

Penerapan *Information Gain* pada *K-Nearest Neighbor* untuk Klasifikasi Tingkat Kognitif Soal pada *Taksonomi Bloom*

Tyas Setiyorini

Abstract— *Bloom's Taxonomy* is a classification system that is used to define and distinguish the level of cognition (thinking, learning, and understanding) human diverse. The initial purpose in making taxonomy is focusing on three main domains of learning, which is cognitive, affective, and psychomotor. Nowadays, academics identify cognitive levels of Bloom of a question manually. However, only few academics that can be identify the cognitive level correctly, so most of them made a mistake to categorize questions. *K-Nearest Neighbor (KNN)* is a simple but effective method for categorization cognitive level questions on the *Bloom's Taxonomy*, however *KNN* has high dimension of text vector. In order to resolve these problems, the *Information Gain (IG)* method is needed to reduce dimension of text vector. Several experiments were conducted to obtain optimal architecture and produce an accurate classification. The results of 10 experiments on the *Question Bank* dataset with *KNN* obtained the biggest accuracy is 59,87% and the biggest kappa is 0,496. Then on *KNN + IG* obtained the biggest accuracy is 66,18% and the biggest kappa is 0,574 It can be concluded that the classification level cognitive questions on *Bloom's taxonomy* using *KNN + IG* method is more accurate than the *KNN* method only.

Intisari— Taksonomi Bloom merupakan sistem klasifikasi yang digunakan untuk mendefinisikan dan membedakan tingkat kognisi (berpikir, belajar, dan memahami) manusia yang berbeda-beda. Tujuan awal dalam pembuatan taksonomi adalah memfokuskan pada tiga domain utama dari pembelajaran, yaitu kognitif, afektif, dan psikomotorik. Saat ini, kalangan akademisi mengidentifikasi tingkat kognitif Bloom sebuah pertanyaan secara manual. Namun, hanya sedikit akademisi yang dapat mengidentifikasi tingkat kognitif dengan benar, sehingga sebagian besar melakukan kesalahan mengkategorisasikan pertanyaan. *K-Nearest Neighbor (KNN)* merupakan metode sederhana namun efektif untuk kategorisasi tingkat kognitif soal pada taksonomi Bloom, namun *KNN* memiliki dimensi vektor yang besar. Untuk menyelesaikan masalah tersebut diperlukan metode *Information Gain (IG)* untuk mengurangi dimensi vektor teks. Beberapa eksperimen dilakukan untuk mendapatkan arsitektur yang optimal dan menghasilkan klasifikasi yang akurat. Hasil dari 10 eksperimen pada dataset *Question Bank* dengan *KNN* didapatkan akurasi terbesar adalah 59,97% dan *kappa* terbesar adalah 0,496. Kemudian pada *KNN+IG* didapatkan akurasi terbesar adalah 66,18% dan *kappa* terbesar adalah 0,574. Maka dapat disimpulkan klasifikasi tingkat kognitif soal pada taksonomi Bloom dengan menggunakan metode *KNN+IG* lebih akurat dibanding dengan metode *KNN* saja.

Kata Kunci — Klasifikasi, Taksonomi Bloom, *K-Nearest Neighbor*, *Information Gain*

Jurusan Teknik Informatika STMIK Nusa Mandiri Jakarta, Jl. Damai No. 8 Warung Jati Barat (Margasatwa) Pasar Minggu Jakarta Selatan DKI Jakarta 12540 INDONESIA (tel: 021-78839502; fax: 021-78839421; e-mail: tyas.setiyorini@gmail.com)

I. PENDAHULUAN

Taksonomi Bloom merupakan sistem klasifikasi yang digunakan untuk mendefinisikan dan membedakan tingkat kognisi (berpikir, belajar, dan memahami) manusia yang berbeda-beda. Pendidik umumnya menggunakan taksonomi ini untuk acuan atau petunjuk dalam penyusunan kurikulum, metode instruksional dan pembuatan ujian atau assessment (test atau evaluasi pembelajaran siswa). Taksonomi Bloom telah diterapkan secara luas sebagai pedoman dalam merancang ujian yang layak, terdiri dari pertanyaan yang memiliki berbagai tingkatan secara kognitif [1]. Taksonomi Bloom telah banyak digunakan untuk tujuan pengukuran dan penilaian dalam proses pembelajaran [2] [3]. Tingkat kognitif terdiri dari enam tingkat. Tingkat yang paling rendah dikenal sebagai tingkat pengetahuan (*knowledge level*), yang bertujuan untuk mengingat data dari pelajaran sebelumnya. Kedua, tingkat pemahaman (*comprehension level*) menuntut siswa untuk memiliki kemampuan untuk menjelaskan pengetahuan yang dipelajari. Ketiga, tingkat aplikasi (*application level*), siswa harus mampu menerapkan pengetahuan yang dipelajari dalam suatu tindakan. Keempat, tingkat analisis (*analysis level*) mengarahkan siswa untuk menyelidiki informasi yang mereka pelajari. Kelima, tingkat sintesis (*synthesis level*), siswa harus menghubungkan semua informasi dan mengintegrasikan menjadi sesuatu hal yang baru. Tingkat terakhir adalah tingkat evaluasi (*evaluation level*) di mana siswa dapat mencapainya apabila mereka mampu bertahan dalam setiap pendapatnya.

