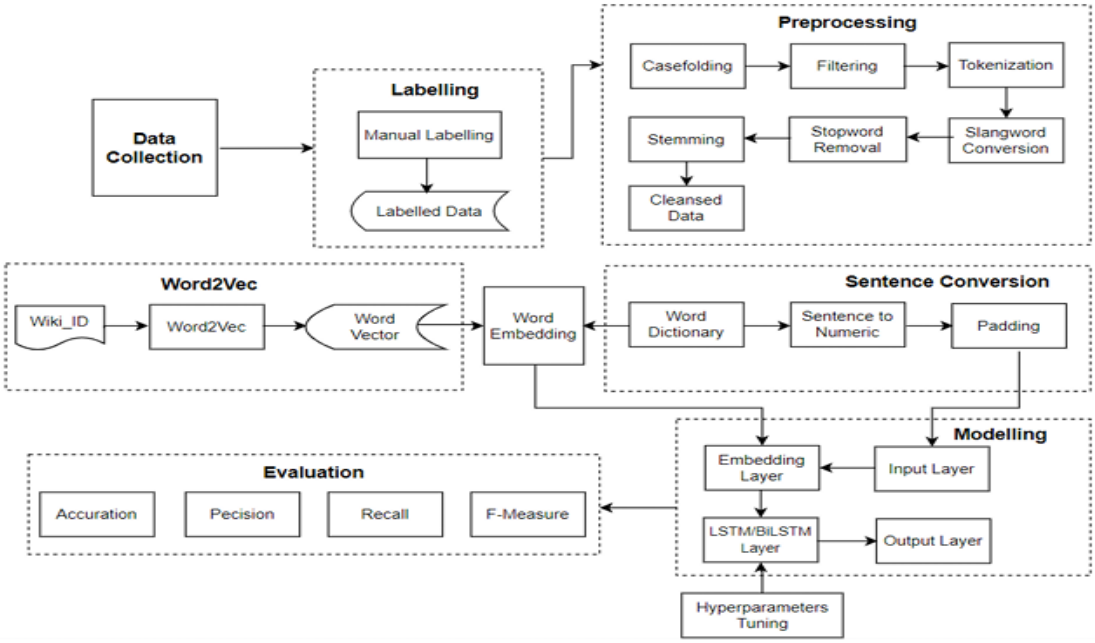
BAB II
KAJIAN PUSTAKA

2.1 Tinjauan Pustaka

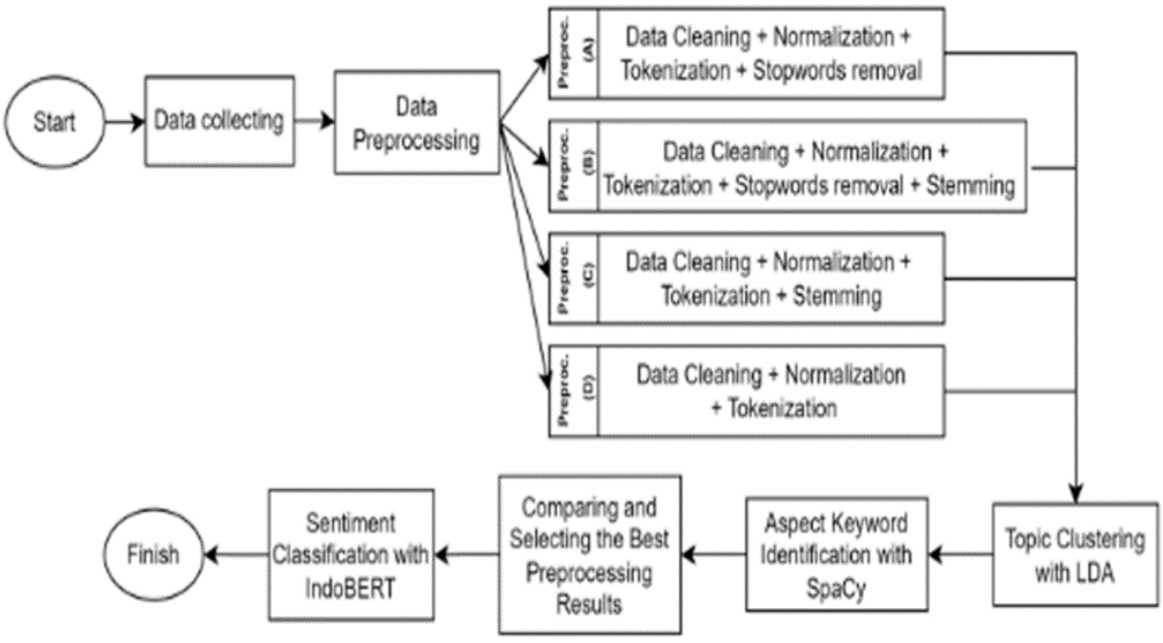
Penelitian yang dilakukan oleh Af'idah, dkk., (2023) yaitu mengembangkan ABSA pada ulasan objek wisata Indonesia menggunakan model Bi-LSTM dan LSTM dengan menerapkan strategi relevansi biner. Hasil penelitian menunjukkan bahwa Bi-LSTM mengungguli LSTM dalam klasifikasi aspek dan sentimen, namun skor F1 masih rendah, terutama pada aspek aksesibilitas**,** fasilitas, dan akomodasi, yang mengindikasikan adanya ketidakseimbangan data dalam dataset. Perbedaan hasil antara LSTM dan Bi-LSTM tidak terlalu signifikan dalam beberapa aspek, menunjukkan bahwa pemrosesan data secara bidirectional tidak selalu memberikan peningkatan besar pada klasifikasi yang lebih sederhana. Kekurangan utama dalam penelitian ini adalah kurangnya penanganan terhadap masalah ketidakseimbangan dataset, yang bisa mempengaruhi kinerja model dalam tugas klasifikasi sentimen dan aspek secara lebih umum sebagaimana ditunjukkan pada Gambar 2.1.



Gambar 2. 1 Pengembangan ABSA LSTM (Af'idah, dkk., 2023).

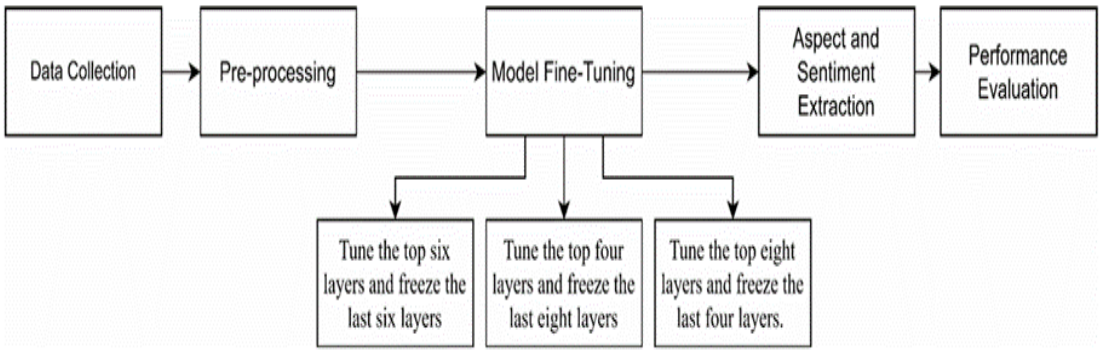
Penelitian yang dilakukan oleh Trisna, dkk, (2024) menerapkan ABSA pada cuitan yang diposting oleh Kementerian Keuangan Indonesia di Twitter. Topik utama dari cuitan dianalisis menggunakan LDA**,** sedangkan LSTM dan Bi-LSTM diterapkan untuk mengklasifikasikan sentimen berbasis aspek. Hasil analisis menunjukkan bahwa Bi-LSTM lebih unggul dibandingkan LSTM, dalam mengklasifikasikan sentimen terkait aspek kebijakan ekonomi, anggaran, dan pegawai, memberikan pemahaman yang lebih baik terhadap persepsi publik terhadap Kemenkeu**.** Penelitian ini memiliki beberapa kelemahan utama yaitu ketidakseimbangan data dalam dataset menyebabkan beberapa kategori sentimen tidak terdeteksi dengan akurat, yang mengurangi efektivitas model dalam menganalisis sentimen secara keseluruhan. Perlu pendekatan yang lebih mendalam dalam menangani masalah ketidakseimbangan dataset. Oleh karena itu, penelitian selanjutnya disarankan untuk menggunakan teknik yang lebih efektif dalam mengatasi ketidakseimbangan data, seperti penerapan teknik oversampling atau undersampling untuk meningkatkan akurasi deteksi sentimen.

Adapun penelitian secara spesifik dapat dilihat pada Gambar 2.2.



Gambar 2. 2 Pengembangan ABSA LDA (Trisna, dkk, 2024)

Penelitian yang dilakukan oleh Perwiran, dkk., (2025) berhasil melakukan fine-tuning IndoBERT, sebuah model *pre-trained* berbasis bahasa Indonesia, untuk melakukan ABSA pada ulasan wisata Indonesia. Proses pre-processing dilakukan dengan tahap normalisasi teks, tokenisasi, penghapusan stopwords, dan stemming. *Fine-tuning* IndoBERT dilakukan dengan eksperimen pada konfigurasi lapisan model, dengan hasil terbaik dicapai pada konfigurasi yang membekukan enam lapisan bawah dan menyesuaikan enam lapisan atas. Kelemahan Penelitian ini yaitu membedakan sentimen netral dan negatif, yang menyebabkan beberapa ulasan dengan sentimen netral salah diklasifikasikan sebagai negatif. Model menghadapi tantangan dalam mengelola ulasan dengan banyak aspek atau sentimen campuran, dimana sentimen pada aspek yang berbeda dalam ulasan tidak selalu tercermin dengan tepat. Hal ini mengindikasikan bahwa model perlu diperbaiki untuk menangani sentimen dengan intensitas rendah atau segi-segi perbedaan dalam perasaan secara lebih efektif. Penggunaan teknik segmentasi aspek dan analisis intensitas sentimen diusulkan untuk meningkatkan kemampuan model dalam mendeteksi sentimen yang lebih kompleks dan beragam. Adapun teknik finetuning ditunjukkan pada Gambar 2.3.



Gambar 2. 3 ABSA *finetuning* IndoBERT (Perwira et al., 2025).

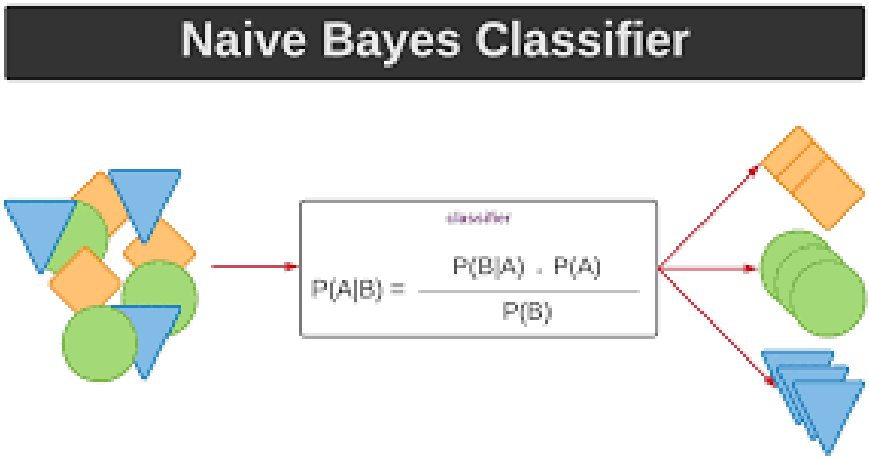
Klasifikasi dengan menggunakan metode *TF-IDF* dan *Naïve Bayes Classifier*, dilakukan pengujian terhadap seluruh data yang akan diklasifikasikan pada setiap semester yang menghasilkan nilai rata – rata yang menghasilkan nilai *accuracy* sebesar 80,10% serta pengujian terhadap *precision* sebesar 80,30%, sedangkan peningkatan terhadap nilai *recall* sebesar 80,30%, selain itu

peningkatan terhadap value *F1-Score* tertinggi dibandingkan dengan uji coba data lain sebesar 80%. Untuk menghasilkan sebuah informasi yang mudah diakses maka dibuatlah *dashboard* yang memuat beberapa tabel yang dapat digunakan untuk menampilkan seluruh komentar dari mahasiswa, untuk menampilkan data tertentu diperlukan penyaringan berdasarkan id dosen dan kode matakuliah, sedangkan untuk memperjelas tampilan data maka diperlukan 2 buah diagram batang yang dapat membedakan penampilan hasil analisis sentiment terhadap komentar mahasiswa berdasarkan id dosen dan kode mata kuliah (Sriwulan U. T., dkk., 2019).

