Techno.COM, Vol. 24, No. 1, Februari 2025: 56-67

Analisis Performa Deep Embedded Clustering untuk Pendeteksian Topik

Performance Analysis of Deep Embedded Clustering for Topic Detection

Danu Julian Cahyadi[1], Hendri Murfi[2], Yudi Satria[3], Sarini Abdullah[4], Yekti Widyaningsih[5]
[1,2,3,4,5]Departemen Matematika, Universitas Indonesia, Depok 16424
E-mail: [1]djcahyadi@sci.ui.ac.id, [2]hendri@ui.ac.id, [3]y-satria@ui.ac.id, [4]sarini@sci.ui.ac.id, [5]yekti@sci.ui.ac.id

Abstrak

Pendeteksian topik adalah solusi untuk mengungkap struktur laten dalam sebuah dokumen. Kerangka umum pendeteksian topik berbasis *clustering* terdiri dari dua langkah: pembelajaran representasi dan pendeteksian topik melalui *clustering*. Dalam penelitian ini, *Bidirectional Encoder Representations from Transformers* (BERT) digunakan untuk pembelajaran representasi karena BERT mampu menangkap konteks setiap kata berdasarkan kata-kata di sekitarnya. Representasi teks yang diperoleh dari BERT digunakan untuk pendeteksian topik dengan *clustering*. *Deep Embedded Clustering* (DEC) dan *Improved* DEC (IDEC) adalah model *clustering* berbasis *deep learning* yang digunakan dalam penelitian ini untuk pendeteksian topik. DEC dan IDEC mampu mengubah data ke dalam ruang dimensi yang lebih rendah serta mengoptimalkan *cluster* secara simultan. *Output* dari teknik *clustering* berupa kata-kata kunci yang menggambarkan setiap topik *cluster*. Setelah mendapat kata kunci yang mewakili topik, evaluasi model dilakukan dengan melakukan perbandingan nilai *topic coherence* menggunakan *Topic Coherence - Word2Vec* (TC-W2V) sebagai analisis kuantitatif. Penelitian ini merupakan perluasan dari penerapan DEC dan IDEC pada pendeteksian topik dengan menambahkan analisis visualisasi dan kata kunci. Simulasi menunjukkan bahwa DEC dan IDEC mengungguli *Uniform Manifold Approximation and Projection* (UMAP)-*based k-means* (UKM) dan *Eigenspace-Based Fuzzy C-Means* (EFCM) dari segi nilai TC-W2V, hasil visualisasi, dan kata kunci.

Kata kunci: analisis teks, *deep clustering*, pemrosesan teks

Abstract

Topic detection is a solution for uncovering the latent structure within a document. The general framework for clustering-based topic detection consists of two steps: representation learning and topic detection through clustering. In this study, Bidirectional Encoder Representations from Transformers (BERT) is used for representation learning because of its ability to capture the context of each word based on surrounding words. The text representations obtained from BERT are then used for topic detection through clustering. Deep Embedded Clustering (DEC) and Improved DEC (IDEC) are the deep learning-based clustering models used in this study for topic detection. DEC and IDEC are capable of simultaneously transforming data into a lower-dimensional space and optimizing clusters. The output of these clustering techniques consists of keywords representing topics from each cluster. After obtaining keywords representing each topic, evaluation is performed by comparing topic coherence values using Topic Coherence - Word2Vec (TC-W2V) for quantitative analysis. This research is an extension of the application of DEC and IDEC for topic detection by incorporating visualization and keyword analysis. Our simulations show that DEC and IDEC also outperform uniform manifold approximation and projection (UMAP)-based k-means (UKM) and Eigenspace-Based Fuzzy C-Means (EFCM) in terms of TC-W2V value, visualization results, and keywords

.

Keywords: deep clustering, text analysis, text processing

1. PENDAHULUAN

Semakin meningkatnya jumlah data teks memberikan tantangan bagi para peneliti dan praktisi untuk mendapatkan informasi pada data teks tersebut secara efisien. Analisis manual terhadap koleksi dokumen menjadi tidak efisien dan membutuhkan waktu yang lama karena terbatasnya kemampuan manusia untuk menganalisis informasi dari dokumen-dokumen tersebut. Pendeteksian topik adalah metode yang dapat menjadi solusi untuk mendapatkan struktur laten dalam dokumen [1]. Metode ini mengungkap struktur semantik laten untuk mendapatkan informasi dari dokumen. Metode pendeteksian topik ini sudah banyak digunakan pada berbagai bidang seperti analisis media [2] dan pada *smart city* [3] dengan memberikan penggambaran topik yang baik berdasarkan kata kunci yang menggambarkan setiap topiknya.

Metode pendeteksian topik terdiri dari dua tahap: pembentukan representasi teks dan dilanjutkan dengan algoritma pendeteksian topik. Ada dua metode pendeteksian topik konvensional: *Nonnegative Matrix Factorization* (NMF) [4] dan *latent Dirichlet allocation* (LDA) [1]. Dua metode tersebut menggunakan metode representasi dengan asumsi bahwa dokumen merupakan kumpulan kata yang setiap katanya memiliki bobot makna yang sama. Dalam dokumen yang lebih kompleks tentu tidak berlaku demikian sebab ada beberapa kata yang memiliki bobot yang lebih bermakna sesuai dengan konteks kalimatnya.

Pendeteksian topik dengan *clustering* dapat menggabungkan berbagai metode representasi teks sebelum dilakukan *clustering* untuk pendeteksian topik. *Output* dari pendeteksian topik dengan *clustering* merupakan kata kunci yang menggambarkan topik setiap *cluster*. Salah satu metode representasi teks lain yang dapat digunakan adalah *Bidirectional Encoder Representations from Transformers* (BERT). BERT merupakan metode representasi teks berbasis *transformer* yang dapat memahami konteks dan makna yang terkandung dalam dokumen [5]. Dengan kemampuannya yang baik untuk mendapatkan informasi dari kumpulan dokumen, BERT telah digunakan untuk analisis sentimen [6–9], membantu proses pembelajaran [10], dan bidang kesehatan [11]. Dalam pengaplikasiannya untuk representasi teks, BERT telah menunjukkan performa yang lebih baik dibandingkan metode representasi teks sebelumnya [6,12]. Representasi teks menggunakan BERT telah dikombinasikan dengan beberapa teknik *clustering* standar seperti *k-means* [13] dan *Fuzzy C-Means* (FCM) [3,14]. Pendeteksian topik yang mengkombinasikan BERT dengan *k-means* dan BERT dengan FCM belum menunjukan hasil yang optimal karena masih menggunakan pereduksian dimensi tanpa mempelajari pola data yang kompleks. Salah satu solusi yang dapat digunakan adalah menggunakan *deep learning.*

Deep learning merupakan sub bagian dari *machine learning* dan *artificial intelligence* yang memiliki kemampuan yang baik dalam mempelajari data yang diberikan [15,16]. Dengan menggunakan *artificial neural network, deep learning* mampu mengekstraksi fitur kompleks dari data secara otomatis, tanpa memerlukan intervensi manusia dalam proses pembelajaran [17,18]. Pendekatan ini telah terbukti efektif dalam berbagai aplikasi, seperti pengenalan gambar, prediksi pada *big data*, dan pemrosesan bahasa alami (*Natural Language Programming*, NLP). Keunggulannya terletak pada kemampuannya untuk bekerja dengan data yang tidak terstruktur, salah satunya seperti teks, serta memberikan hasil yang akurat ketika digunakan dalam tugas-tugas klasifikasi dan pengenalan pola [19–21]. Potensi dari *deep learning* memotivasi penelitian ini untuk mengembangkan metode yang menerapkan *deep learning* pada pendeteksian topik.

