DETEKSI KEMIRIPAN JUDUL SKRIPSI BERBASIS SEMANTIC SIMILARITY MENGGUNAKAN SENTENCE BERT

SKRIPSI

Oleh :
ZULHAM GHINAFIKAR
NIM. 220605110184

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2026

DETEKSI KEMIRIPAN JUDUL SKRIPSI BERBASIS SEMANTIC SIMILARITY MENGGUNAKAN SENTENCE BERT

SKRIPSI

Diajukan kepada:
Universitas Islam Negeri Maulana Malik Ibrahim Malang
Untuk memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)

Oleh :
ZULHAM GHINAFIKAR
NIM. 220605110184

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG
2026

HALAMAN PERSETUJUAN

DETEKSI KEMIRIPAN JUDUL SKRIPSI BERBASIS SEMANTIC SIMILARITY MENGGUNAKAN SENTENCE BERT

SKRIPSI

Oleh :
ZULHAM GHINAFIKAR
NIM. 220605110184

Telah Diperiksa dan Disetujui untuk Diuji:
Tanggal: 26 Juni 2026

Pembimbing I,	Pembimbing II,
Dr. Zainal Abidin M.Kom	Khadijah Fahmi Hayati Holle, M.Kom
NIP. 19760613 200501 1 004	NIP. 19900626 202203 2 002

Mengetahui,
Ketua Program Studi Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang

Supriyono, M.Kom
NIP. 19841010 201903 1 012

HALAMAN PENGESAHAN

DETEKSI KEMIRIPAN JUDUL SKRIPSI BERBASIS SEMANTIC SIMILARITY MENGGUNAKAN SENTENCE BERT

SKRIPSI

Oleh :
ZULHAM GHINAFIKAR
NIM. 220605110184

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal: 26 Juni 2026

Susunan Dewan Penguji

Ketua Penguji	: Dr. Ririen Kusumawati, S.Si, M.Kom NIP. 19720309 200501 2 002	( )
Anggota Penguji I	: Nurizal Dwi Priandani, M.Kom NIP. 19920830 202203 1 001	( )
Anggota Penguji II	: Dr. Zainal Abidin, M.Kom NIP. 19760613 200501 1 004	( )
Anggota Penguji III	: Khadijah Fahmi Hayati Holle, M.Kom NIP. 19900626 202203 2 002	( )

Mengetahui dan Mengesahkan,
Ketua Program Studi Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang

Supriyono, M.Kom
NIP. 19841010 201903 1 012

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama	: ZULHAM GHINAFIKAR
NIM	: 220605110184
Fakultas / Program Studi	: Sains dan Teknologi / Teknik Informatika
Judul Skripsi	: DETEKSI KEMIRIPAN JUDUL SKRIPSI BERBASIS SEMANTIC SIMILARITY MENGGUNAKAN SENTENCE BERT

Menyatakan dengan sebenarnya bahwa Skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambil alihan data, tulisan, atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka.

Apabila dikemudian hari terbukti atau dapat dibuktikan skripsi ini merupakan hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 26 Juni 2026
Yang membuat pernyataan,

ZULHAM GHINAFIKAR
NIM. 220605110184

MOTTO

“Try to move with the good flow”

"Semua arus yang coba diikuti adalah baik, cause you observe, research and even try to follow all the flow.
Jadi mana yang akan menjadi pilihan terakhirmu dari semua arus itu?"

HALAMAN PERSEMBAHAN

Alhamdulillahi rabbil 'alamin, puji syukur senantiasa penulis panjatkan ke hadirat Allah *Subḥānahu wa Ta'ālā* atas segala limpahan rahmat, hidayah, dan karunia-Nya, sehingga skripsi ini dapat diselesaikan dengan baik. Shalawat serta salam semoga senantiasa tercurah kepada junjungan kita, Nabi Muhammad *Ṣallallāhu 'Alaihi Wasallam*, yang telah membawa cahaya kebenaran bagi umat manusia. Melalui lembar ini, penulis ingin menorehkan rasa terima kasih yang tak terhingga kepada pihak-pihak yang telah menjadi pelita dan sandaran selama menempuh perjalanan di perguruan tinggi.

Karya tulis ini secara khusus penulis persembahkan sebagai wujud bakti dan cinta kasih kepada kedua orang tua tercinta. Terima kasih atas doa-doa yang tak pernah lelah dilangitkan, pengorbanan yang tak ternilai, serta kasih sayang yang selalu menjadi sumber kekuatan penulis dalam menghadapi setiap rintangan. Rasa terima kasih yang tulus juga penulis sampaikan kepada saudara-saudara tercinta atas segala dukungan moral, motivasi, dan bantuan yang terus mengalir sejak hari pertama perkuliahan hingga skripsi ini rampung.

Perjalanan akademik ini tidak akan terasa bermakna tanpa kehadiran sahabat dan rekan seperjuangan. Apresiasi penulis dedikasikan untuk teman-teman seperjuangan angkatan 2022 yang saling merangkul agar kita bisa lulus tepat waktu. Terima kasih atas canda tawa dan kebersamaan yang menjadikan angkatan ini sebagai ruang paling nyaman di tengah penatnya masa revisi. Kepada pihak-pihak lain yang tidak dapat disebutkan satu per satu, terima kasih atas setiap uluran tangan dan andil besarnya.

