
Penggunaan Algoritma *Naive Bayes* Dalam *Text Mining* Untuk Klasifikasi Pasal UU ITE

Farhan¹, Triase², Aninda Muliani Harahap³

^{1,2,3}Sistem Informasi, Universitas Islam Negeri Sumatera Utara Medan
Email: ¹faarhan7@gmail.com, ²triase@uinsu.ac.id, ³anindamh@gmail.com
Email Penulis Korespondensi: faarhan7@gmail.com

Article History:

Received April 05th, 2023

Revised May 15th, 2023

Accepted June 20th, 2023

Abstrak

Saat ini, kemajuan informasi dan teknologi terus mengalami perkembangan yang pesat. Meskipun memberikan kemudahan bagi manusia, pengguna internet juga bisa mengalami risiko dalam keamanan dan privasi data mereka. Untuk mengatasi hal ini, UU ITE dibuat sebagai bentuk perlindungan bagi pengguna internet. Namun, masih banyak masyarakat yang melanggar peraturan dalam UU ITE. Untuk meningkatkan kesadaran pengguna internet, teknologi Machine Learning digunakan dengan metode Text Mining untuk mengklasifikasikan pelanggaran UU ITE berdasarkan teks kronologi pelanggaran dengan menggunakan algoritma Naive Bayes. Penelitian ini difokuskan pada pasal 27 dan 28 dengan total data sebanyak 245 kronologi pelanggaran. Pada implementasinya pada aplikasi *web*, menggunakan *micro web framework* Python yaitu *Flask* yang membantu dalam mengeksekusi teks kronologi dari *text pre-processing*, *weighting*, *learning*, dan *testing* sehingga menghasilkan informasi mengenai ancaman pasal yang mempunyai probabilitas tinggi untuk dilanggar. Hasil penelitian menunjukkan akurasi sebesar 80% dengan 196 total data latih dan 49 total data uji. Hal ini menunjukkan bahwa algoritma Naive Bayes mampu menghasilkan tingkat akurasi yang tinggi dalam mengklasifikasikan sebuah teks.

Kata Kunci : UU ITE, *Machine Learning*, *Web*, *Flask*, *Python*, *Naive Bayes*, *Text Mining*

Abstract

Currently, the advancement of information and technology continues to experience rapid development. Despite providing convenience for humans, internet users can also experience risks in the security and privacy of their data. To overcome this, the ITE Law was created as a form of protection for internet users. However, there are still many people who violate the regulations in UU ITE. To increase internet user awareness, Machine Learning technology uses the Text Mining method to classify violations of the ITE Law based on text chronology violations using the Naive Bayes algorithm. This research focuses on articles 27 and 28 with a total data of 245 violation chronologies. In its implementation on web applications, it uses a Python micro web framework called Flask which helps in executing chronological text from text pre-processing, weighting, learning, and testing so as to produce information about article threats that have a high risk of being violated. The results showed an accuracy of 80% with 196 total training data and 49 total test data. This shows that the Naive Bayes algorithm is capable of producing a high level of accuracy in classifying a text.

Keyword : UU ITE, *Machine Learning*, *Web*, *Flask*, *Python*, *Naive Bayes*, *Text Mining*

1. PENDAHULUAN

Perkembangan teknologi di era digital 4.0 terus berlangsung pesat. Banyak pekerjaan manusia yang bisa terbantu dengan memanfaatkan teknologi, mulai dari sektor pertanian, kesehatan, hingga industri. Salah satu kemudahan yang diberikan oleh teknologi adalah dalam pencatatan data, yang sebelumnya dilakukan secara manual namun kini semakin banyak berubah menjadi digital. Dalam proses perkembangan teknologi tersebut, data yang dihasilkan semakin besar dan

berkembang dengan kecepatan yang sangat cepat. Karena data yang begitu besar, perusahaan, institusi, dan organisasi menyimpan data dalam sistem secara elektronik. Saat jumlah data semakin besar dan informasi penting yang dibutuhkan semakin banyak, maka diperlukan sebuah cabang ilmu Data Science yang disebut dengan Text Mining. Text mining adalah suatu proses untuk mendapatkan informasi yang berkualitas tinggi dari teks. Proses ini dilakukan dengan mempelajari pola dan tren yang ada dalam teks menggunakan metode statistik [1].

