KLASIFIKASI BERITA MENGGUNAKAN METODE MULTINOMIAL NAÏVE BAYES

Findra Kartika Sari Dewi, Tri Purnomo Aji
Program Studi Informatika, Fakultas Teknologi Industri, Universitas Atma Jaya Yogyakarta
Email: findra.dewi@uajy.ac.id

Abstrak. *Berita pada awalnya disampaikan melalui media surat kabar, majalah, radio dan televisi, namun sekarang bergeser menggunakan sistem berbasis internet. Berita memiliki kategori berita, seperti polhukam, bisnis, olahraga, hiburan, teknologi, otomotif, kesehatan, dll. Saat ini kategorisasi artikel berita online masih dilakukan secara manual, sehingga hal ini sangat merepotkan dan membutuhkan banyak waktu. Untuk mengatasi masalah tersebut, dibutuhkan sebuah sistem yang dapat mengkategorikan atau mengklasifikasi artikel berita secara otomatis. Sistem klasifikasi ini dibangun menggunakan metode Text Mining dan Multinomial Naïve Bayes untuk membentuk dataset dan model klasifikasi artikel berita. Pengujian dilakukan menggunakan 10.500 dataset dan tujuh kategori. Pengujian diukur dengan menggunakan confusion matrix. Hasil pengujian menunjukkan bahwa implementasi Multinomial Naïve Bayes pada sistem klasifikasi artikel berita memiliki tingkat accuracy 96%, precision 96%, recall 96% dan f1-score 96%.*

Kata Kunci: artikel berita online, klasifikasi, text mining, multinomial naïve bayes, confusion matrix.

Menurut Kamus Besar Bahasa Indonesia, berita adalah cerita atau keterangan mengenai kejadian atau peristiwa yang hangat. Di dalam berita terdapat fakta dan opini yang menarik untuk disampaikan kepada masyarakat, tetapi tidak semua fakta dan opini bisa diangkat menjadi suatu berita oleh media [1]. Dalam perkembangannya, media penyaluran berita yang pada awalnya adalah media surat kabar, majalah, radio, dan televisi, sekarang sudah banyak bergeser menggunakan sistem berbasis internet [2] [3]. Pada umumnya, berita yang ada dalam portal berita dikelompokkan dalam beberapa kategori, seperti berita politik, olahraga, ekonomi, hiburan, teknologi, kesehatan dan lain-lain [4]. Masalahnya adalah pengelompokan berita ke dalam kategori-kategori tersebut untuk saat ini masih dilakukan secara manual, artinya dalam proses mengelompokkan berita tersebut harus didahului dengan membaca isi berita tersebut secara keseluruhan agar pengelompokan tepat. Hal ini sangat merepotkan, apalagi jumlah berita yang ingin dikategorikan berjumlah sangat banyak.

Salah satu teknik yang dapat digunakan untuk melakukan klasifikasi tersebut adalah *text mining* [5] [6]. *Text mining* merupakan variasi dari *data mining* yang berusaha menemukan pola yang menarik dari sekumpulan data tekstual yang berjumlah besar [7]. Selain klasifikasi, *text mining* juga digunakan untuk masalah *clustering* dan *information extraction* [8]. Klasifikasi merupakan proses untuk menemukan model atau fungsi yang menjelaskan atau membedakan konsep atau kelas data, dengan tujuan untuk dapat memperkirakan kelas dari suatu obyek [9]. Oleh karena itu, kelas yang ada tentulah lebih dari satu. Klasifikasi telah banyak dilakukan oleh para peneliti dengan menerapkan berbagai metode, salah satunya adalah *Naïve Bayes* [10]. Berdasarkan penelitian terdahulu, metode *Naïve Bayes Classifier* merupakan metode yang unggul dalam hal *robustness,* yaitu dapat mengenali kata spesifik yang berkorelasi erat terhadap suatu kategori, karena menggunakan model probabilistik yang dapat memperlihatkan perbedaan antar kata dengan jelas, sehingga dapat mengklasifikasi suatu dokumen uji dengan tingkat akurasi cukup tinggi [11]. Klasifikasi mengacu kepada aktifitas menganalisis atau mempelajari himpunan dokumen teks *preclasified* untuk memperoleh suatu model atau fungsi yang dapat digunakan untuk mengelompokkan dokumen teks lain yang belum diketahui kelasnya ke dalam satu kelas atau lebih kelas *predifined* [12].

Melihat permasalahan yang ada, maka perlu dibuat sistem yang dapat mengklasifikasikan berita secara otomatis sesuai dengan kategori-kategori berita yang ada, sehingga mempermudah dalam proses pengkategorian artikel berita [13]. Penulis mengembangkan sebuah sistem klasifikasi berita secara otomatis dengan menerapkan metode *text mining* dan *multinomial naïve bayes* yang mampu melakukan klasifikasi terhadap berita yang belum diketahui kelasnya.

