

## **Penerapan *Naive Bayes Classifier* dengan Algoritma Nazief dan Adriani Untuk Deteksi Hoaks**

**Vidya Oktaviana Yamin<sup>1</sup>, Andi Tenriawaru<sup>2\*</sup>, La Ode Saidi<sup>3</sup>, Gusti Arviana Rahman<sup>4</sup>,**

<sup>1,2,3,4</sup>Studi Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Halu Oleo

Jl. H.E.A Mokodompit, Kambu, Kendari, Sulawesi Tenggara

e-mail: [vidya.okta21@gmail.com](mailto:vidya.okta21@gmail.com)<sup>1</sup>, [andi.tenriawaru@uho.ac.id](mailto:andi.tenriawaru@uho.ac.id)<sup>2\*</sup>, [laode.saidi@uho.ac.id](mailto:laode.saidi@uho.ac.id)<sup>3</sup>,  
[arviana.rahman@uho.co.id](mailto:arviana.rahman@uho.co.id)<sup>4</sup>

### **Abstrak**

Hoaks adalah informasi palsu yang disebarkan secara sengaja sebagai kebenaran. Hoaks dapat menimbulkan dampak negatif bagi kehidupan masyarakat, seperti kegelisahan, ketidaknyamanan, amarah, perpecahan, kebencian, dan hasutan. Media sosial adalah salah satu platform utama yang digunakan untuk menyebarkan hoaks. Menurut survei MASTEL tahun 2019, media sosial memiliki persentase penyebaran hoaks tertinggi yaitu 87,50%, diikuti oleh aplikasi chatting, website, media cetak, dan lainnya. Oleh karena itu, diperlukan suatu sistem yang dapat mendeteksi hoaks secara otomatis dan akurat. Penelitian ini bertujuan untuk meningkatkan akurasi pendeteksian hoaks menggunakan *Naive Bayes Classifier* dengan penyempurnaan *stemming*. *Stemming* adalah teknik yang menghilangkan imbuhan kata untuk mengurangi variasi kata yang memiliki makna sama, meningkatkan efisiensi dan efektivitas sistem klasifikasi teks. *Data set* penelitian berisi artikel berita yang diklasifikasikan sebagai hoaks atau bukan oleh website pengecekan fakta, sebagai sumber terpercaya untuk menentukan validitas berita. Hasil penelitian menunjukkan sistem mencapai akurasi 84%, presisi 88%, *recall* 83%, dan *F1-Score* 85%. Hasil yang tinggi menandakan sistem dapat dengan baik membedakan hoaks dari berita valid. Dengan tingkat akurasi yang tinggi, sistem ini membantu masyarakat memverifikasi informasi secara online dan mencegah penyebaran hoaks yang merugikan.

**Kata kunci**— Deteksi Hoaks, *Naive Bayes*, Nazief Adriani, *Stemming*

### **Abstract**

*Hoax is false information intentionally spread as truth. Hoaxes can have negative impacts on society, such as causing anxiety, discomfort, anger, division, hatred, and incitement. Social media is one of the main platforms used to disseminate hoaxes. According to the MASTEL survey in 2019, social media had the highest percentage of hoax dissemination, reaching 87.50%, followed by messaging applications, websites, print media, and others. Therefore, a system that can automatically and accurately detect hoaxes is needed. This research aims to improve the accuracy of hoax detection using the Naive Bayes Classifier with the refinement of stemming. Stemming is a technique that removes word affixes to reduce variations of words with the same meaning, thereby enhancing the efficiency and effectiveness of text classification systems. The research dataset consists of news articles classified as hoaxes or non-hoaxes by fact-checking websites, considered a reliable source for determining news validity. The results show that the system achieves an accuracy of 84%, precision of 88%, recall of 83%, and F1-Score of 85%. These high scores indicate that the system can effectively differentiate*

*hoaxes from valid news. With its high accuracy, the system helps the public verify information online and prevent the dissemination of harmful hoaxes.*

**Keywords**—*Hoax Detection, Naïve Bayes, Nazief Adriani, stemming*

## 1. PENDAHULUAN

Setiap hari ada ratusan bahkan ribuan informasi tersebar melalui media sosial. Namun, jagad media informasi di Indonesia masih marak akan teresebarnya hoaks. Hoaks merupakan informasi tidak benar yang dibuat secara sengaja untuk disebarluaskan sebagai kebenaran[1].