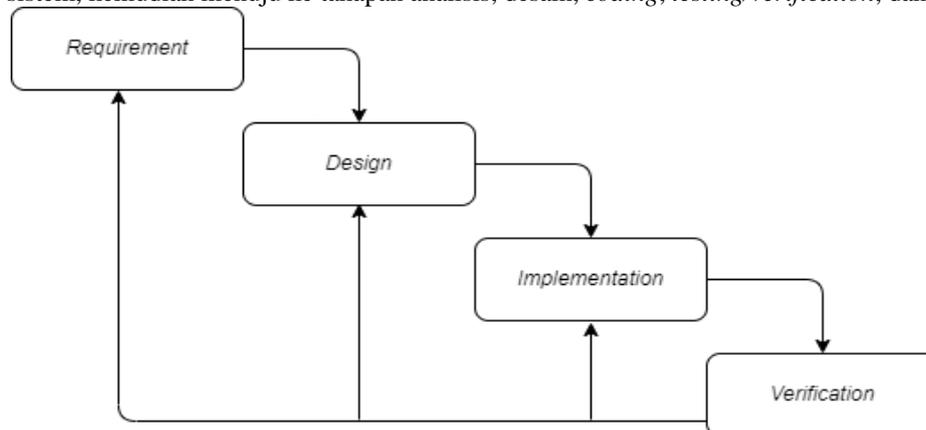
Pengadilan Negeri Medan berlokasi di Jl. Pengadilan Kelurahan No.8, Petisah Tengah, Kec. Medan Petisah, Kota Medan. Selain sebagai peradilan umum untuk perkara perdata dan pidana, pengadilan ini juga memiliki pengadilan khusus dalam lingkup peradilan umum. Menurut Bambang Joko Winarno, SH, seorang hakim di Pengadilan Negeri Medan, kebanyakan kasus pelanggaran UU ITE di Medan melibatkan pasal-pasal seperti 27 ayat 1, 27 ayat 2, 27 ayat 3, 27 ayat 4, dan 28 ayat 2. Oleh karena itu, penelitian ini akan berfokus pada pelanggaran UU ITE. UU ITE dibuat pada tahun 2008 dan direvisi pada tahun 2016 untuk menjaga etika, produktivitas, dan keadilan dalam ruang digital. Pasal 27 dan 28 UU ITE memuat larangan terkait konten yang melanggar kesusilaan, perjudian, pencemaran nama baik, pemerasan, dan menyebarkan berita bohong dan sara. Data survei oleh APJII pada tahun 2018 menunjukkan bahwa jumlah pengguna internet di Indonesia meningkat menjadi 171,1 juta dan jumlah ini terus bertambah setiap tahunnya. Sebanyak 1440 laporan berkaitan dengan konten negatif dilaporkan dalam siaran pers No.17/HM/KOMINFO/01/2019 oleh Ferdinandus Setu pada tahun 2018, dan kategori laporan yang paling banyak adalah konten yang meresahkan atau hoaks dengan 733 laporan. Fakta ini menunjukkan bahwa tingkat kejahatan siber yang melanggar UU ITE tergolong tinggi. Kasus pelanggaran UU ITE yang dijadikan objek penelitian sebanyak 245 dan diperoleh dari *website* putusan Mahkamah Agung. Untuk membantu masyarakat dalam mengklasifikasikan dan memahami jenis pasal UU ITE yang terlibat dalam kasus pelanggaran, metode pembelajaran mesin *Text Mining* akan digunakan.

Salah satu dari beberapa algoritma yang dapat diimplementasikan pada *Text Mining* adalah Naive Bayes. Metode klasifikasi Naive Bayes didasarkan pada probabilitas dan statistik yang dikembangkan oleh ilmuwan Inggris bernama Thomas Bayes. Tujuan dari metode ini adalah untuk memprediksi peluang di masa depan berdasarkan pengalaman di masa sebelumnya [2]. Algoritma Naive Bayes memungkinkan pembangunan model secara cepat dan menjadi algoritma prediksi yang paling cepat untuk dipelajari. Algoritma ini didasarkan pada penggunaan probabilitas dari sebuah objek [3].