Penelitian oleh Rahman, dkk. (2017) tentang *online news classification using multinomial naïve bayes* menjelaskan bahwa

metode *multinomial naïve bayes* sangat membantu untuk klasifikasi artikel berita dengan hasil akurasi yang cukup tinggi. Penelitian ini menggunakan tiga kategori dan 1.011 *dataset*, dengan berita dalam teks Bahasa Indonesia Penggunaan fitur seleksi *DF-Thresholding* menghasilkan akurasi akhir sebesar 93,33%. Sementara penggunaan *TF-IDF* pada metode *multinomial naïve bayes* mendapatkan hasil akurasi akhir sebesar 94,29%. Penggunaan pada *multinomial naïve bayes* dengan *TF-IDF* pun menunjukan nilai akurasi akhir yang lebih rendah dari *Multinomial Naive Bayes* dengan *TF-IDF*, yaitu sebesar 92,38% namun dengan penggunaan jumlah fitur yang lebih sedikit dalam proses klasifikasi [8].

Penelitian oleh Syahnur, dkk. (2016) tentang kategorisasi topik *tweet* di kota Jakarta, Bandung dan Makassar dengan metode *multinomial naïve bayes classifier,* dengan jumlah *dataset* sebanyak 601 dan 13 kategori. Dalam penelitian tersebut disebutkan bahwa untuk klasifikasi sebuah kata atau *tweet* dibutuhkan beberapa tahap sebelum dilakukan proses klasifikasi. Misalnya *tweet* yang dijadikan bahan pelatihan harus dilakukan tahap *preprocessing* dengan tujuan untuk meningkatkan hasil analisis terkait masalah waktu, biaya dan kualitas. Terdapat tiga tahapan praproses yaitu *tokenization, stopword* dan *stemming*. Penelitian ini mendapatkan nilai *f1-measure* rata-rata yang cukup tinggi, yaitu 77% [11].

Penelitian oleh Karunia, dkk. (2017) tentang *online news classification using naïve bayes classifier with mutual information for feature selection* bertujuan untuk membuat sistem klasifikasi berita online menggunakan metode *naive bayes* dengan fitur seleksi *mutual information*. Hasil penelitian dari 2.386 data yang terdiri dari tujuh kategori mendapatkan nilai akurasi sebesar 70%, *precision* sebesar 33,33%, *recall* sebesar 9,40% dan *f-measure* sebesar 14,35%. Hasil pengujian mengalami penurunan dikarenakan adanya fitur yang terbuang dari proses seleksi fitur. Meskipun hasil pengujian mengalami penurunan dari hasil pengujian sebelumnya, namun seleksi fitur *mutual information* menghasilkan efisiensi jumlah fitur mencapai 52% dari sebelumnya [14].

Penulis melakukan penelitian dengan topik klasifikasi berita menggunakan metode *multinomial naïve bayes*. Sistem yang dibuat mampu mengklasifikasikan artikel berita secara otomatis yang kemudian disimpan ke dalam *database* dan digunakan kembali dalam keadaan terstruktur berdasarkan kategori klasifikasi, sehingga para editor berita tidak perlu lagi bersusah payah melakukan pengkategorian berita secara manual. Tabel 1 menampilkan perbandingan antara penelitian-penelitian terdahulu dengan penelitian yang dilakukan oleh penulis.

I. Metodologi

Metodologi penelitian yang digunakan untuk melakukan pembangunan sistem klasifikasi artikel berita ini terdiri dari enam tahapan, yaitu:

1. **Studi Pustaka.** Tahap ini dilakukan dengan mempelajari hal-hal terkait topik penelitian dari buku, jurnal, dan artikel di internet.
2. **Pengumpulan Data.** Data yang digunakan dalam penelitian ini diambil dari portal *web* berita Kompas dan Tempo. Data yang diambil merupakan data dengan kategori polhukam, bisnis, hiburan, olahraga, teknologi, otomatif dan kesehatan. Data diambil dengan menggunakan teknik *web scrapping,* yaitu sebuah proses ekstraksi data dari sebuah halaman *website*. Keuntungan *web scraping* adalah efisiensi waktu, memungkinkan pengumpulan data secara teratur dengan interval waktu yang singkat, pengurangan biaya dan jumlah informasi yang diperoleh lebih banyak [15]. *Web Scraping* memiliki beberapa tahapan, yaitu membuat *template scraping*, eksplorasi navigasi situs, mengotomatisasi navigasi, mengekstraksi informasi dan menyimpan histori [16]. Pengumpulan data bertujuan untuk memperoleh *dataset* atau *corpus* untuk klasifikasi artikel berita.
3. **Preprocessing Data.** *Preprocessing* dilakukan untuk menghilangkan *noise* yang terdapat dalam teks, sehingga diperoleh data yang berkualitas untuk proses klasifikasi [17]. *Preprocessing* dapat meningkatkan hasil analisis terkait masalah waktu, biaya dan kualitas [11]. *Preprocessing data* terdiri dari empat tahapan yaitu *case folding, tokenize, filtering/stopword removal* dan *stemming* [18]. Proses *case folding* berfungsi untuk mengubah semua huruf dalam sebuah

