

## 2. Studi Terkait

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.

### 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:

- Kognitif, aspek yang ditekankan seperti keterampilan berfikir dan pengetahuan.
- Afektif, ranah ini mencakup perasaan, motivasi dan sikap sebagai perilaku yang terkait dengan emosi.
- 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:

- Pengetahuan (C1), kompetensi dalam menyebutkan atau menjelaskan kembali terkait hal yang sudah dipelajari.
- Pemahaman (C2), kompetensi dalam menginterpretasi dan menyatakan kembali berdasarkan pemahaman sendiri serta memahami instruksi / masalah yang diberikan.
- Penerapan (C3), kompetensi dalam mengaplikasikan konsep ke dalam situasi yang baru.
- Analisa (C4), kompetensi dalam memecahkan suatu konsep ke dalam beberapa komponen sehingga memahami korelasi komponen - komponen terhadap konsep tersebut secara utuh.
- Sintesa (C5), kompetensi dalam menciptakan struktur yang baru dari komponen - komponen yang tersedia.
- 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 penelitian Setyaningsih et al. [22] dan dapat dilihat pada Tabel 1.

| Tingkatan Kognitif | Contoh Kata Kerja  |
|--------------------|--|
| mengingat (C1)     | menamai, menulis, mengutip, menyebutkan, menghafal, melabeli, mendaftar, menunjukkan, memasang, 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. |

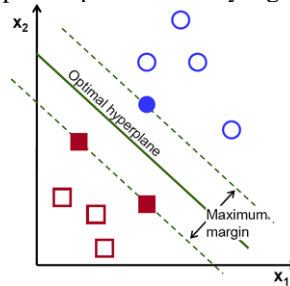
|                   |  |
|-------------------|--|
| menganalisa (C4)  | membandingkan, mengkaji ulang, membedakan, mengkontraskan, memecah ke dalam beberapa bagian, menunjukkan korelasi antar variabel, memisahkan, menyisihkan, menghubungkan, mempertimbangkan, menduga. |
| mengevaluasi (C5) | menilai, 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.                         |

**Tabel 1** Contoh Kata Kerja Tingkatan Kognitif

### 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.

### 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:

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

$x$  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  $P(x_i | y)$

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.

### 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.

#### 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:

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

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:

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