Masyarakat Telematika (MASTEL) melakukan survey wabah hoaks nasional pada tahun 2019, menyebutkan bahwa penyebaran saluran penyebaran berita hoaks tertinggi berasal dari sosial media dengan persentase 87,50% kemudian diikuti oleh Aplikasi Chatting (67,00%), Website (28,20%), Media Cetak (6,40%), dan lainnya. Hasil penelitian ini juga menyebutkan bahwa andalan utama masyarakat untuk memeriksa kebenaran suatu berita masih bertumpu pada search engine (82,80%), sebanyak 27,40% responden menyatakan tidak mengetahui cara memeriksa kebenaran suatu berita, dan sebanyak 63,30% responden berpendapat mengira berita hoaks bukan hoaks karena mendapatkan berita tersebut dari orang yang dapat dipercaya. Kesadaran penanggulangan berita hoaks dengan persentase tertinggi berasal dari diri sendiri (81,90%) namun secara implisit responden tetap menginginkan penegakan hukum yang tegas kepada pelaku penyebaran hoaks[2].

Penyebaran berita tidak benar dapat memicu kegelisahan, ketidaknyamanan, bahkan amarah terlebih jika berita mengandung unsur provokatif dan agitasi negatif. Berita dengan unsur tersebut bisa saja menyebabkan perpecahan, penyulutian kebencian, dan hasutan sana-sini. Maka dari itu, diperlukan suatu langkah untuk mengatasi beredarnya berita hoaks di masyarakat.

Penelitian terkait yang dilakukan untuk mengatasi masalah hoaks pernah dilakukan oleh Candra Surya Sriyano dan Erwin Budi Setiawan Penelitian ini menggunakan metode

*Naïve Bayes* Multinomial untuk klasifikasi berita hoaksnya. Hasil akhir pada pendeteksian berita hoaks menggunakan *Naïve Bayes* Multinomial dan pembobotan TF- IDF ini memperoleh akurasi tertinggi sebesar 72.06%. Namun, nilai akurasi yang didapatkan dari hasil analisis performansi sistem kurang maksimal karena pengaruh dari proses *preprocessing* yang dilakukan terdapat kata yang sama dan terdapat data yang hilang saat melakukan *preprocessing* sehingga sistem tidak dapat menguji secara maksimal yang mengakibatkan penurunannya nilai akurasi[3].

*Naïve Bayes Classifier* adalah algoritma klasifikasi multikelas berbasis supervisi yang mengaplikasikan teorema Bayes dengan asumsi independensi yang "naif" antara setiap pasangan variabel. Keunggulan dari algoritma *Naïve Bayes* terletak pada kemampuannya dalam mengembangkan model dan melakukan prediksi dengan cepat, berkat kesederhanaan algoritmanya[4]. *Stemming* adalah proses mengembalikan kata kerja berimbuhan menjadi kata dasarnya. Di antara beberapa jenis algoritma *stemming*, Algoritma *Stemming* Nazief dan Adriani memiliki keakurasian yang tinggi[5].

Berdasarkan latar belakang di atas maka penulis mengangkat judul “Penerapan *Naïve Bayes Classifier* dengan Algoritma Nazief dan Adriani untuk Deteksi Hoaks” yang diharapkan dapat meningkatkan akurasi pendeteksi hoaks.

*Naïve Bayes Classifier* (NBC) adalah algoritma klasifikasi probabilitas yang bersifat sederhana dan menghitung serangkaian probabilitas dengan mengakumulasikan frekuensi dan nilai kombinasi dari suatu set data yang spesifik. Algoritma menerapkan *Teorema Bayes* dan mengandaikan bahwa semua atribut independen atau independen satu sama lain untuk menetapkan nilai ke variabel kelas [6].

Persamaan dari Teorema *Bayes* dapat dilihat pada Persamaan (1)

$$P(y|x_1, \dots, x_n) = \frac{P(y) \prod_{i=1}^n P(x_i|y)}{P(x_1, \dots, x_n)} \quad (1)$$

Keterangan:

$P(y|x_1, \dots, x_n)$ : *posterior* dari kelas  $y$ .

$P(y)$ : *prior* untuk kelas  $y$ .

$P(x_i|y)$ : *likelihood*

$P(x_1, \dots, x_n)$ : *evidence*

Algoritma Naive Bayes Bernoulli menerapkan proses klasifikasi pada data yang mengikuti distribusi Bernoulli multivariat; artinya, fitur-fitur yang ada mungkin memiliki beberapa variasi, tetapi setiap fitur dianggap sebagai variabel biner (Bernoulli, boolean). Oleh karena itu, kelas ini memerlukan sampel yang diwakili dalam bentuk vektor fitur dengan nilai-nilai biner. [7]. Aturan pengambilan keputusan dalam algoritma Naive Bayes Bernoulli ditunjukkan pada Persamaan (2)

$$P(x_i|y) = P(i|y) * x_i + (1 - P(i|y)) * (1 - x_i) \quad (2)$$

Dimana:

$P(x_i|y)$ : Probabilitas kondisional atribut  $x_i$  diberikan kelas  $y$

$P(i|y)$ : Probabilitas atribut  $x_i$  untuk kelas  $y$ .

$x_i$ : nilai atribut  $x_i$  (0 atau 1).