dokumen teks menjadi huruf kecil (*lowercase*), dan menghapus karakter selain huruf, seperti tanda baca koma, titik, dll., serta angka. Proses *tokenize* berfungsi untuk mengubah sebuah kalimat menjadi potongan-potongan kata. Proses *filtering/stopword removal* berfungsi untuk menghapus kata-kata yang tidak diperlukan atau kata-kata yang sering muncul, seperti kami, saya, kamu, dll. Terakhir, proses *stemming* yang berfungsi untuk mengembalikan kata-kata yang ada di *token list* menjadi bentuk awal atau kata dasar dari kata tersebut, dengan menghilangkan imbuhan awalan, sisipan, akhiran ataupun kombinasi awalan dan akhiran.

a. **Pembobotan Term Frequency-Inverse Document Frequency *(*TF-IDF*).*** Data yang telah melalui tahap *preprocessing* harus berbentuk numerik. Agar data tersebut menjadi numerik, digunakan metode pembobotan *TF-IDF*. Metode ini digunakan untuk menentukan seberapa jauh keterhubungan kata (*term*) terhadap dokumen dengan memberikan bobot pada setiap kata. Metode ini menggabungkan dua konsep, yaitu frekuensi kemunculan sebuah kata di dalam dokumen dan invers frekuensi dokumen yang mengandung kata tersebut. Rumus untuk *TF-IDF* dapat dilihat pada (1), (2) dan (3) [19], dimana D = dokumen ke-d, t = *term* ke-t dari dokumen, W = bobot dokumen ke-d terhadap *term* ke-t, tf = banyaknya *term* i pada sebuah dokumen, idf = inversed document frequency, dan df = banyak dokumen yang mengandung *term* i.

$$tf = 0{,}5 + 0{,}5 \times \frac{tf}{\max(tf)} \quad (1)$$

$$idf_t = log\left(\frac{D}{df_t}\right) \quad (2)$$

$$W_{dt} = tf_{d.t} \times idf_{d.t} \quad (3)$$

b. **Klasifikasi menggunakan Multinomial Naïve Bayes.** *Multinomial Naïve Bayes* merupakan metode klasifikasi dengan pembelarajan *supervised* menggunakan model probabilistik [11]. *Multinomial Naïve Bayes* dipengaruhi oleh serangkaian *term*, dengan kata lain jumlah *term* diperhitungkan. Peluang antara *term* satu dengan yang lain adalah independen (tidak bergantung) [14]. Model *Multinomial Naïve Bayes* memperhitungkan frekuensi setiap kata yang muncul pada dokumen. Rumus *Multinomial Naïve Bayes* yang digunakan untuk pembobotan kata *TF-IDF* dapat dilihat pada persamaan (4) [8], dimana W_{ct} = nilai pembobotan *tf-idf* atau W dari *term* di kategori c, $\sum \boldsymbol{w'} \in \boldsymbol{VW'ct}$ = Jumlah total W dari keseluruhan term yang berada di kategori c, dan *B'*= jumlah W kata unik (nilai *idf* tidak dikali dengan *tf*) pada seluruh dokumen.

$$P(tn|c) = \frac{W_{ct}+1}{\left(\sum_{w' \in VW'ct}\right)+B'} \quad (4)$$

c. **Proses Stemming.** *Stemming* adalah proses untuk menemukan kata dasar dari sebuah kata, dengan menghilangkan semua imbuhan, mencakup kata sisipan, kata awalan, kata akhiran ataupun keduanya. Tidak seperti Bahasa Inggris, Bahasa Indonesia memiliki kategori afiks yang lebih kompleks. Algoritma *stemming* yang dikembangkan oleh Nazhief dan Adriani berdasarkan aturan morfologi Bahasa Indonesia mengelompokkan imbuhan menjadi awalan (*prefix*), sisipan (*infix*), akhiran (*suffix*) dan gabungan awalan akhiran (*confix*) [20].