Naive Bayes adalah algoritma klasifikasi probabilistik, memanfaatkan teorema Bayes untuk menentukan probabilitas suatu kelas berdasarkan fitur-fitur yang diamati. Algoritma ini mengasumsikan bahwa setiap fitur dalam dataset tidak saling bergantung satu sama lain, sebuah asumsi yang disederhanakan namun seringkali efektif dalam praktiknya. Teorema ini menjelaskan hubungan antara probabilitas Prior dan probabilitas Posterior suatu kejadian. Penghitungan ini melibatkan perkalian probabilitasPrior kelas dengan probabilitasLikelihood, kemudian dibagi dengan probabilitas Evidence. *Naive Bayes* bekerja dengan membangun model probabilitas untuk setiap kelas. Model ini mencakup probabilitasPrior kelas dan *probabilitasLikelihood* setiap *fiturGiven* kelas. Proses pelatihan model melibatkan penghitungan frekuensi kemunculan setiap fitur dalam setiap kelas dalam data pelatihan. Frekuensi ini kemudian dinormalisasi untuk menghasilkan probabilitas. Setelah model terbentuk, model tersebut dapat digunakan untuk mengklasifikasikan data baru. Data baru dievaluasi probabilitasnya untuk setiap kelas, dan kelas dengan probabilitas tertinggi ditetapkan sebagai kelas data baru tersebut. Salah satu keunggulan utama Naive Bayes terletak pada kemampuannya menangani data berdimensi tinggi.

Keunggulan lain *Naive Bayes* adalah kecepatan dan efisiensinya. Proses pelatihan dan klasifikasi dengan *Naive Bayes* relatif cepat dibandingkan dengan beberapa algoritma klasifikasi lainnya. Hal ini menjadikannya pilihan yang tepat untuk aplikasi yang membutuhkan respons cepat, seperti klasifikasi email spam atau analisis sentimen real-time. Kecepatan ini juga memungkinkan penggunaan *Naive Bayes* pada dataset yang sangat besar. Meskipun memiliki keunggulan,

Naive Bayes juga memiliki keterbatasan. Asumsi independensi fitur, meskipun menyederhanakan perhitungan, dapat menjadi masalah jika fitur-fitur tersebut sebenarnya saling bergantung. Dalam kasus seperti itu, kinerja *Naive Bayes* dapat menurun. Selain itu, *Naive Bayes* sensitif terhadap data pelatihan. Jika data pelatihan tidak representatif atau mengandung bias, model yang dihasilkan juga akan bias. Adapun proses algoritma *Naïve Bayes* dapat dilihat pada Gambar 2.4.



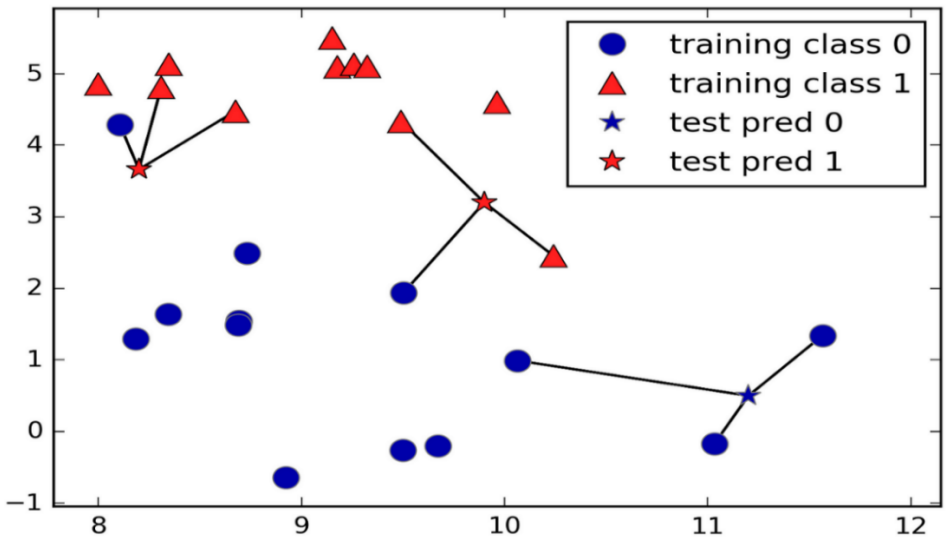
Gambar 2. 4 Struktur algoritma *naïve bayes*.

K-Nearest Neighbor (KNN) merupakan algoritma pembelajaran mesin yang sederhana namun kuat, termasuk dalam kategori pembelajaran supervised. KNN bekerja berdasarkan prinsip kedekatan data. Algoritma ini mengklasifikasikan data baru berdasarkan mayoritas kelas dari k tetangga terdekatnya dalam data pelatihan. Konsep ini intuitif dan mudah dipahami, menjadikannya pilihan populer untuk berbagai masalah klasifikasi. Inti dari KNN terletak pada konsep "jarak". Algoritma ini mengukur jarak antara data baru dan semua data dalam data pelatihan. Pengukuran jarak ini dapat dilakukan dengan berbagai cara, tergantung pada jenis data.

Jarak *Euclidean* adalah salah satu metode yang umum digunakan untuk data numerik. Metode lain seperti jarak Manhattan atau Minkowski juga dapat diterapkan sesuai kebutuhan. Setelah jarak antara data baru dan semua data pelatihan dihitung, algoritma KNN memilih k data terdekat. Nilai k merupakan

parameter penting dalam KNN. Pemilihan nilai k yang tepat akan mempengaruhi kinerja algoritma. Nilai k yang terlalu kecil dapat membuat model terlalu sensitif terhadap noise, sementara nilai k yang terlalu besar dapat mengaburkan batas antar kelas. Proses klasifikasi data baru melibatkan identifikasi kelas dari k tetangga terdekat. Jika masalahnya adalah klasifikasi biner (dua kelas), kelas yang paling banyak muncul di antara k tetangga terdekat akan ditetapkan sebagai kelas data baru. Untuk masalah klasifikasi multi-kelas, prinsip yang sama berlaku, dengan kelas mayoritas sebagai penentu.

Salah satu keunggulan KNN adalah kesederhanaannya. Algoritma ini mudah diimplementasikan dan dipahami. Tidak ada proses pelatihan model yang rumit seperti pada beberapa algoritma lain. KNN hanya menyimpan data pelatihan dan melakukan perhitungan jarak saat ada data baru yang perlu diklasifikasikan. Keunggulan lain KNN adalah fleksibilitasnya. Algoritma ini dapat digunakan untuk berbagai jenis data, baik numerik maupun kategorikal. KNN juga tidak membuat asumsi khusus tentang distribusi data. Hal ini menjadikannya cocok untuk data yang tidak terstruktur atau memiliki pola yang kompleks. Meskipun memiliki keunggulan, KNN juga memiliki keterbatasan. Salah satu keterbatasannya adalah kompleksitas komputasi. Adapun proses kinerja algoritma K-NN dapat dilihat pada Gambar 2.5.

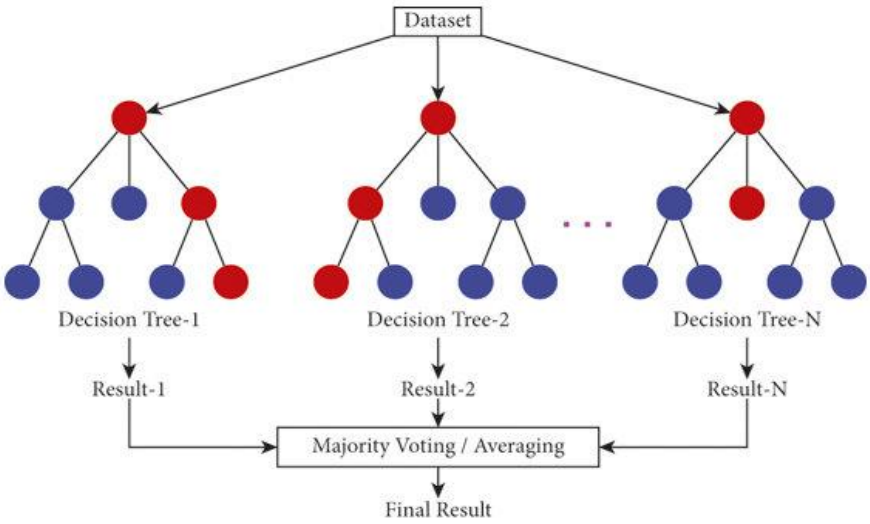


Gambar 2. 5 Struktur algoritma K-Nearest Neighbors.

Decision Tree atau pohon keputusan merupakan algoritma pembelajaran mesin yang populer karena kemampuannya untuk memvisualisasi proses pengambilan keputusan. Algoritma ini bekerja dengan membangun struktur pohon yang menyerupai diagram alir, di mana setiap simpul internal merepresentasikan sebuah fitur, setiap cabang merepresentasikan sebuah keputusan, dan setiap simpul daun merepresentasikan hasil atau kelas.

Struktur pohon keputusan dimulai dari simpul akar, yang merepresentasikan seluruh dataset. Proses ini diulang secara rekursif untuk setiap subset data, menciptakan cabang-cabang dan simpul-simpul internal lainnya. Pemisahan data didasarkan pada kriteria tertentu, seperti *Information Gain* atau *Gini Index*, yang mengukur seberapa baik sebuah fitur memisahkan data berdasarkan kelas atau nilai target.

Salah satu keunggulan utama pohon keputusan adalah interpretasinya yang mudah. Algoritma ini tidak memerlukan asumsi khusus tentang distribusi data, sehingga dapat digunakan untuk berbagai jenis dataset. Salah satu keterbatasannya adalah kerentanan terhadap overfitting. Keterbatasan lain pohon keputusan adalah ketidakstabilan. Adapun proses kinerja algoritma decision tree dapat dilihat pada Gambar 2.6.



Gambar 2. 6 Struktur algoritma *Decision Tree*.

2.2 Keaslian Penelitian

Penelitian ini menggunakan pendekatan orisinalitas dalam analisis sentimen berbasis aspek pada ulasan mahasiswa dengan mengintegrasikan model hibrid BERT dan aspect index generator. Penelitian ini memanfaatkan kemampuan BERT untuk memahami konteks kata secara bidirectional dan aspect index generator untuk mengorganisasi data tidak terstruktur menjadi lebih terstruktur. Pendekatan ini berbeda dari penelitian sebelumnya yang cenderung hanya menggunakan metode klasik seperti TF-IDF atau Word2Vec tanpa menangani tantangan analisis teks kompleks.

Penelitian yang dilakukan oleh Ramasamy & Elangovan, (2024) dalam analisis sentimen menggunakan SVM dan Word2Vec hanya berfokus pada sentimen umum tanpa menangani kompleksitas kalimat majemuk atau kata dengan arti ganda. Penelitian ini memperbaiki kekurangan tersebut dengan menggunakan aspect index generator yang mampu menangani teks kompleks dan memberikan hasil analisis yang lebih akurat dan relevan terhadap aspek tertentu.

Penelitian yang dilakukan oleh Putri, D.I. dkk., (2024) menggunakan IndoBERT untuk analisis sentimen di media sosial berfokus pada pendekatan zero-shot learning tanpa menyelesaikan subtask aspect extraction dan sentimen polarity secara terpadu. Penelitian ini berbeda karena mengintegrasikan kedua subtask ABSA dalam satu framework yang komprehensif, menjadikannya lebih efektif untuk analisis sentimen berbasis aspek pada ulasan mahasiswa.

Penelitian yang dilakukan oleh Kristiyanti dkk., (2024) menggunakan pendekatan sampling teknik SMOTE untuk mengatasi ketidakseimbangan data pada analisis sentimen, namun tidak fokus pada pengolahan aspek-aspek tertentu seperti dosen, kurikulum, sarana prasarana, dan layanan. Penelitian ini melampaui batasan tersebut dengan memberikan solusi untuk analisis yang lebih spesifik pada aspek-aspek terkait evaluasi pendidikan.

Penelitian yang dilakukan oleh Raftopoulos dkk., (2024) menggunakan teknologi cloud computing untuk membangun model evaluasi kualitas siswa di pendidikan vokasi. Pengumpulan data ulasan mahasiswa melalui cloud computing dan menghitung bobot evaluasi menggunakan metode kombinasi AHP dan *entropy*

weight. Hasil penelitian menunjukkan bahwa model evaluasi ini mampu meningkatkan akurasi hingga mencapai kategori atas dalam evaluasi kualitas siswa, memberikan kontribusi signifikan dalam optimalisasi pendidikan vokasi.

Sistem evaluasi kualitas pendidikan tinggi memerlukan pendekatan yang holistik untuk menilai berbagai aspek yang mempengaruhi keberhasilan institusi pendidikan. Salah satu pendekatan yang digunakan dalam penelitian ini adalah analisis vektor, yang mengintegrasikan hiponim, kinerja akademik, hasil penelitian, dan dampak sosial dalam kerangka evaluasi berbasis vektor. Dengan memperhatikan penggunaan hiponim, analisis ini memungkinkan penilaian yang lebih mendalam terhadap aspek-aspek spesifik dalam pengalaman pendidikan mahasiswa, seperti pengajaran, kurikulum, fasilitas, dan layanan, yang mencerminkan kualitas pendidikan secara keseluruhan.