*Text preprocessing* merupakan langkah awal dari *data mining*[8]. Pada tahap ini, persiapan dokumen dan data dilakukan untuk memastikan kesiapan dalam proses selanjutnya, termasuk klasifikasi. Tahap-tahap dalam pra-pemrosesan teks meliputi:

- Tokenizing*: langkah menggabungkan atau memisahkan kalimat menjadi kata-kata dengan menggunakan spasi.
- Case folding*: tahapan di mana semua karakter dalam teks diolah dan diubah menjadi huruf kecil.
- Filtering*: tahap menghapus tanda baca.
- Stemming*: proses untuk memperoleh kata dasar dari suatu kata dengan menghapus akhiran dan atau akhiran dari kata tersebut.
- TF - IDF *weighting*: TF (*Term Frequency*) adalah frekuensi Munculnya suatu istilah dalam suatu dokumen. Semakin sering kata tersebut muncul (TF tinggi), semakin

signifikan bobotnya. IDF (*Inverse Dokumen Frequency*) mengukur sejauh mana kata tersebut tersebar di seluruh koleksi dokumen. IDF mencerminkan pentingnya kata tersebut dalam konteks semua dokumen.

*Stemming* merupakan proses yang digunakan untuk mencari kata dasar dari proses penyaringan, proses penyaringan adalah proses pengambilan kata yang dianggap relevan atau bermakna. Beberapa metode telah dikembangkan untuk melakukan stemming dalam Bahasa Indonesia, termasuk Metode Nazief dan Adriani, Algoritma Porter, dan Metode Arifin dan Setiono. Metode stemming berbasis aturan dalam Bahasa Indonesia cenderung memiliki tingkat kesalahan yang cukup signifikan[5].

*Confusion Matrix* adalah tabel matriks  $N \times N$  (N jumlah kelas/label/kategori) berisi nilai *True* dan *False* dari model klasifikasi yang telah dibangun. Fungsinya adalah untuk membandingkan nilai aktual dengan nilai yang diprediksi oleh model.[9]. Dalam matriks ini, baris menggambarkan nilai kelas sebenarnya, sementara kolom menggambarkan nilai prediksi. *Confusion Matrix* terdiri dari empat kategori nilai yang berbeda yakni:

- True Negatif (TN)*: Nilai aktual dan prediksi keduanya negatif.
- False Positif (FP)*: Prediksi positif padahal nilai aktual negatif.
- False Negatif (FN)*: Prediksi negatif meskipun nilai aktual positif.
- True Positif (TP)*: Keduanya, nilai prediksi dan aktual, positif.

Hasil evaluasi yang dihasilkan sulit untuk diukur seberapa akurat model klasifikasi yang telah dibuat. Oleh karena itu data dari *Confusion Matrix* akan digunakan untuk menghitung nilai-nilai yang disimpan pada *Classification Report* (nilai kinerja dari model klasifikasi yang dibuat). Berikut nilai-nilai *classification report* yang digunakan:

- Accuracy* : Kecocokan antara hasil prediksi pengujian dengan nilai yang sebenarnya

dapat dilihat pada persamaan 3.

$$accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad (3)$$

2. *Precision* : Kesesuaian antara permintaan data dari pengguna dan hasil yang disediakan oleh sistem dapat dilihat pada persamaan 4.

$$precision = \frac{TP}{TP+FP} \quad (4)$$

3. *Recall* : Jumlah kesesuaian data dari hasil eksperimen berdasarkan perspektif kelas yang digunakan dapat dilihat pada persamaan 5.

$$recall = \frac{TP}{TP+FN} \quad (5)$$

4. *F1-Score* : Gambaran rata-rata atau *harmonic mean* pada presisi dan *recall* dapat dilihat pada persamaan 6.

$$f1 - score = 2 \times \frac{recall \times precision}{recall + precision} \quad (6)$$

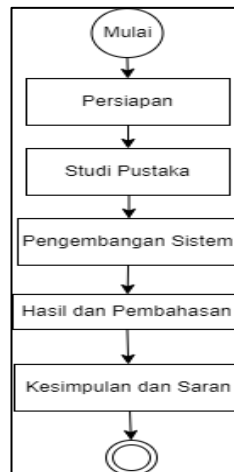
## 2. METODE PENELITIAN

### 2.1 Waktu dan Tempat Penelitian

Penelitian ini dilakukan pada bulan Juni-Agustus 2023 di Laboratorium Penelitian Mahasiswa Fakultas Matematika dan Ilmu Pengetahuan Universitas Halu Oleo Kendari. Dalam penelitian ini, objek penelitian yang digunakan adalah data berupa berita yang didapatkan dari beberapa portal penyedia berita di Indonesia. *Data Training* untuk berita hoaks diperoleh dari website [turnbackhoax.com](http://turnbackhoax.com) sementara itu data training untuk berita benar diperoleh dari website CNN Indonesia.