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Metode Waterfall menerapkan pendekatan yang sistematis dan berurutan dalam pengembangan sistem, dimulai dari level kebutuhan sistem, kemudian menuju ke tahapan analisis, desain, *coding*, *testing/verification*, dan *maintenance* [4].



Gambar 1. Metode *Waterfall*

Metode *Waterfall* memiliki keterangan sebagai berikut:

1. Tahap *Requirement* Pada tahap ini, dilakukan analisis kebutuhan untuk proses Text Mining dan pengklasifikasian pasal UU ITE.
2. Tahap *Design*
Pada tahap ini, terdapat beberapa tahap, yaitu:
 - a. Membuat *Use Case Diagram*
Diagram ini menjelaskan hubungan antara pengguna dan sistem.
 - b. Membuat *Activity Diagram*
Diagram ini menggambarkan urutan aktivitas pada sistem.
 - c. Membuat *Sequence Diagram*
Diagram ini menjelaskan langkah-langkah interaksi dan operasi yang dilakukan.
3. *Implementation*

Tahap ini melibatkan pengetikan kode program sesuai dengan rancangan yang telah dibangun sebelumnya dalam bahasa Python.

4. *Verification*

Tahap ini melibatkan pengujian kode program yang telah dibuat dan memastikan bahwa program yang dibuat sesuai dengan tujuan akhir program. Jika ditemukan anomali dalam kode program, maka akan dilakukan revisi.

5. *Maintenance*

Tahap ini melibatkan perawatan sistem setelah program berjalan sesuai dengan tujuan yang ditentukan dan dapat menyelesaikan masalah. Hal ini bertujuan untuk menjaga agar kode program tetap terpelihara.

2.2 Algoritma Naïve Bayes

Naive Bayes adalah metode klasifikasi probabilistik yang sederhana, di mana ia menghitung serangkaian probabilitas dengan menjumlahkan frekuensi dan kombinasi nilai dari dataset yang diberikan. Metode ini menggunakan teorema Bayes dan mengasumsikan bahwa semua atribut dalam dataset adalah independen atau tidak saling ketergantungan dengan nilai pada variabel kelas [5].

3. HASIL DAN PEMBAHASAN

3.1 Analisis Algoritma Naïve Bayes

Untuk mengklasifikasikan pelanggaran UU ITE dari teks kronologi, perlu dilakukan beberapa tahapan seperti *preprocessing*, *weighting*, *learning*, dan evaluasi. Berikut adalah urutan tahapan yang harus dilakukan [6]:

3.1.1 *Preprocessing*

Preprocessing adalah tahap awal dalam pengolahan data yang harus dilakukan sebelum dapat dihitung oleh algoritma. Tujuan dari tahap ini adalah untuk membersihkan data dari anomali sehingga data menjadi terstruktur.

1. *Casefolding* adalah tahap dimana seluruh huruf pada data diubah menjadi huruf kecil agar memudahkan pada proses pengindeksan.
2. *Tokenizing* adalah proses pemisahan *string input* menjadi beberapa bagian berdasarkan kata-kata yang membentuknya. Proses *tokenizing* juga sering dilakukan untuk menghapus beberapa karakter seperti tanda baca yang dianggap tidak perlu.
3. *Stopword Removal* adalah proses penyaringan kata-kata yang dianggap tidak relevan dalam penentuan topik suatu dokumen, seperti kata "dan", "atau", "dari", dan lain-lain.
4. *Stemming* adalah proses mengubah kata-kata ke bentuk dasarnya atau menghilangkan imbuhan pada kata-kata tersebut [7].

3.1.2 *Weighting*

Pada tahap *weighting* atau pembobotan, teks diubah menjadi nilai numerik sehingga dapat dihitung oleh algoritma Naive Bayes. Proses ini dilakukan setelah *data training* melewati tahap *preprocessing*. Kemudian, *data training* akan dibobotkan menggunakan metode *Term Frequency-Inverse Document Frequency* (TF-IDF) [8]. Berikut merupakan sample data *training* yang digunakan.