4. **Pelabelan Data.** Sebelum dilakukan tahap klasifikasi, dibutuhkan pelabelan data secara manual berdasarkan tujuh kategori yang sudah ditentukan untuk dijadikan sebagai *corpus* atau *dictionary* yang nantinya menjadi data awal pembelajaran atau yang disebut dengan *dataset*. Dengan begitu, *dataset* yang memiliki label kategori dapat dijadikan sebagai bahan *data training* dan *data testing* untuk memprediksi kategori dari artikel berita yang baru.
5. **Penerapan Algoritma.** Pada tahap ini dilakukan penerapan algoritma atau metode dengan *dataset* yang sudah terbentuk, menggunakan Bahasa pemrograman Python.
6. **Pegujian Aplikasi.** Pada tahap ini dilakukan pengujian fungsionalitas aplikasi, baik secara otomatis oleh sistem

menggunakan *confusion matrix* maupun secara langsung kepada pengguna.

II. Hasil dan Pembahasan

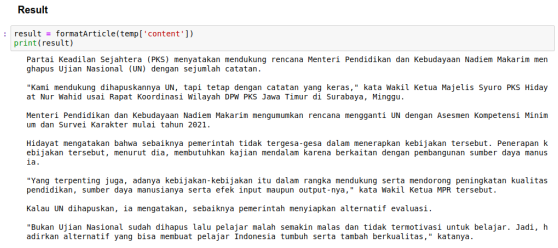
Pengumpulan Data

Data dalam penelitian ini diperoleh dari dua sumber media berita *online,* yaitu Kompas dan Tempo. Data yang digunakan berasal dari tujuh kategori, yaitu kategori polhukam, bisnis, olahraga, hiburan, teknologi, otomotif dan kesehatan, masing-masing sebanyak 1.500 artikel, sehingga totalnya sebanyak 10.500 artikel. Proses pengumpulan data dilakukan secara otomatis menggunakan teknik *web scraping*, dengan begitu data yang ingin dikumpulkan sangat banyak maka dapat dilakukan secara cepat dan dapat menghemat waktu. Penulis menggunakan dua *library* utama untuk mengambil atau mengekstrak informasi yang ada di halaman *website* media berita *online*, yaitu menggunakan *library* BeautifulSoup dan Requests.

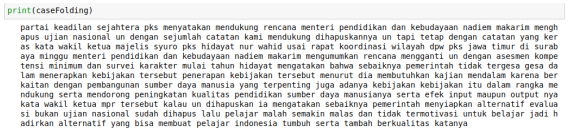
Tahap Preprocessing

Proses *preprocessing* memiliki tiga tahap pembersihan data, yaitu proses *case folding*, proses *stopword removal* dan proses *stemming*. Data yang dibersihkan hanya isi dari artikel berita, dikarenakan isi artikel berita tersebut akan digunakan sebagai data utama dalam *training*. Berikut adalah penjelasan mengenai hasil *preprocessing text*.

1. **Tahap case folding.** Tahap ini menggunakan *library preprocessing_text* dari *Textacy* (sebuah pustaka yang digunakan di bahasa pemrograman Python). Pembersihan data pada tahap ini mencakup mengubah isi dokumen atau kalimat menjadi huruf kecil, dan menghilangkan karakter selain huruf, *url* atau alamat sebuah *website*, *email*, angka, simbol mata uang dan tanda baca. Gambar 1 menampilkan isi dari sebuah artikel berita yang belum dilakukan proses *case folding*, yang masih terdapat banyak *noise*, sedangkan Gambar 2 menampilkan hasil proses *case folding* dari isi artikel berita dari Gambar 1.

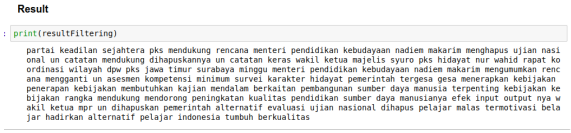


Gambar 1. Isi Artikel Sebelum Proses *Case Folding*



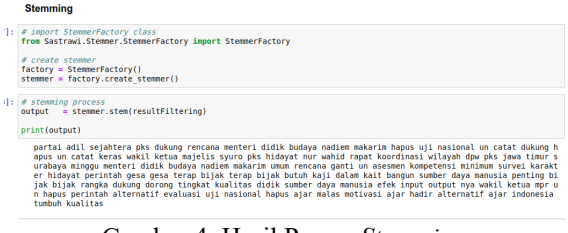
Gambar 2. Hasil Proses *Case Folding*

2. **Tahap stopword removal.** Tahap ini bertujuan untuk menghilangkan kata-kata tidak mempunyai arti atau kata-kata yang sering diulang. Untuk melakukan proses *stopword removal* penulis menggunakan kata-kata *stopword* dari https://github.com/masdevid/ID-Stopwords yang memiliki 758 kata *stopword*. Gambar 3 menampilkan hasil dari proses *stopword removal* dari artikel berita yang ada di Gambar 2.