### 2.2 Prosedur Penelitian

Berikut ini adalah prosedur penelitian yang dilakukan sebagai acuan dalam penelitian yang dilakukan, dapat dilihat pada Gambar 1.



Gambar 1 Prosedur Penelitian

### 2.3 Metode Pengumpulan Data

Metode pengumpulan data yang digunakan untuk memperoleh data keperluan penelitian adalah Metode Pustaka. Dengan metode ini, Penulis dapat mengumpulkan informasi dari berbagai sumber seperti literatur, jurnal, dan referensi yang relevan dengan penelitian.

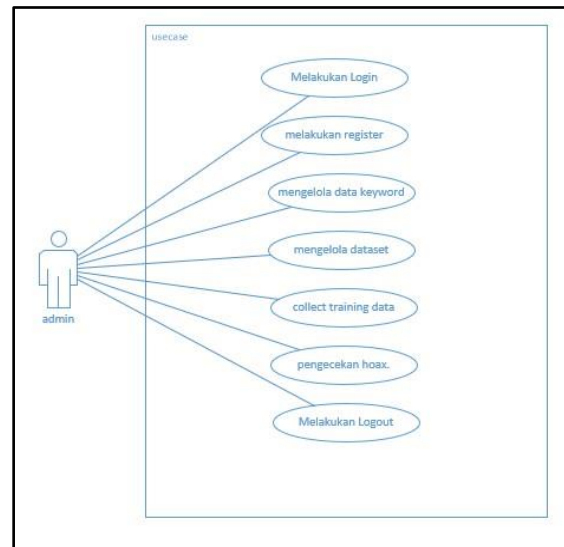
### 2.4 Metode Pengembangan Sistem

Dalam penelitian ini, peneliti menggunakan metode *Rational Unified Process* (RUP) untuk metode pengembangan sistem. Dalam metode ini, melibatkan empat tahapan pengembangan perangkat lunak yang dapat dilihat pada Tabel 1.

Tabel 1 Fase RUP

Fase RUP	Proses yang dilakukan
<i>Inception</i>	Pada langkah ini, dilakukan pengidentifikasian batasan aktivitas, analisis kebutuhan dari pengguna, dan perancangan awal perangkat lunak yang mencakup perancangan arsitektur dan <i>use case</i> .
<i>Elaboration</i>	Pada fase ini, dilakukan proses perancangan perangkat lunak yang melibatkan spesifikasi fitur-fitur perangkat lunak hingga pembuatan prototipe yang akan

Fase RUP	Proses yang dilakukan
	digunakan dalam pengembangan aplikasi.
Construction	Pelaksanaan rancangan perangkat lunak yang telah dirancang dilakukan pada fase ini. Pada akhir fase ini, perangkat lunak versi final yang telah mendapatkan persetujuan dari administrator diimplementasikan dan dirilis bersama dengan dokumentasi perangkat lunak.
Transition	Pada fase ini, dilakukan pemasangan, penggunaan, dan penyebaran perangkat lunak.



Gambar 2 Use Case Diagram

2.5 Perancangan Sistem

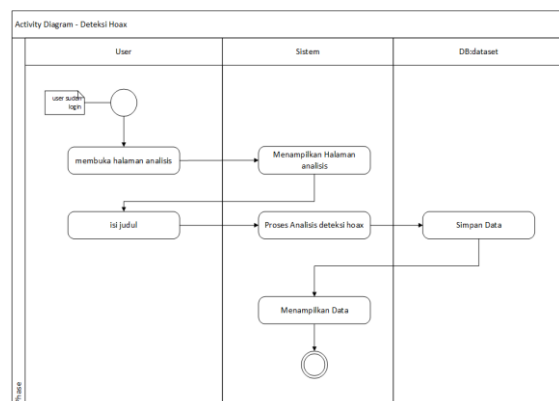
Pada penelitian ini, digunakan pendekatan desain UML (*Unified Modeling Language*), yang merupakan kerangka kerja pengembangan yang berfokus pada objek. Diagram-diagram UML yang diterapkan dalam proses perancangan mencakup *Use Case Diagram*, *Class Diagram*, *Sequence Diagram*, dan *Activity Diagram*.

1) Use Case Diagram

Diagram Use Case digunakan untuk menggambarkan proses atau langkah-langkah yang dilakukan oleh sistem kepada pihak eksternal yang berada di luar sistem. Berikut dijabarkan dalam Gambar 2.

2) Activity Diagram

*Activity Diagram* menyajikan gambaran mengenai aktivitas yang terjadi dalam sistem. Mulai dari awal hingga akhir, diagram ini menggambarkan langkah-langkah dalam proses kerja sistem yang telah dibuat. Berikut ini adalah *activity diagram* yang digunakan oleh peneliti. Berikut dijabarkan dalam Gambar 3.