Pada akhirnya, sebuah apresiasi yang paling jujur dan mendalam penulis tujukan untuk diri sendiri. Terima kasih karena telah memilih untuk terus bertahan, berjuang, dan menolak menyerah di fase-fase yang penuh tekanan. Terima kasih atas keteguhan hati serta malam-malam panjang di depan layar yang telah dilalui demi menuntaskan tanggung jawab ini hingga ke garis akhir.

Penulis menyadari sepenuhnya bahwa keberhasilan ini adalah buah dari kebaikan kolektif yang tak mungkin mampu terbalas dengan sepadan. Oleh karena itu, penulis hanya dapat berdoa semoga Allah *Subḥānahu wa Ta'ālā* membalas setiap kebaikan, waktu, dan energi yang telah diberikan dengan ganjaran yang berlipat ganda. Semoga langkah kita ke depannya senantiasa diberkahi, dan karya sederhana ini dapat memberikan manfaat yang nyata bagi perkembangan keilmuan, khususnya di bidang Teknik Informatika.

KATA PENGANTAR

Alhamdulillahi rabbil 'alamin, puji syukur senantiasa penulis panjatkan ke hadirat Allah *Subḥānahu wa Ta'ālā* atas segala limpahan rahmat, hidayah, serta kasih sayang-Nya yang telah memberikan kemudahan dan kelancaran bagi penulis untuk merampungkan karya ilmiah yang berjudul "DETEKSI KEMIRIPAN JUDUL SKRIPSI BERBASIS SEMANTIC SIMILARITY MENGGUNAKAN SENTENCE BERT". Shalawat seiring salam semoga terus tercurahkan kepada junjungan besar Nabi Muhammad *Ṣallallāhu 'Alaihi Wasallam*, teladan utama umat manusia yang syafaatnya selalu kita harapkan di hari akhir nanti, Aamiin.

Penulis menyadari sepenuhnya bahwa perjalanan panjang menyelesaikan skripsi ini tidak akan pernah terwujud tanpa adanya bimbingan, ketulusan, serta dukungan kolektif dari berbagai pihak. Oleh karena itu, dengan segala kerendahan hati, penulis ingin memberikan rasa terima kasih dan penghormatan yang setinggi-tingginya kepada:

1. Prof. Dr. Hj. Ilfi Nurdiana, M.Si., CAHRM., CRMP selaku Rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang, atas kesempatan dan fasilitas yang diberikan selama penulis menempuh pendidikan.
2. Dr. Agus Mulyono, M.Kes. selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang, atas segala dukungan dan kebijakan selama masa studi penulis.
3. Dr. Ririen Kusumawati, S.Si, M.Kom. selaku Ketua Penguji, yang telah meluangkan waktu serta memberikan arahan dan evaluasi yang sangat berharga demi kesempurnaan skripsi ini.

4. Nurizal Dwi Priandani, M.Kom. selaku Anggota Penguji I, atas segala masukan, koreksi, dan saran yang membangun untuk menyempurnakan penelitian ini.
5. Dr. Zainal Abidin, M.Kom. selaku Pembimbing I, yang telah memberikan evaluasi mendalam serta arahan yang sangat bermanfaat agar karya ilmiah ini menjadi lebih baik.
6. Khadijah Fahmi Hayati Holle, M.Kom. selaku Pembimbing II, atas waktu, perhatian, dan bimbingannya selama proses pengujian skripsi berlangsung.
7. Kedua orang tua serta saudara-saudara tercinta, pilar kekuatan utama yang tiada henti melangitkan doa, memberikan pengorbanan, dan motivasi agar penulis dapat menuntaskan studi dengan lancar.

Terakhir, sebuah apresiasi yang paling jujur penulis tujukan untuk diri sendiri. Terima kasih karena telah memilih untuk terus bertahan, berjuang, dan menolak untuk menyerah di setiap fase yang melelahkan. Terima kasih atas segala kerja keras, keteguhan hati, serta malam-malam panjang yang telah dilalui di depan layar demi menyelesaikan tanggung jawab ini hingga ke garis akhir. Semoga karya tulis ini dapat membawa keberkahan dan manfaat bagi perkembangan keilmuan ke depannya.

Malang, 26 Juni 2026

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

ABSTRAK

Ghinafikar, Zulham. 2026. **Deteksi Kemiripan Judul Skripsi Berbasis Semantic Similarity Menggunakan Sentence BERT**. Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Dr. Zainal Abidin, M.Kom (II) Khadijah Fahmi Hayati Holle, M.Kom.

.

Kata Kunci: Semantic Similarity, Sentence-BERT, Cosine Similarity, Deteksi Kemiripan Teks, Skripsi.