Tabel 1. *Sample Data Training*

No	Kronologi	Pasal
1	korban ida royanti lepas baju telanjang turun celana onani	Pasal 27 ayat 1
2	orang orang main judi taruh sport slot casino togel online	Pasal 27 ayat 2
3	ujar benci presiden joko widodo jokowi milik akun nama imam	Pasal 27 ayat 3
4	sky motor sedia uang rp puluh juta rupiah bayar hapus postingan	Pasal 27 ayat 4
5	orang lawan hukum pakai nama palsu martabat palsu tipu muslihat	Pasal 28 ayat 1
6	maksud busuk ajar agama islam postingan kalimat makan tu islam	Pasal 28 ayat 2
7	gelap mata ambil kain sarung Kasur lanta lilit jerat leher	Pasal lainnya

1. Tahap pertama dalam *weighting* adalah menghitung nilai *Term-Frequency* (TF) yang merupakan jumlah kata dalam dokumen *data training*, yang akan digunakan dalam menghitung nilai IDF [9].

2. Selanjutnya adalah menghitung nilai *Inverse Document Frequency* (IDF) dalam tahap *weighting* dengan rumus berikut [10].

$$IDF = \log\left(\frac{D}{DF}\right)$$

Keterangan:

D = Jumlah dokumen

DF = Dokumen frekuensi

3. Tahap ketiga dalam *weighting* adalah mengalikan nilai TF dan nilai IDF untuk menghitung nilai TF-IDF, lalu melakukan normalisasi nilai tersebut. Proses ini dilakukan pada 7 dokumen *sample data training*.
 4. Langkah berikutnya adalah mencari nilai untuk menormalisasikan angka dengan rumus berikut.

$$||V|| = \sqrt{V1^2 + V2^2 + V3^2 + \dots + Vn^2}$$

Keterangan:

V = nilai fitur pada satu dokumen

Hasil nilai normalisasi dari tiap dokumen contoh adalah sebagai berikut:

Tabel 2. Nilai Normalisasi

No	Kalimat	Nilai
1	korban ida royanti lepas baju telanjang turun celana onani	7.158883083
2	orang orang main judi taruh sport slot casino togel online	7.826234365
3	ujar benci presiden joko widodo jokowi milik akun nama imam	7.427872611
4	sky motor sedia uang rp puluh juta rupiah bayar hapus postingan	7.801774946
5	orang lawan hukum pakai nama palsu martabat palsu tipu muslihat	8.049309091
6	maksud busuk ajar agama islam postingan kalimat makan tu islam	8.158559498

5. Langkah terakhir adalah melakukan normalisasi pada nilai perkalian TF-IDF. Karena nilai yang dihasilkan dapat melebihi 1 dan dokumentasi resmi TF-IDF mensyaratkan nilai antara 0-1, maka perlu dilakukan normalisasi menggunakan rumus sebagai berikut:

$$TF-IDF = TF-IDF / ||V||$$

Dengan rumus tersebut didapatkan hasil akhir nilai TF-IDF pada *dataset* contoh adalah sebagai berikut :

Tabel 3. Nilai Final TF-IDF Dokumen 1

No	Kata	TF-IDF
1	korban	0.333333333
2	ida	0.333333333
3	royanti	0.333333333
4	lepas	0.333333333
5	baju	0.333333333
6	telanjang	0.333333333
7	turun	0.333333333
8	celana	0.333333333
9	onani	0.333333333

3.1.3 Learning

Pada tahap *Learning*, dilakukan proses perhitungan model klasifikasi dengan menggunakan algoritma Naive Bayes. Data *training* yang telah melalui tahap *pre-processing* dan *weighting* akan digunakan sebagai bahan pembelajaran pada tahap testing untuk menentukan kronologi masuk ke pasal dan ayat keberapa. Sama halnya dengan *data training*, *data testing* juga telah melalui tahap *pre-processing*. Rincian data uji dapat dilihat pada tabel berikut:

Tabel 4. Data *Testing*

No	Kalimat
1	menang taruh judi togel online menang uang taruh

Pada *dataset* ini kata yang akan di klasifikasi adalah kata yang ada pada tabel 4. Langkah selanjutnya adalah menghitung prior probabilitas dari tiap kata kepada tiap label yang tersedia menggunakan rumus dibawah ini [11].

$$P(t_n|c) = \frac{w_{tc} + 1}{(\sum w'_{vw'ct}) + B'}$$

Keterangan:

w_{tc} = Nilai pembobotan tf-idf atau W dari term t di kategori c.