Penelitian ini memperluas dan menyempurnakan makalah konferensi [22] yang menggabungkan BERT dengan metode *deep learning-based clustering*, yaitu *Deep Embedded Clustering* (DEC) [23] dan *Improved* DEC (IDEC) [24] untuk pendeteksian topik. Penyempurnaan model dilakukan melalui analisis nilai *topic coherence* yang lebih mendalam untuk menghasilkan perbandingan kuantitatif yang lebih komprehensif. Perluasan analisis juga dilakukan dengan menambahkan analisis visualisasi *cluster* dan analisis kata kunci.

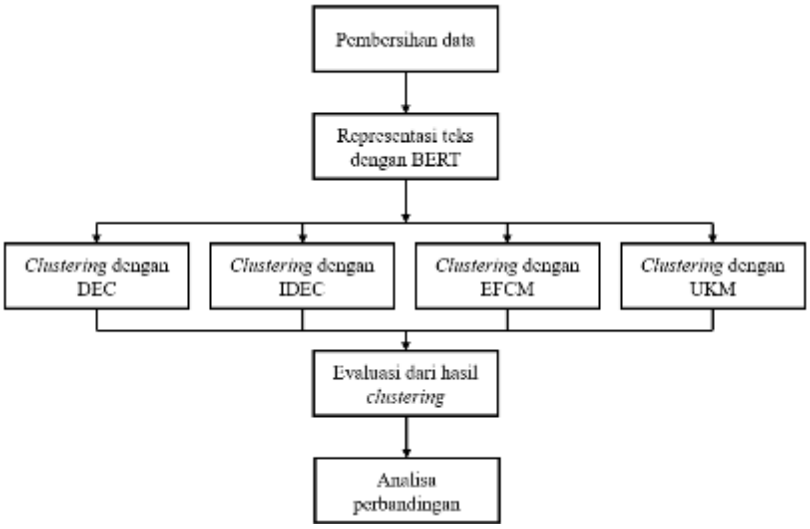
Penelitian ini bertujuan membandingkan kinerja DEC dan IDEC dengan metode pembanding, yaitu *Eigenspace-based Fuzzy C-Means* (EFCM) dan *Uniform Manifold Approximation and Projection* (UMAP)-*based K-Means* (UKM). Perbandingan dilakukan berdasarkan analisis nilai *topic coherence*, visualisasi *cluster*, dan analisis kata kunci. Sebagai

metode *clustering* yang telah mengintegrasikan pereduksian dimensi dan optimasi *clustering* secara simultan, DEC dan IDEC diharapkan memberikan performa yang lebih baik berdasarkan nilai *topic coherence* serta memberikan penggambaran visualisasi dan kata kunci yang lebih jelas.

Bagian selanjutnya dari makalah ini mencakup: Bagian 2 yang menjelaskan alur dan metode penelitian secara singkat. Bagian 3 memaparkan hasil dan pembahasan eksperimen. Terakhir, kesimpulan umum dibahas pada Bagian 4.

2. METODE PENELITIAN

Penelitian ini diawali dengan menggunakan BERT untuk membentuk representasi teks. Proses dilanjutkan dengan pendeteksian topik dengan *clustering* menggunakan DEC dan IDEC sebagai metode utama dan dibandingkan dengan EFCM dan UKM. Hasil dari *clustering* dievaluasi menggunakan *Topic Coherence – Word2Vec* (TC-W2V), lalu dilakukan perbandingan visualisasi *cluster* dan kata kunci dari model dengan nilai TC-W2V tertinggi.



Gambar 1 *Flowchart* penelitian

2.1 Representasi teks dengan BERT

BERT adalah model *pre-trained* bahasa yang dapat memahami makna kata-kata dalam sebuah dokumen berdasarkan konteksnya. BERT merupakan model dengan arsitektur *deep learning* yang menggunakan lapisan-lapisan *transformer encoder* sebagai arsitektur utamanya. BERT memiliki dua jenis model, yaitu BERT-*base* dan BERT-*large*. BERT-*base* terdiri atas 12 lapisan *encoder* yang setiap lapisannya memiliki *hidden layer* dengan ukuran dimensi 768 dan menggunakan 12 lapisan *multi-head self-attention*. Diperkirakan BERT-*base* menghasilkan 110 juta parameter. Sementara itu, BERT-*large* terdiri dari 24 lapisan *encoder*, dengan ukuran dimensi 1024 dan menggunakan 16 lapisan *multi-head self-attention*. Diperkirakan BERT-*large* menghasilkan 340 juta parameter [5].

Pemrosesan dokumen dengan BERT diawali dengan mengubah setiap kata dari dokumen menjadi kumpulan *token*. Setelah itu, kumpulan *token* diberikan *token* unik [CLS], [SEP], dan [PAD]. [CLS] diberikan pada awal dokumen, sedangkan [SEP] pada akhir dokumen atau di antara 2 buah dokumen. Apabila dokumen yang telah diubah memiliki jumlah *token* kurang dari 100, maka *token* [PAD] akan ditambahkan setelah *token* [SEP] hingga memiliki jumlah 100 *token*. Apabila dokumen tersebut memiliki lebih dari 100 *token*, maka hanya akan diambil 100 *token* pertama. Setelah itu, proses dilanjutkan ke tahap *embedding* untuk mengubah kumpulan *token* menjadi representasi vektor. Hasil akhir dari proses *embedding* berupa hasil penjumlahan dari tiga tahap *embedding*, yaitu *tokenization*, *segment* dan *position embedding*.

Dengan menggunakan *contextualized word embeddeding* pada model BERT-*base*, BERT dapat mengolah data *input* berupa teks dan hasilnya berupa vektor representasi dengan

ukuran dimensi 768. Hal ini juga disebut sebagai pendekatan berbasis fitur. Kata-kata yang memiliki makna serupa akan dipetakan dalam jarak yang dekat. Keuntungan dari pendekatan ini adalah BERT menjadi efisien dalam pengolahan data [12]. Dengan menggunakan model BERT-*base*, jika *input* teks ada sebanyak n, maka *output* akan berupa vektor dengan ukuran $n \times 768$.

