Klasifikasi Teks Soal Ujian Berbahasa Indonesia Berdasarkan Ranah Kognitif Taksonomi Bloom

1st Justisio Yan Prawira Adam
Fakultas Informatika
Universitas Telkom
Bandung, Indonesia
prawiraadam@student.telkomuniversity.ac.id

2nd Ade Romadhony
Fakultas Informatika
Universitas Telkom
Bandung, Indonesia
aderomadhony@telkomuniversity.ac.id

3rd Erwin Budi Setiawan
Fakultas Informatika
Universitas Telkom
Bandung, Indonesia
erwinbudisetiawan@telkomuniversity.ac.id

Abstrak
Ujian tertulis merupakan bentuk ujian yang paling umum digunakan untuk mengukur capaian belajar siswa, baik dari jenjang SD, SMP ataupun SMA. Tingkat kesulitan dari ujian dapat bervariasi antar soal, sehingga hasil ujian dari siswa dapat dianalisis lebih jauh dengan melihat pada tingkat kesulitan apa siswa mampu dan tidak mampu menjawab dengan benar. Taksonomi Bloom memiliki ranah kognitif yang dapat dijadikan acuan dalam menentukan tingkat kesulitan dari soal ujian. Dalam ranah kognitif tersebut, terdapat 6 kelas berbeda yang secara urut diantaranya mengingat, memahami, menerapkan, menganalisa, mengevaluasi dan mencipta. Pada Tugas Akhir ini, latihan soal diklasifikasikan ke dalam 6 kelas dari ranah kognitif Taksonomi Bloom. Data yang digunakan berupa teks soal dalam Bahasa Indonesia dari jenjang pendidikan Sekolah Dasar, Sekolah Menengah Pertama, dan Sekolah Menengah Atas. Metode yang digunakan adalah Support Vector Machine dan Naive Bayes, karena terbukti pada penelitian sebelumnya mampu menghasilkan performa yang cukup baik dalam melakukan klasifikasi pada bidang yang sama. Selain itu, ekstraksi fitur dilakukan menggunakan TF-IDF yang telah dimodifikasi berdasarkan nilai bobot dari POS Tag. Metode ekstraksi fitur tersebut terbukti memiliki performa yang lebih baik dibandingkan TF-IDF reguler pada penelitian sebelumnya.

Kata kunci : Taksonomi Bloom, Support Vector Machine, Naïve Bayes, TF-IDF

Abstract
Written exam is the common type of exam to measure student's learning achievement, from Elementary School, Junior High School and Senior High School. The difficulty level of the exam may vary from one question to another, so that the exam result could be analyzed further by observing in which difficulty students can or cannot answer correctly. Bloom's Taxonomy has a cognitive domain that can be used as reference for determining the exam questions' difficulty. The cognitive domain has 6 different classes which in order of them are remember, understand, apply, analyze, evaluate, create. This final project aims to do a classification of exam questions into 6 classes of Bloom's Taxonomy cognitive field. The data used will be a text in Bahasa Indonesia from Elementary School, Junior High School and Senior High School. Methods used in this final project are Support Vector Machine and Naive Bayes, since both are proven in previous study to perform well in the same task. As for the feature extraction, this final project will be using TF-IDF that has been modified based on weight value from POS Tag. Such feature extraction method is already proven in previous study to perform better than the regular TF-IDF.

Keywords: Bloom Taxonomy, Support Vector Machine, Naïve Bayes, TF-IDF

I. PENDAHULUAN

A. Latar Belakang

Dalam lingkup pendidikan, ujian tulis menjadi hal yang umum diberikan untuk menguji capaian belajar pada siswa dan memiliki peran yang penting dalam mengidentifikasi kemampuan kognitif [3]. Selain itu, identifikasi kemampuan kognitif siswa perlu dilakukan untuk memastikan pemahaman siswa atas apa yang telah diajarkan. Hal tersebut dapat dilakukan dengan memberikan soal ujian dengan tingkat kesulitan yang mengacu pada Taksonomi Bloom [1].

Taksonomi Bloom diperkenalkan oleh Benjamin Bloom pada tahun 1956 dengan tujuan untuk mengklasifikasikan soal-soal yang ada pada sistem pendidikan [4]. Taksonomi Bloom memiliki 3 domain di antaranya kognitif, afektif dan psikomotorik dimana domain kognitif berfokus pada kemampuan berpikir seseorang [4]. Kemudian, domain kognitif terbagi menjadi 6 tingkatan yang diurutkan berdasarkan kompleksitasnya, diantaranya pengetahuan, pemahaman, penerapan, penguraian, pemaduan dan penilaian [2][4]. Keenam tingkatan tersebut dapat dijadikan acuan untuk menentukan tingkat kesulitan ujian yang diberikan, sehingga hasil ujian dapat digunakan sebagai patokan untuk mengukur kemampuan kognitif siswa [1].

Klasifikasi soal ujian menggunakan Taksonomi Bloom dapat dilakukan secara manual oleh pengajar.

Akan tetapi, menurut Kusuma et al. [1] hal tersebut dapat memerlukan waktu yang cukup banyak. Selain itu, klasifikasi secara manual rentan akan perbedaan persepsi antar pengajar. Hal tersebut dapat memicu terjadinya perbedaan dari hasil klasifikasi [3].

Berkaitan dengan masalah diatas, pada Tugas Akhir ini akan dilakukan klasifikasi otomatis pada soal ujian menggunakan 2 metode pembelajaran mesin, yaitu Support Vector Machine (SVM) dan Naive Bayes (NB). Kedua metode tersebut dipilih karena mampu menghasilkan performa yang baik pada penelitian sebelumnya yang dilakukan oleh Patil et al. [5] dan Aninditya et al. [3]. Data yang digunakan bersifat tekstual yang berisi 600 latihan soal dalam Bahasa Indonesia untuk mata pelajaran Bahasa Indonesia, Ilmu Pengetahuan Alam dan Matematika dari tingkat Sekolah Dasar (SD), Sekolah Menengah Pertama (SMP) dan Sekolah Menengah Atas (SMA). Mata pelajaran IPA untuk SMP dan SMA mencakup soal tentang biologi, fisika dan kimia.