Gambar 3. Hasil Proses *Stopword Removal*

3. **Tahap stemming.** Tahap ini bertujuan untuk menghilangkan kata imbuhan yang meliputi awalan kata, akhiran kata, sisipan kata, ataupun keduanya. Untuk melakukan proses *stemming,* digunakan *library* dari https://github.com/sastrawi/sastrawi. Gambar 4 menampilkan hasil dari tahap *stemming*.

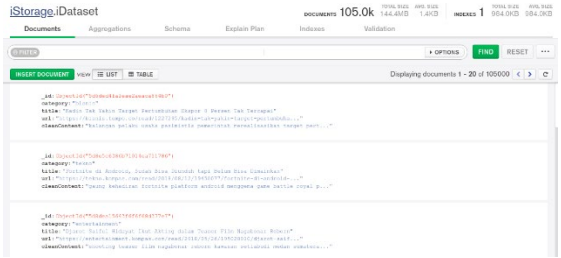


Gambar 4. Hasil Proses *Stemming*

Pelabelan Data

Data yang sudah dikumpulkan pada tahap pengumpulan data merupakan data yang sudah bersih dan sudah tidak terdapat *noise*

yang dapat mengurangi kualitas dari data tersebut, akan tetapi data tersebut belum bisa digunakan sebagai *dataset* awal atau *corpus*, dikarenakan data tersebut belum memiliki label kategori. Setelah tahap *preprocessing* selesai, data diberi label kategori secara manual, sesuai dengan isi artikel berita tersebut. Gambar 5 menampilkan data akhir yang dikumpulkan untuk dijadikan sebagai *dataset* atau *corpus*. Variabel-variabel yang terdapat pada *dataset* terdiri dari judul artikel berita, kategori artikel berita, *url* atau alamat artikel berita, dan isi artikel berita yang sudah bersihkan. Terdapat 10.500 data artikel berita yang diberi label kategori secara manual.



Gambar 5. Contoh *Dataset* atau Corpus

Pengujian Metode

Pada tahap pengujian metode, penulis menggunakan metode *Multinomial Naive Bayes* untuk membuat model klasifikasi artikel berita, yang nantinya model tersebut akan digunakan untuk melakukan klasifikasi artikel berita secara otomatis. Penulis menggunakan sebuah *library* dari *Scikit Learn* untuk dapat melakukan pengujian terhadap metode tersebut. Tahapan pengujian dilakukan dengan (1) pembuatan model *pipeline* (Gambar 6) yang dapat membantu proses pengujian metode secara lebih efisien, (2) pembuatan *data training* dan *data testing* menggunakan *library train_test_split* dari *Scikit Learn*, dengan cara membagi *dataset* dengan porsi 90% *data training* dan 10% *data testing* (Gambar 7), (3) *training data* menggunakan metode *multinomial naive bayes* dan *data training* yang sudah dibagi pada tahap sebelumnya (Gambar 8).

Create Pipeline MNB

```
nb = Pipeline([('vect', CountVectorizer()),
               ('tfidf', TfidfTransformer()),
               ('clf', MultinomialNB()),
              ])
```

Gambar 6. *Model pipeline*

Split Dataset

```
X1_train, X1_test, y1_train, y1_test = train_test_split(df['cleanContent'], df['category'], test_size=0.1)

print('X_traint : {}'.format(len(X1_train)))
print('X_test   : {}'.format(len(X1_test)))
print('Total    : {}'.format(len(X1_train) + len(X1_test)))
```

```
X_traint : 94500
X_test   : 10500
Total    : 105000
```

Gambar 7. *Split Dataset*

Use MNB

```
nb.fit(X1_train, y1_train)
```

```
Pipeline(memory=None,
     steps=[('vect', CountVectorizer(analyzer='word', binary=False, decode_error='strict',
        dtype=<class 'numpy.int64'>, encoding='utf-8', input='content',
        lowercase=True, max_df=1.0, max_features=None, min_df=1,
        ngram_range=(1, 1), preprocessor=None, stop_words=None,
        strip...inear_tf=False, use_idf=True)), ('clf', MultinomialNB(alpha=1.0, class_prior=None, fit_prior=Tru
e))])
```

Gambar 8. *Training Dataset*

Gambar 9 menampilkan *accuracy score* dari model klasifkasi *multinomial naive bayes* sebesar 96,67% dengan 10.500 *dataset* yang digunakan. Gambar 10 menampilkan hasil *confusion matrix* dari model tersebut, dengan skor akurasi 96%, *precession* 96%, *recall* 96% dan *f1-score* 96% dengan 10.500 *dataset* yang digunakan. Gambar 11 menampilkan visualisasi *heatmap confusin matrix* menggunakan *library Seaborn*. Hasil yang ditujukkan merupakan hasil yang cukup bagus untuk sebuah model prediksi klasifikasi teks. Setelah dilakukan pengujian metode dan hasilnya ternyata sudah bagus untuk sebuah model klasifikasi, maka model klasifkasi tersebut akan disimpan sehingga nantinya dapat digunakan kembali.