$\sum w'_{vw'ct}$ = Jumlah total W dari keseluruhan term yang berada di kategori c.

B' = Jumlah W kata unik (nilai idf tidak dikali dengan tf) pada seluruh dokumen

Proses perhitungan prior dapat dilihat sebagai berikut:

$$p(\text{menang}|27\text{ayat1}) = 0+1/3+151.51 = 0.006472073$$

$$p(\text{taruh}|27\text{ayat1}) = 0+1/3+151.51 = 0.006472073$$

$$p(\text{judi}|27\text{ayat1}) = 0+1/3+151.51 = 0.006472073$$

$$p(\text{togel}|27\text{ayat1}) = 0+1/3+151.51 = 0.006472073$$

$$p(\text{online}|27\text{ayat1}) = 0+1/3+151.51 = 0.006472073$$

$$p(\text{uang}|27\text{ayat1}) = 0+1/3+151.51 = 0.006472073$$

$$p(\text{menang}|27\text{ayat2}) = 0+1/2.945479565+151.51 = 0.006474358$$

$$p(\text{taruh}|27\text{ayat2}) = 0.304909647+1/2.945479565+151.51 = 0.008448452$$

Dengan nilai probabilitas kata yang didapat maka dapat dilanjutkan dengan menghitung probabilitas dokumen dengan tiap-tiap kelas menggunakan rumus berikut [12].

$$P(c|\text{term dokumen } d) = P(c) \times P(t_1|c) \times P(t_2|c) \times P(t_3|c) \times \dots \times P(t_n|c)$$

Keterangan:

$P(c)$ = Probabilitas *prior* dari dokumen c.

t_n = Kata dokumen d ke-n.

$P(c|\text{term dokumen } d)$ = Probabilitas suatu dokumen termasuk kelas c.

$P(t_n|c)$ = Probabilitas kata ke-n dengan diketahui kelas c

Contoh perhitungan probabilitas data testing pada tabel 3 adalah sebagai berikut:

$$p(27\text{ayat1}|\text{data_testing}) = 1/7 \times 0.006472073^6 = 1.04993 \times 10^{-14}$$

$$p(27\text{ayat2}|\text{data_testing}) = 1/7 \times 0.006474358^2 \times 0.008448452^4 = 3.05073 \times 10^{-14}$$

$$p(27\text{ayat3}|\text{data_testing}) = 1/7 \times 0.00646546^6 = 1.04351 \times 10^{-14}$$

$$p(27\text{ayat4}|\text{data_testing}) = 1/7 \times 0.006459007^5 \times 0.008434595 = 1.35455 \times 10^{-14}$$

$$p(28\text{ayat1}|\text{data_testing}) = 1/7 \times 0.006477781^6 = 1.0555 \times 10^{-14}$$

$$p(28\text{ayat2}|\text{data_testing}) = 1/7 \times 0.006477305^6 = 1.05504 \times 10^{-14}$$

$$p(\text{lainnya}|\text{data_testing}) = 1/7 \times 0.006465283^6 = 1.04334 \times 10^{-14}$$

Dari nilai yang didapat dapat dilihat bahwa probabilitas data testing pada tabel 3 terhadap label 27ayat2 lebih tinggi, maka dapat dinyatakan bahwa data testing pada tabel 3 akan diklasifikasikan dengan label 27ayat2.

Berdasarkan perhitungan *dataset* kronologi asli dengan algoritma Naive Bayes menggunakan *Jupyter Notebook* pada klasifikasi jenis pasal UU ITE maka didapat akurasi sebesar 0.8 atau 80%. Berikut merupakan hasil akurasi yang didapat.