2.2 DEC

DEC merupakan metode *clustering* berbasis *deep learning* yang menggunakan arsitektur *autoencoder* untuk mengelompokkan dan mempelajari data secara simultan. Langkah utama dari DEC adalah mereduksi dimensi data berdimensi tinggi menjadi data yang berdimensi lebih rendah lalu dilanjutkan dengan proses *clustering*. Dengan melakukan *clustering* pada dimensi data yang rendah, proses *clustering* menjadi lebih efisien.

Misalkan terdapat himpunan data $\{\boldsymbol{x}_i\}_{i=1}^{N}$ yang akan dikelompokan menjadi k *cluster*. Sebuah *cluster j* akan direpresentasikan oleh *centroid* $\{\boldsymbol{x}_i\}_{i=1}^{N}$ sebagai titik pusat. DEC diawali dengan melakukan transformasi non-linear $f_\theta: X \rightarrow Z$, dengan θ merupakan kumpulan parameter-parameter DEC yang akan dioptimasi dan Z adalah ruang yang memiliki dimensi lebih kecil dibandingkan dengan X. Pemetaan ini adalah pemetaan bagian *encoder* dari *autoencoder*. Hasil pemetaan dalam ruang Z tersebut akan digunakan untuk proses *clustering*.

Clustering menggunakan DEC dilakukan dengan cara mempelajari himpunan $\{\boldsymbol{\mu}_j\}_{j=1}^{k}$ dalam ruang laten Z serta mempelajari parameter θ dari *encoder* secara simultan. Dalam penggunaannya, DEC terdiri dari dua tahapan utama, yaitu tahap inisialisasi parameter *autoencoder* dan tahap optimasi [23]. Dalam proses optimasinya, DEC menghilangkan bagian *decoder* dan mempertahankan bagian *encoder* [24]. Arsitektur DEC memungkinkan model mempelajari pola kompleks pada data berdimensi tinggi. Pendekatan ini diharapkan memberikan hasil *clustering* yang lebih akurat dibandingkan dengan model lain yang belum mengintegrasikan proses pereduksian dimensi data dan *clustering* secara bersamaan, seperti EFCM dan UKM.

2.3 IDEC

IDEC merupakan pengembangan dari DEC dengan tidak menghilangkan bagian *decoder* pada arsitektur *autoencoder*. Penggunaan *decoder* akan mengurangi kemungkinan terjadinya distorsi pada ruang Z. Secara umum, IDEC memiliki langkah yang sama dengan DEC, yaitu proses diawali dengan mereduksi dimensi data menjadi data yang berdimensi rendah lalu dilanjutkan dengan *clustering*.

Misalkan terdapat himpunan data $\{\boldsymbol{x}_i\}_{i=1}^{N}$ yang akan dikelompokan menjadi k *cluster*. Sebuah *cluster j* akan direpresentasikan oleh *centroid* $\{\boldsymbol{x}_i\}_{i=1}^{N}$ sebagai titik pusat. IDEC diawali dengan melakukan transformasi non linier $f_\theta: X \rightarrow Z$, dengan θ merupakan kumpulan parameter-parameter DEC yang akan dioptimasi dan Z adalah ruang yang memiliki dimensi lebih kecil. Proses diteruskan dengan pemetaan $g_\theta: Z \rightarrow X'$ yang merupakan rekonstruksi X dari hasil reduksi dimensi ruang Z. Kedua pemetaan tersebut akan dioptimasi selama proses *clustering* [24]. Dengan mempertahankan arsitektur *autoencoder*, IDEC diharapkan mampu memberikan peningkatan hasil dibandingkan DEC.

2.4 EFCM

EFCM merupakan algoritma yang mengelompokkan data menjadi beberapa *cluster* menggunakan FCM yang dimensi datanya telah direduksi dengan *truncated singular value decomposition* (TSVD). Berbeda dengan DEC dan IDEC, EFCM melakukan pereduksian dimensi dan *clustering* secara terpisah. Misal, diberikan data $\{\boldsymbol{x}_i\}_{i=1}^{n}$ yang membentuk matriks $X = (\boldsymbol{x}_1\ \boldsymbol{x}_2\ \ldots \boldsymbol{x}_n)$. EFCM memanfaatkan TSVD untuk mengaproksimasi matriks X di dimensi yang lebih kecil. Secara umum aproksimasi matriks X digambarkan pada Persamaan 1.

$$X \approx U_{m\times p} \sum_{p\times p} V^T_{p\times q} \tag{1}$$

dengan $U_{m\times p}$ adalah matriks *truncated* dari matriks ortogonal U, $\sum_{p\times p}$ adalah matriks *truncated* dari matriks diagonal non negatif $\sum$, dan $V^T_{p\times q}$ adalah matriks *truncated* dari matriks V^T yang

mengandung nilai *eigen*, dimana berlaku $X = U\sum V^T$. Matriks $\sum_{p\times p} V^T_{p\times q}$ akan digunakan sebagai hasil pereduksian dimensi matriks X. Selanjutnya, FCM digunakan untuk proses *clustering* [14].

2.5 UKM

UKM merupakan kombinasi algoritma pereduksian dimensi UMAP dengan *k-means*. Misalkan diberikan sebuah himpunan data $\{\boldsymbol{x}_i\}_{i=1}^n$ yang membentuk matriks $X = (\boldsymbol{x}_1\ \boldsymbol{x}_2 \dots \boldsymbol{x}_n)$. X akan dipetakan oleh UMAP ke dimensi yang lebih rendah untuk digunakan pada proses *clustering* menggunakan *k-means*. Penggabungan ini telah digunakan pada penelitian yang dilakukan oleh [13] dan telah terbukti memberikan hasil yang baik dari segi akurasi dan waktu. Sama seperti EFCM, UKM melakukan pereduksian dimensi dan *clustering* secara terpisah.

2.6 Representasi Cluster

Representasi *cluster* menggunakan model C-TFIDF. Setiap *cluster* akan digabung menjadi satu lalu C-TFIDF akan digunakan sesuai dengan Persamaan 2 untuk mendapatkan kata dengan nilai tertinggi:

$$W_{t,c} = tf_{t,c} \times \log\left(1 + \frac{A}{tf_t}\right) \quad (2)$$

dengan $tf_{t,c}$ adalah jumlah kata t pada kelas c, A merupakan rata-rata banyaknya kata pada seluruh *cluster*, dan tf_t adalah jumlah kata pada sebuah *cluster* [13].

3. HASIL DAN PEMBAHASAN

Bagian ini menjelaskan data dan langkah eksperimen yang digunakan dalam penelitian. Penelitian ini menggunakan BERT untuk menghasilkan representasi teks serta menggunakan DEC, IDEC, EFCM, dan UKM untuk *clustering* dan membandingkan setiap hasilnya. Proses optimasi model dimulai dengan menentukan parameter untuk mencapai nilai *topic coherence* tertinggi, metrik ini akan dijelaskan pada bagian berikutnya. Hasil model terbaik dari setiap teknik *cluster* digunakan untuk analisis kualitatif berupa analisis kata kunci.