Saat ini, kalangan akademisi mengidentifikasi tingkat kognitif Bloom sebuah pertanyaan secara manual. Namun, hanya sedikit akademisi yang dapat mengidentifikasi tingkat kognitif dengan benar, sehingga sebagian besar melakukan kesalahan mengkategorisasikan pertanyaan [4]. Selain itu, proses penentuan tingkat kognitif Bloom sangat memakan waktu dan melelahkan, karena tingginya jumlah pertanyaan yang harus diklasifikasikan dalam kertas ujian. Oleh karena itu dibutuhkan metode yang dapat mengidentifikasi tiap-tiap soal menurut tingkat kognitifnya dengan secara otomatis.

Text mining adalah proses menemukan pengetahuan dari data tekstual yang juga dikenal sebagai *text data mining* [5]. Sebagaimana diketahui oleh masyarakat, data mining adalah proses ekstraksi informasi yang berguna dari fakta yang tersimpan di dalam database terstruktur, sedangkan *text mining* mencari pola dari teks bahasa alami [6]. Umumnya, pekerjaan pada *text mining* dimulai dengan memilih koleksi dokumen, yang tidak jauh berbeda dengan pekerjaan pada *data mining* [7]. Seperti yang disebutkan Milner [8] pada

bukunya, *text mining* terdiri dari tujuh tipe yaitu klasifikasi dokumen (*document classification*), pengelompokan dokumen (*document clustering*), ekstraksi informasi (*information extraction*), pemulihan informasi (*information retrieval*), web mining, pemrosesan bahasa alami (*natural language processing*) dan ekstraksi konsep. Kategorisasi teks merupakan bagian dari klasifikasi dokumen. Kategorisasi teks merupakan proses untuk menempatkan sekumpulan dokumen ke dalam kategori yang telah ditetapkan [9].

Dalam beberapa tahun terakhir, teknik kategorisasi teks telah menarik perhatian dan telah banyak digunakan di berbagai aplikasi. Beberapa penelitian memperkenalkan kategorisasi teks secara komprehensif [10] [11] dan berbagai metode kategorisasi teks telah diusulkan, yaitu decision tree [12], K-Nearest Neighbor (KNN) [13], Naïve Bayes (NB) [14], neural network (NN) [15], Rocchio [16], Support Vector Machine (SVM) [17], dan centroid-based approaches [18]. Selain itu, kategorisasi teks telah digunakan secara luas pada pemfilteran email *spam* [19] dan analisis sentimen [20]. Saat ini tantangan utama dari kategorisasi teks dalam memecahkan masalah dunia nyata adalah klasifikasi hirarkis, ketidakseimbangan klasifikasi, dan mengelompokkan data teks besar secara efisien [11]. Ada hubungan yang rumit antara kategori dalam masalah klasifikasi multi-kategori. Untuk lebih memahami hubungan ini, beberapa klasifikasi teks hierarkis telah diteliti [21] [22]. Dalam masalah kategorisasi teks, performa KNN dan SVM terbukti lebih baik jika dibandingkan dengan NB, Rocchio dan NN [23].

KNN merupakan metode sederhana namun efektif untuk kategorisasi teks, tetapi memiliki tiga kecacatan yang fatal. Pertama, kompleksitas komputasi kemiripan datanya besar. Kedua, kinerjanya mudah dipengaruhi oleh data pelatihan tunggal, seperti data *noise*. Ketiga, KNN tidak membangun model klasifikasi karena merupakan metode *lazy learning*. Banyak peneliti mencari cara untuk mengurangi kompleksitas KNN, yang dapat dibagi menjadi tiga metode umum, yaitu mengurangi dimensi vektor teks [24], mengurangi jumlah data pelatihan [25], dan mempercepat proses menemukan k tetangga terdekat [26].