Peningkatan volume skripsi di perguruan tinggi memunculkan tantangan besar dalam memverifikasi kebaruan judul guna menghindari duplikasi penelitian. Sistem deteksi kemiripan leksikal konvensional seringkali gagal mengenali kesamaan makna pada dua judul yang menggunakan susunan kata yang berbeda. Penelitian ini bertujuan mengevaluasi kinerja *pipeline* deteksi kemiripan judul skripsi berbasis semantic similarity menggunakan model *Sentence-BERT* dan *Cosine Similarity*. Guna memastikan validitas evaluasi, penelitian ini menyusun dataset dari 1.887 judul skripsi melalui teknik *pairwise combination*, yang kemudian disaring menggunakan *Stratified Under-sampling* dan divalidasi oleh *domain expert* hingga menghasilkan 700 pasangan *ground truth*. Pengujian performa sistem dilakukan melalui skema *Nested K-Fold Cross Validation* dengan membandingkan metode pencarian *threshold* berbasis *Grid Search* dari penelitain Ghinassi dan Holis. Hasil uji coba menunjukkan bahwa skenario pencarian Ghinassi pada rentang 0.05 - 0.95 menggunakan skema *5-Fold Cross Validation* memberikan performa paling optimal, dengan mencetak tingkat Akurasi 82,00%, Presisi 83,20%, Recall 95,44%, F1-Score 88,76%, serta specificity 42,39%. Melalui pengujian lanjutan *Flat K-Fold CV*, ditetapkan bahwa *threshold* 0,50 merupakan titik keseimbangan tertinggi dengan tingkat Akurasi 82,00% dan F1-Score 89,03%. Penelitian ini menyimpulkan bahwa model *SBERT* terbukti sangat efektif dan peka dalam menangkap konteks semantik, sementara konfigurasi *threshold* 0,50 mampu memberikan titik deteksi yang stabil.

ABSTRACT

Ghinafikar, Zulham. 2026. **Similarity Detection of Thesis Titles Based on Semantic Similarity Using Sentence BERT.** Thesis. Department of Informatics Engineering, Faculty of Science and Technology, State Islamic University of Maulana Malik Ibrahim Malang. Supervisor: (I) Dr. Zainal Abidin, M.Kom (II) Khadijah Fahmi Hayati Holle, M.Kom.
.

Keywords: Semantic Similarity, Sentence-BERT, Cosine Similarity, Text Similarity Detection, Thesis.

The increasing volume of theses in universities poses a major challenge in verifying the novelty of titles to avoid research duplication. Conventional lexical similarity detection systems often fail to recognize the similarity of meaning in two titles that use different wording. This study aims to evaluate the performance of a semantic similarity-based thesis title similarity detection pipeline using the Sentence-BERT and Cosine Similarity models. To ensure the validity of the evaluation, this study compiled a dataset of 1,887 thesis titles through a pairwise combination technique, which was then filtered using Stratified Under-sampling and validated by domain experts to produce 700 ground truth pairs. System performance testing was carried out using the Nested K-Fold Cross Validation scheme by comparing the Grid Search-based threshold search method from the Ghinassi and Holis research. The test results show that the Ghinassi search scenario in the range of 0.05 - 0.95 using the 5-Fold Cross Validation scheme provides the most optimal performance, with an Accuracy level of 82.00%, Precision of 83.20%, Recall of 95.44%, F1-Score of 88.76%, and specificity 42.39%. Through further Flat K-Fold CV testing, it was determined that the threshold of 0.50 was the highest balance point with an Accuracy level of 82.00% and an F1-Score of 89.03%. This study concludes that the SBERT model is proven to be very effective and sensitive in capturing semantic context, while the threshold configuration of 0.50 is able to provide stable detection points.

البحث مستخلص

*غنافيكر، زولهام. 2026. **اكتشاف التشابه في عناوين الأطروحات بناء على التشابه الدلالي باستخدام SBERT**. أطروحة. قسم هندسة المعلوماتية، كلية العلوم والتكنولوجيا، جامعة الدولة الإسلامية في مولانا مالك إبراهيم مالانغ. المشرف: (الأولى) الدكتور زين العابدين، م. كوم (الثاني) خديجة فهمي حياة هولي، م. كوم.*

.

***الكلمات المفتاحية:** التشابه الدلالي، SBERT، تشابه جيب التمام، اكتشاف تشابه النص، الأطروحة.*

يشكل حجم الرسائل المتزايدة في الجامعات تحديا كبيرا في التحقق من حداثة العناوين لتجنب تكرار الأبحاث. غالبا ما تفشل أنظمة اكتشاف التشابه المعجمي التقليدية في التعرف على تشابه المعنى في عنوانين يستخدمان صياغة مختلفة. تهدف هذه الدراسة إلى تقييم أداء خط كشف تشابه عنوان أطروحة يعتمد على التشابه الدلالي باستخدام نموذجي (SBERT) و(Cosine Similarity). لضمان صحة التقييم، جمعت هذه الدراسة مجموعة بيانات تضم 1,887 عنوان أطروحة عبر تقنية الدمج الزوجي، والتي تم تصفيتها بعد ذلك باستخدام التخفيض الطبقي والتحقق منها من قبل خبراء المجال لإنتاج 700 زوج حقيقة أرضية. تم إجراء اختبار أداء النظام باستخدام مخطط (Nested CV K-Fold) من خلال مقارنة طريقة البحث العتبي المعتمدة على البحث الشبكي من أبحاث غيناسي وهوليس. تظهر نتائج الاختبار أن سيناريو البحث في غيناسي في النطاق من 0.05 إلى 0.95 باستخدام نظام التحقق المتقاطع (5-Fold) يوفر الأداء الأمثل، مع مستوى (دقة) 82.00٪، (دقة) 83.20٪، استرجاع 95.44٪، و(درجة F1) 88.76٪ و(خصوصية) 42.39٪. ومن خلال اختبارات إضافية (Flat K-Fold CV)، تم تحديد أن عتبة 0.50 هي أعلى نقطة توازن مع مستوى (دقة) 82.00٪ ودرجة F1 تبلغ 89.03٪. خلصت هذه الدراسة إلى أن نموذج (SBERT) ثبت فعاليته وحساسيته جدا في التقاط السياق الدلالي، بينما تكوين العتبة 0.50 قادر على توفير نقاط كشف مستقرة.