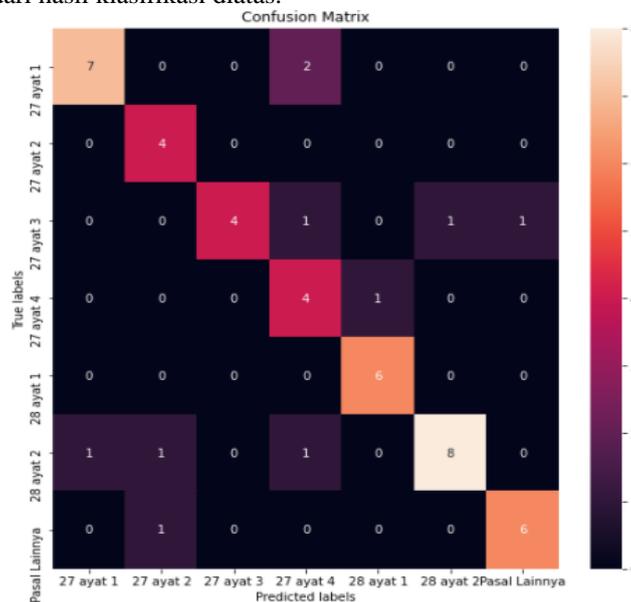
	precision	recall	f1-score	support
pasal 27 ayat 1	0.88	0.78	0.82	9
pasal 27 ayat 2	0.67	1.00	0.80	4
pasal 27 ayat 3	1.00	0.57	0.73	7
pasal 27 ayat 4	0.50	0.80	0.62	5
pasal 28 ayat 1	0.86	1.00	0.92	6
pasal 28 ayat 2	0.89	0.73	0.80	11
pasal lainnya	0.86	0.86	0.86	7
accuracy			0.80	49
macro avg	0.81	0.82	0.79	49
weighted avg	0.84	0.80	0.80	49

Gambar 2. Hasil Akurasi Model

Berdasarkan gambar diatas diketahui bahwa nilai *precision* yang didapat dari seluruh *data training* dalam rentang 0.67 sampai dengan 1.0. Dan diketahui juga nilai dari *recall* yang didapat pada seluruh *data training* dalam rentang 0.57 dan 1.0. Serta nilai *f1-score* yang didapat dari seluruh *data training* berada dalam rentang 0.62 sampai 0.92, Maka dari ini didapatlah hasil akurasi berupa 0.80 atau 80%.

3.1.4 Evaluasi

Teknik yang digunakan untuk melakukan evaluasi dalam klasifikasi pada penelitian ini adalah dengan menghitung *recall*, *precision*, dan *f-measure*. Teknik ini menggunakan *confusion matrix* sebagai acuan perhitungan [13]. Berikut merupakan *confusion matrix* dari hasil klasifikasi diatas.



Gambar 3. Confusion Matrix

Terlihat bahwa pada Pasal 27 ayat 1 terdapat skor *precision*, *recall* dan *f1-score*, skor tersebut didapat melalui tabel *Confusion Matrix* diatas. Berikut merupakan perhitungan *Confusion Matrix* tiap label.

a. Pasal 27 ayat 1

1. *Precision*

$$= TP / (TP + FP) = 7 / (7 + 1) = 0.88$$

2. *Recall*

$$= TP / (TP + FN) = 7 / (7 + 2) = 0.78$$

3. *f1-score*

$$= 2 * precision * recall / (precision + recall) = 2 * 0.88 * 0.78 / (0.88 + 0.78) = 0.82$$