Score Model MNB

```
# Calculate the accuracy score and predict target values
score = nb.score(X1_train, y1_train)
print("Test score: {0:.2f} %".format(100 * score))
```

```
Test score: 96.67 %
```

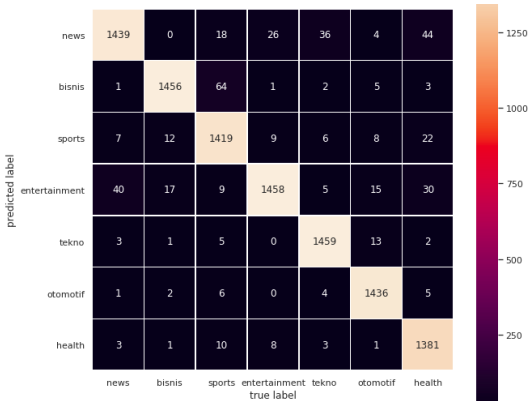
Gambar 9. Skor Akurasi Model Klasifikasi

Confusion Matrix MNB

```
print("Table Confusion Matrix")
print("==========================\n")
print('Accuracy %s' % accuracy_score(y_pred, y1_test))
print('==========================\n')
print(classification_report(y1_test, y_pred,target_names=categories))
```

```
Table Confusion Matrix
==========================

Accuracy 0.956952380952381
==========================

                precision    recall  f1-score   support

          news       0.92      0.96      0.94      1494
        bisnis       0.95      0.98      0.96      1489
        sports       0.96      0.93      0.94      1531
entertainment       0.93      0.97      0.95      1502
         tekno       0.98      0.96      0.97      1515
      otomotif       0.99      0.97      0.98      1482
        health       0.98      0.93      0.95      1487

   avg / total       0.96      0.96      0.96     10500
```

Gambar 10. *Confusion Matrix MNB*

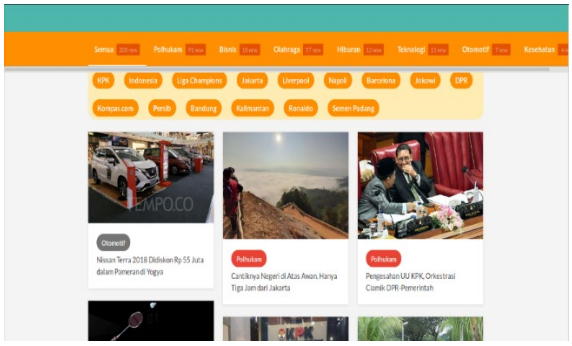


Gambar 11. *Heatmap Confusin Matrix*

Implementasi Antarmuka

Antarmuka untuk aplikasi *web* ini mencakup halaman login, halaman beranda, halaman pencarian berita, halaman dashboard admin, halaman chart harian, halaman artikel sebelum klasifikasi, halaman artikel sesudah klasifikasi dan halaman pengujian manual, namun karena keterbatasan halaman, maka pada makalah ini hanya akan ditampilkan beberapa halaman saja.

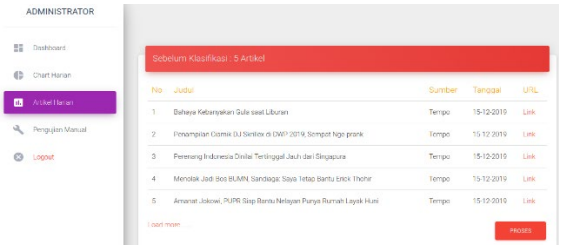
Gambar 12 menampilkan halaman beranda dengan artikel berita terbaru. Gambar 13 menampilkan hasil pencarian berita dengan kata kunci "CPNS" dari semua kategori yang terkait dengan kata kunci dan tanggal penerbitan 10 Oktober 2019 hingga 20 November 2019.