BAB I
PENDAHULUAN

1.1 Latar Belakang

Produksi karya ilmiah di lingkungan perguruan tinggi khususnya skripsi menunjukkan volume yang terus meningkat secara signifikan pertahunnya. Sebagai contoh kecil di program studi Teknik Informatika UIN Malang, dihasilkan 165 judul skripsi yang terbit khusus pada tahun 2024 dan 111 judul skripsi sebelum penghujung tahun 2025. Seiring bertambahnya volume judul skripsi di repositori yang tersebar luas ke dalam berbagai rumpun penjurusan keilmuan, muncul masalah dalam proses penjaminan mutu, deteksi awal potensi kemiripan antarjudul dan verifikasi judul baru yang akan di*input*kan kedepannya (Irawan *et al.,* 2021).

Setiap pengajuan judul skripsi baru memerlukan tahap verifikasi yang ketat guna mendukung proses identifikasi awal terhadap kemiripan dengan penelitian terdahulu. Pada tahap ini, dosen pembimbing maupun komite penilai dihadapkan pada satu pertanyaan krusial, yakni sejauh mana topik yang diajukan memiliki kemiripan semantik dengan penelitian yang telah ada. Mengingat volume data yang terus bertambah, proses pengecekan secara manual ribuan dokumen menjadi tidak efisien, memakan waktu, dan rentan terhadap *human error*. Oleh karena itu, diperlukan sebuah sistem pendukung yang mampu mendeteksi kemiripan judul secara otomatis dan akurat untuk membantu proses verifikasi awal ini (Sanjaya *et al.,* 2016).

Prinsip fundamental untuk senantiasa mengevaluasi masa lalu guna menghasilkan karya yang lebih baik di masa depan merupakan pilar penting yang ditekankan dalam ajaran Islam. sistem pendukung untuk deteksi awal kemiripan semantik judul skripsi sejatinya adalah manifestasi dari prinsip tersebut. Allah SWT berfirman dalam Q.S. Al-Hasyr ayat 18:

يٰٓاَيُّهَا الَّذِيْنَ اٰمَنُوا اتَّقُوا اللّٰهَ وَلْتَنْظُرْ نَفْسٌ مَّا قَدَّمَتْ لِغَدٍۚ وَاتَّقُوا اللّٰهَۗ اِنَّ اللّٰهَ خَبِيْرٌۢ بِمَا تَعْمَلُوْنَ

"*Wahai orang-orang yang beriman, bertakwalah kepada Allah dan hendaklah setiap orang memperhatikan apa yang telah diperbuatnya untuk hari esok (akhirat). Bertakwalah kepada Allah. Sesungguhnya Allah Mahateliti terhadap apa yang kamu kerjakan*." (Q.S Al- Hasyr : 18)

Penafsiran dalam kitab Tafsir Al-Mishbah jilid 14 memberikan penjelasan bahwa lafaz *waltandzur* bermakna anjuran bagi manusia untuk memikirkan secara mendalam dan melakukan *muhasabah*. Adapun kalimat *mâ qaddamat lighad* menegaskan bahwa setiap individu harus menelaah perbuatan atau rekam jejak masa lalunya sebagai pijakan merencanakan masa depan. Secara substansial, ayat ini adalah perintah untuk mempersiapkan hari esok yang lebih baik berdasarkan hasil evaluasi dari apa yang telah berlalu (Shihab, 2005).

Jika prinsip tafsir ini ditarik ke ranah akademik, perintah mengevaluasi capaian masa lalu demi hari esok yang lebih baik merupakan esensi dari *state of the art*. Mahasiswa dituntut untuk secara cermat memverifikasi judul-judul yang sudah ada agar penelitian saat ini tidak sekadar mengulang penelitian terdahulu. Ketiadaan sistem pendukung yang mampu memberikan deteksi awal terhadap kemiripan semantik judul akan membuat proses evaluasi masa lalu ini menjadi kurang optimal,

sehingga berpotensi menyulitkan proses identifikasi kebaruan penelitian dan pada akhirnya dapat memengaruhi mutu akademik institusi.

Sistem deteksi kemiripan yang umum digunakan seringkali masih terbatas pada pencocokan kata kunci seperti *string matching* oleh Sanjaya *et al.,* (2016). Sistem seperti ini memiliki kelemahan fundamental, sebagai contoh sistem leksikal mungkin akan menganggap judul Analisis Dampak Pembelajaran Daring terhadap Motivasi Mahasiswa dan Studi Pengaruh Kuliah Online pada Semangat Belajar sebagai dua hal yang sangat berbeda karena keduanya menggunakan kata yang tidak identik. Padahal secara makna atau maksud kedua judul tersebut sangat mirip.