3.1 Data

Penelitian dimulai dengan pengumpulan dan pembersihan data. Data yang digunakan pada penelitian ini adalah: AGNews, R2, dan Yahoo! Answers. Data tersebut merupakan yang juga digunakan dalam [22]. Informasi data ditampilkan pada Tabel 1.

Tabel 1 Deskripsi Data Penelitian

Data	Deskripsi	Topik	Banyak Dokumen
AGNews	Terdiri dari judul, isi, dan topik berita dari AGNews	4	4000
Reuters (R2)	Terdiri dari dokumen dan topik berita Reuters-21578	2	5859
Yahoo! Answers	Terdiri dari topik, pertanyaan, serta jawaban dari Yahoo! Answers	10	10000

Setelah dibersihkan, data akan diproses oleh BERT untuk menghasilkan representasi teks. Model BERT yang digunakan adalah BERT-*base uncased*. Model ini tidak sensitif terhadap perbedaan kata dengan huruf kapital (misal, *english* dengan *English* dianggap memiliki makna yang sama), sehingga setiap kata harus diubah ke dalam bentuk *lowercase*. Data akan masuk ke dalam model tersebut yang terdiri dari 12 lapisan encoder, 768 ukuran tersembunyi, dan 12 kepala dari lapisan *multi-head self-attention*. Dengan melakukan *forward* data ke dalam model BERT, representasi data teks diperoleh dalam bentuk matriks berukuran $(n, 768)$ di mana n adalah jumlah dokumen dalam data dan 768 adalah ukuran lapisan tersembunyi. Representasi teks yang diperoleh digunakan untuk simulasi *clustering* dengan DEC, IDEC, EFCM, dan UKM.

3.2 Metode Clustering

Setiap metode *clustering* disimulasikan dengan 5 kali pengulangan untuk setiap kombinasi parameter untuk menangkap stabilitas model. Fase pra-pelatihan model DEC dan IDEC menggunakan parameter yang sama. Proses optimasi DEC dan IDEC dilakukan dengan memanfaatkan 30 kombinasi parameter yang dapat dilihat pada Tabel 2. Misalnya, arsitektur saat menggunakan parameter *default* dengan dimensi jaringan 3 lapis adalah: dimensi jaringan d-500-500-2000-5-2000-500-500, dengan d adalah dimensi ruang data, 500-500-2000 adalah jumlah *neuron autencoder* dalam jaringan 3 lapis, 5 adalah dimensi data setelah reduksi dimensi, dan 2000-500-500 adalah jumlah *neuron decoder* dalam jaringan 3 lapis. Dalam arsitektur tersebut, model menggunakan nilai *dropout rate* 0.2. Hasil pembelajaran digunakan pada DEC dan IDEC dengan parameter $\alpha = 1$ dan interval pembaruan 100 sebagai parameter *default* untuk *clustering*.

Fungsi aktivasi yang digunakan dalam arsitektur adalah ReLU. Selama pra-pelatihan, 10% data dialokasikan untuk validasi. Pelatihan akan berhenti ketika *validation loss* meningkat, dengan nilai *patience* sebesar 10, dan bobot terbaik yang memberikan *validation loss* terkecil akan disimpan sebagai keluaran akhir setelah pelatihan selesai. *Optimizer* yang digunakan adalah *Adaptive moment estimation* (Adam).

Tabel 2 Tabel deskripsi parameter DEC dan IDEC

No	Parameter	Pilihan Nilai Parameter
1	Dimensi laten *autoencoder*	**5**, 10, 15, 20, 25
2	Jumlah *neuron autoencoder*	[500, 500], [500,1000], [**500**, **2000**], [1000, 500], [1000,1000], [1000, 2000], [2000,500], [2000,1000], [2000,2000]
3	*Dropout rate autoencoder*	0.1, **0.2**, 0.3, 0.4, 0.5
4	α untuk *cluster assignment* DEC dan IDEC	**1**, 10, 100, 1000, 10000
5	Interval pembaruan DEC dan IDEC	**100**, 200, 300, 400, 500, 600

Angka yang ditebalkan merupakan parameter *default.*

Untuk UKM, parameter yang digunakan adalah: jumlah iterasi maksimum adalah 300 dan toleransi 10^{-4}. Untuk EFCM, jumlah dimensi yang diekstraksi dari SVD adalah 10, toleransi adalah 10^{-4}, dan nilai *fuzziness* yang disimulasikan dalam rentang (1, 1.1, 1.2, …, 2.5) untuk menangkap sensitivitas EFCM.

3.3 Metrik Penilaian

Untuk Evaluasi hasil simulasi dari metode yang telah dipaparkan pada Bagian 3 dijelaskan pada bagian ini. Dengan ketiga data yang digunakan, algoritma pendeteksian topik menghasilkan beberapa topik yang setiap topiknya direpresentasikan oleh 10 kata teratas dari setiap *cluster* [14]. Metode kuantitatif yang umum digunakan untuk mengukur interpretabilitas topik adalah *topic coherence*. *Topic coherence* yang digunakan dalam penelitian ini adalah *Topic Coherence – Word2Vec* (TC-W2V) [25]. Misalkan sebuah topik *t* terdiri dari *w* kata yang didefinisikan sebagai $\{t_1, t_2, \dots, t_w\}$. TC-W2V dari topik *t* dihitung sebagai berikut:

$$TC(t) = \frac{1}{\binom{w}{2}} \sum_{j=2}^{w} \sum_{i=1}^{j-1} \text{similarity}(\boldsymbol{wv}_j, \boldsymbol{wv}_i) \tag{3}$$

dengan $\boldsymbol{wv}_j$ dan $\boldsymbol{wv}_i$ merupakan vektor dari kata yang bersesuaian dari setiap topik. Vektor dari setiap kata dibentuk oleh model *word2vec* yang telah dilatih menggunakan data dari Google News yang berisi sekitar 100 triliun kata. TC-W2V digunakan untuk analisis kuantitatif berupa optimasi

parameter dan perbandingan antar model. Model dengan nilai TC-W2V tertinggi setelah proses optimasi dipilih untuk digunakan dalam analisis kualitatif berupa analisis kata kunci setiap model.

3.4 Hasil

Representasi teks yang didapat dari BERT digunakan untuk simulasi setiap teknik *clustering*, yaitu DEC, IDEC, EFCM, dan UKM. Simulasi akan diawali dengan proses optimasi DEC dan IDEC untuk mendapatkan model terbaik dari setiap kombinasi parameter. Lalu, proses dilanjutkan dengan perbandingan nilai TC-W2V, visualisasi *cluster*, dan perbandingan kualitatif dari setiap hasil model *clustering*.

Langkah pertama adalah melakukan optimasi parameter DEC dan IDEC. Untuk efisiensi proses komputasi, setiap parameter dioptimasi sesuai dengan Tabel 2, sementara parameter lainnya didefinisikan menggunakan nilai *default*. Selanjutnya, hasil dari DEC, IDEC, EFCM, dan UKM akan dibandingkan pada setiap data yang digunakan. Analisis dimulai dari perbandingan nilai TC-W2V.