Seleksi fitur (*feature selection*) dapat digunakan untuk mengurangi dimensi vektor teks pada dataset pertanyaan berbasis taksonomi Bloom. Dalam mengoptimalkan performa *classifier*, seleksi fitur merupakan salah satu bagian terpenting [27]. Seleksi fitur berdasarkan pada pengurangan fitur yang besar, yaitu dengan menghapus atribut yang tidak relevan [28]. Akurasi dapat ditingkatkan apabila menggunakan algoritma seleksi fitur yang tepat [29] [30]. Algoritma seleksi fitur dapat dibagi menjadi dua tipe, yaitu *filter* dan *wrapper* [31]. Contoh dari tipe *filter* adalah log likelihood ratio, IG, dan chi-square. Contoh dari tipe *wrapper* adalah *backward elimination* dan *forward selection* [32]. Hasil presisi dari tipe *filter* lebih rendah daripada tipe *wrapper*, tetapi hasil tipe *wrapper* ini dicapai dengan tingkat kompleksitas yang tinggi yang dapat menimbulkan masalah [28].

Yang dan Perderson [33] melakukan perbandingan lima algoritma seleksi fitur pada klasifikasi dokumen. Lima algoritma tersebut antara lain mutual information, term

strength, chi-square, IG, dan document frequency. Hasil penelitian tersebut membuktikan bahwa chi-square dan IG paling efisien. Forman [30] membandingkan 12 algoritma seleksi fitur pada 229 klasifikasi teks menjadi dua kategori. Hasil penelitian menunjukkan bahwa metode Bi-Normal Separation yang diusulkan peneliti mendapatkan hasil yang tidak lebih baik dibandingkan IG dan chi-square. Tan dan Zang [34] menggunakan algoritma seleksi fitur untuk analisis sentimen dokumen yang menggunakan bahasa Cina. Hasil yang ditunjukkan dalam penelitian tersebut yaitu IG mendapatkan hasil yang paling baik.

Dari paparan di atas tampak bahwa IG memiliki potensi yang lebih baik dalam proses mengurangi dimensi atau menghilangkan fitur yang kurang relevan sehingga dapat mengurangi dimensi dari vektor teks sebelum dilakukannya klasifikasi. Untuk itu penelitian ini akan menekankan pada kombinasi kedua metode tersebut untuk memungkinkan klasifikasi soal ujian berdasarkan tingkat kognitif yang berbasis taksonomi Bloom dapat dilakukan secara otomatis dengan tingkat akurasi yang optimal.

II. KAJIAN LITERATUR

A. K-Nearest Neighbor (KNN)

KNN merupakan algoritma *supervised learning*, di mana hasil dari *query instance* yang baru diklasifikasikan berdasarkan mayoritas dari kategori pada KNN. Kelas yang paling banyak muncul yang akan menjadi kelas hasil klasifikasi. KNN merupakan salah satu metode pengklasifikasian data berdasarkan similaritas dengan label data [35]. Algoritma KNN merupakan sebuah metode untuk melakukan klasifikasi terhadap objek baru berdasarkan k tetangga terdekatnya [36]. Menurut M. Han, J., & Kamber [36], KNN adalah algoritma pembelajaran berbasis instan yang menggunakan jarak terdekat dalam menentukan kategori vektor baru dalam set data training.

Model KNN memiliki masalah untuk menemukan tetangga terdekat k pada titik *query* dari dataset yang digunakan [37] [38]. Penggunaan metode ini banyak digunakan untuk menangani masalah dalam bidang ilmiah dan rekayasa perangkat lunak seperti pengenalan pola, pengenalan objek, pengelompokan data, fungsi *approximate*, kuantisasi vektor, klasifikasi pola.

Untuk menentukan jumlah data atau tetangga terdekat ditentukan oleh user yang dinyatakan dengan k.

$$d_i = \sqrt{\sum_{i=1}^p (x_{1i} - x_{2i})^2} \quad (1)$$

Di mana $x_1 = (x_{11}, x_{12}, \dots, x_{1n})$ dan $x_2 = (x_{21}, x_{22}, \dots, x_{2n})$. Dengan kata lain, untuk setiap atribut numerik, kita mengambil perbedaan antara yang sesuai nilai-nilai atribut yang dalam vektor x_1 dan x_2 dari matriks dengan ukuran dimensi. Akar kuadrat diambil dari akumulasi jumlah total jarak. Biasanya, kita menormalkan nilai masing-masing atribut sebelum digunakan [39]. Prinsip kerja KNN adalah mencari jarak terdekat antara data yang dievaluasi dengan k tetangga terdekatnya dalam data pelatihan. Persamaan

