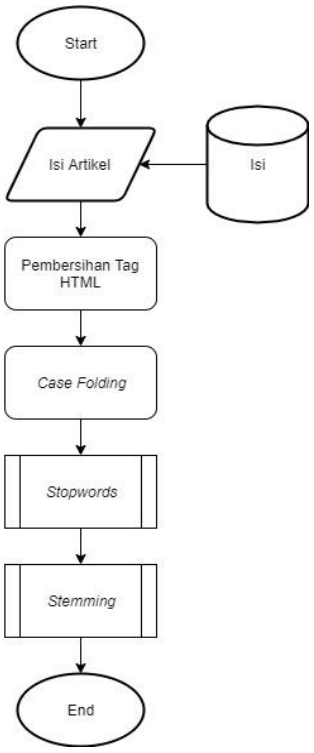


Gambar 2. Flowchart Proses Web Scraping

Pada proses *Web Scraping* ini akan dilakukan pengecekan terlebih dahulu apakah *div class* artikel pada website yang dituju apakah sudah ada pada database, jika sudah ada maka proses akan dihentikan dan dilanjutkan ke langkah *Text Preprocessing*, tetapi jika belum ada maka akan diambil judul dan isi artikel pada *div class* tersebut kemudian disimpan ke database. Setelah itu akan dilakukan pengecekan pada *row* berikutnya apakah *div class* artikel pada website sudah ada atau belum dan proses tersebut berulang hingga ditemukan *div class* yang sama.

3.3. Proses Text Preprocessing

Pada proses *Text Preprocessing* akan memiliki langkah seperti pada Gambar 3.



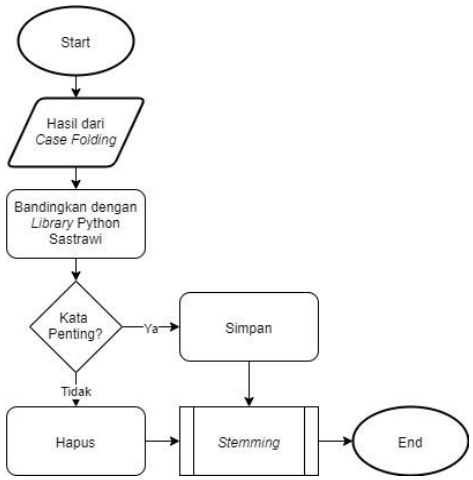
Gambar 3. Flowchart Proses Text Preprocessing

Pada proses *Text Preprocessing* akan dilakukan pembersihan *Tag HTML* terlebih dahulu, kemudian dilakukan proses pengubahan semua huruf yang ada menjadi huruf kecil semua pada proses *Case Folding*.

3.4. Proses Stopwords

Pada proses *Stopwords* akan memiliki langkah seperti pada Gambar 4.

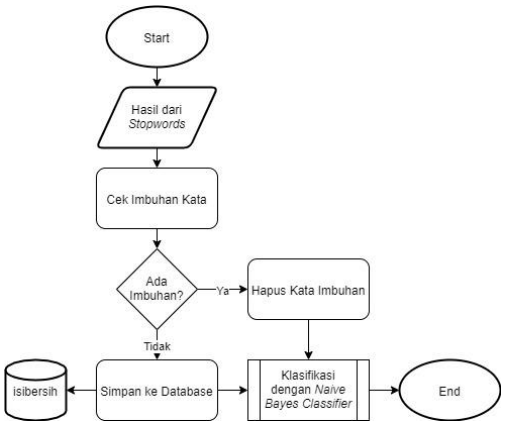
Pada proses *Stopwords* akan membandingkan kata yang didapat dari hasil *Case Folding* dengan Library Python Sastrawi apakah kata tersebut merupakan kata penting / kata bermakna atau tidak. Jika kata penting maka akan disimpan sedangkan jika kata tidak bermakna maka akan dihapus. Daftar kata tidak penting yang terdapat pada *Library* Python Sastrawi adalah 'yang', 'untuk', 'pada', 'ke', 'para', 'namun', 'menurut', 'antara', 'dia', 'dua', 'seperti', 'jika', 'jika', 'sehingga', 'kembali', 'dan', 'tidak', 'ini', 'karena', 'kepada', 'oleh', 'saat', 'harus', 'sementara', 'setelah', 'belum', 'kami', 'sekitar', 'bagi', 'serta', 'di', 'dari', 'telah', 'sebagai', 'masih', 'hal', 'ketika', 'adalah', 'itu', 'dalam', 'bisa', 'bahwa', 'atau', 'hanya', 'kita', 'dengan', 'akan', 'juga', 'ada', 'mereka', 'sudah', 'saya', 'terhadap', 'secara', 'agar', 'lain', 'anda', 'begitu', 'mengapa', 'kenapa', 'yaitu', 'yakni', 'daripada', 'itulah', 'lagi', 'maka', 'tentang', 'demi', 'dimana', 'kemana', 'pula', 'sambil', 'sebelum', 'sesudah', 'supaya', 'guna', 'kah', 'pun', 'sampai', 'sedangkan', 'selagi', 'sementara', 'tetapi', 'apakah', 'kecuali', 'sebab', 'selain', 'seolah', 'seraya', 'seterusnya', 'tanpa', 'agak', 'boleh', 'dapat', 'dsb', 'dst', 'dll', 'dahulu', 'dulunya', 'anu', 'demikian', 'tapi', 'ingin', 'juga', 'nggak', 'mari', 'nanti', 'melainkan', 'oh', 'ok', 'seharusnya', 'sebetulnya', 'setiap', 'setidaknya', 'sesuatu', 'pasti', 'saja', 'toh', 'ya', 'walau', 'tolong', 'tentu', 'amat', 'apalagi', 'bagaimanapun'.