Penggabungan beberapa mata pelajaran ke dalam sebuah dataset bertujuan supaya dataset berisi soal dengan karakteristik yang berbeda-beda. Kemudian akan dilihat apakah algoritma klasifikasi tetap mampu melakukan klasifikasi dengan baik.

B. Topik dan Batasannya

Topik yang dibahas dalam tugas akhir ini adalah bagaimana melakukan klasifikasi teks berdasarkan Taksonomi Bloom dengan metode SVM dan NB serta mengukur performansi dari metode yang digunakan.

Batasan masalah pada tugas akhir ini adalah sebagai berikut: Pertama, data yang digunakan berupa teks Berbahasa Indonesia. Kedua, mata pelajaran yang digunakan hanya Bahasa Indonesia, Matematika dan Ilmu Pengetahuan Alam dari jenjang pendidikan SD, SMP, dan SMA. Ketiga, klasifikasi yang dilakukan hanya untuk menentukan tingkatan kognitif yang sesuai dari sebuah soal berdasarkan Taksonomi Bloom.

C. Tujuan dan Manfaat

Tujuan dari tugas akhir ini adalah melakukan klasifikasi teks Berbahasa Indonesia dengan menggunakan metode SVM dan NB serta mengukur performansi dari masing-masing metode. Mengenai manfaat dari penelitian ini, hasil ujian bisa digunakan untuk menganalisis capaian belajar siswa berdasarkan jawaban yang mereka berikan [5]. Kemudian, pengajar juga dapat menyesuaikan pertanyaan yang dibuat untuk ujian sehingga bisa mengukur pemahaman siswa berdasarkan capaian belajarnya [3]. Lebih lanjut, hasil klasifikasi dapat dibuat menjadi alur pembelajaran untuk masing-masing siswa, sehingga ke depannya siswa mengetahui urutan dari materi-materi yang harus dipelajari [7].

D. Organisasi Tulisan

Struktur penulisan dari tugas akhir ini disusun sebagai berikut: Bagian pertama berisi pendahuluan terkait tugas akhir ini. Bagian kedua menjelaskan studi yang terkait dengan tugas akhir ini. Bagian ketiga menjelaskan pemodelan dari sistem yang dibangun dan data yang digunakan. Bagian keempat menjelaskan hasil dan evaluasi hasil pengujian yang telah dilakukan pada bagian ketiga. Kemudian, pada bagian terakhir menjelaskan kesimpulan dan saran berdasarkan hasil pengujian yang dilakukan pada tugas akhir ini.

II. KAJIAN TEORI

Saat tugas akhir ini disusun, terdapat beberapa penelitian yang telah dilakukan pada bidang yang sama. Penelitian yang dilakukan oleh Kusuma et al. [1] bertujuan untuk mengajukan pendekatan baru dalam melakukan klasifikasi soal ujian berbahasa Indonesia yang mengacu pada Taksonomi Bloom. Metode yang digunakan berupa SVM dengan kernel linear, dan diaplikasikan pada dataset yang berisi 130 soal berbahasa Indonesia. Dataset yang digunakan mencakup 5 mata pelajaran pada tingkat sekolah dasar. Fitur leksikal dan sintaktik digunakan untuk ekstraksi fitur. Penelitian ini berhasil mendapatkan rata-rata nilai akurasi sebesar 88,6%. Penelitian lainnya dilakukan oleh Aninditya et al. [2] menggunakan metode NB dalam melakukan klasifikasi soal berdasarkan tingkatan kognitif dari Taksonomi Bloom. Ekstraksi fitur dilakukan dengan metode Term Frequency — Inverse Document Frequency (TF-IDF). Dataset yang digunakan berupa naskah soal ujian semester berbahasa Indonesia dari Departemen Sistem Informasi Universitas Telkom. Setiap soal pada dataset tersebut dilabeli *Lower Order* (LO) atau *High Order* (HO). Hasil dari penelitian ini adalah NB dengan N-Gram TF-IDF mampu menghasilkan nilai *precision* sebesar 85%.

Penelitian terkait klasifikasi teks dengan latihan soal berbahasa inggris dilakukan oleh Patil et al. [5] menggunakan metode SVM dan K-Nearest Neighbor (KNN). Dataset yang digunakan berupa 1000 pertanyaan yang berkaitan dengan kursus sistem operasi, dan dilabeli berdasarkan 6 tingkatan kognitif Taksonomi Bloom. Hasil dari penelitian ini adalah performansi metode SVM mengungguli KNN dengan nilai akurasi masing-masing sebesar 0.923 dan 0.666. Sementara itu, penelitian yang dilakukan oleh Mohammed et al. [12] menggunakan metode ekstraksi fitur yang dimodifikasi dari TF-IDF dan word2vec dalam melakukan klasifikasi berdasarkan Taksonomi Bloom. Dataset yang digunakan berupa teks yang berisi pertanyaan terbuka dengan 6 label berbeda, dan terdapat 2 dataset berbeda yang digunakan pada penelitian ini. Dataset pertama dikumpulkan dari beberapa situs, buku dan penelitian sebelumnya sebanyak 141 pertanyaan, sementara dataset kedua bersumber dari Yahya et al. (2012) berupa pertanyaan terbuka sebanyak 600 buah. TF-

IDF dimodifikasi dengan cara dikalikan dengan bobot yang menyesuaikan dengan POSTag yang dimiliki tiap kata, yang kemudian diberi nama TFPOS-IDF. Kemudian word2vec dan TFPOS-IDF, yang diberi nama W2V-TFPOSIDF, akan dikombinasikan sehingga akan menghasilkan satu vektor. Pengujian yang dilakukan dengan algoritma SVM menunjukkan W2V-TFPOSIDF mampu menghasilkan F1-measure yang paling tinggi pada kedua dataset, diikuti dengan TFPOS-IDF dan terakhir TF-IDF.