Penghitungan untuk mencari Euclidean dengan d adalah jarak dan p adalah dimensi data dengan:

$$d_i = \sqrt{\sum_{i=1}^p (x_{1i} - x_{2i})^2} \quad (2)$$

di mana:

x1: sample data uji

x2: data uji

d: jarak

p: dimensi data

B. Information Gain (IG)

IG sering digunakan sebagai penentu atribut pada *field machine learning* [33]. IG dari sebuah *term* diukur dengan menghitung jumlah bit informasi yang diambil dari prediksi kategori dengan ada atau tidaknya *term* dalam suatu dokumen. Jain & Richariya [40] menjelaskan penghitungan IG. Misalkan ada kelas m dan training set berisi sampel SI, yaitu kelas I dan S adalah jumlah sampel dalam *training set* diharapkan merupakan informasi yang dibutuhkan untuk mengklasifikasikan sampel yang diberikan akan dihitung.

$$I(S_1, S_2, \dots, S) = \sum_{i=1}^m \frac{S_i}{S} \log_2 \frac{S_i}{S} \quad (3)$$

Atribut F dengan nilai-nilai {f 1, f 2, f 3, ..., fv} dapat membagi data pelatihan yang ditetapkan dalam subset {S1, S2, S3, ..., Sv}, dimana Sj merupakan subset yang memiliki nilai fj untuk atribut F. Selanjutnya Sj berisi sampel Sij kelas i. Entropi atribut F diberikan dalam:

$$E(F) = - \sum_{i=1}^m \frac{S_{1j} + S_{2j} + S_{3j} + \dots + S_{ij}}{S} I(S_{1j} + S_{2j} + S_{3j} + \dots + S_{ij}) \quad (4)$$

sehingga IG untuk atribut F dapat dihitung dengan menggunakan rumus, sebagai berikut:

$$\text{Gain}(F) = I(S_{1j} + S_{2j} + S_{3j} + \dots + S_{ij}) - E(F) \quad (5)$$

Nilai dari Gain yang tersebut di atas, yaitu menghitung IG dari atribut F yang berkaitan dengan semua kelas. Jika kita ingin mengukur Gain dari atribut untuk kelas k yang diberikan, kita akan mempertimbangkan masalah sebagai klasifikasi biner satu. Kami mempertimbangkan dua kelas: kelas normal dan sisanya merupakan anomali kelas lain. Jadi perkiraan IG yang diperlukan untuk mengklasifikasi sampel yang diberikan akan menjadi:

$$I(S, S) = - \frac{S}{S} \log_2 \left(\frac{S}{S} \right) - \frac{S}{S} \log_2 \frac{S}{S} \quad (6)$$

Di mana k menunjukkan kelas dikomplemenkan kelas k. Entropi atribut F menurut kelas k adalah:

$$E(F) = - \sum_{i=1}^m \frac{S_{kj} + S_{k'j}}{S} I(S_{kj}, S_{k'j}) \quad (7)$$

IG untuk F dapat dihitung dengan menggunakan rumus sebagai berikut:

$$\text{Gain}(F) = I(S_k, S_{k'}) - E(F) \quad (8)$$

Mengukur Gain ini memberikan arti penting dari atribut. Algoritma berikut memilih atribut yang lebih besar dari nilai batas dari kumpulan data. Algoritma seleksi atribut menggunakan IG F(I) digunakan untuk menyimpan set atribut yang dipilih.

III. METODE PENELITIAN

A. Pengumpulan Data

Dataset yang digunakan pada penelitian ini mengacu pada penelitian yang dilakukan oleh Chai Jing Hui [41]. *Detail* kalimat soal yang diujikan dapat dilihat di lampiran. Dataset menggunakan bahasa Inggris, sehingga *stopword* yang digunakan untuk *preprocessing* data juga berbahasa Inggris.

Tabel 1 menunjukkan *dataset* terdiri dari 274 butir soal yang secara keseluruhan terbagi dalam berbagai tingkat kesulitan. Terdapat 28 butir soal yang diklasifikasikan *Knowledge*, 44 butir soal diklasifikasikan *Comprehension*, 41 butir soal diklasifikasikan *Application*, 48 butir soal diklasifikasikan *Analysis*, 59 butir soal diklasifikasikan *Synthesis* dan 54 butir soal diklasifikasikan *Evaluation*.