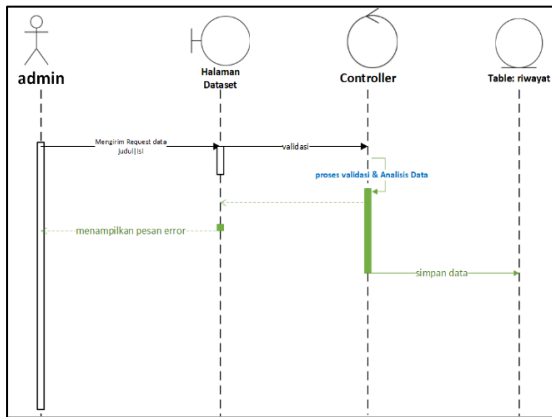


Gambar 3 Activity Diagram Deteksi Hoaks

*Activity Diagram* ini menjabarkan alur kerja pengembang yang bertujuan untuk melakukan deteksi hoaks. Setelah user berhasil login, user akan dapat mengakses laman analisis. Kemudian user dapat menginput judul hoaks yang hendak dicek. Sistem akan melakukan proses analisis terhadap data untuk dideteksi sebagai hoaks atau bukan dan menyimpan data tersebut di *database*. Data yang telah melalui proses tersebut kemudian akan ditampilkan.

3) *Sequence Diagram*

*Sequence Diagram* digunakan untuk memodelkan interaksi antara objek-objek dalam sistem yang dikembangkan. Diagram ini membantu dalam memahami dan memvisualisasikan urutan langkah-langkah yang terjadi dalam interaksi objek, memperjelas aliran informasi, dan mengidentifikasi potensi masalah atau kebutuhan perubahan dalam sistem. Berikut dijabarkan dalam Gambar 4.



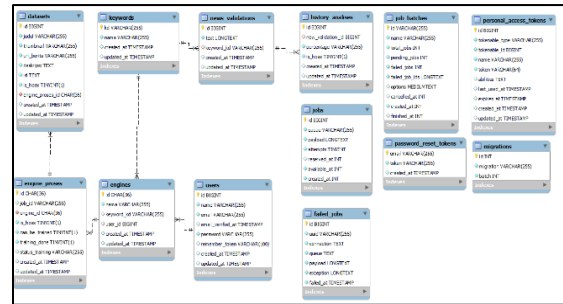
Gambar 4 *Sequence Diagram* Deteksi Hoaks

*Sequence Diagram* Deteksi Hoaks menjabarkan interaksi antara *admin*, halaman dataset, *Controller*, dan tabel riwayat. Proses dimulai ketika *actor* melakukan *request* data pada judul/jisi. Data akan diteruskan ke *controller*. *Controller* akan memvalidasi dan melakukan proses analisis data. Jika data valid maka data yang telah diproses akan disimpan ke tabel riwayat namun apabila data tidak valid maka akan menampilkan pesan *error*. Validasi yang dilakukan pada *diagram* ini adalah untuk memeriksa apakah data yang *diinput* oleh user yakni teks berita dapat diproses atau telah diisi dengan benar sesuai dengan ketentuan yang ada.

4) *Class Diagram*

Class diagram adalah diagram dalam pemodelan perangkat lunak yang digunakan untuk menggambarkan struktur dan hubungan antara kelas-kelas dalam suatu sistem. *Class diagram* berfokus pada pemetaan objek-objek yang ada dalam sistem ke dalam kelas-kelas yang merepresentasikan entitas-entitas dalam

sistem tersebut. Berikut dijabarkan dalam Gambar 5.



Gambar 5 *Class Diagram*

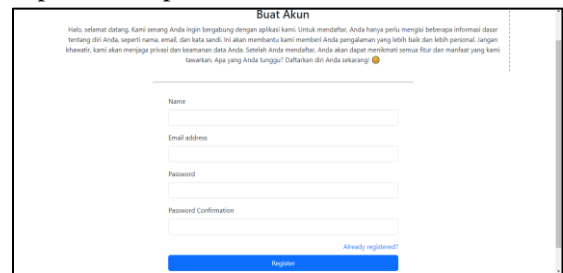
3. HASIL DAN PEMBAHASAN

3.1 Implementasi *Interface* Sistem Deteksi Hoaks

*Interface* ini bertujuan untuk menyediakan antarmuka yang mudah digunakan bagi pengguna untuk melakukan analisis dan verifikasi keaslian berita. Implementasi *interface* ini mencakup pengembangan antarmuka pengguna (*user interface*) serta integrasi dengan komponen inti sistem deteksi hoaks.