Untuk mengatasi kelemahan tersebut pendekatan deteksi kemiripan berbasis makna atau semantik menjadi sangat relevan. Pendekatan semantik tidak hanya membandingkan kata per kata, tetapi berupaya memahami konteks dan maksud dari keseluruhan kalimat judul. Salah satu metode tepat di bidang pemahaman bahasa alami yang mampu melakukan ini adalah *Sentence-BERT* (Reimers & Gurevych, 2019), sebuah model yang efektivitasnya dalam pemahaman semantik terus divalidasi dalam penelitian terbaru (Ranasinghe *et al.,* 2025). *SBERT* dirancang khusus untuk membaca kalimat dan memetakannya ke dalam sebuah representasi matematis yang menangkap makna semantiknya. Dengan cara ini dua judul yang berbeda susunan kata namun serupa maknanya akan diidentifikasi memiliki tingkat kemiripan yang tinggi.

Berdasarkan paparan permasalahan ini, terlihat adanya kesenjangan antara kebutuhan untuk mendukung proses verifikasi awal kemiripan semantik judul skripsi secara efisien dan keterbatasan sistem deteksi kemiripan yang ada saat ini

yang masih bersifat leksikal. Pemanfaatan teknologi pemahaman bahasa seperti SBERT untuk analisis kemiripan semantik menawarkan solusi menjanjikan.

1.2 Pernyataan Masalah

Kebutuhan akan sistem yang mampu mendeteksi kemiripan semantik antarjudul skripsi secara akurat diperlukan evaluasi yang komprehensif terhadap kinerja metode yang digunakan untuk mengetahui tingkat efektivitasnya. Oleh karena itu, bagaimana performa pipeline Sentence-BERT yang dipadukan dengan Cosine Similarity dalam mendeteksi kemiripan semantik antarjudul skripsi berdasarkan metrik accuracy, precision, recall, F1-score, dan specificity.

1.3 Batasan Masalah

a. Fokus utama sistem adalah mengukur *semantic similarity* antar judul skripsi.
b. Dataset untuk pengujian adalah kumpulan judul skripsi Teknik Informatika dari web E-Thesis UIN Malang tahun 2008 - 2026.
c. Label yang digunakan Adalah *Similar atau Not Similar*.
d. Sistem hanya melakukan deteksi awal *semantic similarity* teks judul.

1.4 Tujuan Penelitian

penelitian ini bertujuan untuk mengevaluasi performa pipeline Sentence-BERT yang dipadukan dengan Cosine Similarity dalam mendeteksi kemiripan semantik antarjudul skripsi berdasarkan metrik accuracy, precision, recall, F1-score, dan, specificity.

1.5 Manfaat Penelitian

a. Manfaat Akademis

   a) Menghasilkan sistem aplikasi yang dapat membantu mahasiswa dalam melakukan proses identifikasi awal terhadap kemiripan makna judul skripsi.

   b) Membantu mahasiswa dan dosen pembimbing memastikan kebaruan topik.

b. Manfaat Keilmuan

   a. Memberikan kontribusi pengetahuan mengenai performa dan efektivitas pipeline *SBERT + Cosine Similarity* pada NLP dan studi kasus judul skripsi.

BAB II
STUDI PUSTAKA

2.1 Penelitian Terkait

Penelitian dalam bidang pendeteksian kemiripan teks telah mengalami evolusi signifikan dengan pendekatan yang bervariasi, mulai dari metode fundamental berbasis *String Matching* hingga model representasi semantik modern. Tiap metode menawarkan sebuah *trade-off* antara kecepatan pemrosesan dan kedalaman pemahaman semantic. Rangkuman point pembahasan penelitian terdahulu dapat dilihat pada tabel 2.1.

Salah satu penelitian dasar adalah karya Devlin *et al.,* (2019) yang memperkenalkan arsitektur BERT. Model ini diuji pada tugas deteksi parafrasa menggunakan dataset *Microsoft Research Paraphrase Corpus* untuk menentukan apakah dua kalimat memiliki makna yang sama atau tidak. Hasil evaluasi menunjukkan model *BERT-Large* mencapai akurasi 86,6% dan F1-score 90,1%.

Penelitian lanjutan oleh Reimers & Gurevych, (2019) memperkenalkan *Sentence-BERT*, sebuah varian dari *BERT* yang dioptimalkan untuk menghasilkan representasi vektor kalimat bermakna, sehingga dapat digunakan secara efisien dalam pengukuran kemiripan teks. Melalui pengujian pada *benchmark Semantic Textual Similarity*, model *SBERT-base* mencapai korelasi Pearson sebesar 85,29% terhadap penilaian manusia, jauh melampaui performa BERT-base standar yang hanya mencapai 54,81%.

Dalam Bahasa Indonesia sendiri Cahyawijaya *et al.,* (2021) memperkenalkan *benchmark IndoNLU* yang mencakup beragam tugas *Natural Language Understanding*, termasuk pengukuran kemiripan teks. Proses *preprocessing* dilakukan secara minimal karena mekanisme tokenization internal dan mampu menangani variasi linguistik secara otomatis. Melalui pelatihan dan *fine-tuning* model IndoBERT pada korpus Bahasa Indonesia mereka memperoleh korelasi *Spearman* sebesar 0,814 pada tugas *Indo-STS*.