Pendekatan ini mampu memberikan evaluasi yang lebih komprehensif dan adaptif terhadap perubahan dinamika pendidikan tinggi. Hasil penelitian menunjukkan bahwa penggunaan hiponim dalam analisis vektor menghasilkan evaluasi yang lebih terperinci dan relevan mengenai kualitas pendidikan tinggi. Dengan memanfaatkan hiponim, sistem ini dapat memberikan gambaran yang lebih akurat mengenai sentimen mahasiswa terhadap berbagai aspek pendidikan yang sering kali terlewatkan dalam analisis sentimen tradisional (Zhu et al., 2022).

Membangun model evaluasi berbasis lima dimensi untuk pelatihan guru prajabatan. Peneliti menggunakan data perilaku dari sistem kampus online dan melatih model dengan enam algoritma machine learning. Hasil penelitian menunjukkan bahwa model ini mampu menemukan hubungan non-linear antara perilaku guru dan efektivitas pendidikan, memberikan wawasan mendalam terhadap pengembangan kualitas guru, mengintegrasikan algoritma War Strategy Optimization dengan Analytic Hierarchy Process untuk meningkatkan evaluasi pendidikan tinggi. Peneliti mengoptimalkan bobot evaluasi melalui kerangka WSO-AHP, yang menunjukkan peningkatan akurasi analisis dan konsistensi dalam bobot evaluasi pendidikan tinggi (Ummah, 2024).

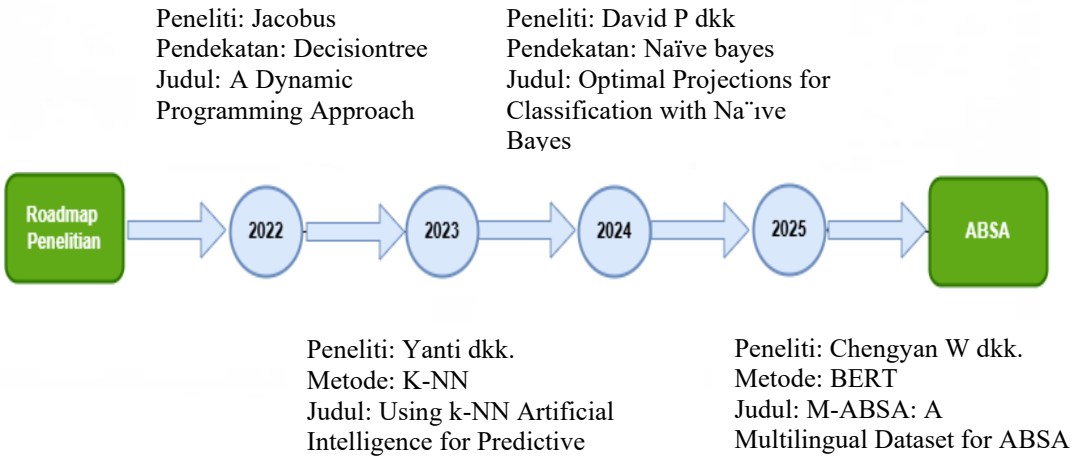
Penelitian yang dilakukan oleh Ma, (2024) yaitu membandingkan berbagai algoritma machine learning, termasuk logistic regression, decision tree, dan LSTM, untuk mengevaluasi dan memberi skor dalam pendidikan. Peneliti

mengembangkan pendekatan multi-classifier yang mampu meningkatkan akurasi evaluasi hingga 98,50%, menjadikannya model paling unggul dibandingkan algoritma tunggal dalam prediksi kualitas pendidikan.

Penelitian ini mengintegrasikan model hibrid BERT dan *aspect index generator* untuk menciptakan metode inovatif yang mampu menangani ulasan mahasiswa yang tidak terstruktur, khususnya yang mengandung kalimat majemuk bertingkat dan kata-kata dengan makna hiponim. Metode ini dirancang untuk menganalisis sentimen berbasis aspek secara lebih akurat dengan memahami kompleksitas struktur bahasa dan konteks makna yang terdapat dalam ulasan. Pendekatan ini memberikan kontribusi signifikan bagi pengembangan evaluasi pendidikan yang lebih mendalam dan komprehensif.

2.3 *Roadmap* Penelitian

Berikut adalah gambaran roadmap perkembangan ABSA dan BERT dari tahun 2022 hingga 2025 menunjukkan roadmap penelitian yang mencakup perkembangan topik dan metode yang digunakan dari tahun 2022 hingga 2025. Pada tahun 2025, fokus penelitian beralih ke ABSA, dengan menggunakan metode BERT. Roadmap ini menggambarkan perkembangan bertahap metode dalam ABSA untuk analisis sentimen berbasis aspek di masa depan ditunjukkan Gambar 2.7 sebagai berikut:



Gambar 2. 7 *Roadmap* penelitian.

Adapun daftar penelitian yang dilakukan oleh peneliti sebelumnya serta dapat dilihat pada Tabel 2.1 di bawah ini.

Tabel 2. 1 Daftar penelitian yang dilakukan oleh peneliti sebelumnya.

PENELITIAN	METODE	DATASET	HASIL
Van der Linden dkk., (2022)	BERT dan Decision Tree	Ulasan Mahasiswa	Accuracy 62,03 %
Musril dkk., (2023)	BERT dan K-NN	Ulasan Mahasiswa	Accuracy 77.01 %
Daqiqil dkk, (2024)	BERT dan Naïve Bayes	Ulasan mahasiswa	Accuracy 88,40 %
Razaq dkk., (2024)	BERT Model dan LSTM	Ulasan Mahasiswa	Accuracy 85,30 %
Kristiyanti dkk., (2024)	pendekatan sampling teknik BERT SMOTE	Ulasan Twitter	*F1 score* 86.00%.
Putri Elisa, (2024)	Naïve bayes, SVM	Crawling website	F1 Score 78,00 %
Chengyan W. dkk, (2025)	BERT	Hotel review	F1-Score 70,76

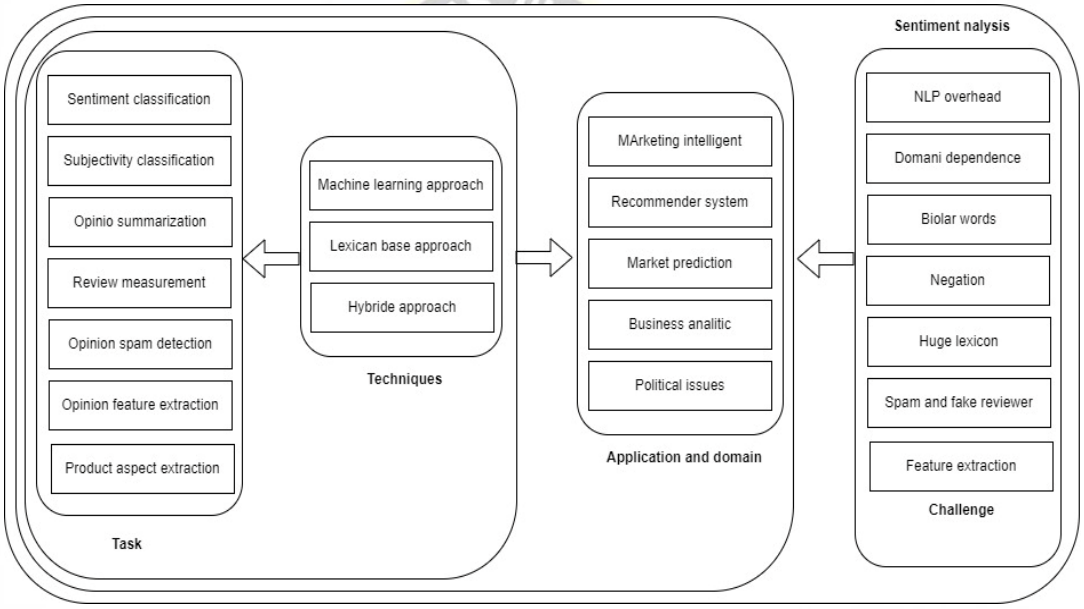
2.3 Landasan Teori

2.3.1 Analisis Sentimen

Analisis sentimen adalah tugas dari *Natural Language Processing* (NLP) yang bertujuan untuk mengekstrak sentimen dan opini dari teks. Selain itu teknik Analisis sentiment baru mulai menggabungkan informasi dari teks dan modalitas lain seperti data visual. Menurut Birjali et al., (2021) komputasi afektif dan analisis sentimen adalah kunci pengembangan *Artificial Intelligence* (AI). Analisis sentimen memiliki potensi besar bila diterapkan ke berbagai domain atau sistem.

Tugas analisis sentimen dapat dianggap sebagai masalah klasifikasi tek karena prosesnya mencakup beberapa operasi yang berakhir dengan mengklasifikasikan apakah teks yang diberikan mengekspresikan sentimen positif atau negatif, namun analisis sentimen mungkin tampak sebagai proses yang mudah tetapi pada kenyataannya, ini membutuhkan pertimbangan banyak *subtask* NLP

seperti *sarkasme* dan deteksi subjektivitas. Selain itu teks tidak selalu terorganisir seperti dalam buku atau surat kabar yang dapat mengandung banyak kesalahan *ortografis*, ungkapan idiomatik, atau singkatan. Tugas utama dalam analisis sentimen meliputi klasifikasi sentimen, klasifikasi subjektivitas, rangkuman opini, pengukuran ulasan, deteksi spam opini, ekstraksi fitur opini, dan ekstraksi aspek produk. Teknik yang digunakan untuk menyelesaikan tugas-tugas ini antara lain pendekatan berbasis pembelajaran mesin, pendekatan berbasis leksikon, dan pendekatan hibrida. Adapun *taxonomi* dari analisis sentimen dapat dilihat pada Gambar 2.8.



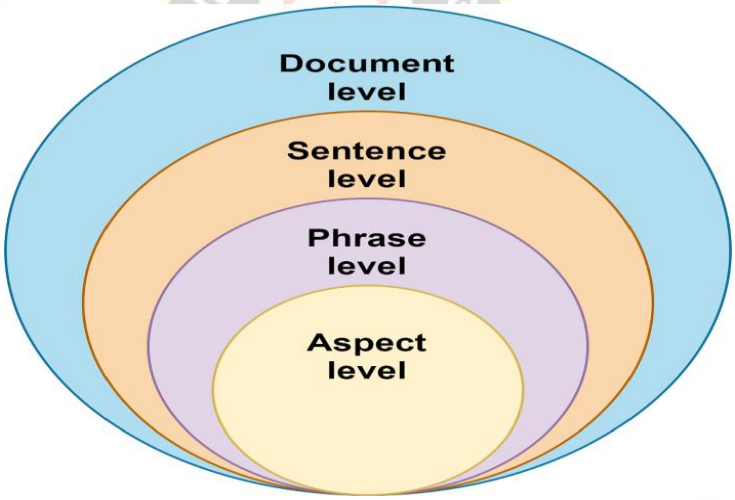
Gambar 2. 8 Riset pada analisis sentimen (Chen & Xie, 2020).

Metodologi dalam menganalisa sentimen memerlukan pemrosesan awal dan ekstraksi fitur *leksikal* dari *tweet*. Langkah-langkah prapemrosesan dapat memiliki efek signifikan pada kinerja model (Antonakaki et al., 2021) dan mencakup tokenisasi, perluasan singkatan dan penghapusan kata henti dan elemen lain tanpa nilai *leksikal*, seperti URL dan penyebutannya untuk mengatasi permasalahan ini dengan cara melakukan *query* di mesin pencari dengan konten *tweet* dan menambahnya dengan hasil teratas. Metode lain adalah dengan menambahkan informasi dari kumpulan leksikon khusus *twitter* yang dibuat dengan *Machine*

Learning atau hanya menggabungkan *tweet* dari pengguna yang sama dan membangun model topik berbasis aspek.

2.3.2 Analisis Sentimen Berbasis Aspek

Birjali dkk., (2021) mengemukakan bahwa analisis sentimen dibagi menjadi tiga tingkatan: tingkat kalimat, tingkat dokumen dan tingkat aspek. Dalam hal ini, asumsi bahwa sebuah dokumen atau kalimat hanya mengandung satu topik dibahas oleh (Zulqarnain et al., 2021). Untuk melakukan analisis yang komprehensif, diperlukan penanganan pada tingkat entitas dan aspek. Proses klasifikasi sentimen yang berkaitan dengan entitas dan aspek tersebut dapat diobservasi dalam Gambar 2.9.