b. Pasal 27 ayat 2

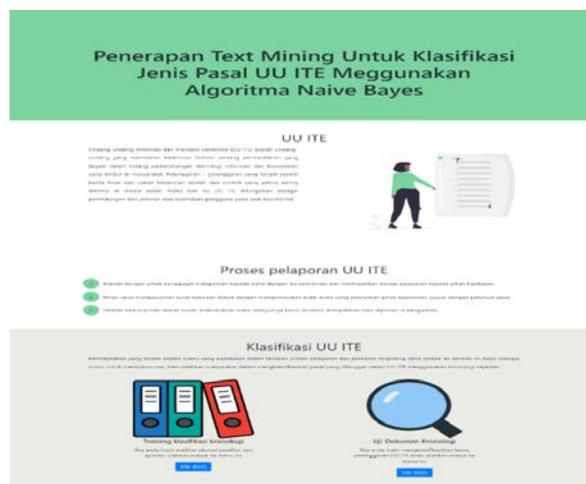
1. *Precision* = 0.67
 2. *Recall* = 1.0
 3. *f1-score* = 0.80
- c. Pasal 27 ayat 3
1. *Precision* = 1.0
 2. *Recall* = 0.57
 3. *f1-score* = 0.73
- d. Pasal 27 ayat 4
1. *Precision* = 0.86
 2. *Recall* = 0.8
 3. *f1-score* = 0.62
- e. Pasal 28 ayat 1
1. *Precision* = 0.5
 2. *Recall* = 1.0
 3. *f1-score* = 0.92
- f. Pasal 28 ayat 2
1. *Precision* = 0.89
 2. *Recall* = 0.73
 3. *f1-score* = 0.80
- g. Pasal Lainnya
1. *Precision* = 0.86
 2. *Recall* = 0.86
 3. *f1-score* = 0.86
- h. Akurasi
- Skor akurasi = jumlah data benar/banyaknya data latihan
= 39/49
= 0.80

3.2 Implementasi

Berikut ini adalah hasil implementasi pada sistem yang dibangun untuk mengklasifikasi pasal UU ITE.

a. Halaman beranda

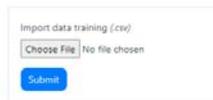
Di halaman ini, pengguna dapat menemukan informasi mengenai definisi UU ITE, proses pelaporan UU ITE, opsi untuk melatih klasifikasi kronologi dan melihat laporan akurasi aplikasi, serta opsi untuk menguji dokumen dan mengklasifikasikan teks kronologi pelanggaran UU ITE.



Gambar 4. Beranda

- b. Halaman *training* klasifikasi kronologi

Untuk melihat report akurasi klasifikasi, pengguna dapat mengunggah data training atau data latih yang telah disediakan pada halaman ini.



Gambar 5. Halaman *training* klasifikasi kronologi

- c. Halaman hasil *training* klasifikasi kronologi

Pada halaman tersebut, pengguna dapat melihat laporan akurasi klasifikasi pada model *machine learning* untuk mengetahui seberapa akurat aplikasi yang telah dibuat. Selain itu, terdapat tombol untuk kembali ke halaman utama.



Naive Bayes Classification Report

	Precision	Recall	F1-score	Support
Pasal 27 ayat 1	0.88	0.78	0.82	9
Pasal 27 ayat 2	0.67	1.00	0.80	4
Pasal 27 ayat 3	1.00	0.57	0.73	7
Pasal 27 ayat 4	0.50	0.80	0.62	5
Pasal 28 ayat 1	0.86	1.00	0.92	6
Pasal 28 ayat 2	0.89	0.73	0.80	11
Pasal lainnya	0.86	0.86	0.86	7

[Kembali](#)

Gambar 6. Halaman hasil *training* klasifikasi kronologi

4. KESIMPULAN

Kesimpulan pada penelitian ini membuktikan bahwa model klasifikasi *Text Mining* dengan algoritma Naive Bayes efektif dalam mengklasifikasikan kronologi pelanggaran UU ITE dengan menghitung probabilitas pasal tertinggi dengan akurasi model sebesar 80%, yang dihitung berdasarkan jumlah dataset kronologi pelanggaran UU ITE yang terdiri dari 245 dokumen dengan 196 data latih dan 49 data uji. Hasil akurasi tersebut menunjukkan bahwa model *Machine Learning* memiliki performa yang baik dan dapat membantu masyarakat umum dalam mengklasifikasikan pelanggaran pasal UU ITE. Dengan menggunakan bahasa pemrograman Python dan *mini-framework flask*, model *Machine Learning* berhasil mengeksekusi dokumen dari *dataset* kronologi pelanggaran UU ITE melalui uji dokumen. Halaman *web* dapat memberikan informasi mengenai probabilitas pasal tertinggi yang kemungkinan besar dilanggar, baik dari *file* yang diunggah maupun memasukkan teks secara manual.