Penelitian oleh Maulida, (2024) mengembangkan sistem pencarian judul dan pendeteksi kemiripan abstrak tugas akhir menggunakan metode *TF-IDF* dan *Cosine Similarity*. Data penelitian berupa kumpulan judul dan abstrak mahasiswa program studi Teknologi Informasi yang diuji tingkat kesamaannya. Hasil pengujian menunjukkan bahwa sistem mampu mengidentifikasi kesamaan antar judul dengan akurasi 91%, presisi 83%, dan recall 62,5%.

Studi oleh Sunandar & Muiz, (2025) membangun sistem deteksi kemiripan judul skripsi berbasis *Levenshtein Distance* dan *Cosine Similarity*. Model ini membandingkan teks secara karakter dan semantik untuk menentukan tingkat kedekatan antar judul. Evaluasi terhadap sistem dilakukan dengan menghitung *confusion matrix*, di mana hasil tertinggi dicapai oleh kombinasi kedua metode dengan akurasi 89%.

Penelitian oleh Chamidah *et al.,* (2022) menilai jawaban esai pendek otomatis berbasis *semantic similarity* menggunakan *SBERT*. Model yang digunakan mentransformasi teks jawaban esai mahasiswa dan kunci jawaban menjadi vektor berdimensi 512 yang mewakili makna semantik dari teks. Objek

penelitian berupa 144 jawaban esai dari 36 mahasiswa. Hasil penelitian menunjukkan bahwa pendekatan ini memberikan rata-rata *0.78 Pearson Correlation* dan *MAE* sebesar *0.26*.

Penelitian oleh Monir *et al.*, (2024) mengusulkan sistem *VectorSearch* berbasis *semantic embedding* untuk pencarian dokumen menggunakan model *MiniLM, BERT, dan RoBERTa*. Penentuan *threshold similarity* dilakukan melalui *Grid Search* pada beberapa nilai threshold. Hasil penelitian menunjukkan bahwa threshold optimal berbeda pada setiap model, dengan model terkait *MiniLM-L6-V2* yang menghasilkan F1-Score 0.95.

Penelitian oleh Holis *et al.*, (2025) mengembangkan sistem *chatbot FAQ* akademik menggunakan *Sentence-BERT* dan *Cosine Similarity* untuk mengukur kemiripan semantik antara pertanyaan pengguna dan basis pengetahuan. Penelitian ini menggunakan data pertanyaan yang dipasangkan dengan jawaban *ground truth* dan melakukan pengujian beberapa nilai *threshold similarity*. Hasil evaluasi menunjukkan bahwa threshold 0,5 memberikan F1-score sebesar 88,4%.

Tabel 2. 1 Penelitian Terdahulu

No.	Author	Preprocessing					Feature Extraction Model	Evaluation Metrix	Result
		Case Folding	Cleansing	Stopword	Tokenizing	Stemming			
1.	(Devlin et al., 2019)	✓	✓	-	-	-	BERT	Confusion Matrix	Acc 86,6% F1-Score 90,1%
2.	(Reimers & Gurevych	✓	✓	-	-	-	SBERT	Pearson Correlation	Pearson 85,29%

3.	(Cahyawijaya et al.,	✓	✓	-	-	-	IndoBERT	Spearman Correlation	Spearman 0,814
4.	(Maulida, 2024)	✓	✓	✓	✓	✓	TF-IDF	Confusion Matrix	Acc 91% Prec 83% Rec 62,5%
5.	(Sunandar & Muiz,	✓	✓	✓	✓	✓	TF-IDF	Confusion Matrix	Acc 89%
6.	(Chamidah et al., 2022)	✓	-	-	-	-	SBERT	Pearson Corr & MAE	Pearson 0,78 MAE 0,26
7.	(Monir et al., 2024)	✓	✓	-	-	-	MiniLM-L6-V2	Confusion Matrix	Perc 0.97 Rec 0,93 F1-Score 0,95
8.	(Holis et al., 2025)	✓	✓	-	-	-	SBERT	Confusion Matrix	Acc 0,792 Prec 0,817 Rec 0,933 F1-Score 0,884

Penelitian awal seperti BERT (Devlin *et al.,* 2019), *SBERT* (Reimers & Gurevych, 2019), dan IndoBERT (Koto *et al.,* 2020) berfokus pada pemahaman semantik. Keunggulan utama metode ini adalah mereka tidak memerlukan proses *preprocessing* manual yang rumit seperti *stemming* atau *cleansing*, karena sudah memiliki tokenisasi internal. Hasilnya terbukti efektif dengan metrik seperti akurasi 86,6% dan F1-score 90,1% untuk BERT, serta korelasi 85,29% untuk *SBERT* dan 0,814 untuk IndoBERT.

Untuk metode yang lebih tradisional juga menunjukkan hasil yang cukup baik dengan Penelitian oleh Maulida (2024) yang memakai TF-IDF, serta Sunandar *et al.* (2025), semuanya sangat bergantung pada tahapan *preprocessing* seperti *case folding*, *stopword removal*, dan *stemming*. Meski butuh proses lebih, metode ini tetap menghasilkan akurasi yang tinggi antara 91% dan 87%.