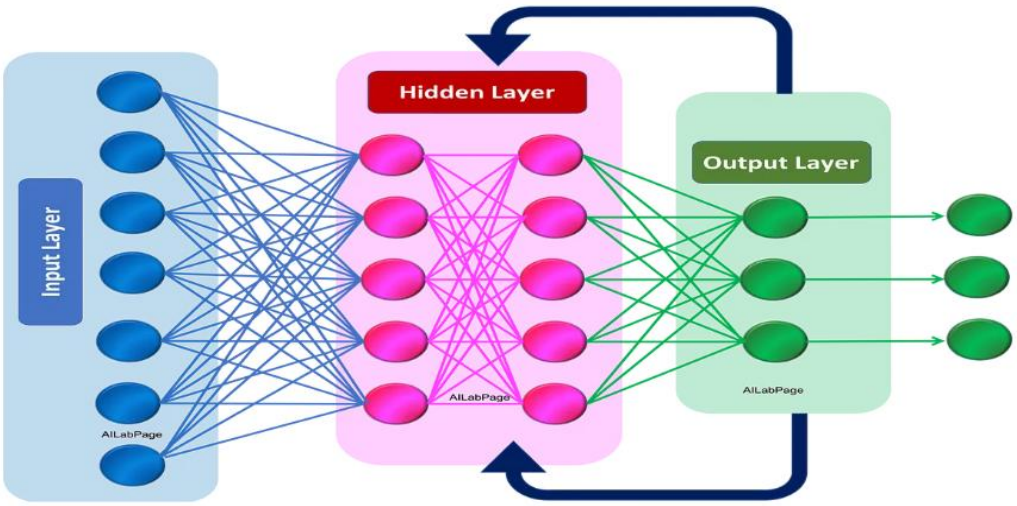
Gambar 2. 9 *Level aspect based sentiment analysis* (Salazar et al., 2021).

Analisis sentimen berbasis aspek adalah *subtask* terpenting dalam *analisis sentimen* untuk mengidentifikasi sentimen pengguna pada level aspek dalam teks, dimana aspek tersebut dapat berupa karakteristik atau properti apa pun dari entitas tersebut (Kastrati et al., 2021). *Subtask* sebelumnya pada ABSA melibatkan dua *subtask* yang berbeda menurut aspek yang paling kecil, yaitu, analisis sentimen dengan istilah *aspek detection* dan *sentiment polarity*. Pertama berfokus menganalisis sentimen pada aspek target entitas yang muncul dalam teks. Terakhir menganalisis sentimen pada tingkat yang lebih tinggi yang menyangkut aspek serupa sebagai salah satu dari beberapa kategori yang telah ditentukan sebelumnya

Sutherland et al., (2020) dalam *subtask* ini, mengusulkan model yang efektif untuk menangani berbagai jenis tugas ABSA yang berarti bahwa metode ini lebih fleksibel dan akurat.

2.3.3 Recurrent Neural Networks

Jaringan saraf telah menarik perhatian yang meningkat dalam beberapa dekade ini terutama dibidang domain ABSA. Metode dan model banyak yang dikembangkan terutama dalam jaringan saraf *rekursif neural network*, *recurrent neural networks* dan jaringan saraf *convolutional* (Saab et al., 2021). Adapun model *Recurrent Neural Network* dapat dilihat pada Gambar 2.10.



Gambar 2. 10 Arsitektur *Recurrent Neural Network* (Seob et al., 2020).

Struktur sintaksis dalam natural language program direpresentasikan sebagai struktur pohon yang dapat dimodelkan secara rekursif (Zheng & Zheng, 2019). *RecNeural Networks* diterapkan pada *sub task* ABSA yang didukung oleh metode ini dan dapat menyimpulkan ekspresi semantik dan emosional kalimat yang mirip dengan analisis sintaksis (Wang X. et al., 2016). Namun, *RecNN* dalam tugas ABSA membutuhkan pohon sintaks yang telah ditentukan untuk mengkodekan kalimat yang masih membatasi ekstensibilitas yang ada. *Recurrent Neural Network* dapat menangani masalah tekstual secara berurutan hanya memerlukan sedikit metode pemrosesan teks dan informasi aspek khusus untuk *subtask* ABSA (Aydln

& Gungor, 2020). (Seob et al., 2020) memanfaatkan unit *long short-term memory* berdasarkan *recurrent neural network* untuk memprediksi *aspect sentiment* dan *aspect polarity*. Model ini memiliki dua strategi yang dimasukkan untuk memproses informasi aspek. Strategi pertama memotong kalimat pada kata dan posisi. Strategi kedua menggabungkan kata aspek ke setiap kata kalimat. Mekanisme *Subtask* ABSA mencegah model dari pengkodean informasi yang tidak relevan. Dengan cara ini model dapatkan fokus pada bagian-bagian tertentu dari kalimat yang relevan dengan aspek yang diinginkan. Model *recurrent neural network* ini telah membuktikan keefektifan mekanisme perhatian dalam tugas analisis sentiment (Al-Smadi et al., 2018). Namun, mekanisme perhatian diperlakukan hanya sebagai pelengkap *Recurrent Neural Network* untuk meningkatkan akurasi (Gruber, 2021).

2.3.4 Text Mining

Text mining didefinisikan sebagai cabang ilmu yang berhubungan dengan metode dari pengolahan, pendeteksian dan mengekstraksi data yang berbentuk teks baik secara terstruktur maupun tidak terstruktur (Skarpathiotaki & Psannis, 2022). *Opinion mining* adalah teknik pemrosesan ulang informasi yang bermakna melalui analisis opini publik yang muncul di situs web dan media sosial dengan menganalisis teks untuk mendigitalkan opini publik pengguna internet menjadi informasi yang objektif (Estrada et al., 2020). Hal ini juga disebut *sentiment analysis* karena mengategorikan opini pelanggan kuantitatif menjadi ide positif dan negatif. *Opinion mining* adalah studi tentang metodologi saran dalam membuat evaluasi kepuasan pelanggan yang akurat, (Raffel et al., 2020) untuk memperoleh data yang terstruktur memerlukan berbagai macam teknik untuk mengekstraksi data agar data dapat digunakan dengan baik. Data yang tidak terstruktur membutuhkan penanganan yang khusus dan rumit sehingga perlu adanya metodologi untuk mempermudah penanganan data, yaitu menggunakan data mining baik secara konvensional maupun otomatisasi untuk mendapatkan informasi dari konten yang tidak terstruktur.

Penelitian ini sangat penting dalam penanganan *text mining*. Menurut *International Educational Data Mining Society* adalah disiplin ilmu yang berkaitan dengan sebuah pengembangan metode yang digunakan untuk mengeksplorasi data secara unik dan mengklasifikasikan data secara sistematik. Makin besar skala data yang digunakan, makin menghasilkan klasifikasi dan training data makin akurat. Secara garis besar metode *text mining* digunakan untuk menangani berbagai macam riset operasi seperti pengambilan kesimpulan atau meringkas dari suatu teks yang berskala besar serta mengklasifikasikan berbagai macam dokumen, analisis sentiment serta bentuk data yang lain.

2.3.5 Preprocessing pada Text Mining

Proses Text Preprocessing merupakan tahap awal dalam mempersiapkan teks agar dapat diproses lebih lanjut. Teks yang tidak diproses tidak dapat langsung digunakan oleh algoritma pencarian, oleh karena itu, langkah preprocessing diperlukan untuk mengubah teks menjadi format data numerik yang siap untuk dianalisis dan diolah (Jacinto et al., 2020).

2.3.6 Autoencoder

Autoencoder dapat digunakan untuk pemrosesan teks (*text processing*) dalam beberapa tugas seperti pengurangan dimensi, generasi teks, dan denoising teks. Dalam pemrosesan teks, representasi teks yang sederhana dan padat dapat membantu mengatasi masalah dengan data teks yang komplek dan ukurannya besar. Pada tugas pengurangan dimensi dalam pemrosesan teks, *autoencoder* dapat digunakan untuk mengekstrak fitur-fitur penting dari teks yang kompleks sehingga dapat menghasilkan representasi teks yang lebih sederhana dan mudah dipahami oleh mesin. Representasi teks yang lebih sederhana dan padat ini kemudian dapat digunakan dalam tugas-tugas seperti klasifikasi teks atau *clustering*.

Pada tugas generasi teks, *autoencoder* dapat digunakan untuk membuat model bahasa yang dapat digunakan untuk menghasilkan teks yang baru. Dalam tugas ini, *autoencoder* dilatih untuk merekonstruksi teks inputan kemudian *decoder* dari *autoencoder* digunakan untuk menghasilkan teks yang baru berdasarkan representasi yang diberikan. Representasi ini dapat diambil dari layer terdalam pada

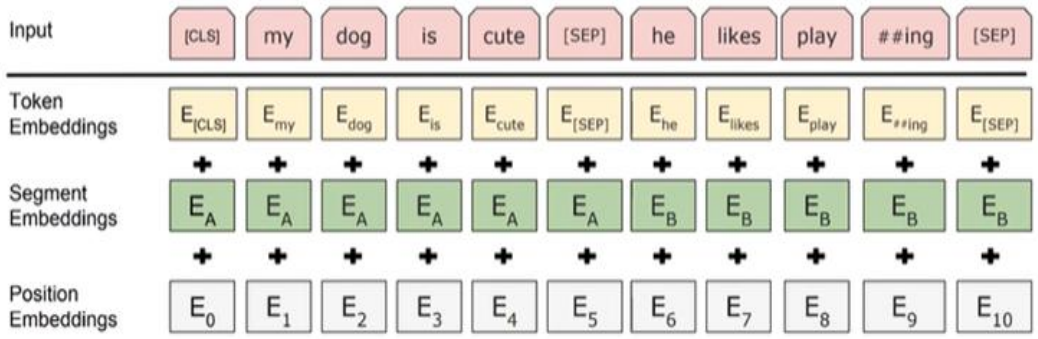
encoder. Pada tugas *denoising text*, *autoencoder* dapat digunakan untuk menghilangkan *noise* atau gangguan pada teks yang diberikan. Dalam tugas ini *autoencoder* dilatih dengan menggunakan teks yang telah terganggu dengan *noise*, kemudian *decoder* dari *autoencoder* digunakan untuk merekonstruksi teks yang lebih bersih dan jernih. *Autoencoder* dapat membantu menghilangkan *noise* pada teks untuk meningkatkan kualitas teks yang dihasilkan.

Autoencoder dapat digunakan untuk menangani analisis sentimen berbasis aspek dengan memperoleh representasi kalimat yang sederhana dan padat. Analisis sentimen berbasis aspek bertujuan untuk menemukan sentimen atau opini yang terkait dengan aspek-aspek atau fitur-fitur dari suatu produk atau layanan yang diulas dalam kalimat. Salah satu tantangan dalam tugas ini adalah kompleksitas dan variasi yang tinggi dalam data berbentuk teks. Proses untuk menangani tugas *aspect-based sentiment analysis*, *autoencoder* dapat digunakan untuk menghasilkan representasi kalimat yang lebih sederhana dan padat dari kalimat majemuk yang sangat komplek. Proses training dilakukan dengan memasukkan kalimat ulasan yang telah diberi label ke dalam autoencoder kemudian *autoencoder* dilatih untuk menghasilkan representasi kalimat yang sesuai. Representasi ini kemudian dapat digunakan untuk menemukan sentimen atau opini yang terkait dengan aspek atau fitur tertentu dalam teks. Selain itu, *autoencoder* juga dapat digunakan untuk mengatasi masalah data yang tidak seimbang *(imbalanced data*) pada tugas *aspect-based sentiment analysis*. Masalah ini muncul ketika jumlah data pada sentimen positif, negatif atau netral tidak seimbang. *Autoencoder* dapat digunakan untuk menyeimbangkan data dengan menghasilkan representasi kalimat yang lebih sederhana dan padat dari setiap aspek atau fitur dalam teks sehingga dapat meningkatkan kualitas klasifikasi sentimen pada setiap aspek atau fitur.

2.3.7 Bidirectional Encoder Representations for Transformers.

Penggunaan kalimat majemuk bertingkat yang dilakukan oleh Wu & Ong, (2020). memperkenalkan konsep penggunaan *Adaptive Bidirectional Attention* dalam pemahaman bahasa mesin. Pemahaman *machine reader* seringkali digunakan dalam membaca kalimat majemuk bertingkat yang mengharuskan model untuk memahami hubungan antara kalimat dalam ulasan. Gambar 2.11 menunjukkan cara

pemrosesan input dalam model BERT. Setiap kata atau subkata dalam input diubah menjadi representasi vektor yang disebut Token Embeddings. Vektor ini menunjukkan makna dari kata atau subkata tersebut. Selanjutnya, untuk membedakan antara dua segmen yang ada dalam input, digunakan Segment Embeddings. Segment Embeddings ini membedakan antara segmen A dan segmen B, digunakan dalam tugas seperti tanya jawab. Selain itu, untuk mempertahankan urutan kata dalam kalimat, digunakan Position Embeddings, menunjukkan posisi masing-masing token dalam urutan kalimat. Ketiga jenis embedding ini kemudian dijumlahkan untuk membentuk representasi akhir yang akan diproses oleh BERT. Representasi akhir ini BERT memahami konteks dan hubungan antar token dalam input secara lebih mendalam sebagaiman terlihat pada gambar 2.11.