Penggunaan algoritma SVM didasari oleh penelitian Kusuma et al. dan Patil et al. [1, 5], namun yang menjadi pembeda pada Tugas Akhir ini adalah metode ekstraksi fitur yang digunakan yaitu menggunakan TFPOS-IDF. Kemudian, pemilihan algoritma NB didasari oleh penelitian Aninditya et al. [2] dengan pembeda berupa dataset yang digunakan yaitu dibagi menjadi 6 kelas daripada 2 kelas berbeda. Selain itu, metode ekstraksi fitur berupa TFPOS-IDF yang diambil dari penelitian Mohammed et al. [12] akan diuji performansinya pada algoritma NB. Hal ini dikarenakan algoritma NB tidak diuji menggunakan TFPOS-IDF pada penelitian tersebut.

A. Taksonomi Bloom

Taksonomi Bloom merupakan kerangka konsep kemampuan berpikir yang mengidentifikasi kompetensi dari tingkat paling rendah hingga tingkat paling tinggi [2]. Terdapat tiga ranah kemampuan intelektual dalam Taksonomi Bloom diantaranya:

a. Kognitif, aspek yang ditekankan seperti keterampilan berfikir dan pengetahuan.
b. Afektif, ranah ini mencakup perasaan, motivasi dan sikap sebagai perilaku yang terkait dengan emosi.
c. Psikomotorik, aspek yang ditekankan berupa keterampilan motorik, seperti berenang dan mengoperasikan mesin.

Pada umumnya, ranah kognitif dapat diukur dengan membuat evaluasi berupa ujian tertulis. Berdasarkan hal tersebut, Tugas Akhir ini akan berfokus pada ranah kognitif, karena dataset yang digunakan berbentuk soal ujian tertulis. Dalam ranah kognitif, terdapat 6 tingkatan, dimana 3 tingkatan pertama disebut *Lower Order Thinking Skills* (LOTS), sedangkan tiga level berikutnya *Higher Order Thinking Skill* (HOTS). Siswa harus melalui tingkatan LOTS terlebih dahulu sebelum mulai memasuki tingkat HOTS. Tingkatan tersebut diantaranya adalah sebagai berikut:

a. Pengetahuan (C1), kompetensi dalam menyebutkan atau menjelaskan kembali terkait hal yang sudah dipelajari.
b. Pemahaman (C2), kompetensi dalam menginterpretasi dan menyatakan kembali berdasarkan pemahaman sendiri serta memahami instruksi / masalah yang diberikan.
c. Penerapan (C3), kompetensi dalam mengaplikasikan konsep ke dalam situasi yang baru.
d. Analisa (C4), kompetensi dalam memecahkan suatu konsep ke dalam beberapa komponen sehingga memahami korelasi komponen - komponen terhadap konsep tersebut secara utuh.
e. Sintesa (C5), kompetensi dalam menciptakan struktur yang baru dari komponen - komponen yang tersedia.
f. Evaluasi (C6), kompetensi dalam menilai dan mengevaluasi sesuatu berdasarkan kriteria tertentu.

Pada Tugas Akhir ini, Taksonomi Bloom yang digunakan adalah versi revisi [16]. Perbedaan versi ini dibandingkan versi sebelumnya adalah perubahan nama untuk setiap tingkatan kognitif. Selain itu, dilakukan penukaran untuk tingkatan C5 dan C6 pada versi sebelumnya, sehingga urutan tingkatan kognitif menjadi seperti berikut: mengingat (C1), memahami (C2), menerapkan (C3), menganalisa (C4), Mengevaluasi (C5) dan Mencipta (C6). Pada setiap tingkatan, terdapat beberapa kata kunci yang dapat membantu menggambarkan karakteristik dari masing-masing tingkatan. Daftar kata kunci diambil dari peneilitian Setyaningsih et al. [22] dan dapat dilihat pada Tabel 1.

TABEL 1 *Contoh Kata Kerja Tingkatan Kognitif*

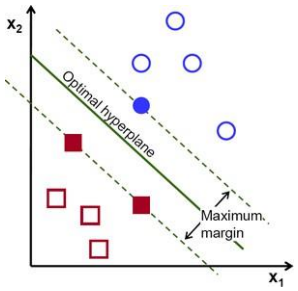
Tingkatan Kognitif	Contoh Kata Kerja
mengingat (C1)	menamai, menulis, mengutip, menyebutkan, menghafal, melabeli, mendaftar, menunujukkan, memasangkan, mengidentifikasi, menandai, membaca, menyadari, mencatat, mengulang, memilih.
memahami (C2)	mengartikan, menerangkan, menyatakan kembali, menjelaskan, menguraikan, menterjemahkan, menginterpretasikan, menafsirkan, mendiskusikan, menyeleksi, mendeteksi, melaporkan, mengelompokkan, memberi, menduga.
menerapkan (C3)	menerapkan, menggunakan, memilih, melaksanakan, mempraktekkan, mengubah, mendemonstrasikan, memodifikasi, menginterpretasikan, membuktikan, menunjukkan, menggambarkan, mengoperasikan, memulai,

	menjalankan, memprogramkan.
menganalisa (C4)	membandingkan, mengkaji ulang, membedakan, mengkontraskan, memecah ke dalam beberapa bagian, menunjukan korelasi antar variabel, memisahkan, menyisihkan, menghubungkan, mempertimbangkan, menduga.
mengevaluasi (C5)	menilai, membenarkan, mempertahankan, menyalahkan, mengkaji ulang, mempertahankan, mendukung, menyeleksi, mengevaluasi, menjustifikasi, mengkritik, mengecek, memprediksi.
mencipta (C6)	membangun, merakit, merancang, membuat, menemukan, menciptakan, memperoleh, mengembangkan, memformulasikan, membentuk, melengkapi, melakukan, mendisain, menghasilkan karya.