Untuk mendapatkan nilai evaluasi dan akurasi yang valid, sebuah model sangat bergantung pada ketersediaan dataset *pair* yang dilengkapi dengan *ground truth* sebagai standar kebenarannya (Devlin *et al.*, 2019; Reimers & Gurevych, 2019). Tanpa adanya pasangan teks beserta *ground truth* seperti label kesamaan makna (Devlin *et al.*, 2019) atau skor penilaian manusia (Reimers & Gurevych, 2019) sistem tidak memiliki parameter pasti untuk memvalidasi keakuratan hasil komputasinya. Berdasarkan literatur sebelumnya pada tabel 2.2 terdapat penelitian yang secara spesifik menggunakan dataset berpasangan.

Tabel 2. 2 Penelitian Dengan *Pair* Teks

Penelitian	**Data *Pair***
(Devlin *et al.*, 2019)	*Microsoft Research Paraphrase Corpus*
(Reimers & Gurevych, 2019)	*benchmark Semantic Textual Similarity*
(Cahyawijaya *et al.*, 2021)	*benchmark IndoNLU*
(Chamidah *et al.*, 2022)	Jawaban VS Kunci Jawaban

Berdasarkan berbagai literatur yang telah diuraikan, dapat disimpulkan bahwa penelitian terkait deteksi kemiripan teks telah berkembang dari metode pencocokan leksikal tradisional, seperti *TF-IDF* (Maulida, 2024) dan *Levenshtein Distance* (Sunandar & Muiz, 2025), menuju pendekatan berbasis pemahaman semantik menggunakan arsitektur *deep learning* seperti *BERT, SBERT, dan IndoBERT* (Cahyawijaya *et al.*, 2021; Devlin *et al.*, 2019; Reimers & Gurevych,

2019). Meskipun metode leksikal terbukti memberikan hasil yang baik melalui tahapan prapemrosesan yang ketat, model semantik menawarkan keunggulan komputasi dalam menangkap konteks makna secara otomatis. Lebih lanjut, seluruh studi ini menegaskan bahwa untuk menghasilkan evaluasi akurasi yang terukur dan valid, pengembangan sistem deteksi kemiripan sangat bergantung pada penggunaan *pair text dataset* yang dilengkapi dengan *ground truth* sebagai standar kebenaran absolutnya (Chamidah *et al.*, 2022).

2.2 Landasan Teori

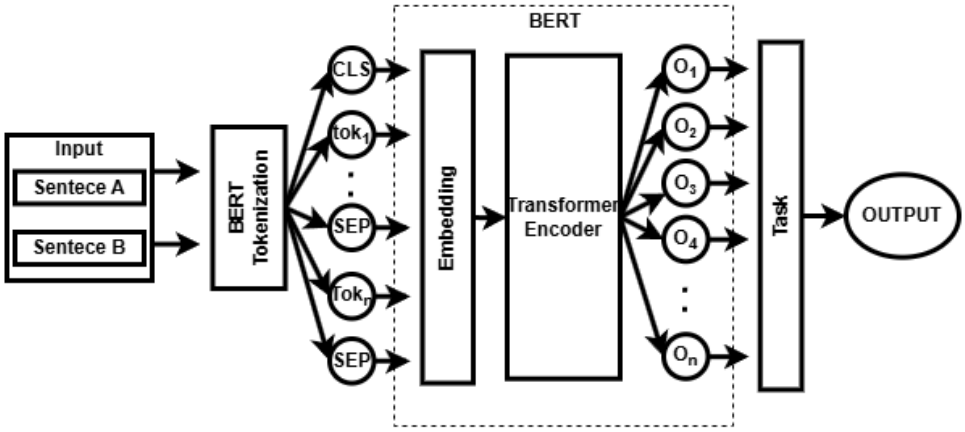
2.2.1 Semantic Similarity

Semantic Similarity atau kesamaan semantik adalah konsep dalam ilmu linguistik pemrosesan bahasa alami yang digunakan untuk menggambarkan seberapa mirip makna dari dua unit teks (Boudaa *et al.,* 2024; Reilly *et al.,* 2024). Unit teks ini dapat berupa kata, frasa, kalimat maupun dokumen. Ide utamanya adalah bahwa dua teks dapat dianggap mirip jika memiliki makna yang berdekatan (Ranasinghe *et al.,* 2025), meskipun bentuk kata atau struktur kalimat yang digunakan berbeda (Sunilkumar & Shaji, 2019).

Untuk mengukur *Semantic Similarity* sebenarnya banyak pendekatan, dari pendekatan manual yaitu menggunakan sumber daya daya linguistik seperti ontologi atau kamus digital untuk melihat hubungan antar kata (Pilehvar & Camacho-Collados, 2020). Sampai tingkat pemogramman dengan pendekatan berbasis representasi vektor yang memanfaatkan teknik *embedding* untuk memproyeksikan teks ke dalam ruang vektor sehingga makna dapat dihitung dalam bentuk matematis (Reimers & Gurevych, 2019).