Gambar 2. 11 Segmentasi dan posisi *embedding* dan token.

Token embeddings adalah representasi numerik dari kata-kata dalam bentuk vektor. Setiap kata dalam korpus teks akan diberikan kode unik yang merepresentasikan makna dan konteksnya. Dengan cara ini, model dapat memahami hubungan antara kata-kata dalam kalimat. Segment embeddings digunakan untuk membedakan antara dua kalimat dalam satu input. Misalnya saat mengolah dua kalimat dalam satu konteks, segment embeddings menandai bagian yang berasal dari kalimat pertama dan kalimat kedua, sehingga sangat penting untuk tugas-tugas seperti pertanyaan dan jawaban. Position embeddings memberikan informasi tentang posisi setiap kata dalam urutan kalimat, karena model tidak memproses urutan kata secara berurutan, position embeddings membantu mempertahankan konteks dan urutan kata. Misalnya, jika kata “dosen”

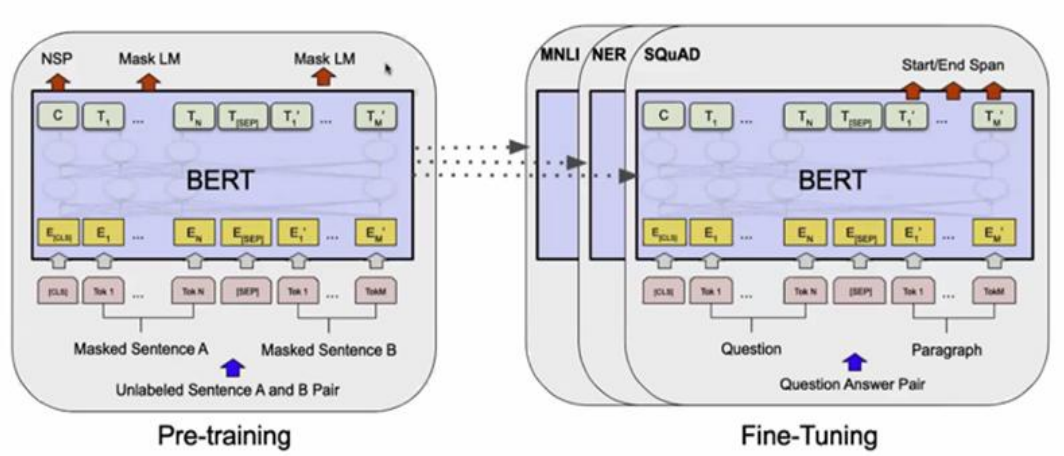
muncul sebelum “di taman”, model dapat memahami bahwa “dosen” adalah subjek dan “di taman” adalah keterangan tempat. Ketiga *embedding* ini yaitu token, segment, dan position bekerja bersama untuk meningkatkan kemampuan pemahaman model bahasa, sehingga proses analisis teks menjadi lebih efektif dan akurat. Dengan kemajuan ini, berbagai aplikasi dalam kecerdasan buatan, seperti penerjemahan dan analisis sentimen, menjadi lebih optimal dan dapat diandalkan.

Hiponimitas dalam kalimat majemuk merujuk pada situasi dimana kata yang sama memiliki makna yang berbeda dalam konteks kalimat yang berbeda. BERT dapat memahami konteks kata dalam kalimat sehingga dapat membantu dalam mengatasi masalah hiponimisme dalam kalimat dengan lebih baik. BERT dapat membantu mengidentifikasi makna yang tepat dari sebuah kalimat dengan mempertimbangkan konteks sekitarnya sehingga mengurangi risiko kesalahan akibat hiponimisme kalimat. Penelitian yang dilakukan oleh (Liu et al., (2021), BERT digunakan dalam analisis sentimen pada kumpulan ulasan konsumen dari berbagai domain seperti obat-obatan, hotel, produk, dan sekolah. BERT membantu memahami makna kata-kata dalam konteks domain yang berbeda, sehingga dapat mengurangi risiko kesalahan akibat hiponimisme terhadap kata didalam kalimat.

Penelitian yang dilakukan oleh Lin dkk., (2022), BERT digunakan dalam konteks kesamaan teks semantik yang melibatkan representasi kalimat yang komplek. Hal ini menciptakan sebuah sistem yang menggabungkan representasi kalimat terdistribusi dan *representasi one-hot* menggunakan jaringan berbasis gerbang. BERT membantu dalam memahami makna kata-kata dalam konteks yang lebih dalam dapat mengurangi risiko kesalahan akibat hiponimisme kata, hasilnya menunjukkan bahwa BERT dapat memberikan korelasi yang lebih tinggi dalam tugas kesamaan teks semantik, mencerminkan kemampuannya dalam mengatasi kompleksitas pada kalimat.

Penggunaan BERT dalam berbagai tugas pemrosesan bahasa alami seperti pemahaman teks *multidomain* dan kesamaan teks semantik memiliki implikasi yang signifikan dalam mengatasi hiponimisme kalimat. Kemampuan BERT untuk memahami konteks kata-kata dalam kalimat memungkinkannya untuk mengidentifikasi makna yang tepat, bahkan ketika kata-kata tersebut memiliki arti ganda dalam konteks yang berbeda, hal ini dapat membantu dalam meningkatkan

akurasi dalam berbagai aplikasi pemrosesan bahasa yang dapat meningkatkan kualitas hasil dan pengalaman pengguna. Penggunaan BERT telah menjadi bagian integral dalam pengembangan solusi natural language program yang efektif dapat dilihat pada Gambar 2.12.

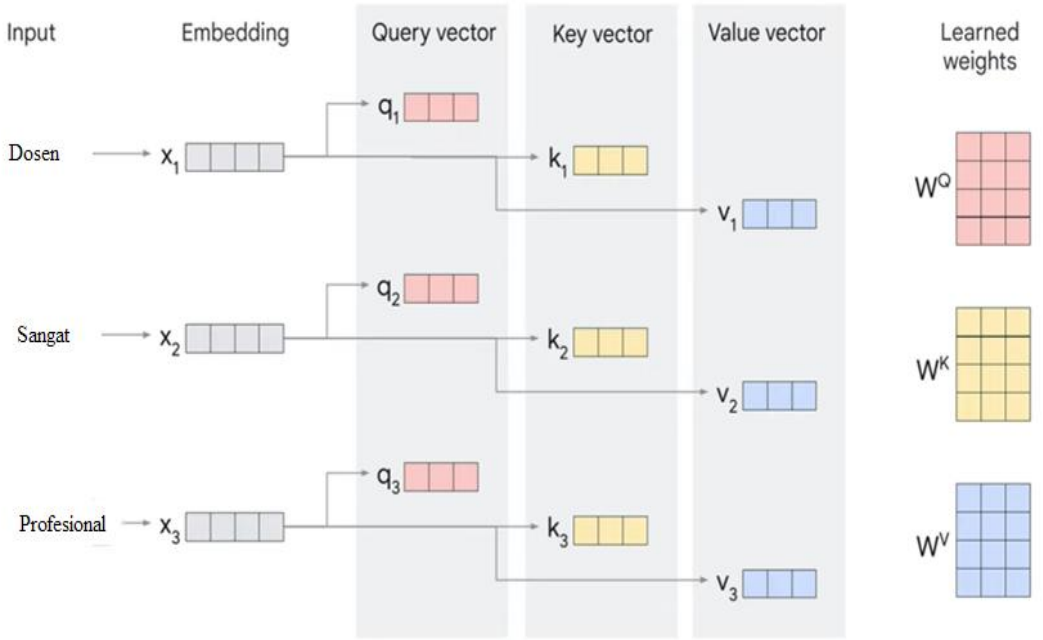


Gambar 2. 12 *Pretraining* dan *fine tuning* pada BERT.

BERT adalah model *deep learning* yang digunakan untuk memproses teks dan menghasilkan representasi numerik dari teks tersebut. Model ini didasarkan pada arsitektur *Transformer*, di mana setiap elemen output terhubung ke setiap elemen input, dan bobot elemen dihitung secara dinamis berdasarkan hubungan antar elemen. Transformer adalah arsitektur jaringan saraf yang digunakan dalam pemrosesan bahasa alami. Transformer memungkinkan model untuk memproses urutan input secara paralel, sehingga mempercepat waktu pelatihan dan inferensi. Transformer terdiri dari dua bagian utama: *encoder* dan *decoder*. *Encoder* digunakan untuk memproses input, sedangkan *decoder* digunakan untuk menghasilkan output.

Encoder dan *decoder* pada Transformer adalah dua bagian utama dari arsitektur jaringan saraf yang digunakan dalam pemrosesan bahasa alami. *Encoder* digunakan untuk memproses input, sedangkan *decoder* digunakan untuk menghasilkan *output*. *Encoder* pada Transformer terdiri dari beberapa blok *encoder*, di mana setiap blok *encoder* terdiri dari dua sub-bagian: *multi-head attention* dan *feed-forward neural network*. *Multi-head attention* digunakan untuk

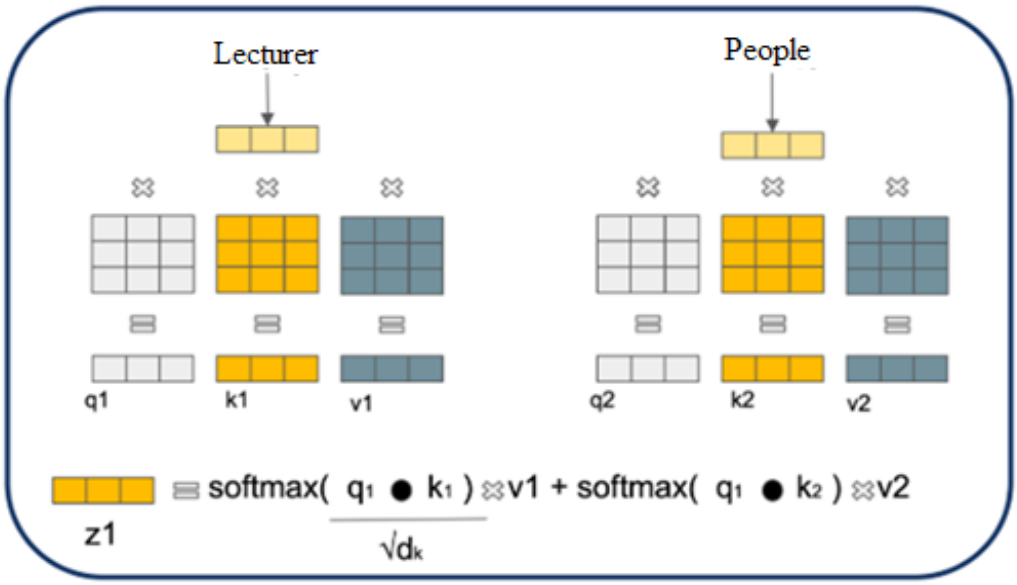
memperhatikan hubungan antara setiap elemen input, sedangkan *feed-forward neural network* digunakan untuk memproses setiap elemen input secara independen. *Decoder* pada *Transformer* juga terdiri dari beberapa blok *decoder*, di mana setiap blok *decoder* terdiri dari tiga sub-bagian yaitu *masked multi-head attention*, *multi-head attention*, dan *feed-forward neural network*. *Masked multi-head attention* digunakan untuk memperhatikan hubungan antara setiap elemen *output*, sedangkan *multi-head attention* digunakan untuk memperhatikan hubungan antara setiap elemen *input* dan *output*. Konsep dasar dari *attention mchanisme* dalam model pembelajaran mesin dapat dilihat pada gambar 2.13.