Gambar 4. Flowchart Proses Stopwords

3.5. Proses Stemming

Pada proses *Stemming* akan memiliki langkah seperti pada Gambar 5.



Gambar 5. Flowchart Proses Stemming

Pada proses *Stemming* akan membandingkan kata dengan *Library* Python Sastrawi, jika bukan kata dasar dan memiliki imbuhan, maka akan diganti menjadi kata dasarnya.

3.6. Proses N-Gram

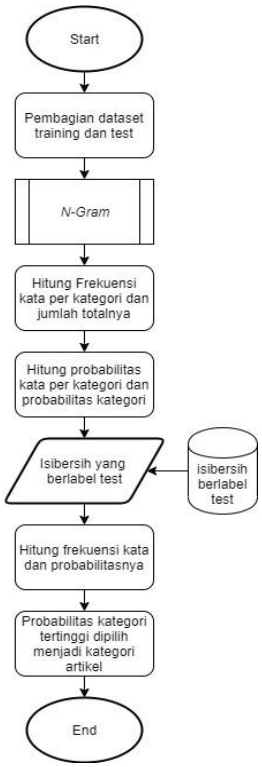
Pada proses implementasi metode *N-Gram* akan memiliki langkah seperti pada Gambar 6.

Gambar 6. *Flowchart* Langkah Implementasi Metode *N-Gram*

Pada langkah ini akan dilakukan pengelompokan kata berdasarkan Unigram dan Bigram.

3.7. Proses Klasifikasi Dengan Naïve Bayes Classifier

Pada proses implementasi metode Naïve Bayes Classifier untuk mengklasifikasikan hasil output dari metode N-Gram akan memiliki langkah seperti pada Gambar 7.



Gambar 7. *Flowchart* langkah klasifikasi berita menggunakan metode *Naïve Bayes Classifier*

Pertama yang dilakukan adalah membagi dataset menjadi training dan test. Kemudian akan dipilah datanya berdasarkan Unigram dan Bigram berdasarkan proses N-Gram. Setelah itu pada proses training akan dihitung frekuensi kata per kategori beserta jumlah totalnya. Setelah didapat semua frekuensi kata dan jumlah totalnya, selanjutnya akan dihitung probabilitas kata dan probabilitas kategori. Selanjutnya pada proses testing akan diambil data artikel yang sudah bersih dan kemudian akan dihitung frekuensi tiap kata pada artikel tersebut dan juga dihitung probabilitasnya dengan mengkalikannya dengan probabilitas kata pada proses training yang sebelumnya dilakukan. Setelah didapatkan probabilitas tiap kategorinya, akan dipilih probabilitas tertinggi menjadi kategori artikel tersebut.

4. PENGUJIAN SISTEM

Pada bab ini akan dijelaskan tentang pengujian terhadap sistem dan aplikasi yang telah dibuat. Pengujian dilakukan dengan menggunakan dataset yang berbeda rasionya. Rasio yang dipakai untuk dataset *Training* dan *Test* adalah 50 : 50, 60 : 40, 70 : 30, 80 : 20. Jumlah dataset yang digunakan adalah 3809, dibagi menjadi 5 kategori yaitu News (1111), Superskor (780), Seleb (736), Sport (656), dan Lifestyle (526).