DAFTAR PUSTAKA

- [1] A. Deolika, Kusriani, and E. T. Luthfi, "Analisis Pembobotan Kata Pada Klasifikasi Text Mining," *J. Teknol. Inf.*, vol. 3, no. 2, 2019.
- [2] M. Simanjuntak, Nurfalinda, and M. R. Rathomi, "Penerapan Metode Naive Bayes Untuk Memprediksi Status Kehadiran Masyarakat Dalam Pemilihan GuberNUR," *Student Online J. Umr.*, vol. 3, no. 1, 2022.
- [3] A. Ridwan, "Penerapan Algoritma Naive Bayes Untuk Klasifikasi Penyakit Diabetes Mellitus," *J. Sist. Komput. dan Kecerdasan Buatan*, vol. IV, no. 1, 2020.
- [4] Y. D. Wijaya, "Sistem Informasi Penjualan Tiket Wisata Berbasis Web Menggunakan Metode Waterfall," *Pros. Semin. Nas. Teknol. Inf. Dan Komun.*, vol. 2, no. 1, 2019.
- [5] M. F. Rifai, H. Jatnika, and B. Valentino, "Penerapan Algoritma Naive Bayes Pada Sistem Prediksi Tingkat Kelulusan Peserta Sertifikasi Microsoft Office Specialist (MOS)," *PETIR (Jurnal Pengkaj. Dan Penerapan Tek. Inform.*, vol. 12, no. 2, 2019.
- [6] A. Kurniawan, Indriati, and S. Adinugroho, "Analisis Sentimen Opini Film Menggunakan Metode Naive Bayes dan Lexicon Based Features," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 3, no. 9, 2019.
- [7] L. A. Andika, P. A. N. Azizah, and Respatiwan, "Analisis Sentimen Masyarakat terhadap Hasil Quick Count Pemilihan Presiden Indonesia 2019 pada Media Sosial Twitter Menggunakan Metode Naive Bayes Classifier," *Indones. J. Appl. Stat.*, vol. 2, no. 1, 2019.
- [8] D. F. Zhafira, B. Rahayudi, and Indriati, "No Title Analisis Sentimen Kebijakan Kampus Merdeka Menggunakan Naive Bayes dan Pembobotan TF-IDF Berdasarkan Komentar pada Youtube," *J. Sist. Informasi, Teknol. Informasi, dan Edukasi Sist. Inf.*, vol. 2, no. 1, 2021.
- [9] Y. Yunefri, Y. E. Fadrial, and Sutejo, "Chatbot Pada Smart Cooperative Oriented Problem Menggunakan Natural Language Processing Dan Naive Bayes Classifier," *J. Inf. Technol. Comput. Sci.*, vol. 4, no. 2, 2021.
- [10] S. W. U. Vitandy, A. A. Supianto, and F. A. Bachtiar, "Analisis Sentimen Evaluasi Kinerja Dosen menggunakan Term Frequency_Inverse Document Frequency dan Naive Bayes Classifier," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 3, no. 6, 2019.
- [11] L. Mayasari and D. Indarti, "Klasifikasi Topik Tweet Mengenai Covid Menggunakan Metode Multinomial Naive Bayes Dengan Pembobotan Tf-Idf," *J. Ilm. Inform. Komput.*, vol. 27, no. 1, 2022.
- [12] G. Subroto, N. Sulistiyowati, and A. A. Ridha, "Klasifikasi Jenis Kekerasan Pada Perempuan Dan Anak Dengan Algoritma Multinomial Naive Bayes," *INTECOMS J. Inf. Technol. Comput. Sci.*, vol. 5, no. 1, 2022.
- [13] Muljono, D. P. Artanti, A. Syukur, A. Prihandono, and D. R. I. M. Setiadi, "Analisa Sentimen Untuk Penilaian Pelayanan Situs Belanja Online Menggunakan Algoritma Naive Bayes," in *Konferensi Nasional Sistem Informasi (KNSI) 2018*, 2018.