B. Support Vector Machine (SVM)

Support Vector Machine (SVM) merupakan algoritma untuk *supervised learning* yang dapat digunakan untuk mengklasifikasikan data dengan dimensi yang besar [6, 7]. Metode ini diperkenalkan oleh Vapnik untuk mengklasifikasikan data ke dalam 2 kelas yang berbeda [8]. Walaupun demikian, SVM juga dapat digunakan untuk mengklasifikasikan data ke dalam beberapa kelas yang berbeda [6].

Pada SVM, setiap data akan dipetakan sebagai titik yang kemudian ditempatkan pada ruang berdimensi *n* (jumlah fitur pada data) yang kemudian akan dipisahkan secara linear menggunakan *hyperplane*. Akan terdapat banyak *hyperplane* yang dapat digunakan untuk memisahkan data, oleh karena itu *hyperplane* yang dipilih adalah *hyperplane* dengan margin yang paling besar dari titik data terjauh masing-masing kelas [9]. Hal tersebut dilakukan untuk memastikan algoritma mampu memberikan klasifikasi yang tepat pada titik data yang baru. Jika data tidak dapat dipisahkan secara linear, maka data akan ditempatkan pada dimensi yang lebih besar dengan bantuan fungsi kernel.



GAMBAR 1. *Ilustrasi SVM [9]*

Pada Tugas Akhir ini, SVM akan diimplementasikan menggunakan bahasa pemrograman Python dengan bantuan *library* dari Scikit Learn.

C. Naïve Bayes

Naive Bayes (NB) merupakan salah satu algoritma *supervised learning* yang mengaplikasikan teorema Bayes dengan asumsi 'naif' berupa tidak adanya keterkaitan pada setiap pasang fitur yang ada [10]. NB umum digunakan untuk melakukan klasifikasi pada dokumen dan deteksi spam. Rumus yang digunakan adalah sebagai berikut:

$$\hat{y} = \arg max\ P(y) \prod_{i=1}^{n} P(x_i \mid y) \qquad (1)$$

x merupakan fitur pada data, sementara y adalah kelas dari data. Algoritma ini akan menentukan kelas dengan mengambil nilai yang paling besar dari y setelah menghitung nilai probabilitas dari sebuah data untuk semua kelas yang ada. Yang membedakan *classifier* dari NB adalah asumsi yang dibuat terkait distribusi dari $P(x_i \mid y)$

Walaupun dengan asumsi yang disederhanakan, NB mampu memberikan performa yang baik dalam kasus di dunia nyata. Selain itu, proses klasifikasi NB dilakukan dengan sangat cepat jika dibandingkan dengan algoritma lain yang lebih mutakhir. Akan tetapi, NB tidak mampu menghasilkan performa yang baik sebagai *estimator* [10].

Pada Tugas Akhir ini, NB diimplementasikan menggunakan bahasa pemrograman Python dengan bantuan *library* dari Scikit Learn.

D. Term Frequency – Inverse Document Frequency (TF-IDF)

TF-IDF merupakan salah satu metode untuk melakukan pembobotan kata yang tergabung dari 2 istilah berbeda, yaitu *Term Frequency* (TF) dan *Inverse Document Frequency* (IDF). Tujuan dari metode ini adalah memberikan bobot untuk setiap kata, yang mengindikasikan seberapa penting kata tersebut dalam sebuah dokumen. Metode ini diperkenalkan oleh Sparck Jones dengan intuisi heuristik bahwa kata yang sering muncul dalam banyak dokumen yang berbeda tidak dapat dijadikan pembeda, sehingga harus diberikan bobot yang lebih

kecil dibandingkan kata yang sedikit kemunculannya pada dokumen [11]. Berikut adalah rumus yang digunakan untuk menghitung nilai TF-IDF:

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right) \quad (2)$$

$w_{i,j}$ merupakan bobot kata i pada dokumen j, N merupakan jumlah dokumen pada korpus, $tf_{i,j}$ merupakan TF dari kata i pada dokumen j, dan df_i merupakan *document frequency* dari kata i pada korpus.

E. Modifikasi TF-IDF (TFPOS-IDF)

Metode ini diperkenalkan oleh Mohammed M, et al. [12]. Tujuan dari metode ini adalah memberikan bobot pada kata yang berdasarkan tagar Part-of-Speech (POS)nya masing-masing. Berikut adalah nilai bobot yang optimal setelah dilakukan eksperimen:

$$w_{pos}(t) = \begin{cases} w_1 & if\ t\ is\ verb \\ w_2 & if\ t\ is\ noun\ or\ adjective \\ w_3 & otherwise \end{cases} \quad (3)$$

Urutan dari bobot tersebut adalah $w_1 > w_2 > w_3 > 0$ dengan asumsi $w_1 = 5$, $w_2 = 3$, $w_3 = 1$. Kemudian, berikut adalah rumus dari TFPOS:

$$TFPOS(t,d) = \frac{c(t,d) * w_{pos}(t)}{\sum_i c(t_i,d) * w_{pos}(t)} \quad (4)$$

Selanjutnya, TFPOS-IDF dapat dihitung dengan rumus sebagai berikut:

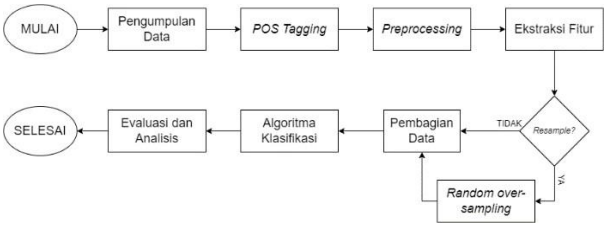
$$TFPOS - IDF(t,d) = TFPOS(t,d).IDF(t) \quad (5)$$