Pengujian sistem dilakukan dengan melakukan pengujian akurasi yang dihasilkan menggunakan metode Naïve Bayes Classifier dan perbandingan penggunaan dataset Training dan Test yang berbeda. Cara penghitungan akurasi disini adalah dengan membagi hasil total prediksi yang benar dengan jumlah dataset Test yang dipakai dalam klasifikasi, hasil bagi tersebut adalah akurasi ketepatan prediksi untuk mengkategorikan berita dalam database. Pada Unigram melakukan pengelompokan tiap 1 kata, sedangkan pada Bigram melakukan pengelompokan tiap 2 kata. Sebagai contoh kumpulan kata “olahraga hidup sehat”, pada Unigram menjadi “olahraga”, “hidup”, “sehat”. Sedangkan pada Bigram “olahraga hidup”, “hidup sehat”, dan “olahraga sehat”.

Tabel 1. Akurasi Ketepatan Prediksi

	Naïve Bayes Classifier	
Rasio	**Unigram**	**Bigram**
80 : 20	0.887	0.507
70 : 30	0.947	0.519
60 : 40	0.904	0.498
50 : 50	0.901	0.508

Pada tabel 1 berisi rasio dataset training dan test, beserta dengan akurasi yang didapat dengan menggunakan unigram dan bigram. Angka pada kolom Unigram dan Bigram dikalikan dengan 100 untuk mendapatkan berapa besar persentase akurasi yang didapat. Sebagai contoh pada rasio dataset training dan test 80 : 20, pada Unigram memiliki akurasi sebesar 88.7%, didapat dari mengkalikan 0.887 dengan 100, setelah itu didapat hasilnya yaitu 88.7%.

5. KESIMPULAN DAN SARAN

Dari hasil perancangan dan pembuatan sistem dan aplikasi, dapat diambil kesimpulan antara lain:

- Metode Naïve Bayes Classifier memiliki akurasi yang terbilang tinggi yaitu antara 88.7% hingga 94.7% dengan variasi rasio dataset Training dan Test sebesar 80 : 20, 70 : 30, 60 : 40, 50 : 50.
- Fitur N-Gram Unigram memiliki tingkat akurasi yang tinggi dibandingkan dengan dengan Bigram.

- Rasio dataset Training dan Test yang memiliki tingkat akurasi tertinggi pada penelitian ini adalah 70 : 30, dengan akurasi Unigram sebesar 0.947 dan akurasi Bigram sebesar 0.519. Oleh karena itu bisa diambil kesimpulan bahwa rasio terbaik pada penelitian ini adalah 70 : 30 dan persentase dataset training yang semakin besar tidak selalu menghasilkan akurasi yang lebih tinggi.

Saran yang dapat diberikan untuk menyempurnakan dan mengembangkan Implementasi ini adalah:

- Menambah dataset untuk mengetes penambahan akurasi.
- Mencari fitur alternatif selain N-Gram yang mungkin bisa menambah akurasi.

6. DAFTAR REFERENSI

[1] A. S., Santoso, B. P., D. R., Wiraswari, N. M. A. K., & Sari, T. R. Klasifikasi dokumen bahasa Jawa menggunakan metode N-Gram. https://docplayer.info/37613251-Klasifikasi-dokumen-bahasa-jawa-menggunkan-metode-n-gram.html

[2] Destuardi & Sumpeno, S. 2009. Klasifikasi emosi untuk teks bahasa Indonesia menggunakan metode Naive Bayes. http://digilib.its.ac.id/ITS-Article-91105120000039/19046

Draxl, V. (2018). *Web Scraping Data Extraction from Websites*. https://www.academia.edu/35901535/BACHELOR_PAPER_Web_Scraping_Data_Extraction_from_websites

[3] Holm, J. & Gustavsson, M. 2018. *XML Parser – A Comparative Study with Respect to Adaptability*. http://www.diva-portal.org/smash/get/diva2:1220705/FULLTEXT01.pdf

[4] Huang, O. 2017. *Applying Multinomial Naïve Bayes to NLP Problems: A Practical Explanation*. https://medium.com/syncedreview/applying-multinomial-naive-bayes-to-nlp-problems-a-practical-explanation-4f5271768ebf

[5] Naive Bayes Classifier. 2018. http://www.statsoft.com/textbook/naive-bayes-classifier

[6] Shaoul, C., Westbury, C. F., Baayen, R. H. 2013. *The Subjective Frecuency of Word N-Grams*. https://www.academia.edu/33832265/The_subjective_frequency_of_word_n-grams

[7] Wijaya, A. P., & Santoso, H. A. 2016. Naïve Bayes Classification pada klasifikasi dokumen untuk identifikasi konten E-Government. In *Journal of Applied Intelligent System*. 1(1), 48-55. https://publikasi.dinus.ac.id/index.php/jais/article/view/1032/772

[8] Yulio, A. P. 2019. Text Preprocessing dengan Python NLTK. https://devtrik.com/python/text-preprocessing-dengan-python-nltk/