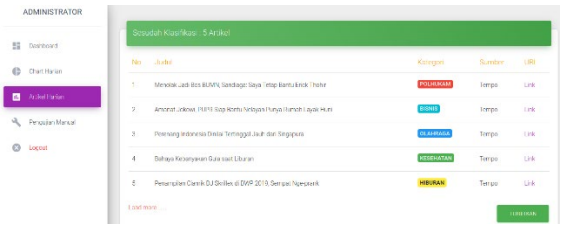
Gambar 12. Halaman Beranda dengan Artikel

Gambar 13. Hasil Pencarian Berita

Gambar 14 menampilkan halaman artikel sebelum klasifikasi pada aplikasi *web*. Di halaman ini admin dapat melakukan klasifikasi manual terhadap artikel. Admin dapat melakukan proses klasifikasi manual ini dengan menekan tombol Proses. Gambar 15 menampilkan halaman artikel sesudah klasifikasi. Halaman ini menampilkan data artikel berita yang sudah dilakukan klasifikasi oleh sistem dan dapat dilakukan penerbitan artikel berita secara manual oleh admin. Proses penerbitan ini dapat dilakukan oleh admin melalui tombol Terbitkan.

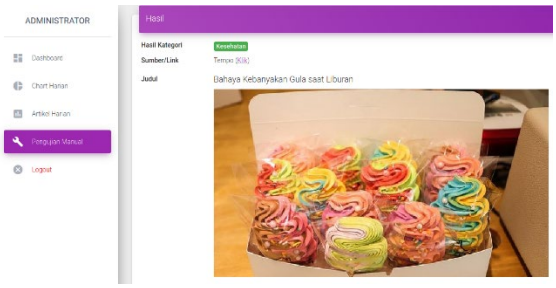


Gambar 14. Halaman Artikel Sebelum Klasifikasi



Gambar 15. Halaman Sesudah Klasifikasi

Gambar 16 menampilkan hasil pengujian manual pengkategorian artikel berita. Informasi yang ditampilkan meliputi hasil kategori artikel berita, sumber artikel berita, judul artikel berita, gambar artikel berita, hasil proses *case folding*, hasil proses *stopword removal* dan hasil proses *stemming*.



Gambar 16. Halaman Hasil Pengujian Manual

Hasil Pengujian Aplikasi oleh Pengguna

Pengujian aplikasi dilakukan secara otomatis menggunakan *confusion matrix* dan juga dilakukan secara langsung oleh pengguna, dan hasilnya dikumpulkan menggunakan

kuisioner. Pengujian oleh pengguna melibatkan 30 responden dari pihak perusahaan media berita *online* Beritagar, dan hasilnya dapat dilihat pada Tabel 2.

Tabel 2. Pengujian oleh Pengguna

No	Pertanyaan	SS	S	B	TS	STS
1	Tampilan keseluruhan dari aplikasi nyaman dilihat.	20	10	0	0	0
2	Aplikasi mudah digunakan dan dipahami.	6	24	0	0	0
3	Aplikasi ini membantu klasifikasi kategori artikel berita online secara otomatis.	2	19	9	0	0
4	Aplikasi ini dibutuhkan oleh perusahaan berita online.	2	21	7	0	0
5	Isi informasi yang diberikan aplikasi sesuai dengan judul.	7	23	0	0	0
6	Secara umum anda merasa puas menggunakan aplikasi ini.	6	19	5	0	0

III. Kesimpulan

Dari seluruh tahapan yang sudah dilakukan dalam penelitian ini, dapat ditarik kesimpulan bahwa sistem klasifikasi artikel berita *online* secara otomatis menggunakan metode *text mining* dan *multimonial naïve bayes* telah berhasil dibangun, dengan hasil pengujian yang sangat bagus, dengan skor akurasi, skor *precission,* skor *recall* dan skor *f1-score* masing-masing sebesar 96% dengan 10.500 *dataset* yang digunakan. Model klasifikasi ini dapat dijadikan model untuk digunakan secara terus-menerus, untuk menentukan kategori dari sebuah artikel berita.

Untuk lebih menyempurnakan sistem klasifikasi ini, penulis menyarankan untuk menambahkan lagi sampel kategori artikel berita sesuai dengan kriteria kategori yang berkembang pada saat ini. Selain itu, model klasifikasi dapat diperbarui secara berkala dengan menggunakan *dataset* terbaru.

IV. Daftar Pustaka

[1] Chandra, D. N., Indrawan, G. & Sukajaya, I. N.. (2016). Klasifikasi Berita Lokal Radar Malang Menggunakan Metode Naïve Bayes Dengan Fitur N-Gra. *Jurnal Ilmiah Teknologi dan Informasia ASIA (JITIKA)* , vol. 10, p. 11.

[2] Mansah, A. M. (2019). Tren Pergeseran Media Konvensional Ke Era Digitalisasi (Studi Kasus Konvergensi Media Di Lembaga Kantor Berita Nasional Antara Biro Sulawesi Selatan-Sulawesi Barat). *Al-MUNZIR*, *12*(1), 121-130.