1) Halaman *Register*

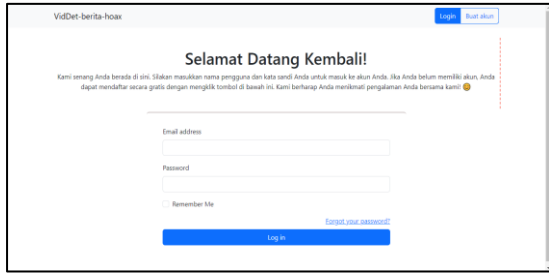
Halaman *Register* memungkinkan pengguna untuk membuat akun baru dan mendaftarkan diri sebagai anggota di *website*, dapat dilihat pada Gambar 6.



Gambar 6 Halaman *Register*

2) Halaman *Login*

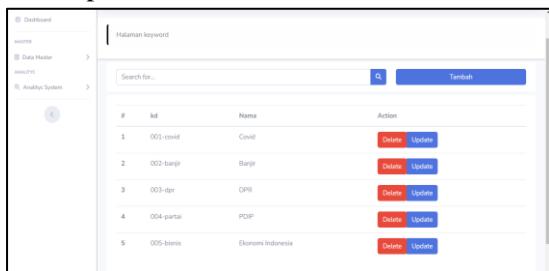
Pada halaman ini, pengguna dapat melakukan proses *login* untuk masuk ke halaman utama *web*, dapat dilihat pada Gambar 7.



Gambar 7 Halaman Login

3) Halaman Keyword

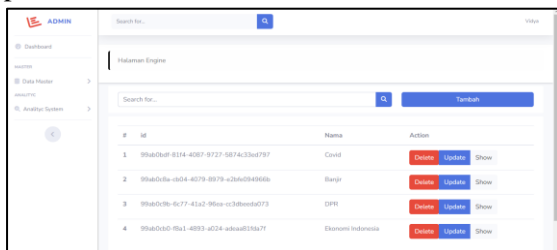
Pada laman ini, terdapat fitur khusus yang memungkinkan admin untuk menginputkan keyword yang sesuai dengan topik berita, dapat dilihat pada Gambar 8.



Gambar 8 Halaman Keyword

4) Halaman Engine

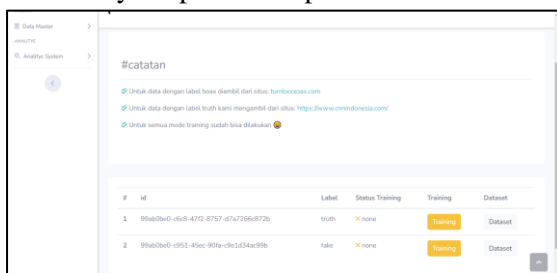
Pada laman ini, pengguna memiliki kemampuan untuk menginputkan engine sesuai dengan keyword yang diinginkan, dapat dilihat pada Gambar 9.



Gambar 9 Halaman Engine

5) Halaman Crawling Data

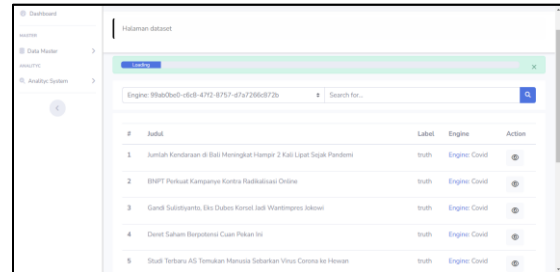
Pada laman ini, pengguna dapat melakukan crawling data berita hoaks dan non-hoaks yang sesuai dengan engine yang telah dipilih sebelumnya dapat dilihat pada Gambar 10.



Gambar 10 Halaman Crawling Data

6) Halaman Dataset

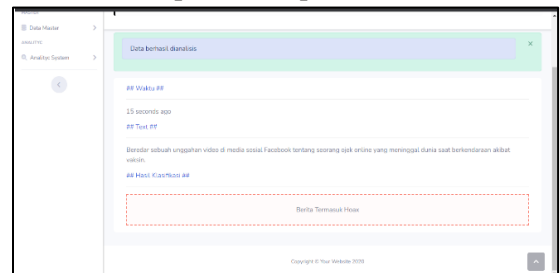
Pada laman ini, pengguna dapat melihat hasil dari proses crawling data berita hoaks dan non-hoaks yang sesuai dengan engine yang telah dipilih sebelumnya, dapat dilihat pada Gambar 11.



Gambar 11 Halaman Dataset

7) Halaman Deteksi Hoaks

Pada laman ini, pengguna dapat melakukan deteksi terhadap suatu berita untuk mengetahui apakah berita tersebut tergolong hoaks atau non-hoaks, dapat dilihat pada Gambar 12.



Gambar 12 Halaman Deteksi Hoaks

3.2 Uji Coba Sistem Deteksi Hoaks

Pada pengujian sistem ini, dilakukan tiga pengujian yaitu pengujian performansi stemming, pengujian performansi klasifikasi dan Blackbox.