Hasil dari perhitungan diatas adalah berupa *sparse matrix*, atau vektor dengan dimensi besar. Untuk mengurangi kompleksitas dari segi komputasi, hasil tersebut akan dinormalisasi menggunakan L2 norm dengan rumus sebagai berikut:

$$||\vec{v}||_2 = \left(\sum_{i=1}^{n} |v_i|^2\right)^{\frac{1}{2}} \quad (6)$$

III. METODE

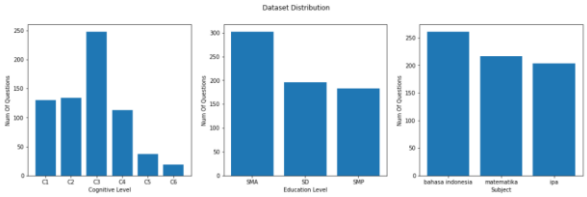
Sistem dibangun menggunakan bahasa pemrograman Python dengan alur seperti pada gambar 2.



GAMBAR 2 *Alur Kerja Sistem*

F. Pengumpulan Data

Dataset berupa teks latihan soal Berbahasa Indonesia dikumpulkan secara manual dari berbagai sumber daring seperti EduBox, Blog Ruangguru [14] dan penelitian oleh Syarifah et al. [15]. Dataset dilabeli secara manual berdasarkan tingkatan kognitif dalam Taksonomi Bloom. Data berhasil terkumpul sebanyak 682 soal dengan persebaran seperti pada gambar 3.



GAMBAR 3 *Distribusi Dataset*

G. *POS Tagging*

POS Tagging diimplementasikan menggunakan *library* FlairNLP [17]. Untuk Bahasa Indonesia, FlairNLP menyediakan corpus dan 2 *pre-trained word embedding* yang bersumber dari FastText Wikipedia dan *crawling* situs. Model dilatih menggunakan corpus dan kedua *pre-trained word embedding* dengan parameter learning_rate = 0.1, mini_batch_size = 32 dan max_epochs = 10. Skor yang dihasilkan model setelah dilatih tertera pada gambar 4.

```
Results:
- F-score (micro) 0.9251
- F-score (macro) 0.8646
- Accuracy 0.9251

By class:
              precision    recall  f1-score   support

        NOUN     0.8748    0.9152    0.8945      2511
       PROPN     0.9271    0.8996    0.9131      2162
       PUNCT     0.9982    1.0000    0.9991      1623
        VERB     0.9454    0.9342    0.9398      1261
         ADP     0.9334    0.9560    0.9446      1114
        PRON     0.9321    0.9798    0.9553       644
         ADJ     0.8614    0.7131    0.7803       488
         NUM     0.9352    0.9766    0.9554       384
       CCONJ     0.9783    0.9945    0.9863       362
         ADV     0.8433    0.7775    0.8090       346
         DET     0.9599    0.9120    0.9353       341
         AUX     0.9306    0.9956    0.9620       229
       SCONJ     0.8500    0.7887    0.8182       194
        PART     0.9149    0.9663    0.9399        89
         SYM     1.0000    1.0000    1.0000         6
           X     0.0000    0.0000    0.0000         2

   micro avg     0.9251    0.9251    0.9251     11756
   macro avg     0.8678    0.8631    0.8646     11756
weighted avg     0.9247    0.9251    0.9243     11756
 samples avg     0.9251    0.9251    0.9251     11756
```

GAMBAR 4 *Hasil Training Model untuk POSTagging*

H. Preprocessing

Preprocessing perlu dilakukan untuk memastikan dataset siap digunakan untuk pelatihan dan pengujian. Perangkat yang digunakan pada proses ini adalah Microsoft Excel dan bahasa pemrograman Python dengan *library* Scikit Learn [18]. Pada Microsoft Excel, *preprocessing* yang dilakukan adalah sebagai berikut:

a. Pemeriksaan ejaan kata pada setiap latihan soal.
b. Penghapusan spasi yang berjumlah lebih dari 1.

Sementara itu, *preprocessing* yang dilakukan dengan bahasa pemrograman Python adalah sebagai berikut:

a. *Case folding*
Pada bagian ini, semua huruf kapital pada teks diubah menjadi huruf kecil.

b. Penghapusan tanda baca
Pada bagian ini, semua tanda baca pada teks dihapus.

c. Penghapusan *stopwords*
Pada bagian ini, ada 2 daftar *stopwords* yang akan dijadikan acuan yaitu *stopwords* dari library PySastrawi (default), dan modifikasi dari PySastrawi. Modifikasi *stopwords* mengacu pada penelitian Mohammed et al. [12] yang mengatakan bahwa *stopwords* tertentu dapat memiliki dampak yang signifikan dalam menentukan tingkat kesulitan sebuah soal. Daftar *stopwords* yang dikecualikan dari *stopwords* PySastrawi dapat dilihat pada gambar 5. Proses ini akan menghasilkan data yang berbeda, dan akan dijadikan pembanding untuk menentukan mana yang lebih baik.

```
sw_keep = ['adalah', 'apa', 'arti', 'artinya', 'berapa', 'berapakah', 'beri',
           'berikan', 'diantaranya', 'disebut', 'jelaskan', 'karena',
           'mengapa', 'menunjukkan', 'merupakan', 'rupa', 'sebut']
```

GAMBAR 5 *Pengecualian Stopwords*

d. *Stemming*
Mengubah kata ke dalam bentuk dasar dari kata tersebut.