Gambar 2. 13 Konsep kerja dari *attention mechanism.*

Gambar 2.13 menjelaskan konsep dasar dari *attention mechanism* dalam model pembelajaran mesin, khususnya dalam konteks pengolahan bahasa alami. Terdapat beberapa kata sebagai inputan. Setiap kata diwakili oleh vektor embedding (x) yang mengkodekan informasi semantik dari kata tersebut. *Vektor embedding* untuk setiap input menghasilkan representasi makna dari setiap kata dalam dimensi yang lebih rendah. *Query Vector* (q) merupakan representasi dari kata yang sedang diproses digunakan untuk mencari informasi relevan dari kata lainnya. *Key Vector* (k) digunakan untuk menilai seberapa relevan informasi dari kata tersebut dengan

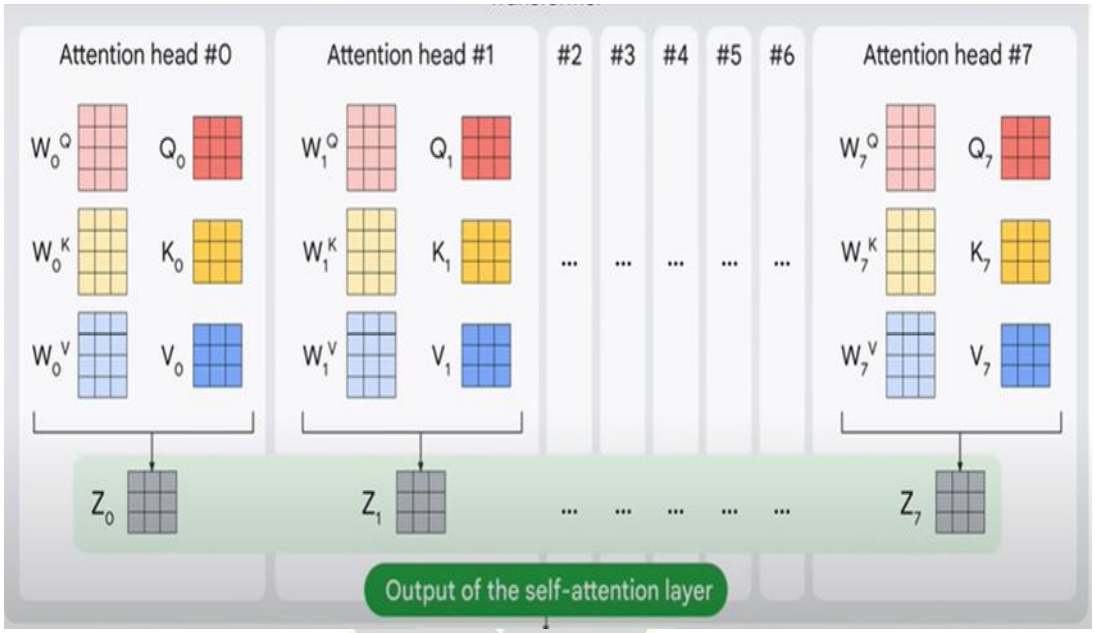
query yang diberikan. *Value Vector* jika suatu kata dianggap relevan maka vektor nilai inilah yang akan diambil untuk digunakan dalam output. *Learned Weights* yang mewakili bobot yang dipelajari oleh model selama proses pelatihan. Bobot ini menghubungkan input dengan output dan dioptimalkan untuk meningkatkan akurasi. *Attention mechanism* mengukur kesamaan antara *query* dan *key* untuk mengidentifikasi informasi mana yang paling relevan dari kata lainnya, kemudian informasi tersebut dikombinasikan untuk membentuk output yang lebih kaya dan informatif hal ini menggambarkan bagaimana informasi dapat diambil dan diperkuat dari beberapa elemen input untuk menghasilkan pemahaman yang lebih baik dalam konteks kalimat secara keseluruhan. Hubungan antara q, k dan v dapat dilihat pada Gambar 2.14.



Gambar 2. 14 Proses pemodelan hubungan antara queri dan kunci.

Gambar 2.14 menunjukkan diagram yang memperjelas fungsi *attention mechanism* dalam model pembelajaran mesin, terdapat dua kategori yang berbeda yaitu "*Lecturer*" dan "*People*." Setiap kategori memiliki representasi yang menunjukkan bagaimana istilah terkait diolah menggunakan fungsi *softmax*. Fungsi *softmax* adalah untuk menggambarkan proses pemodelan hubungan antara queri dan kunci, serta bagaimana bobot perhatian diperoleh dan dihitung dalam konteks analisis data

serta memvisualisasikan cara data terdistribusi dan saling berinteraksi, sehingga memudahkan pemahaman tentang pembelajaran representasi dalam jaringan saraf. Adapun output dari layer *self attention* pada NLP dapat dilihat pada pada Gambar 2.15 berikut ini.



Gambar 2. 15 Output *self attention* layer.

Gambar 2.15 menunjukkan output dari layer *self attention* pada NLP bertindak sebagai representasi vektor yang memperhitungkan interaksi antar kata-kata dalam sebuah kalimat. Dengan menggunakan mekanisme attention, model dapat fokus pada kata-kata yang relevan untuk memahami konteks kalimat secara lebih baik. Hasil dari layer ini membantu meningkatkan kualitas representasi teks yang dihasilkan oleh model, sehingga memungkinkan model untuk memahami hubungan antar kata secara lebih baik dan memperbaiki kinerja pada tugas-tugas NLP seperti pemrosesan bahasa alami, terjemahan, atau analisis sentimen.

Cara menghitung output dari multi-head attention adalah sebagai berikut:

1. Hitung skalar produk titik antara *query* dan *key*.
2. Bagi hasil dari langkah 1 dengan akar kuadrat dari dimensi *key*.
3. Terapkan fungsi *softmax* pada hasil dari langkah 2.
4. Hitung hasil perkalian antara hasil dari langkah 3 dan *value*.

Berikut adalah rumus matematika yang terkait dengan *multi-head attention*:

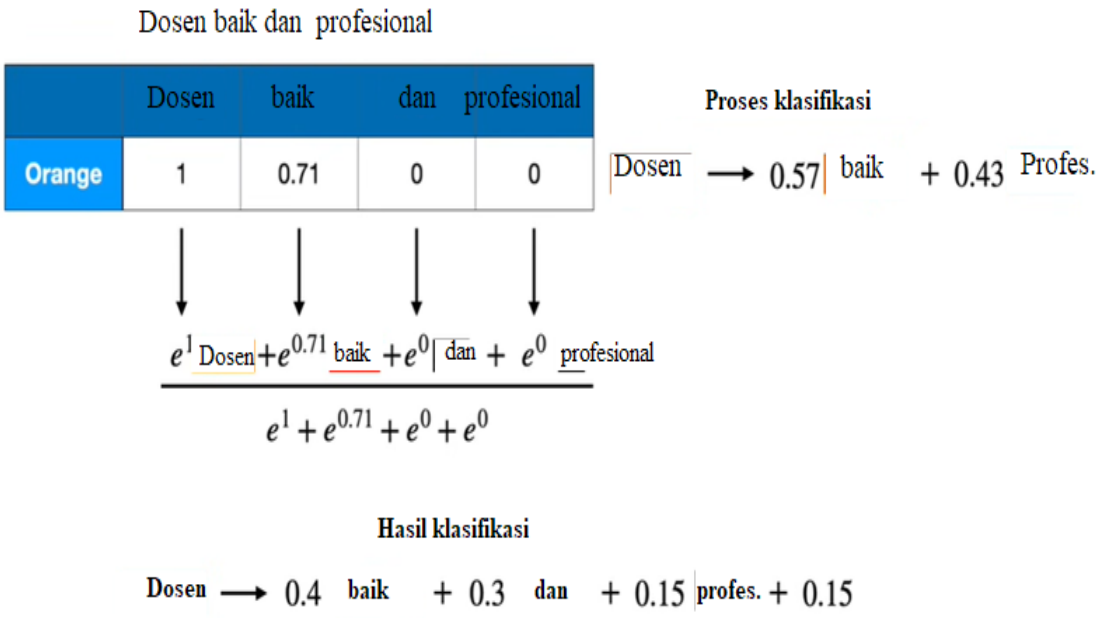
MultiHead (Q, K, V) = Concate (head$_1$, …, head$_h$) W^o (2.1)

Head1 = Attention (QW_i^Q , KW_i^K , VW_i^V)

Penjelasan dari rumus tersebeut adalah:

a) Q adalah queri
b) K adakah key
c) V adalah value
d) W_i^Q , W_i^K , W_i^V adalah bobot query, key dan volume pada head ke i
e) W^o adalah bobot untuk menghubungkan hasil dari setiap head

Proses perhitungan menggunakan softmax dapat dilihat pada Gambar 2.16.



Gambar 2. 16 Klasifikasi multi kelas menggunakan *softmax*.

Gambar 2.16 menjelaskan konsep fungsi *softmax* dalam konteks klasifikasi multikelas. *Softmax* adalah fungsi matematika yang mengubah vector angka menjadi probabilitas di mana setiap angka dalam vector mewakili kemungkinan klasifikasi tertentu. Contoh penggunaan kelas sebagaimana terlihat bahwa terdapat dua kelas, yaitu "Orange" dan "Apple," dengan nilai eksponensial dari masing-masing kelas dihitung dan dinormalisasi. Hasilnya menunjukkan bahwa probabilitas untuk kelas "Orange" adalah 0.57 dan untuk "Apple" adalah 0.43. Ini

menandakan bahwa, berdasarkan nilai input tersebut, model lebih cenderung untuk memprediksi kelas "Orange." Peringatan pada gambar menunjukkan pentingnya memahami output *softmax* karena dapat mempengaruhi perubahan signifikan dalam distribusi probabilitas tergantung pada nilai input yang diberikan. Hal ini menunjukkan bagaimana perubahan kecil dalam input dapat mempengaruhi hasil prediksi akhir dalam model klasifikasi.

Encoder layer: BERT menggunakan beberapa lapisan *encoder* dari *transformer*, setiap lapisan *encoder* mengandung sub lapisan *multi head self attention* dan sub lapisan *feed-forward network*. Rumus untuk *self attention* dalam satu head adalah:

$$Attention(Q, K, V) = softmax\left(\frac{QK^t}{\sqrt{d_k}}\right)V \quad (2.2)$$

Q, K, V: Ini adalah matriks Query (Q), Key (K), dan Value (V). Dalam konteks transformer, setiap vektor input diubah menjadi ketiga representasi ini menggunakan matriks berbeda. Q adalah representasi query, K adalah representasi key, dan V adalah representasi value.

QK^t adalah produk dot antara Query dan transpose dari Key. Langkah ini digunakan untuk mengukur tingkat kesesuaian antara berbagai query dan key. Hal ini secara esensial menunjukkan seberapa relevan setiap bagian dari input terhadap bagian lain. $\sqrt{d_k}$ adalah faktor skala, di mana d_k adalah dimensi kunci. Skala ini digunakan untuk menghindari gradient yang sangat besar selama pelatihan, yang dapat menyebabkan masalah dalam optimisasi.

Setiap encoder juga memiliki sebuah *feed-forward network* sebagai berikut:

$$FFN(x) = max(0, \chi W_1 + b_1)W_2 + b_2 \quad (2.3)$$

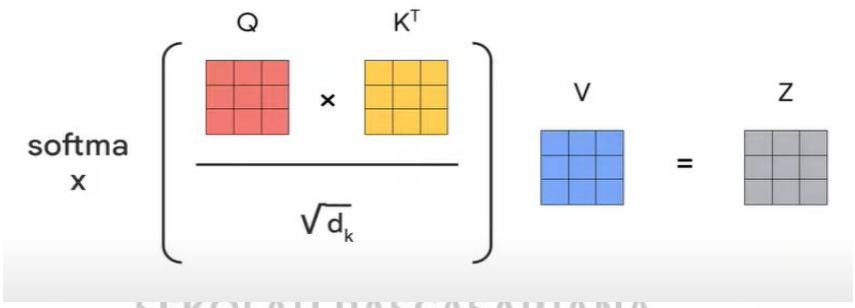
Dengan:

a) χ adalah input dari feed forward network.
b) W_1, W_2 matrik bobot. W_1 mengubah dimensi input ke dimensi yang lebih tinggi, dan W_2 mengubahnya kembali ke dimensi yang lebih rendah.
c) b_1, b_2 adalah matrik bias.

d) $Fungsi\ max(0, z)$ adalah fungsi aktivasi *ReLU* (*Rectified Linear Unit*), yang menghasilkan 0 untuk setiap nilai input z yang negatif dan z itu sendiri jika positif.

Dalam arsitektur *Transformer,* FFN ini digunakan setelah langkah *multi-head attention* di setiap lapisan *encoder*.

Fungsi softmax merupakan suatu fungsi matematika yang digunakan untuk mengonversi nilai-nilai numerik menjadi probabilitas. Fungsi ini umum diterapkan dalam machine learning dan deep learning untuk menghasilkan output yang dapat diinterpretasikan sebagai probabilitas dari berbagai kelas. Dengan menerima nilai numerik yang mewakili berbagai kelas, fungsi softmax menghasilkan distribusi probabilitas yang menggambarkan kemungkinan setiap kelas terpilih. Proses perhitungan *softmax* dapat dilihat pada Gambar 2.17.



Gambar 2. 17 Proses perhitungan fungsi *softmax*.

Output dari fungsi *softmax* adalah vektor dengan probabilitas dari setiap kemungkinan. Adapun untuk menghitung fungsi *softmax* sebagai berikut:

$$P_{ij}^{s} = softmax(W_s h_{ij}^{s} + b_s) \quad (2.4)$$

dengan $W_s \in R^{4xd_h} and\ b_s \in R^4$ learnable parameter

rumus yang digunakan untuk menghitung loss pada setiap iterasi. Rumus ini dikenal sebagai *Cross Entropy Loss* sebagai berikut:

$$\mathcal{L}^T = -\frac{1}{2}\sum_{i=1}^{N} P_i^T \ log(P_i^T) \quad (2.5)$$

dengan:

$\mathcal{L}^T$ adalah nilai kerugian (loss) pada tahap atau untuk tugas tertentu yang ditandai dengan *T*. Dalam konteks pembelajaran mesin, nilai kerugian mengukur seberapa

baik model melakukan tugasnya dengan nilai yang lebih rendah menunjukkan performa yang lebih baik.