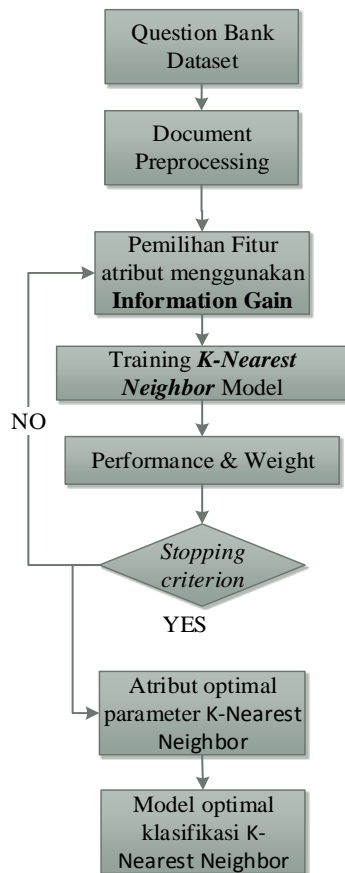
TABEL I.
KLASIFIKASI SOAL

Klasifikasi	Jumlah
<i>Knowledge</i>	28
<i>Comprehension</i>	44
<i>Application</i>	41
<i>Analysis</i>	48
<i>Synthesis</i>	59
<i>Evaluation</i>	54

B. Metode

Gambar 1 menggambarkan metode yang diusulkan dalam penelitian ini yaitu metode IG pada KNN. Pada pengolahan awal, semua pertanyaan pada *Question Bank* diubah menjadi dataset yang dapat diklasifikasikan dengan menggunakan teknik *text processing*. *Text processing* yang digunakan antara lain adalah *tokenizer*, *filter stopwords*, dan *stem*. *Tokenize* merupakan proses untuk memisah-misahkan kata. Tahapan *tokenize* dimulai dari memisah-misahkan bagian review yang dipisahkan dengan karakter spasi. Selanjutnya, bagian yang hanya memiliki karakter *non alphabet* dan angka akan dibuang. Bagian yang masuk dalam daftar *emoticon* akan dikonversi. Terakhir, bagian yang memiliki karakter selain alfabet, angka, dan garis bawah akan dipecah sesuai posisi karakter tersebut. Pada proses *filter stopwords* akan dihilangkan kata-kata yang sering muncul namun tidak memiliki pengaruh apapun dalam ekstraksi sentimen suatu

review. Tahap stemming adalah tahap untuk mencari kata dasar dari tiap kata hasil filter.



Gambar 1. Metode KNN+IG

Fitur dari dataset yang dihasilkan pada tahap *text processing* diseleksi menggunakan metode IG. Setelah diseleksi fitur-fitur tersebut akan digunakan pada proses klasifikasi menggunakan algoritma KNN untuk menghasilkan fitur yang optimal, kemudian akan dihitung nilai akurasi dan *kappa*.

IV. HASIL DAN PEMBAHASAN

Penelitian yang dilakukan menggunakan komputer dengan spesifikasi CPU Intel Core i5 1.6GHz, RAM 8GB, dan sistem operasi Microsoft Windows 7 Professional 64-bit. Aplikasi yang digunakan adalah RapidMiner 7.3. Penelitian ini menggunakan dataset *Question Bank*. Dataset yang digunakan pada penelitian ini mengacu pada penelitian yang dilakukan oleh Chai Jing Hui [41].

Setelah eksperimen yang dilakukan dengan KNN dan KNN+IG, kemudian dikomparasi hasil akurasi dan *kappa* pada metode KNN dengan KNN+IG dari 10 eksperimen. Pada Tabel 2 dari 10 eksperimen dan rata-rata keseluruhan eksperimen secara konsisten menunjukkan kenaikan nilai akurasi dan *kappa* yang signifikan antara KNN dengan KNN+IG.

TABEL 2.
HASIL AKURASI DAN KAPPA DENGAN KNN DAN KNN+IG

KNN		KNN + IG	
Akurasi	Kappa	Akurasi	Kappa
53,13%	0,404	66,18%	0,574
47,95%	0,345	56,82%	0,47
47,45%	0,34	62,50%	0,532
53,76%	0,419	56,32%	0,463
53,16%	0,409	58,87%	0,483
59,97%	0,496	58,92%	0,481
58,89%	0,482	64,03%	0,542
59,42%	0,49	62,50%	0,519
55,76%	0,444	64,08%	0,538
56,71%	0,456	63,97%	0,539

Kenaikan nilai akurasi dan *kappa* yang signifikan dan konsisten dari 10 eksperimen menunjukkan bahwa penggunaan KNN+IG memiliki nilai akurasi dan *kappa* yang lebih besar dibanding dengan penggunaan KNN saja. Dari hasil pengujian tersebut menunjukkan bahwa IG pada KNN mampu mengurangi dimensi vektor teks, sehingga menghasilkan kinerja atau tingkat akurasi klasifikasi tingkat kognitif soal pada taksonomi Bloom yang lebih baik dibanding dengan menggunakan metode KNN saja. Hal ini membuktikan penelitian Yang dan Pedersen [33], Forman [30], Tan dan Zang [34] bahwa IG mampu mengurangi dimensi vektor teks. Selain itu hasil tersebut juga membuktikan penelitian Supriyanto et al. [41] bahwa KNN mampu mengklasifikasi tingkat kognitif soal pada taksonomi Bloom.