I. Ekstraksi Fitur

Ekstraksi fitur dilakukan dengan metode TF-IDF reguler dan TFPOS-IDF. TF-IDF reguler diimplementasikan secara penuh menggunakan *library* Scikit Learn. Sementara itu, untuk TFPOS-IDF program akan dimodifikasi menyesuaikan dengan rumus (5) dan dinormalisasi menggunakan rumus (6). Proses ini akan menghasilkan data yang berbeda, dan akan dijadikan pembanding untuk menentukan mana yang lebih baik. Contoh hasil TF-IDF dan TFPOS-IDF dari dokumen nomor 6 pada dataset dapat dilihat pada Tabel 2.

TABEL 2 *Hasil Ekstraksi Fitur Pada Dokumen nomor 6*

	Teladan (*NOUN*)	Tokoh (*NOUN*)	Dasar (*ADP*)	Kutip (*NOUN*)
TF-IDF	0.402	0.521	0.638	0.402
TFPOS-IDF	0.689	0.563	0.145	0.434

*nilai dibulatkan ke atas.

J. Random OverSampling

Random Oversampling merupakan salah satu metode *resampling* yang bertujuan untuk mengurangi kesenjangan ukuran kelas pada dataset, sehingga algoritma klasifikasi tidak membias pada kelas mayoritas [20]. *Random oversampling* diimplementasikan menggunakan *library* Imbalanced-learn [19], dengan parameter sampling_strategy = ‘not majority’, dan random_state = 10. Mengacu pada distribusi data dari dataset yang digunakan, maka kelas yang bukan mayoritas, yaitu selain C3, akan diduplikasi hingga jumlahnya setara dengan kelas C3. Hasil *random oversampling* dapat dilihat pada Tabel 3.

TABEL 3 *Hasil Random Oversampling*

Kelas	Sebelum *random oversampling*	Setelah *random oversampling*
C1	130	248
C2	134	248
C3	248	248
C4	113	248
C5	37	248
C6	19	248

K. Pembagian Data

Menurut penelitian Gholamy et al. [21], pembagian data untuk pelatihan dan pengujian dengan rasio 80:20 merupakan rasio yang terbaik secara empiris. Maka dari itu, pembagian dataset untuk pelatihan dan pengujian pada tugas akhir ini dibagi dengan rasio 80:20 dan parameter random_state = 23 untuk hasil pembagian data yang konsisten pada setiap eksekusi. Data yang digunakan untuk pengujian berjumlah 8 data dengan spesifikasi yang berbeda antar data. Skenario pengujian pada Tugas Akhir ini dapat dilihat pada Tabel 4.

TABEL 4 *Skenario Pengujian*

Skenario	Algoritma	Feature Extraction	Stopwords	Random Over-Sampling
1	SVM	TF-IDF	Default	N
	NB	TF-IDF	Default	N
2	SVM	TF-IDF	Modifikasi	N
	NB	TF-IDF	Modifikasi	N
3	SVM	TF-IDF	Default	Y
	NB	TF-IDF	Default	Y
4	SVM	TF-IDF	Modifikasi	Y
	NB	TF-IDF	Modifikasi	Y
5	SVM	TFPOS-IDF	Default	N
	NB	TFPOS-IDF	Default	N
6	SVM	TFPOS-IDF	Modifikasi	N
	NB	TFPOS-IDF	Modifikasi	N
7	SVM	TFPOS-IDF	Default	Y
	NB	TFPOS-IDF	Default	Y
8	SVM	TFPOS-IDF	Modifikasi	Y
	NB	TFPOS-IDF	Modifikasi	Y

L. Algoritma Klasifikasi

Klasifikasi dilakukan dengan algoritma SVM dan NB. Kedua algoritma tersebut akan mengeksekusi beberapa skenario pengujian untuk menentukan spesifikasi yang terbaik pada dataset yang digunakan. Setiap algoritma selesai memberi prediksi dan mendapatkan skor, parameter algoritma tersebut akan dioptimasi menggunakan GridSearchCV dari library Scikit Learn dengan parameter scoring = 'f1-micro'. Sebelum dioptimasi, SVM akan dijalankan dengan parameter C = 1 dan kernel = 'linear'. Sementara untuk NB akan menggunakan MultinomialNB dengan parameter *default*.

M. Evaluasi dan Analisis

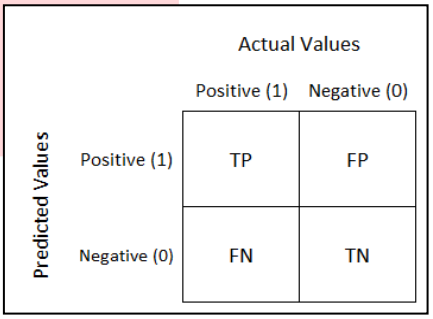
Evaluasi hasil klasifikasi dari algoritma SVM dan NB akan diukur menggunakan metrik utama yaitu F1-Measure. Perhitungan metrik berikut dapat dilakukan dengan bantuan *confusion matrix*. Skor dari *accuracy*, *precision*, *recall* dan *F1-Measure* memiliki rentang nilai 0 hingga 1. Skor dengan nilai mendekati 0 menandakan performa yang buruk, sementara skor dengan nilai mendekati 1 menandakan performa yang baik [12].

$$Accuracy = \frac{TP + TN}{n} \quad (7)$$

$$Precision = \frac{TP}{TP + FP} \quad (8)$$

$$Recall = \frac{TP}{TP + FN} \quad (9)$$

$$F1\text{-}Measure = \frac{2 \times (Precision + Recall)}{(Precision + Recall)} \quad (10)$$



GAMBAR 6 *Confusion Matrix [13]*

IV. HASIL DAN PEMBAHASAN

A. Hasil Pengujian

Hasil eksekusi dari masing-masing skenario pengujian dapat dilihat pada tabel 5. Perhitungan F1-Measure dan akurasi dibulatkan dengan 3 desimal. Algoritma dengan performa terbaik ditandai dengan *shading* berwarna hijau.