Hasil optimasi parameter DEC pada ketiga data dapat dilihat pada Gambar 2. Parameter optimal untuk data AGNews tercapai ketika jumlah *neuron* pada *autoencoder* adalah [2000,2000] dengan TC-W2V sebesar 0.3511, untuk data R2 tercapai dengan jumlah neuron pada *autoencoder* [1000,500] dengan TC-W2V sebesar 0.3797, dan untuk data Yahoo! Answers tercapai ketika *dropout rate* pada *autoencoder* adalah 0.5 dengan TC-W2V sebesar 0.4141.

Untuk hasil optimasi parameter IDEC pada ketiga dataset dapat dilihat pada Gambar 3. Parameter optimal untuk data AGNews tercapai ketika nilai α adalah 10 dengan TC-W2V sebesar 0.3823, untuk data R2 dengan parameter *default* menghasilkan TC-W2V sebesar 0.4117, dan untuk data Yahoo! Answers tercapai ketika jumlah *neuron* pada *autoencoder* adalah [1000,2000] dengan TC-W2V sebesar 0.4112.

Hasil optimasi tersebut dibandingkan dengan hasil *clustering* menggunakan EFCM dan UKM. Tabel perbandingan antara DEC, IDEC, EFCM, dan UKM dapat dilihat pada Tabel 3. DEC dan IDEC memberikan TC-W2V yang lebih tinggi dibandingkan dengan EFCM dan UKM pada ketiga data. DEC dan IDEC menghasilkan TC-W2V setidaknya 25% lebih tinggi jika dibandingkan dengan EFCM, dan setidaknya 85% lebih tinggi jika dibandingkan dengan UKM pada ketiga data. Jika dibandingkan dengan DEC, IDEC memberikan nilai TC-W2V yang lebih tinggi pada ketiga data.

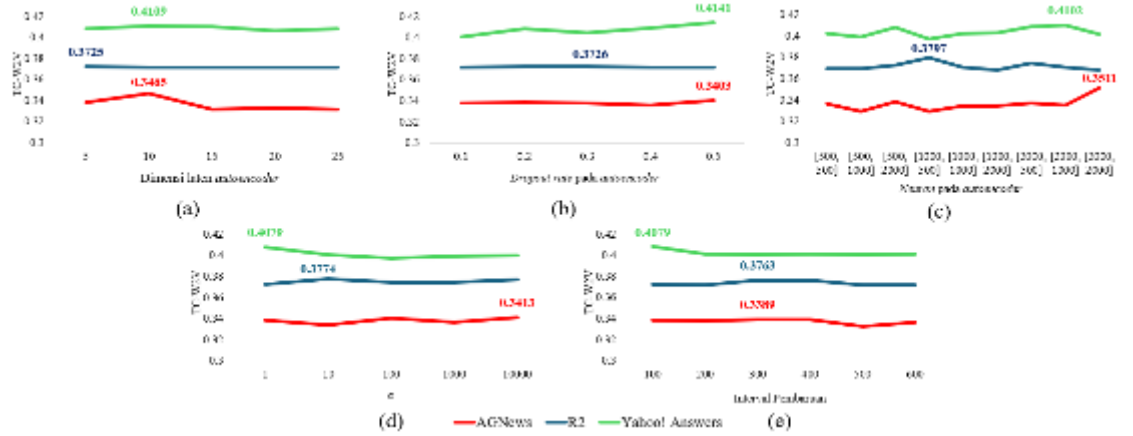
Selain itu, pada dataset yang sama, perubahan nilai TC-W2V dalam optimasi parameter DEC dan IDEC lebih kecil dibandingkan dengan EFCM. Perubahan terbesar nilai TC-W2V pada DEC dan IDEC adalah 0.02639 atau sekitar 7.4%, sedangkan EFCM mengalami perubahan nilai TC-W2V sebesar 0.0748 ketika menggunakan nilai *fuzziness* 1.1 dibandingkan 1.6, atau sekitar 27.5%. Hal ini menunjukkan bahwa DEC dan IDEC lebih tahan atau *robust* terhadap perubahan parameter yang digunakan.

Tabel 3 Tabel nilai TC-W2V dari model DEC, IDEC, EFCM, UKM dengan menggunakan data AGNews, R2, dan Yahoo! Answers (disingkat menjadi Yahoo!)

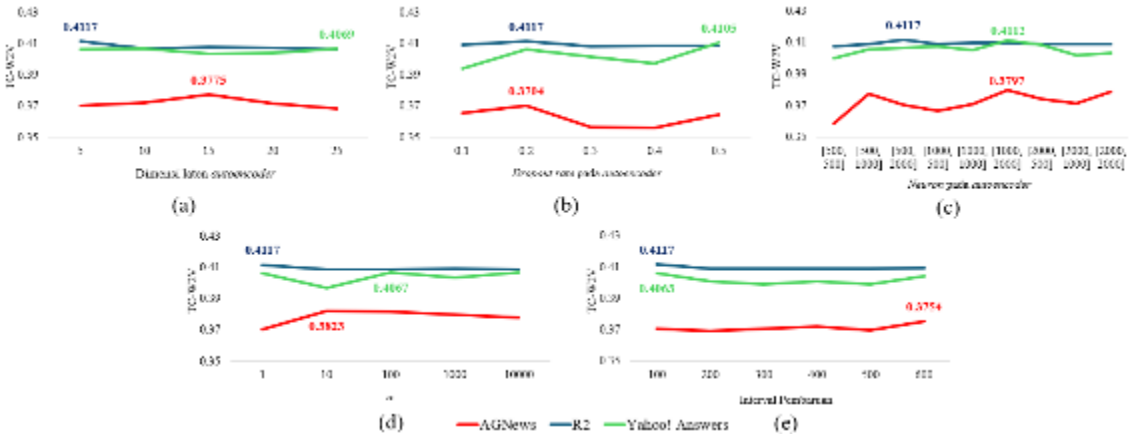
Model	**Nilai TC-W2V**		
	AGNews	**R2**	**Yahoo!**
DEC *dropout rate* 0.5	0.3403	0.3715	0.4141
IDEC *neuron* [1000,2000]	0.3797	0.4100	0.4112
DEC *default*	0.3385	0.3725	0.4080
IDEC *default*	0.3704	0.4117	0.4063
EFCM *fuzziness* 1.1	0.2724	0.2724	0.2775
EFCM *fuzziness* 1.5	0.2805	0.3344	0.3045
EFCM *fuzziness* 1.6	0.2988	0.3472	0.2968
UKM	0.1823	0.2111	0.2714

Dari hasil analisis TC-W2V, didapat bahwa DEC dan IDEC memberikan hasil yang lebih baik dari EFCM dan UKM pada tiga data yang digunakan. Model dengan TC-W2V tertinggi

selanjutnya digunakan untuk perbandingan hasil visualiasi *cluster* dari setiap hasil model pada tiga data dan contoh kata kunci pada data AGNews.