Untuk menjamin evaluasi hipotesis penelitian ini, dibutuhkan pengujian dengan metode statistik untuk menguji hubungan antara penggunaan metode KNN dengan KNN+IG, apakah terdapat hubungan di antara keduanya. Pengujian hipotesis ini menggunakan metode t-Test. Metode ini termasuk yang paling umum dalam metode statistik tradisional, yaitu t-Test [42]. Ada atau tidaknya perbedaan antara dua model membutuhkan pengujian, salah satunya dengan uji t-Test [43], dengan melihat nilai P. Jika nilai $P < 0,05$ maka menunjukkan hipotesis nol ditolak atau disebut hipotesis alternatif [44]. Hipotesis nol menyatakan tidak ada pengaruh atau perbedaan antara dua buah variabel, sedangkan hipotesis alternatif menyatakan adanya pengaruh atau perbedaan antara dua buah variabel [44].

TABEL III.
HASIL T-TEST AKURASI DENGAN KNN DAN KNN+IG

	Akurasi	
	KNN	KNN+IG
Mean	0,5462	0,61419
Variance	0,001961691	0,001170034
Observations	10	10
Pearson Correlation	0,258675754	
Hypothesized Mean Difference	0	
df	9	
t Stat	-4,437129131	
P(T<=t) one-tail	0,000815125	
t Critical one-tail	1,833112933	
P(T<=t) two-tail	0,00163025	
t Critical two-tail	2,262157163	

TABEL IV.
HAIL T-TEST KAPPA DENGAN KNN DAN KNN+IG

Kappa		
	KNN	KNN+IG
Mean	0,4285	0,5141
Variance	0,003119167	0,001391211
Observations	10	10
Pearson Correlation	0,124678997	
Hypothesized Mean Difference	0	
df	9	
t Stat	-4,284858928	
P(T<=t) one-tail	0,001017689	
t Critical one-tail	1,833112933	
P(T<=t) two-tail	0,002035377	
t Critical two-tail	2,262157163	

Pada Tabel 3 menampilkan t-Test untuk hasil akurasi KNN dan KNN+IG yang menunjukkan hipotesis nol ditolak (hipotesis alternatif) yaitu dengan nilai $P < 0,05$ yaitu 0,0008 Pada Tabel 4 menampilkan t-Test untuk hasil *kappa* KNN dan KNN+IG juga menunjukkan hipotesis nol ditolak (hipotesis alternatif) yaitu dengan nilai $P < 0,05$ yaitu 0,0010.

Hasil t-Test dengan hipotesis nol ditolak (hipotesis alternatif) tersebut menunjukkan bahwa antara penggunaan metode KNN dengan KNN+IG menunjukkan adanya pengaruh atau perbedaan yang signifikan. KNN+IG menghasilkan kinerja atau tingkat akurasi yang lebih baik dibanding dengan menggunakan metode KNN saja. Hal tersebut seperti dikatakan pada penelitian Yang dan Pedersen [33], Forman [30], Tan dan Zang [34] bahwa IG secara konsisten menunjukkan kemampuan meningkatkan akurasi klasifikasi pada KNN, hal ini berarti bahwa seleksi fitur yang diusulkan dapat menjadi alat yang efektif untuk meningkatkan kinerja KNN.

V. KESIMPULAN

Hasil dari 10 eksperimen pada dataset *Question Bank* dengan KNN didapatkan akurasi terbesar adalah 59,97% dan *kappa* terbesar adalah 0,496. Kemudian pada KNN+IG didapatkan akurasi terbesar adalah 66,18% dan *kappa* terbesar adalah 0,574.

Dari hasil pengujian di atas maka dapat disimpulkan bahwa IG mampu mengurangi dimensi vektor teks pada KNN, sehingga menghasilkan kinerja atau tingkat akurasi klasifikasi tingkat kognitif soal pada taksonomi Bloom yang lebih baik dibanding dengan menggunakan metode KNN saja.

UCAPAN TERIMA KASIH

Terima kasih saya sampaikan kepada pihak-pihak yang telah mendukung dalam pembuatan *paper* ini.