TABEL 5 *Hasil Pengujian*

Skenario	Algoritma	Feature Extraction	Stopwords	Random Over-Sampling	Akurasi & F1-Measure	Akurasi & F1-Measure optimized
1	SVM	TF-IDF	Default	N	0.453 0.429	0.445 0.438
	NB	TF-IDF	Default	N	0.423 0.421	0.445 0.434

2	SVM	TF-IDF	Modifikasi	N	0.474 0.452	0.438 0.434
	NB	TF-IDF	Modifikasi	N	0.401 0.421	0.467 0.479
3	SVM	TF-IDF	Default	Y	0.799 0.798	0.839 0.837
	NB	TF-IDF	Default	Y	0.772 0.772	0.815 0.815
4	SVM	TF-IDF	Modifikasi	Y	0.819 0.818	0.842 0.842
	NB	TF-IDF	Modifikasi	Y	0.782 0.781	0.829 0.829
5	SVM	TFPOS-IDF	Default	N	0.438 0.43	0.438 0.43
	NB	TFPOS-IDF	Default	N	0.431 0.463	0.453 0.45
6	SVM	TFPOS-IDF	Modifikasi	N	0.445 0.431	0.504 0.491
	NB	TFPOS-IDF	Modifikasi	N	0.401 0.43	0.467 0.472
7	SVM	TFPOS-IDF	Default	Y	0.815 0.814	0.836 0.836
	NB	TFPOS-IDF	Default	Y	0.735 0.732	0.795 0.793
8	SVM	TFPOS-IDF	Modifikasi	Y	0.826 0.825	0.846 0.846
	NB	TFPOS-IDF	Modifikasi	Y	0.752 0.748	0.812 0.81

B. Analisis Hasil Pengujian

Berdasarkan hasil pengujian, penggunaan *stopwords* versi modifikasi PySastrawi secara rata-rata berpengaruh baik pada hasil F1-measure dan akurasi untuk algoritma SVM dan NB. Hal ini didukung oleh pernyataan Mohammed et al. [12] bahwa beberapa *stopwords* dapat berdampak signifikan dalam menentukan tingkat kesulitan sebuah soal. Ekstraksi fitur menggunakan metode TFPOS-IDF juga secara rata-rata berdampak baik pada hasil F1-measure dibandingkan dengan metode TF-IDF. Melakukan pembobotan pada kata berdasarkan POSTag dapat membantu algoritma klasifikasi untuk meningkatkan performansinya.

Dataset yang melalui proses *random oversampling* mampu menghasilkan skor F1-measure dan akurasi yang lebih baik pada semua skenario pengujian dibandingkan data yang tidak melalui proses *random oversampling*. Hal ini dikarenakan algoritma klasifikasi dapat dilatih dengan data yang lebih banyak, sehingga dapat melakukan klasifikasi pada data pengujian secara lebih baik. Sementara itu, menggunakan parameter hasil optimasi GridSearchCV secara rata-rata mampu meningkatkan skor F1-measure dan akurasi untuk kedua algoritma. Untuk hasil optimasi yang menunjukkan penurunan skor seperti pada skenario 2 dengan algoritma SVM dan skenario 5 dengan algoritma NB, dapat disebabkan karena cara kerja GridSearchCV yang menentukan parameter terbaik berdasarkan rata-rata tertinggi dari hasil *cross validation*.

SVM menghasilkan performa paling baik pada skenario 8 dengan parameter C = 10 dan kernel = ‘linear’. Sementara itu, NB menghasilkan performa paling baik pada skenario 4 dengan parameter alpha = 0. Hasil kesalahan klasifikasi kelas pada kedua algoritma tersebut dapat dilihat pada Tabel 6, sementara untuk kesalahan klasikasi berdasarkan mata pelajaran dapat dilihat pada Tabel 7.

Berdasarkan Tabel 6, kelas C3 merupakan kelas dengan kesalahan klasifikasi terbanyak untuk kedua algoritma. Hal ini dapat disebabkan oleh strategi random oversampling yang digunakan adalah ‘not majority’, sehingga kelas C3 yang merupakan mayoritas kelas pada dataset ini tidak dilakukan random oversampling. Urutan kelas berikutnya yang dengan kesalahan klasifikasi terbanyak secara urut adalah C2, C4 dan C1 untuk kedua algoritma. Kemudian, pada Tabel 7 dapat dilihat bahwa urutan mata pelajaran yang paling banyak terdapat kesalahan klasifikasi secara urut adalah bahasa indonesia, matematika dan ipa untuk kedua algoritma.

TABEL 6 *Jumlah Salah Prediksi per Kelas*

	C1	C2	C3	C4	C5	C6
SVM	4	16	21	8	0	0
NB	5	14	21	9	0	3

TABEL 7 *Jumlah Salah Prediksi per Mata Pelajaran*

	Bahasa Indonesia	IPA	Matematika
SVM	22	9	18
NB	26	9	19

V. KESIMPULAN

Pada Tugas Akhir ini, algoritma klasifikasi dengan performa yang terbaik adalah SVM dengan nilai akurasi dan F1-measure sebesar 0.846, disusul dengan algoritma NB dengan nilai akurasi dan F1-measure sebesar 0.829. Kedua algoritma dapat dikategorikan memiliki performa yang baik karena nilai akurasi dan F1-measure sama-sama lebih mendekati 1 daripada mendekati 0 [12]. Walaupun karakteristik soal pada dataset beragam, algoritma SVM dan NB masih dapat melakukan klasifikasi dengan baik. Ekstraksi fitur dengan TFPOS-IDF dapat memberikan performansi yang lebih baik dibandingkan TF-IDF pada algoritma SVM. Sementara itu, TF-IDF memiliki performansi yang lebih baik dibandingkan dengan TFPOS-IDF pada algoritma NB. Kemudian, memodifikasi *stopwords* dapat membantu memberikan performansi yang baik untuk kedua algoritma. Selain itu, melakukan *random oversampling* pada data dapat meningkatkan performa yang dihasilkan untuk algoritma SVM dan NB. Penelitian ini masih dapat dikembangkan dari sisi ketersediaan jumlah data yang digunakan dan menggunakan kombinasi Word2Vec dan TFPOS-IDF sebagai metode ekstraksi fitur.