Gambar 2 Hasil Simulasi Sensitivitas Parameter: (a) Dimensi Laten *Autoencoder*, (b) *Dropout Rate* pada *Autoencoder*, (c) Neuron pada A*utoencoder*, (d) Nilai α pada DEC, dan (e) Interval Pembaruan pada IDEC dengan Data AGNews, R2, dan Yahoo! Answers.



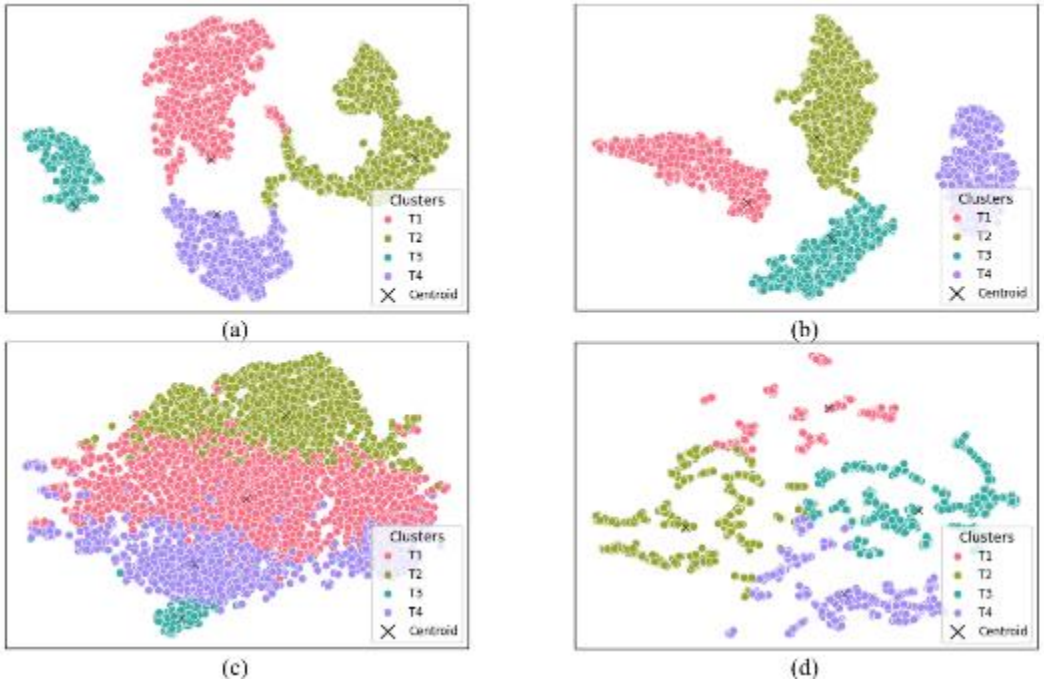
Gambar 3 Hasil Simulasi Sensitivitas Parameter: (a) Dimensi Laten *Autoencoder*, (b) *Dropout Rate* pada *Autoencoder*, (c) Neuron pada A*utoencoder*, (d) Nilai α pada IDEC, dan (e) Interval Pembaruan pada IDEC dengan Data AGNews, R2, dan Yahoo! Answers.

Dari hasil visualisasi pada Gambar 4 – Gambar 6, DEC dan IDEC mampu memisahkan setiap *cluster* dengan lebih baik dibandingkan dengan EFCM dan UKM pada seluruh data. DEC dan IDEC menunjukan celah yang jelas dan luas untuk memisahkan setiap *cluster*, meskipun masih terdapat sebagian kecil area *cluster* yang tercampur, terutama di bagian tengah visualisasi *cluster* pada Gambar 4 dan beberapa area kecil pada Gambar 6. Sebaliknya, EFCM dan UKM memiliki lebih banyak area yang tercampur dan tidak menunjukkan batasan area yang jelas.

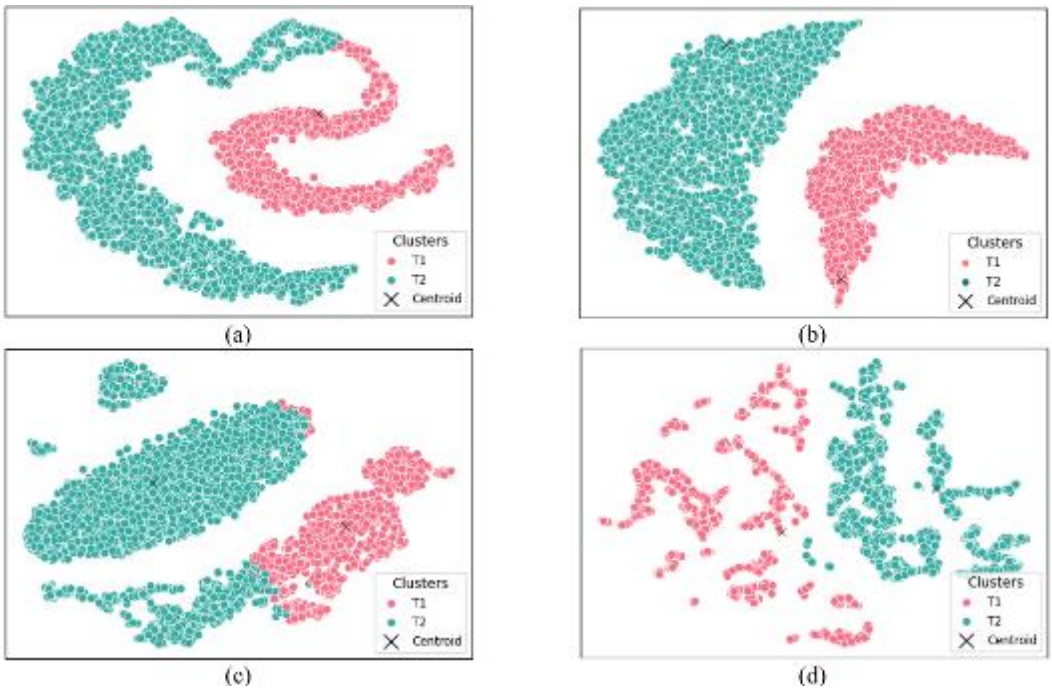
Secara umum, DEC dan IDEC mampu menghasilkan visualisasi *cluster* yang baik dengan memberikan celah yang jelas sebagai pembeda antar *cluster*, bahkan pada data dengan jumlah *cluster* yang lebih banyak. Selain itu, performa DEC dan IDEC yang stabil dalam memisahkan *cluster* pada seluruh data menjadi keunggulan penting. Di sisi lain, EFCM dan UKM mampu memetakan *cluster* dengan baik pada data AGNews dan R2, tetapi tidak dapat menghasilkan pemisahan yang serupa pada data Yahoo! Answers dengan jumlah topik yang lebih banyak. Selanjutnya, hasil ekstraksi kata kunci dari setiap teknik *clustering* dengan data AGNews dapat dilihat pada Tabel 4.