REFERENSI

[1] K. O. Jones, J. Harland, J. M. V Reid, and R. Bartlett, "Relationship between examination questions and bloom's

taxonomy," in *Proceedings - Frontiers in Education Conference, FIE*, 2009.

- [2] N. N. Khairuddin and K. Hashim, "Application of Bloom's taxonomy in software engineering assessments," *Proc. 8th Conf. Appl. Comput. Sci.*, pp. 66–69, 2008.
- [3] E. Thompson, A. Luxton-Reilly, J. L. Whalley, M. Hu, and P. Robbins, "Bloom's taxonomy for CS assessment," *Conf. Res. Pract. Inf. Technol. Ser.*, vol. 78, pp. 155–161, 2008.
- [4] N. Yusof and C. J. Hui, "Determination of Bloom's cognitive level of question items using artificial neural network," in *Proceedings of the 2010 10th International Conference on Intelligent Systems Design and Applications, ISDA'10*, 2010, pp. 866–870.
- [5] U. et al Fayyad, "From Data Mining to Knowledge Discovery in Databases," *Comput. J.*, vol. 58, no. 1, pp. 1–6, 1996.
- [6] S. M. Weiss et al., "Maximizing text-mining performance," *Intell. Syst. their Appl. IEEE*, vol. 14, no. 4, pp. 63–69, 1999.
- [7] H. Cherfi, A. Napoli, and Y. Toussaint, "Towards a text mining methodology using association rule extraction," *Soft Comput.*, vol. 10, no. 5, pp. 431–441, 2005.
- [8] G. Miner, *Practical Text Mining and Statistical Analysis for Non-structured Text Data Applications*. 2012.
- [9] A. Genkin, D. D. Lewis, and D. Madigan, "Large-Scale Bayesian Logistic Regression for Text Categorization," *Technometrics*, vol. 49, no. 3, pp. 291–304, 2007.
- [10] E. Alpaydm, *Introduction to Machine Learning*. MIT press, 2014.
- [11] J.-S. SU, "Advances in Machine Learning Based Text Categorization," *J. Softw.*, vol. 17, no. 9, p. 1848, 2006.
- [12] C. Apté, F. Damerau, and S. M. Weiss, "Automated learning of decision rules for text categorization," *ACM Trans. Inf. Syst.*, vol. 12, no. 3, pp. 233–251, 1994.
- [13] G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer, "An kNN model-based approach and its application in text categorization," in *International Conference on Intelligent Text Processing and Computational Linguistics*, 2004, pp. 559–570.
- [14] E. Frank and R. R. Bouckaert, "Naive bayes for text classification with unbalanced classes," *PKDD'06 Proc. 10th Eur. Conf. Princ. Pract. Knowl. Discov. Databases*, pp. 503–510, 2006.
- [15] P. Srinivasan and M. E. Ruiz, "Hierarchical Text Categorization Using Neural Networks," *Inf. Retr. Boston.*, vol. 5, pp. 87 – 118, 2002.
- [16] J. J. Rocchio, "Relevance feedback in information retrieval," *SMART Retr. Syst.*, pp. 313–323, 1971.
- [17] R. C. Chen and C. H. Hsieh, "Web page classification based on a support vector machine using a weighted vote schema," *Expert Syst. Appl.*, vol. 31, no. 2, pp. 427–435, 2006.
- [18] S. Tan, "An improved centroid classifier for text categorization," *Expert Syst. Appl.*, vol. 35, no. 1–2, pp. 279–285, 2008.
- [19] E. Blanzieri and A. Bryl, "A survey of learning-based techniques of email spam filtering," *Artif. Intell. Rev.*, vol. 29, no. 1, pp. 63–92, 2008.
- [20] Q. Ye, Z. Q. Zhang, and R. D. A.-A. D. O.-10. 1016/j. eswa. 2008. 07. 03. Law, "Sentiment classification of online reviews to travel destinations by supervised machine learning approaches," *Expert Syst. Appl.*, vol. 36, p. 6527–6535 ST–Sentiment classification of online, 2009.
- [21] P.-Y. Hao, J.-H. Chiang, and Y.-K. Tu, "Hierarchically SVM classification based on support vector clustering method and its application to document categorization," *Expert Syst. Appl.*, vol. 33, no. 3, pp. 627–635, 2007.
- [22] A. Esuli, T. Fagni, and F. Sebastiani, "Boosting multi-label