REFERENSI

[1] S. F. Kusuma, D. Siahaan and U. L. Yuhana, "Automatic Indonesia's questions classification based on bloom's taxonomy using Natural Language Processing a preliminary study," 2015 International Conference on Information Technology Systems and Innovation (ICITSI), 2015, pp. 1-6, doi: 10.1109/ICITSI.2015.7437696.

[2] Utari, Retno. 2011. Taksonomi Bloom Apa dan Bagaimana Menggunakannya?

[3] A. Aninditya, M. A. Hasibuan and E. Sutoyo, "Text Mining Approach Using TF-IDF and Naive Bayes for Classification of Exam Questions Based on Cognitive Level of Bloom's Taxonomy," 2019 IEEE International Conference on Internet of Things and Intelligence System (IoTaIS), 2019, pp. 112-117, doi: 10.1109/IoTaIS47347.2019.8980428.

[4] H. S. Bhargav, G. Akalwadi and N. V. Pujari, "Application of Blooms Taxonomy in Day-to-Day Examinations," 2016 IEEE 6th International Conference on Advanced Computing (IACC), 2016, pp. 825-829, doi: 10.1109/IACC.2016.157.

[5] S. K. Patil and M. M. Shreyas, "A Comparative Study of Question Bank Classification based on Revised Bloom's Taxonomy using SVM and K-NN," 2017 2nd International Conference On Emerging Computation and Information Technologies (ICECIT), 2017, pp. 1-7, doi: 10.1109/ICECIT.2017.8453305.

[6] A. B. Prasetijo, R. R. Isnanto, D. Eridani, Y. A. A. Soetrisno, M. Arfan and A. Sofwan, "Hoax detection system on Indonesian news sites based on text classification using SVM and SGD," 2017 4th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE), 2017, pp. 45-49, doi: 10.1109/ICITACEE.2017.8257673.

[7] E. Subiyantoro, A. Ashari and Suprapto, "Cognitive Classification Based on Revised Bloom's Taxonomy Using Learning Vector Quantization," 2020 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM), 2020, pp. 349-353, doi: 10.1109/CENIM51130.2020.9297879.

[8] N. Kalcheva, M. Karova and I. Penev, "Comparison of the accuracy of SVM kernel functions in text classification," 2020 International Conference on Biomedical Innovations and Applications (BIA), 2020, pp. 141-145, doi: 10.1109/BIA50171.2020.9244278.

[9] Gandhi, Rohith. 2018. Support Vector Machine — Introduction to Machine Learning Algorithms. [Online]. Available at: https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47 [Accessed 24 November 2021]

[10] Anonymous. Naive Bayes. [Online]. Available at: https://scikit-learn.org/stable/modules/naive_bayes.html [Accessed 30 November 2021]

[11] W Zhang, T Yoshida, and X Tang. 2011. A comparative study of TFIDF, LSI and multi-words for text classification. Expert Systems with Applications Volume 38 Issue 3 Pages 2758-2765. doi: 10.1016/j.eswa.2010.08.066.

[12] Mohammed M, Omar N (2020) Question classification based on Bloom's taxonomy cognitive domain using modified TF-IDF and word2vec. PLoS ONE 15(3): e0230442. https://doi.org/10.1371/journal.pone.0230442

[13] Narkhede S. 2018. Understanding Confusion Matrix. [Online]. Available at: https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62 [Accessed 12 December 2021]

[14] [Online]. Available at: https://www.ruangguru.com/blog/ [Accessed 21 December 2021]

[15] Syarifah L., Yenni Y., & Dewi W. (2020). Analisis Soal-Soal Pada Buku Ajar Matematika Siswa Kelas XI Ditinjau Dari Aspek Kognitif. *Jurnal Cendekia : Jurnal Pendidikan Matematika*, *4*(2), 1259-1272. https://doi.org/10.31004/cendekia.v4i2.335

[16] David R. Krathwohl (2002) A Revision of Bloom's Taxonomy: An Overview, Theory Into Practice, 41:4, 212-218, DOI: 10.1207/s15430421tip4104_2

[17] Akbik A., Bergmann, T., Blythe, D., Rasul, K., Schweter, S. and Vollgraf, R., 2019, June. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)* (pp. 54-59).

[18] Pedregosa, F. et al., 2011. Scikit-learn: Machine learning in Python. *Journal of machine learning research*, 12(Oct), pp.2825–2830.

[19] Guillaume Lemaître, Fernando Nogueira, and Christos K. Aridas. 2017. Imbalanced-learn: a python toolbox to tackle the curse of imbalanced datasets in machine learning. J. Mach. Learn. Res. 18, 1 (January 2017), 559–563.

[20] Padurariu C., Breaban M.E. 2019. Dealing with Data Imbalance in Text Classification. Procedia Computer Science, Volume 159, Pages 736-745.

[21] Gholamy, Afshin; Kreinovich, Vladik; and Kosheleva, Olga, "Why 70/30 or 80/20 Relation Between Training and Testing Sets: A Pedagogical Explanation" (2018). Departmental Technical Reports (CS). 1209.

[22] E. R. Setyaningsih and I. Listiowarni, "Categorization of Exam Questions based on Bloom Taxonomy using Naïve Bayes and Laplace Smoothing," 2021 3rd East Indonesia Conference on Computer and Information Technology (EIConCIT), 2021, pp. 330-333, doi: 10.1109/EIConCIT50028.2021.9431862.