Dari hasil ekstraksi kata kunci, kata kunci yang dibentuk oleh DEC dan IDEC lebih mencerminkan topik yang relevan dibandingkan dengan EFCM dan UKM. Sebagai contoh, pada topik "*sports*," kata kunci dan urutan kata kunci dari DEC dan IDEC lebih menggambarkan topik

"*sports*" dibandingkan dengan EFCM dan UKM. Secara spesifik, kata kunci pada IDEC mengandung kata "*sports*", yang dengan jelas mencerminkan topik tersebut.

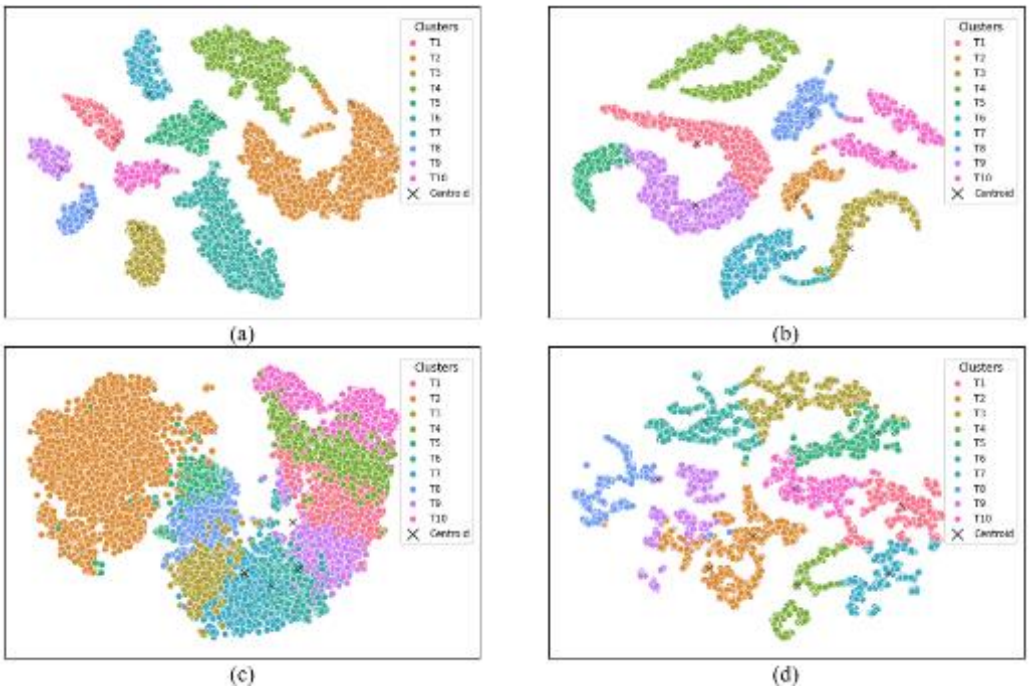


Gambar 4 Visualisasi Hasil *Clustering* dari Model (a) DEC, (b) IDEC, (c) EFCM, dan (d) UKM pada data AGNews



Gambar 5 Visualisasi Hasil *Clustering* dari Model (a) DEC, (b) IDEC, (c) EFCM, dan (d) UKM pada data R2

Berdasarkan perbandingan nilai TC-W2V, perbandingan hasil visualisasi, dan hasil ekstraksi kata kunci, DEC dan IDEC sebagai metode *clustering* berbasis *neural network* memberikan performa yang lebih baik dibandingkan dengan EFCM dan UKM. Kemampuan DEC dan IDEC yang dapat mempelajari fitur untuk pereduksian dimensi serta mengoptimasi hasil *cluster* secara simultan berkontribusi pada baiknya performa DEC dan IDEC jika dibandingkan dengan EFCM dan UKM yang melakukan pereduksian dimensi dan *clustering* secara terpisah.



Gambar 6 Visualisasi Hasil *Clustering* dari Model (a) DEC, (b) IDEC, (c) EFCM, dan (d) UKM pada data Yahoo! Answers

Tabel 4 Kata Kunci dari Model DEC, IDEC, EFCM, dan UKM dengan Nilai TC-W2V Tertinggi pada Data AGNews

Topik	Model	Kata Kunci
Sports	DEC	*Game, team, season, victory, first, win, league, cup, last, new*
Sports	IDEC	*Game, season, team, victory, win, first, league, cup, last, sports*
Sports	EFCM	*Game, world, AP, new, team, season, second, league, win, united*
Sports	UKM	*Team, game, season, first, new, league, win, last, cup, victory*
World	DEC	*President, Iraq, minister, reuters, prime, US, government, killed, people, AP*
World	IDEC	*President, reuters, US, AFP, minister, new, government, people, Iraq, Inc*
World	EFCM	*President, Iraq, minister, reuters, prime, bush, people, killed, Baghdad, government*
World	UKM	*President, Iraq, minister, reuters, prime, people, killed, AP, government, Baghdad*
Business	DEC	*New, inc, company, reuters, corp, york, percent, million, microsoft, software*
Business	IDEC	*Reuters, new, york, inc, company, corp, percent, oil, million, billion*
Business	EFCM	*Oil, reuters, new, york, prices, percent, quarter, stocks, rose, crude*
Business	UKM	*New, reuters, oil, US, york, company, prices, percent, inc, million*
Sci-tech	DEC	*New, US, AP, company, space, internet, music, government, world, first*
Sci-tech	IDEC	*New, US, company, government, AP, first, internet, software, music, space*
Sci-tech	EFCM	*Company, software, microsoft, corp, new, internet, million, business, oracle, billion*
Sci-tech	UKM	*Software, microsoft, new, internet, inc, company, computer, service, music, corp*

4. KESIMPULAN DAN SARAN

Tujuan penelitian ini adalah membandingkan performa DEC dan IDEC dengan EFCM dan UKM berdasarkan hasil analisis kuantitatif serta menambahkan analisis kualitatif berupa analisis kata kunci dari model. Secara kuantitatif, DEC dan IDEC menunjukkan performa yang lebih baik dari nilai TC-W2V yang lebih tinggi serta kestabilan perubahan nilai TC-W2V dibandingkan EFCM dan UKM pada ketiga data. Hal ini juga tercermin pada hasil visualisasi *cluster*. DEC dan IDEC memberikan pemisahan antar *cluster* yang jelas, disertai dengan hanya ada sedikit anggota-anggota *cluster* yang tercampur. Sebaliknya, EFCM dan UKM menghasilkan hasil visualisasi dengan banyak area yang masih tercampur dengan anggota *cluster* lain dan memberikan batasan yang kurang jelas untuk memisahkan setiap *cluster*. Hasil ini menunjukkan kemampuan DEC dan IDEC yang lebih baik dalam memisahkan setiap *cluster* pada ketiga data.