- hierarchical text categorization,” *Inf. Retr. Boston.*, vol. 11, no. 4, pp. 287–313, 2008.
- [23] Y. Yang and X. Liu, “A re-examination of text categorization methods,” in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '99*, 1999, pp. 42–49.
- [24] A. P. de Vries, N. Mamoulis, N. Nes, and M. Kersten, “Efficient k-NN search on vertically decomposed data,” *Proc. 2002 ACM SIGMOD Int. Conf. Manag. data - SIGMOD '02*, p. 322, 2002.
- [25] L. R. Lu and H. Y. Fa, “A Density-Based Method for Reducing the Amount of Training Data in kNN Text Classification [J],” *J. Comput. Res. Dev.*, vol. 4, p. 3, 2004.
- [26] Z. Al Aghbari, “Array-index: a plug & search K nearest neighbors method for high-dimensional data,” *Data Knowl. Eng.*, vol. 52, pp. 333–352, 2005.
- [27] S. Wang, D. Li, X. Song, Y. Wei, and H. Li, “A feature selection method based on improved fisher’s discriminant ratio for text sentiment classification,” *Expert Syst. Appl.*, vol. 38, no. 7, pp. 8696–8702, 2011.
- [28] P. Koncz and J. Paralic, “An approach to feature selection for sentiment analysis,” *2011 15th IEEE International Conference on Intelligent Engineering Systems*. pp. 357–362, 2011.
- [29] T. Xu, Q. Peng, and Y. Cheng, “Identifying the semantic orientation of terms using S-HAL for sentiment analysis,” *Knowledge-Based Syst.*, vol. 35, pp. 279–289, 2012.
- [30] G. Forman, “An extensive empirical study of feature selection metrics for text classification,” *J Mach Learn Res*, vol. 3, pp. 1289–1305, 2003.
- [31] Y. N. Liu, G. Wang, H. L. Chen, H. Dong, X. D. Zhu, and S. J. Wang, “An Improved Particle Swarm Optimization for Feature Selection,” *J. Bionic Eng.*, vol. 8, no. 2, pp. 191–200, 2011.
- [32] C. Vercellis, *Business Intelligence: Data Mining and Optimization for Decision Making*. John Wiley and Sons, 2009.
- [33] Y. Yang and J. O. Pedersen, “A comparative study on feature selection in text categorization,” in *Proceedings of the 14 th International Conference on Machine Learning.*, 1997, pp. 412–420.
- [34] S. Tan and J. Zhang, “An empirical study of sentiment analysis for chinese documents,” *Expert Syst. Appl.*, vol. 34, no. 4, pp. 2622–2629, 2008.
- [35] D. T. Larose, *Data Mining Methodes And Model*. 2006.
- [36] F. Gorunescu, *Data Mining: Concepts, models and techniques*, 2011th ed. Springer, 2011.
- [37] W. C.-M. Liaw, Yi-Ching, Leou Maw-Lin, “Fast exact k nearest neighbors search using anorthogonal search tree,” *Pattern Recognit.*, vol. 43, no. 6, pp. 2351–2358, 2010.
- [38] Y. C. Liaw, C. M. Wu, and M. L. Leou, “Fast k-nearest neighbors search using modified principal axis search tree,” *Digit. Signal Process. A Rev. J.*, vol. 20, no. 5, pp. 1494–1501, 2010.
- [39] J. Han, M. Kamber, and J. Pei, *Data mining: concepts and techniques*, 3rd ed. Morgan Kaufmann, 2012.
- [40] M. M. Jain and P. V. Richariya, “An Improved Techniques Based on Naive Bayesian for Attack Detection,” *Int. J. Emerg. Technol. Adv. Eng. Website www.ijetae.com*, vol. 2, no. 1, pp. 324–331, 2250.
- [41] C. Supriyanto, N. Yusof, B. Nurhadiono, and Sukardi, “Two-level feature selection for naive bayes with kernel density estimation in question classification based on Bloom’s cognitive levels,” in *2013 International Conference on Information Technology and Electrical Engineering (ICITEE)*, 2013, pp. 237–241.
- [42] L. Maimon, Oded&Rokach, *Data mining and knowledge discovery handbook*. New York: Springer, 2010.
- [43] D. T. Larose, *Data Mining Methods and Model*. New Jersey: John Willey & Sons, Inc, 2006.
- [44] Sumanto, *Statistika Deskriptif*. Yogyakarta: Center of Academic Publishing Service, 2014.



Tyas Setiyorini. Batang, 12 Oktober 1986. Menempuh pendidikan S1 Sistem Informasi di STMIK Swadharma, Jakarta, S2 Magister Sistem Informasi di STMIK Nusa Mandiri Jakarta. Saat ini menjadi dosen di STMIK Nusa Mandiri Jakarta. Paper yang pernah dipublikasikan berjudul “Penerapan Metode Bagging untuk Mengurangi Data Noise pada Neural Network untuk Estimasi Kuat Tekan Beton”.