$-\frac{1}{2}$ aktor ini merupakan pengali skala. Penggunaan faktor negatif di sini menunjukkan bahwa kita ingin meminimalkan nilai kerugian karena logaritma dari nilai antara 0 dan 1 adalah negatif, dan mengalikannya dengan -1 membuat nilai kerugian menjadi positif.

$\sum_{i=1}^{N}$ Ini adalah simbol sigma yang menunjukkan penjumlahan. Menjumlahkan atas semua *N* elemen atau sampel dalam kumpulan data atau batch yang sedang diproses.

P_i^T ini mewakili probabilitas prediksi model untuk sampel ke-i dalam tugas *T*. Probabilitas ini biasanya dihasilkan oleh model, misalnya melalui fungsi *softmax* di lapisan output.

$log(P_i^T)$ adalah logaritma alami dari probabilitas P_i^T . Mengambil logaritma dari probabilitas adalah cara umum untuk menghitung kerugian dalam banyak tugas klasifikasi, terutama dalam konteks entropi silang cross-entropy loss.

Akhirnya, fungsi kerugian keseluruhan dihitung sebagai berikut:

$$\mathcal{L} = \mathcal{L}^T + \mathcal{L}^O + \mathcal{L}^S + \lambda \|'\Omega\|^2 \quad (2.6)$$

Keterangan:

L: Ini adalah nilai total kerugian (total loss) yang dicari dalam model. Dalam konteks pembelajaran mesin, ini mengukur seberapa baik model secara keseluruhan dalam melakukan serangkaian tugas atau memenuhi serangkaian kriteria.

$\mathcal{L}^T + \mathcal{L}^O + \mathcal{L}^S$ Hal ini menunjukkan komponen-komponen individu dari fungsi kerugian total. Masing-masing mungkin mewakili kerugian untuk tugas yang berbeda atau aspek yang berbeda dari model. Misalnya, $\mathcal{L}^T$ adalah kerugian untuk tugas klasifikasi, $\mathcal{L}^0$ kerugian untuk optimasi tertentu, dan$\mathcal{L}^S$ kerugian untuk aspek lain seperti regresi atau rekonstruksi. Penambahan ketiga kerugian ini menunjukkan bahwa model tersebut dioptimalkan untuk memenuhi lebih dari satu kriteria.

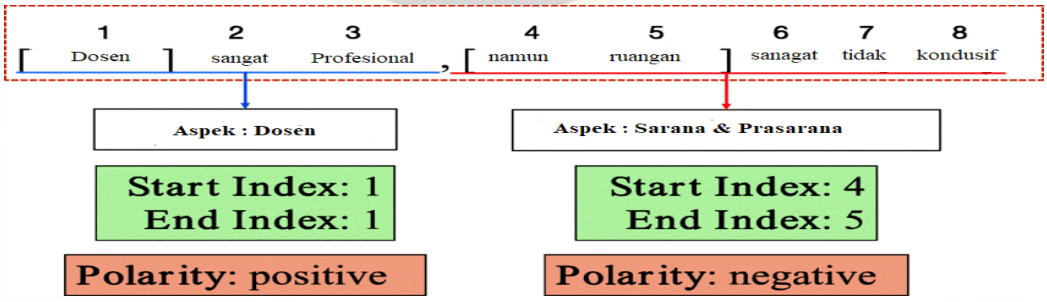
$\lambda'\Omega^2$ adalah istilah regularisasi. Regularisasi adalah teknik yang digunakan untuk mencegah overfitting, yaitu ketika model terlalu sesuai dengan data latih dan tidak mampu melakukan generalisasi dengan baik pada data baru. λ: Ini adalah parameter

yang mengontrol seberapa kuat pengaruh regularisasi terhadap fungsi kerugian total. Nilai yang lebih tinggi berarti regularisasi memiliki pengaruh yang lebih kuat. $\lambda\Omega^2$ adalah istilah khusus dari regularisasi yang digunakan, yang bisa berupa norma L2 dari bobot sering disebut sebagai *weight decay*, atau bentuk regularisasi lainnya seperti L1. Pemilihan istilah ini bergantung pada apa yang ingin dicapai oleh pembuat model.

2.3.8 Aspect Index Generator

Definisi Index Generator

Aspect index generator adalah algoritma yang digunakan untuk memetakan elemen-elemen penting dari data tidak terstruktur menjadi indeks yang lebih terorganisasi dan terstruktur. AIG ini bertujuan untuk mengubah teks atau informasi tidak terstruktur menjadi representasi yang lebih terorganisasi, sehingga memudahkan analisis data lebih lanjut. Dalam konteks pemrosesan teks, AIG membantu menangkap hubungan semantik, sintaksis, dan kontekstual antara elemen-elemen dalam teks dapat dilihat pada Gambar 2.18.



Gambar 2. 18 Proses *aspect index generator.*

Proses pengindekan dalam analisis sentimen menggunakan BERT dan AIG melibatkan beberapa tahapan penting. Pertama, data teks dikumpulkan dari berbagai sumber seperti media sosial, ulasan produk, artikel berita, dan forum diskusi. Data ini kemudian disimpan dalam bentuk yang dapat diolah oleh sistem. Setelah pra-pemrosesan data teks dipecah menjadi unit-unit kecil yang disebut token melalui proses tokenisasi. BERT kemudian digunakan untuk menghasilkan representasi vektor dari token-token ini, yang menangkap konteks dua arah dari

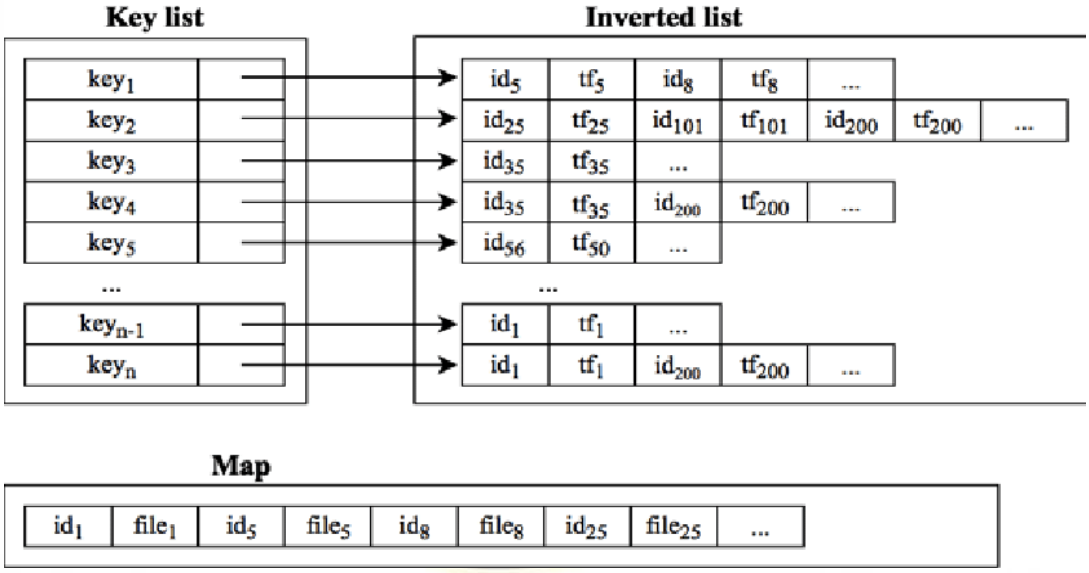
kata-kata dalam kalimat. Dengan menggunakan algoritma pembelajaran mesin, token yang telah diekstraksi dan diubah menjadi representasi vektor oleh BERT diklasifikasikan berdasarkan sentimen yang terkandung seperti positif, negatif, atau netral. Setelah sentimen diklasifikasikan, data teks diindeks menggunakan aspect index generator. Aspect index generator ini membuat indeks yang digunakan untuk proses pencarian dan pengambilan data berdasarkan kategori sentimen yang telah didefinisikan.

Proses tokenisasi dalam aspect index generator dapat dilihat pada Gambar 2.19.

Gambar 2. 19 Tokenisasi dalam indeks generator.

Tokenisasi dalam indeks generator adalah proses penting yang memecah teks menjadi unit-unit kecil yang disebut token, yang biasanya berupa kata atau frasa. Proses ini dimulai dengan mengambil teks mentah dan membersihkannya dari elemen-elemen yang tidak relevan seperti tanda baca dan angka. Setelah itu, teks dipecah menjadi token-token yang lebih kecil. Tokenisasi memungkinkan sistem

untuk mengidentifikasi dan mengkategorikan kata-kata kunci yang relevan, yang kemudian digunakan untuk membuat indeks. Indeks ini membantu dalam pencarian dan pengambilan data yang lebih efisien, karena setiap token dihubungkan dengan informasi yang relevan dalam teks asli. Dengan menggunakan indeks generator, proses tokenisasi menjadi lebih terstruktur dan sistematis, sehingga analisis data teks yang akurat. Proses penetuan key index dapat dilihat pada Gambar 2.20.



Gambar 2. 20 Menentukan key index untuk setiap token.

Menentukan key index untuk setiap token adalah langkah penting dalam proses pengindekan data teks. Proses ini dimulai dengan tokenisasi, di mana teks dipecah menjadi unit-unit kecil yang disebut token. Setiap token kemudian dianalisis untuk menentukan indeks kunci (key index) yang akan digunakan dalam sistem pengindekan. Indeks kunci ini berfungsi sebagai penanda unik untuk setiap token, memungkinkan sistem untuk dengan cepat mengidentifikasi dan mengakses informasi yang terkait dengan token tersebut. Proses penentuan key index melibatkan beberapa tahapan. Pertama setiap token diberi nilai numerik unik berdasarkan urutan kemunculannya untuk memastikan keunikan indeks. Setelah itu token-token ini diorganisir dalam struktur data yang efisien seperti tabel *hash* atau pohon *tree* untuk mempermudah dalam pencarian dan pengambilan data yang cepat dan akurat.

Fungsi dan manfaat aspect index generator

Pentingnya penggunaan AIG dalam penelitian ini terletak pada kemampuannya untuk mendeteksi hiponim, yang sering dijumpai dalam ulasan mahasiswa. Hiponim adalah kata yang memiliki makna lebih spesifik dan terbatas dalam konteks tertentu. Dalam analisis sentimen berbasis aspek, perbedaan makna antara kata-kata yang tampak serupa ini harus dapat dikenali dengan tepat.

AIG berfungsi untuk memisahkan dan mengorganisir makna yang berbeda ini dengan cara menyusun teks menjadi bentuk yang lebih terstruktur. Dengan mengidentifikasi kata-kata dan frasa dalam ulasan mahasiswa yang mengandung konteks semantik yang lebih mendalam, alat ini membantu model untuk memahami serta memetakan makna kata-kata tersebut dengan lebih akurat. Proses ini menjamin bahwa analisis sentimen yang dilakukan lebih tepat sasaran, karena sentimen yang berhubungan dengan aspek tertentu seperti dosen, kurikulum, atau sarana prasarana dapat dianalisis dengan mempertimbangkan konteks yang lebih luas.

AIG tidak hanya berfungsi untuk mengorganisir data, tetapi juga untuk memastikan bahwa makna spesifik dari hiponim dapat dipahami dengan benar dalam analisis sentimen berbasis aspek. Pendekatan ini menjadi sangat penting dalam menangani kompleksitas bahasa yang terkandung dalam ulasan mahasiswa, yang sering kali melibatkan kalimat majemuk dan kata-kata dengan makna ganda. Adapun manfaat dari aspect index generator adalah:

a. Pengorganisasian Data: Aspect index generator mengubah data tidak terstruktur seperti ulasan atau dokumen teks menjadi format yang terstruktur dan dapat diolah oleh mesin.
b. Peningkatan Analisis Semantik: Aspect index generator ini membantu memahami hubungan antar kata, frasa, atau aspek dalam teks, termasuk konteks dan makna ganda.
c. Efisiensi Proses: Dengan membuat data lebih terstruktur, aspect index generator ini memungkinkan proses analisis yang lebih cepat dan efisien.

d. Penanganan Kalimat Kompleks: Aspect index generator ini dapat menangkap makna dari kalimat majemuk atau frasa yang mengandung ambiguitas, seperti kata hiponim, sehingga menghasilkan analisis yang lebih akurat.