Tingginya nilai TC-W2V dan jelasnya hasil pemisahan visualisasi dari DEC dan IDEC juga tercermin pada kata kunci setiap *cluster* yang dihasilkan oleh kedua model ini, yang lebih menggambarkan topik-topik yang relevan. Berdasarkan nilai TC-W2V, visualisasi *cluster*, dan kata kunci yang dihasilkan, dapat disimpulkan bahwa DEC dan IDEC, sebagai model *clustering* berbasis *deep learning*, memiliki performa yang lebih baik dibandingkan dengan EFCM dan UKM. Kemampuan DEC dan IDEC dalam mempelajari data tekstual serta mengoptimalkan hasil *cluster* secara simultan menjadikan DEC dan IDEC unggul dibandingkan dengan metode *clustering* standar seperti EFCM dan UKM.

Meskipun hasil menunjukkan keunggulan DEC dan IDEC, terdapat beberapa langkah yang dapat diambil untuk memperluas cakupan penelitian ini. Salah satunya adalah dengan mencoba optimasi parameter lain pada model DEC dan IDEC untuk mengevaluasi pengaruhnya. Selain itu, penelitian dapat dilanjutkan dengan menggunakan data dengan tingkat *noise* yang lebih tinggi untuk menguji kestabilan model terhadap data yang lebih kompleks dan beragam.

DAFTAR PUSTAKA

[1] Blei DM, Ng AY, Jordan MI. Latent Dirichlet Allocation. Journal of Machine Learning Research 2003;3:993–1022.

[2] Aji PM, Nadhila V, Sanny L. Effect of social media marketing on instagram towards purchase intention: Evidence from Indonesia's ready-to-drink tea industry. International Journal of Data and Network Science 2020;4:91–104. https://doi.org/10.5267/j.ijdns.2020.3.002.

[3] Parlina A, Ramli K, Murfi H. Exposing emerging trends in smart sustainable city research using deep autoencoders-based fuzzy c-means. Sustainability (Switzerland) 2021;13:1–28. https://doi.org/10.3390/su13052876.

[4] Paatero P, Tappert U. Positive Matrix Factorization: A Non-negative Factor Model With Optimal Utilization of Error Estimates of Data Values. Environmetrics 1994;5:111–26.

[5] Devlin J, Chang M-W, Lee K, Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 1, 2019, p. 4171–86.

[6] Murfi H, Gowandi T, Syamsyuriani, Ardaneswari G, Nurrohmah S. BERT-based combination of convolutional and recurrent neural network for indonesian sentiment analysis. Appl Soft Comput 2024;151. https://doi.org/10.1016/j.asoc.2023.111112.

[7] Thomson M, Murfi H, Ardaneswari G. BERT-Based Hybrid Deep Learning with Text Augmentation for Sentiment Analysis of Indonesian Hotel Reviews. Proceedings of the 12th International Conference on Data Science, Technology and Applications (DATA 2023), INSTICC; 2023, p. 468–73. https://doi.org/10.5220/0012127400003541.

[8] Dhanalakshmi P, Reddy UJ, Ravikanth G, Samathoti P, Ramu G. COVID-19 Twitter Data Analysis Using LSTM and BERT Techniques. International Journal of Engineering

Trends and Technology 2024;72:219–28. https://doi.org/10.14445/22315381/IJETT-V72I1P122.

[9] Mandhasiya DG, Murfi H, Bustamam A. The hybrid of BERT and deep learning models for Indonesian sentiment analysis. Indonesian Journal of Electrical Engineering and Computer Science 2024;33:591–602. https://doi.org/10.11591/ijeecs.v33.i1.pp591-602.

[10] Xu Z, Zhu P. Using BERT-Based Textual Analysis to Design a Smarter Classroom Mode for Computer Teaching in Higher Education Institutions. International Journal of Emerging Technologies in Learning (IJET) 2023;18:114–27. https://doi.org/10.3991/ijet.v18i19.42483.

[11] Babu A, Boddu SB. BERT-Based Medical Chatbot: Enhancing Healthcare Communication through Natural Language Understanding. Exploratory Research in Clinical and Social Pharmacy 2024;13. https://doi.org/10.1016/j.rcsop.2024.100419.

[12] Subakti A, Murfi H, Hariadi N. The performance of BERT as data representation of text clustering. J Big Data 2022;9. https://doi.org/10.1186/s40537-022-00564-9.

[13] Grootendorst M. BERTopic: Neural topic modeling with a class-based TF-IDF procedure. ArXiv 2022.

[14] Murfi H, Rosaline N, Hariadi N. Deep autoencoder-based fuzzy c-means for topic detection. Array 2022;13. https://doi.org/10.1016/j.array.2021.100124.

[15] Alzubaidi L, Zhang J, Humaidi AJ, Al-Dujaili A, Duan Y, Al-Shamma O, et al. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. J Big Data 2021;8. https://doi.org/10.1186/s40537-021-00444-8.

[16] Sarker IH. Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. SN Comput Sci 2021;2. https://doi.org/10.1007/s42979-021-00815-1.

[17] Qaddoura R, Al-Zoubi AM, Faris H, Almomani I. A multi-layer classification approach for intrusion detection in iot networks based on deep learning. Sensors 2021;21. https://doi.org/10.3390/s21092987.

[18] Zeng R, Liao M. Developing a Multi-Layer Deep Learning Based Predictive Model to Identify DNA N4-Methylcytosine Modifications. Front Bioeng Biotechnol 2020;8. https://doi.org/10.3389/fbioe.2020.00274.

[19] Choudhary K, DeCost B, Chen C, Jain A, Tavazza F, Cohn R, et al. Recent advances and applications of deep learning methods in materials science. NPJ Comput Mater 2022;8. https://doi.org/10.1038/s41524-022-00734-6.

[20] Taye MM. Understanding of Machine Learning with Deep Learning: Architectures, Workflow, Applications and Future Directions. Computers 2023;12. https://doi.org/10.3390/computers12050091.

[21] Naskath J, Sivakamasundari G, Begum AAS. A Study on Different Deep Learning Algorithms Used in Deep Neural Nets: MLP SOM and DBN. Wirel Pers Commun 2023;128:2913–36. https://doi.org/10.1007/s11277-022-10079-4.

[22] Cahyadi DJ, Murfi H, Satria Y, Abdullah S, Widyaningsih Y. BERT-Based Deep Embedded Clustering for Topic Modeling. International Conference on Computer, Control, Informatics and its Applications (IC3INA), 2024, p. 331–6. https://doi.org/10.1109/IC3INA64086.2024.10732729.

[23] Xie J, Girshick R, Farhadi A. Unsupervised Deep Embedding for Clustering Analysis. Proceedings of The 33rd International Conference on Machine Learning, in Proceedings of Machine Learning Research, vol. 48, 2016, p. 478–87.

[24] Guo X, Gao L, Liu X, Yin J. Improved Deep Embedded Clustering with Local Structure Preservation. Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, 2017, p. 1753–9.

[25] O'Callaghan D, Greene D, Carthy J, Cunningham P. An analysis of the coherence of descriptors in topic modeling. Expert Syst Appl 2015;42:5645–57. https://doi.org/10.1016/j.eswa.2015.02.055.