1) Pengujian Performansi Stemming

Pengujian ini menggunakan 2 skenario dimana data uji yang diambil sebanyak 20% dari total dataset yang berjumlah 160 data yang diacak. Skenario pertama dilakukan dengan menggunakan stemming dan skenario kedua dilakukan tanpa menggunakan stemming. Hasil dari skenario pertama dapat dilihat pada Tabel 2.

Tabel 2 Hasil Confusion Matrix Dengan Stemming

N = 32		Ground Truth	
Prediction	TP: 15	FN: 2	
	FP: 0	TN: 15	

Sistem berhasil mencapai tingkat akurasi sebesar 93%, presisi mencapai 100%, nilai *recall* sebesar 88%, dan *F1-score* mencapai 93%. Sementara itu, hasil dari skenario kedua dapat dilihat pada Tabel 3.

**Tabel 3** Hasil *Confusion Matrix* Tanpa *Stemming*

N = 32	Ground Truth	
Prediction	TP: 12	FN:5
	FP: 0	TN:15

Sistem mencapai tingkat akurasi sebesar 84%, presisi mencapai 100%, *recall* sebesar 70%, dan *F1-score* mencapai 82%.

Berdasarkan hasil analisis, sistem yang menggunakan *stemming* mencapai akurasi 93% dengan presisi 100%, namun memiliki *recall* sebesar 88% dan *F1-score* 93%, menunjukkan kemampuan baik dalam mengklasifikasikan data positif. Di sisi lain, sistem tanpa *stemming* mencapai akurasi 84% dengan presisi 100%, *recall* 70%, dan *F1-score* 82%. Secara keseluruhan, penggunaan *stemming* meningkatkan kemampuan sistem dalam mengidentifikasi data positif dan menjaga keseimbangan *presisi* dan *recall*.

## 2) Pengujian Performansi Klasifikasi

Pengujian ini menggunakan 2 skenario dimana data uji yang diambil sebanyak 20% dari total dataset yang berjumlah 160 data yang diacak. Skenario pertama dilakukan dengan menggunakan *Naïve Bayes Classifier* dan skenario kedua dilakukan dengan menggunakan *Support Vector Machine*. Hasil dari skenario pertama dapat dilihat pada Tabel 4.

**Tabel 4** Hasil *Confusion Matrix Naive Bayes Bernoulie Classifier*

N = 32	Ground Truth	
Prediction	TP: 15	FN:2
	FP: 3	TN:15

Sistem berhasil mencapai tingkat akurasi sebesar 93%, presisi mencapai 100%, nilai *recall* sebesar 88%, dan *F1-score* mencapai 93%. Sementara itu, hasil dari skenario kedua dapat dilihat pada Tabel 5.

**Tabel 5** Hasil *Confusion Matrix Support Vector Machine*

N = 32	Ground Truth	
Prediction	TP: 14	FN:1
	FP: 3	TN:14

Hasil pengujian menunjukkan tingkat akurasi 87%, presisi sebesar 82%, *recall* 93%, dan *F1-score* sebesar 87%.

Sistem yang menggunakan *Naive Bayes Bernoulie Classifier* memiliki tingkat akurasi yang tinggi (93%) dan presisi yang sempurna (100%), namun dengan *recall* yang masih perlu ditingkatkan (88%). Meskipun begitu, *F1-score* mencapai 93%, mencerminkan keseimbangan yang baik antara presisi dan *recall*. Di sisi lain, sistem yang menggunakan SVM mencapai akurasi 87%, presisi 82%, *recall* 93%, dan *F1-score* 87%. Keseluruhan, kedua sistem memiliki performa yang baik dalam mengklasifikasikan data, dengan kemampuan yang berbeda dalam menjaga keseimbangan antara presisi dan *recall*.

## 3) Pengujian *Blackbox* pada Sistem

Pengujian *Blackbox* pada sistem ini dilakukan pengujian menggunakan teknik pengujian *Equivalence Partitioning*. Teknik ini membagi input ke dalam kelompok partisi yang setara atau ekuivalen, di mana setiap partisi dianggap memiliki karakteristik dan perilaku yang sama. Pengujian dilakukan dengan memilih satu input dari setiap partisi untuk diuji, sehingga mengurangi jumlah pengujian yang perlu dilakukan.

Secara keseluruhan, hasil pengujian *Blackbox* pada halaman-halaman yang memiliki validasi khusus dapat dilihat pada Tabel 6.

**Tabel 6** Hasil Pengujian *Blackbox*

No.	Kelompok Pengujian	Validitas Sistem
1	Halaman <i>Register</i>	100%
2	Halaman <i>Login</i>	100%
3	Halaman <i>Keyword</i>	100%
4	Halaman <i>Engine</i>	100%

5	Halaman <i>Crawling Data</i> dan <i>Dataset</i>	100%
6	Halaman Deteksi Hoaks	100%

Hasil dari tabel *blackbox* menunjukkan bahwa semua halaman berfungsi dengan baik, menegaskan bahwa sistem deteksi berita hoaks berjalan dengan kinerja yang memuaskan. Tabel tersebut memberikan indikasi bahwa sistem dapat beroperasi secara efektif dalam mengidentifikasi dan membedakan berita hoaks dari berita non-hoaks.