Komponen utama aspect index generator

a. Ekstraksi Fitur: Proses awal untuk mengidentifikasi elemen penting dalam teks, seperti kata kunci, frasa, atau aspek.
b. Pemberian Bobot: Aspect index generator menetapkan bobot pada elemen yang diekstraksi berdasarkan relevansi atau kepentingannya dalam konteks teks.
c. Normalisasi: Proses untuk memastikan bahwa indeks yang dihasilkan konsisten dan relevan, termasuk menangani sinonim, hiponim, atau variasi linguistik lainnya.
d. Mapping dan Pengelompokan: Aspect index generator memetakan elemen yang relevan ke dalam kelompok atau kategori untuk mempermudah analisis lebih lanjut. Dalam analisis sentimen berbasis aspek, index generator memainkan peran penting dalam memisahkan aspek-aspek spesifik dari teks ulasan (misalnya, dosen, kurikulum, fasilitas) dan menghubungkannya dengan sentimen yang relevan.

Keunggulan aspect index generator

1. Mengatasi Kalimat Majemuk: Index generator mampu memproses kalimat kompleks dengan lebih baik, seperti memisahkan aspek yang berbeda dalam satu kalimat majemuk.
2. Penanganan Makna Hiponim: Aspect index generator ini membantu membedakan makna kata-kata yang sama tergantung pada konteksnya, misalnya "kelas" sebagai ruang belajar atau sebagai kelompok siswa.
3. Fleksibilitas: Index generator dapat digunakan untuk berbagai jenis data teks, termasuk ulasan, komentar media sosial, atau dokumen resmi.

Implementasi aspect index generator

Dalam implementasinya, index generator sering dikombinasikan dengan model machine learning atau deep learning, seperti BERT. Kombinasi ini memungkinkan pemahaman yang lebih dalam terhadap konteks kata dalam kalimat, baik sebelum maupun sesudah kata tersebut. Misalnya, BERT memahami konteks kata "kelas" dalam frasa *"kelas belajar"* dan *"kelas atas"* untuk memberikan interpretasi yang sesuai.

Implementasi aspect index generator dapat dikombinasikan dengan model machine learning atau deep learning seperti BERT untuk meningkatkan pemahaman konteks kata dalam kalimat. Kombinasi ini dapat menganalisis dan memahami kata dalam konteks sebelum dan sesudahnya dalam sebuah kalimat. Sebagai contoh, BERT mampu membedakan arti kata "kelas" dalam "kelas belajar" dan "kelas atas," sehingga memberikan interpretasi yang lebih tepat. Proses implementasi *aspect index generator* melibatkan beberapa langkah utama, dimulai dengan praproses teks di mana teks dibersihkan dan dinormalisasi.

Selanjutnya, fitur-fitur penting seperti kata kunci dan frasa diekstraksi. Fitur-fitur ini kemudian digunakan untuk membangun indeks yang memungkinkan pencarian dan analisis teks yang lebih efisien. Dalam konteks ulasan mahasiswa, Index Generator dapat digunakan untuk menganalisis umpan balik terkait aspek dosen. Dengan menggunakan BERT, sistem dapat mengidentifikasi dan memahami komentar mahasiswa mengenai kinerja dosen, metode pengajaran, dan interaksi dengan mahasiswa. Selain itu, implementasi index generator juga sangat berguna untuk mengevaluasi aspek kurikulum, dengan menganalisis ulasan mahasiswa, sistem dapat mengidentifikasi kekuatan dan kelemahan kurikulum yang diterapkan.

Hubungan antara BERT dan AIG sangatlah penting dalam proses ini. BERT berfungsi sebagai model yang mampu memahami konteks kata dalam kalimat secara mendalam. BERT dapat menghasilkan representasi dari kata yang akan diindeks. AIG kemudian memanfaatkan representasi ini untuk membangun indeks yang lebih kaya dan bermakna. Dalam analisis ulasan mahasiswa, kombinasi ini sistem dapat menangkap nuansa dan konteks dari setiap komentar, menghasilkan analisis yang baik dan relevan. Dengan adanya Index Generator yang didukung oleh

BERT. AIG digunakan untuk membuat indeks yang menyederhanakan pencarian dan pengambilan data terkait dalam ruang lingkup model BERT. Indeks yang dihasilkan dapat berupa representasi vektor dari kata-kata atau frasa dalam dokumen yang telah di-tokenisasi, yang kemudian disimpan dalam struktur indeks yang terorganisir. Dengan cara ini, alih-alih mengakses model BERT setiap kali pencarian dilakukan, indeks generator memungkinkan pencarian dilakukan pada level indeks, mempercepat waktu respons dan mengurangi beban komputasi.

Relevansi dalam penelitian

Dalam penelitian terkait analisis ulasan mahasiswa, AIG memiliki relevansi yang sangat signifikan. AIG ini dirancang untuk mengatasi tantangan analisis data tidak terstruktur, yang sering kali ditemui dalam ulasan-ulasan mahasiswa. Data tidak terstruktur ini dapat mencakup teks panjang, kalimat majemuk, dan makna hiponim, yang semuanya bisa menyulitkan analisis menggunakan metode tradisional.

AIG bekerja dengan mengurai ulasan mahasiswa ke dalam unit-unit yang lebih kecil, seperti kalimat atau kata kunci. Proses ini memungkinkan algoritma untuk menangkap makna di balik setiap elemen teks, bahkan pada kalimat yang kompleks. Sebagai contoh, ketika mahasiswa memberikan ulasan tentang seorang dosen, aspect index generator dapat mengidentifikasi aspek-aspek seperti metode pengajaran, keahlian, dan interaksi dengan mahasiswa. Selanjutnya, aspect index generator ini mampu mengelompokkan dan mengkategorikan ulasan berdasarkan tema yang relevan. Misalnya, ulasan tentang kurikulum dapat dikelompokkan menjadi topik-topik seperti kesesuaian materi, tingkat kesulitan, dan integrasi dengan praktik nyata. Dengan demikian, peneliti dapat memperoleh gambaran yang lebih jelas mengenai aspek mana dari kurikulum yang perlu diperbaiki atau dipertahankan. Fasilitas kampus adalah aspek lain yang sering diulas oleh mahasiswa. Melalui penggunaan index generator, peneliti dapat mengidentifikasi berbagai kategori ulasan fasilitas, seperti kebersihan, ketersediaan peralatan, dan kenyamanan. Proses ini memungkinkan evaluasi yang lebih terperinci dan akurat, membantu manajemen kampus untuk melakukan perbaikan yang diperlukan.

Layanan mahasiswa, seperti bimbingan akademik, administrasi, dan kesehatan, juga menjadi fokus analisis.

Kemampuan aspect index generator untuk menangani data tidak terstruktur memberikan banyak manfaat dalam penelitian. AIG ini membantu peneliti untuk menggali informasi yang mendalam dan mendapatkan wawasan yang berharga dari data yang kompleks dan bervariasi. Dengan demikian, penelitian dapat memberikan kontribusi yang lebih besar dalam memahami dan meningkatkan berbagai aspek kehidupan kampus dan masyarakat pada umumnya. Secara keseluruhan, penggunaan AIG dalam penelitian ulasan mahasiswa menunjukkan potensi besar untuk meningkatkan akurasi dan efisiensi analisis data. Proses indeksasi mempercepat pemanggilan informasi serta meningkatkan performa model BERT untuk digunakan secara lebih efisien dalam konteks aplikasi yang membutuhkan pengolahan teks dalam jumlah besar, tanpa membebani sistem secara berlebihan.

2.3.9 Hiponim

2.3.9.1 Definisi hiponim

Hiponim merupakan salah satu konsep dalam linguistik yang merujuk pada kata atau frasa yang memiliki makna yang lebih spesifik dan sempit dibandingkan dengan kata atau frasa lainnya yang lebih umum. Menurut Cruse (2024), hiponim adalah kata yang termasuk dalam kategori yang lebih luas atau superordinat. Misalnya, kata "penguji" adalah hiponim dari kata "dosen", karena penguji adalah contoh yang lebih spesifik dari kategori umum dosen. Konsep ini penting dalam analisis semantik karena membantu mengklasifikasikan dan memahami hubungan antar kata dalam bahasa. Hiponim adalah kata yang menggambarkan bagian atau jenis dari kategori yang lebih besar. Sebagai contoh, dalam kategori dosen, hiponim seperti "dosen fisika", "dosen kimia", dan "dosen matematika" merupakan turunan dari kata yang lebih umum, yaitu dosen. Dengan demikian, setiap hiponim mempersempit ruang lingkup makna yang dimiliki oleh superordinatnya, tetapi tetap berhubungan erat dengan kategori yang lebih luas.

Dalam pengertian yang lebih luas, hiponim dapat ditemukan dalam banyak aspek kehidupan, termasuk dalam dunia pendidikan. Dalam hal ini, kurikulum

merupakan superordinat yang dapat mencakup berbagai jenis kurikulum spesifik seperti kurikulum matematika, kurikulum bahasa Indonesia, dan kurikulum teknik. Kata-kata ini merupakan hiponim dari kata "kurikulum", menggambarkan bidang studi yang lebih sempit dan lebih terperinci dalam konteks pendidikan.

Menurut Haspelmath (2024), hubungan antara superordinat dan hiponim mencerminkan cara manusia mengelompokkan pengetahuan. Setiap kali manusia memberikan nama atau label untuk sebuah objek atau konsep yang lebih spesifik, mereka menggunakan hiponim. Oleh karena itu, dalam kajian linguistik, hubungan antara hiponim dan superordinat penting untuk memahami bagaimana makna diperoleh dan diterjemahkan dalam konteks yang lebih luas, seperti dalam ulasan mahasiswa terhadap aspek pendidikan. Di dunia pendidikan, pemahaman terhadap hiponim juga mempermudah proses komunikasi dan penilaian. Misalnya, dalam memberikan ulasan tentang sarpras, mahasiswa dapat menggunakan hiponim seperti laboratorium komputer, perpustakaan digital, atau ruang seminar. Penggunaan hiponim dalam ulasan ini memungkinkan mahasiswa untuk memberikan penilaian yang lebih terperinci mengenai fasilitas yang ada. Dengan kata lain, hiponim memperkaya komunikasi dalam konteks akademik dengan memberikan detail yang lebih jelas dan tepat.

2.3.9.2 Ciri-ciri hiponim

Hiponim adalah istilah dalam linguistik yang merujuk pada kata atau frasa yang memiliki makna lebih sempit dan spesifik dibandingkan dengan kata yang lebih umum, yang disebut superordinat. Dalam bahasa Indonesia, hubungan antara hiponim dan superordinat sering dijumpai dalam berbagai konteks, baik dalam kehidupan sehari-hari maupun dalam pembahasan ilmiah. Pemahaman tentang hiponim sangat penting karena memungkinkan pengelompokan kata-kata berdasarkan kesamaan makna, serta membantu memperjelas makna dalam komunikasi, khususnya dalam evaluasi aspek-aspek tertentu, seperti dosen, kurikulum, sarpras, dan layanan dalam dunia pendidikan. Dengan memahami ciri-ciri hiponim, hubungan antara kata-kata yang lebih umum dan kata-kata yang lebih terperinci dapat diidentifikasi dengan lebih jelas dengan pemahaman yang lebih

mendalam dalam analisis linguistik. Adapun ciri – ciri kata atau kalimat yang mengandung makna hiponin sebagai berikut:

1. Hubungan hierarkis dengan superordinat

Hiponim adalah hubungan hierarkis dengan kata yang lebih umum, yaitu superordinat. Dalam linguistik, superordinat adalah kata yang memiliki makna yang lebih luas, sementara hiponim adalah kata yang lebih sempit dan spesifik. Misalnya, dosen merupakan superordinat dari hiponim seperti dosen fisika, dosen kimia, dan dosen matematika. Masing-masing kata tersebut lebih spesifik dalam menggambarkan jenis dosen yang mengajar di bidang tertentu. Begitu pula dengan kurikulum, yang merupakan superordinat bagi hiponim seperti kurikulum matematika, kurikulum fisika, dan kurikulum seni, yang menggambarkan mata pelajaran atau bidang studi yang lebih khusus.

2. Menunjukkan subkategori dalam kategori yang lebih luas

Hiponim memiliki kemampuann untuk menggambarkan subkategori dalam suatu kategori yang lebih besar. Sebagai contoh, dalam dunia pendidikan, sarpras adalah superordinat yang mencakup berbagai fasilitas seperti perpustakaan, laboratorium komputer, dan ruang kelas. Masing-masing fasilitas ini adalah hiponim dari kata sarpras, karena mereka merujuk pada elemen-elemen yang lebih spesifik dalam kategori sarana dan prasarana pendidikan. Penggunaan hiponim ini memudahkan pemahaman dan memberikan detail yang lebih terperinci tentang fasilitas atau fasilitas pendidikan yang dimaksud dalam sebuah ulasan atau evaluasi.

3. Memberikan deskripsi yang lebih spesifik

Hionim memberikan deskripsi yang lebih spesifik dan terperinci dalam ulasan mahasiswa terhadap aspek dosen, misalnya kata dosen fisika akan memberikan informasi yang lebih jelas dan terfokus daripada hanya menyebutkan dosen secara umum. Demikian pula, kata laboratorium komputer memberikan gambaran yang lebih mendetail mengenai sarpras yang dimaksud daripada hanya menggunakan kata sarpras.