[3] Schrøder, K. C., & Steeg Larsen, B. (2010). The shifting cross-media news landscape: Challenges for news producers. *Journalism studies*, 11(4), 524-534.

[4] Usmani, S., & Shamsi, J. A. (2020, March). News headlines categorization scheme for unlabelled data. In 2020 International Conference on Emerging Trends in Smart Technologies (ICETST) (pp. 1-6). IEEE

[5] Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E. D., Gutierrez, J. B., & Kochut, K. (2017). A brief survey of text mining: Classification, clustering and extraction techniques. arXiv preprint arXiv:1707.02919.

[6] Wongso, R., Luwinda, F. A., Trisnajaya, B. C., & Rusli, O. (2017). News article text classification in Indonesian language. Procedia Computer Science, 116, 137-143.

[7] Srivastava, A. N., & Sahami, M. (Eds.). (2009). Text mining: Classification, clustering, and applications. CRC press.

[8] Rahman, A. , Wiranto, W. & Doewes, A.. (2017). Online News Classification Using Multinomial Naive Bayes. ITSMART: *Jurnal Ilmiah Teknologi dan Informasi*, vol. 6, p. 32,.

[9] Mangal, S. B., & Goyal, V. (2014). Text news classification system using Naïve Bayes classifier. *An International Journal of Engineering Sciences*, 3.

[10] Kumar, S., Sharma, A., Reddy, B. K., Sachan, S., Jain, V., & Singh, J. (2021). An intelligent model based on integrated inverse document frequency and multinomial Naive Bayes for current affairs news categorisation. *International Journal of System Assurance Engineering and Management,* 1-15.

[11] Syahnur, M. H., Bijaksana, M. A. & Mubarok, M. S.. (2016). Kategorisasi Topik Tweet di Kota Jakarta, Bandung, dan Makassar dengan Metode Multinomial Naïve Bayes Classifier. *e-Proceeding of Engineering,* vol. 3, p. 3631.

[12] Nurhadi, A. (2016). Implementasi Algoritma Naïve Bayes Classifier Berbasis Particle Swarm Optimization

(PSO) Untuk Klasifikasi Konten Berita Digital Bahasa Indonesia. *Journal Speed – Sentra Penelitian Engineering dan Edukas,* vol. 8, p. 48.

[13] Ariadi, D. & Fithriasari, K.. (2015). Klasifikasi Berita Indonesia Menggunakan Metode Naive Bayesian Classification dan Support Vector Machine dengan Confix Stripping Stemmer. *Jurnal Sains dan Seni ITS,* p. 248.

[14] Karunia, S. A., Saptono, R. & Anggrainingsih, R.. (2017). Online News Classification Using Naive Bayes Classifier with Mutual Information for Feature Selection. *ITSMART: Jurnal Ilmiah Teknologi dan Informasi,* vol. 6, p. 11.

[15] Maulana, A. A., Susanto, A. & Purwanti K, D.. (2019). Rancang Bangun Web Scraping Pada Marketplace di Indonesia," *Journal of Information System,* vol. 4, p. 1.

[16] Mitra, V., Sujaini, H. & Negara, A. B. P.. (2017). Rancang Bangun Aplikasi Web Scraping untuk Korpus Paralel Indonesia-Inggris dengan Metode HTML DOM. *Jurnal Sistem dan Teknologi Informasi (JUSTIN),* vol. 5, p. 36.

[17] Efendi, Z. & Mustakim, M.. (2019). Text Mining Classification Sebagai Rekomendasi Dosen Pembimbing Tugas Akhir Program Studi Sistem Informasi. *Seminar Nasional Teknologi Informasi, Komunikasi dan Industri (SNTIKI),* vol. 3, p. 236.

[18] Nurfikri, F. S., Mubarok, S. M. & Adiwijaya, A.. (2018). Klasifikasi Topik Berita Menggunakan Mutual Information dan Bayesian Network," *e-Proceeding of Engineering,* vol. 5, p. 1579,.

[19] Wahyuni, R. T., Prastiyanto, D. & Supraptono, E.. (2017). Penerapan Algoritma Cosine Similarity dan Pembobotan TF-IDF pada Sistem Klasifikasi Dokumen Skripsi. *Jurnal Teknik Elektro,* vol. 9, p. 18.

[20] Tahitoe, A. D. & Purwitasari, D.. (2010). Implementasi Modifikasi Enhanced Confix Stripping Stemmer Untuk Bahasa Indonesia Dengan Metode Corpus Based Stemming. *Jurnal Ilmiah Teknologi Informasi,* vol. 1, p. 1.