#### 4. KESIMPULAN

Penelitian ini berhasil menerapkan Naïve Bayes Classifier dan Algoritma *Stemming* Nazief dan Adriani. Adapun kesimpulan yang dapat diambil adalah sebagai berikut:

- Hasil pengujian dan evaluasi menunjukkan bahwa sistem deteksi ini berhasil mencapai tingkat *Classification Report* yang cukup tinggi, yaitu nilai akurasi 93%, presisi 100%, *recall* 88% dan *F1-Score* 93%.
- Hasil eksperimen menunjukkan bahwa penerapan algoritma *Naïve Bayes* dengan metode Nazief dan Adriani mampu memberikan kinerja yang baik dalam proses klasifikasi.

#### 5. SARAN

Hasil pengujian menunjukkan tingkat kesesuaian yang cukup baik terhadap harapan yang telah ditetapkan. Namun, untuk meningkatkan kualitas dan efektivitas sistem deteksi hoaks ini, terdapat beberapa saran yang perlu dipertimbangkan yakni:

- Selain menggunakan algoritma *Naïve Bayes*, perlu diinvestigasi penerapan metode *machine learning* lainnya untuk meningkatkan kinerja sistem. Contoh metode yang dapat dipertimbangkan adalah *Support Vector Machine* (SVM), *Random Forest*, atau *Neural Network*.
- Meluaskan cakupan *dataset* dengan mengumpulkan data dari berbagai sumber berita hoaks dan non-hoaks yang lebih beragam.

- Mempertimbangkan penggunaan teknik fitur ekstraksi yang lebih canggih seperti *Word Embedding* atau TF-IDF.

#### UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih sebesar-besarnya kepada Universitas Halu Oleo khususnya jajaran staff dan Dosen Fakultas MIPA, Dosen Pembimbing, serta pihak-pihak yang telah membantu dalam proses penyusunan hasil penelitian ini.

#### DAFTAR PUSTAKA

- [1] H. Allcott dan M. Gentzkow, "Nber Working Paper Series Social Media and Fake News in the 2016 Election," *J. Econ. Perspect.*, vol. 31, no. 2, hal. 211–236, 2017, [Daring]. Tersedia pada: <https://web.stanford.edu/~gentzkow/research/fakenews.pdf>
- [2] MASTEL, "Hasil Survey Wabah Hoax Nasional 2019," *Website Masy. Telemat. Indones.*, hal. 35, 2019, [Daring]. Tersedia pada: <https://mastel.id/hasil-survey-wabah-hoax-nasional-2019/>
- [3] C. S. Sriyano dan E. B. Setiawan, "Pendeteksian Berita Hoax Menggunakan Naive Bayes Multinomial Pada Twitter dengan Fitur Pembobotan TF-IDF," *e-Proceeding Eng. Vol.8, No.2*, vol. 8, no. 2, hal. 3396–3405, 2021.
- [4] F.-J. Yang, "An implementation of naive bayes classifier," in *2018 International conference on computational science and computational intelligence (CSCI)*, 2018, hal. 301–306.
- [5] A. Prasadhatama dan K. M. Suryaningrum, "Perbandingan Algoritma Nazief & Adriani Dengan Algoritma Idris Untuk Pencarian Kata Dasar," *J. Teknol. dan Manaj. Inform.*, vol. 4, no. 1, hal. 1–4, 2018, doi: 10.26905/jtmi.v4i1.1773.
- [6] B. Bustami, "Penerapan algoritma

- Naive Bayes untuk mengklasifikasi data nasabah asuransi,” *TECHSI-Jurnal Tek. Inform.*, vol. 5, no. 2, 2013.
- [7] A. McCallum dan K. Nigam, “A Comparison of Event Models for Naive Bayes Text Classification,” in *AAAI-98 workshop on learning for text categorization*, 1998, vol. 752, no. 1, hal. 41–48.
- [8] Munirul, Ula, M. M. Alvanof, dan R. Triandi, “Analisa Dan Deteksi Konten Hoax Pada Media Berita,” *J. Teknol. Terap. Sains 4.0 Univ. Malikussaleh*, vol. 1, hal. 2, 2020.
- [9] B. K. Palma, D. T. Murdiansyah, dan W. Astuti, “Klasifikasi Teks Artikel Berita Hoaks Covid-19 dengan Menggunakan Algoritma K- Nearest Neighbor,” *eProceedings ...*, vol. 8, no. 5, hal. 10637–10649, 2021.
-