2.2.2 Sentence-BERT

Sentence-BERT merupakan sebuah variasi arsitektur dari model BERT atau *Bidirectional Encoder Representations from Transformers* yang dimodifikasi menggunakan struktur *Siamese Network* untuk menghasilkan representasi vektor kalimat *sentence embedding* berdimensi tetap secara efisien. SBERT dikembangkan untuk mengatasi keterbatasan komputasi pada model BERT standar ketika dihadapkan pada tugas-tugas penataan atau pencarian tekstual berskala besar (Reimers & Gurevych, 2019). Kelemahan mendasar dari arsitektur BERT standar dalam melakukan tugas perbandingan kalimat diilustrasikan pada Gambar 2.1.

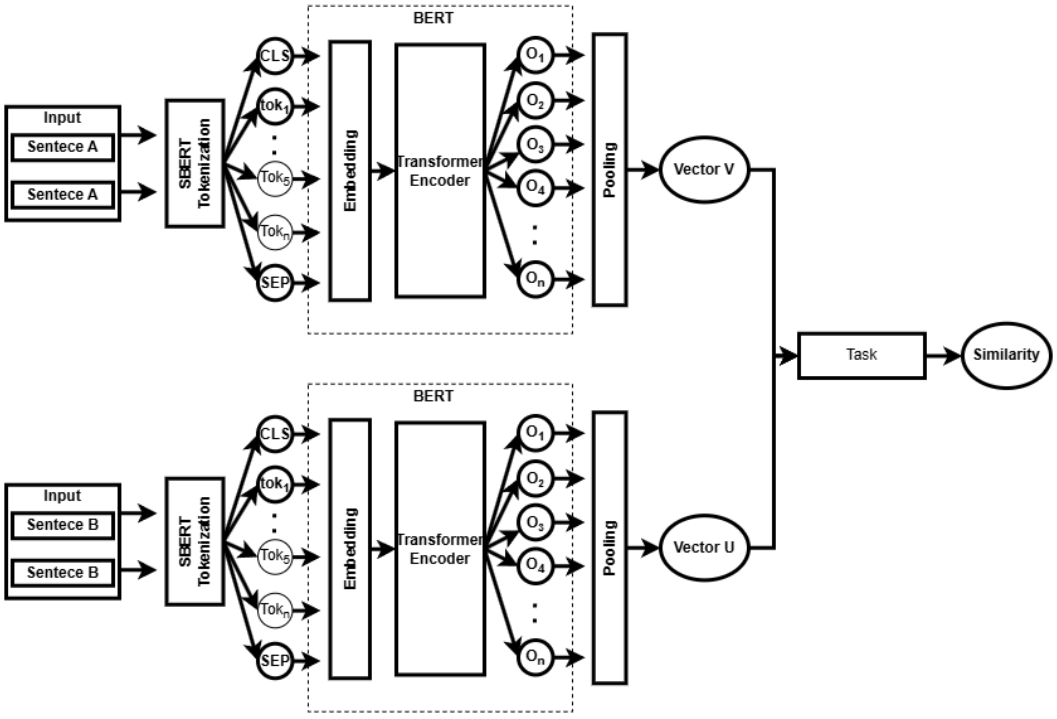


Gambar 2. 1 Arsitektur BERT

Berdasarkan Gambar 2.1, BERT standar bertindak sebagai *Cross-Encoder* yang mengharuskan Kalimat A dan Kalimat B dimasukkan secara bersamaan ke dalam model sebagai satu input tunggal yang digabungkan oleh token khusus [SEP] (Devlin *et al.*, 2019). Konsekuensinya, proses *Self-Attention* harus menghitung hubungan antar-kata dari kedua kalimat tersebut secara masif dari awal. Proses ini harus diulang untuk setiap kombinasi pasangan kalimat, sehingga tidak efisien dan

membutuhkan waktu komputasi yang sangat lama jika diterapkan pada database dengan ribuan judul.

SBERT mengatasi masalah skalabilitas tersebut dengan mengadopsi struktur *Bi-Encoder* di mana Kalimat A dan Kalimat B diproses secara independen melalui dua jalur *encoder* yang sejajar. Melalui pendekatan ini, SBERT mampu mengekstrak fitur dan makna dari masing-masing kalimat menjadi sebuah vektor representasi mandiri yang kaya akan informasi konseptual (Reimers & Gurevych, 2019). Vektor-vektor individual inilah yang nantinya dapat disimpan dan dibandingkan secara instan menggunakan metrik jarak sederhana seperti *Cosine Similarity*, menjadikannya sangat unggul untuk tugas *Semantic Textual Similarity*. Secara lebih rinci, alur kerja pemrosesan data teks dalam arsitektur SBERT ditunjukkan pada Gambar 2.2.



Gambar 2. 2 Arsitektur SBERT

Berdasarkan Gambar 2.2, proses SBERT diawali dengan memisahkan Kalimat A dan Kalimat B ke dalam jalur tokenisasi masing-masing secara individual. Pada tahap *Tokenizing*, teks mentah dikonversi menjadi urutan token sub-kata tunggal, dan secara otomatis disisipkan token khusus *Transformer*, yaitu [CLS] di awal sebagai jangkar urutan, serta token [SEP] di akhir sebagai penanda batas akhir teks tunggal. Setelah melalui proses penyesuaian dimensi deretan token ini dikirim ke dalam *Core BERT Encoder*.

Di dalam *BERT Encoder*, setiap token diproses melalui tumpukan lapisan *Transformer*. Di sinilah letak ekstraksi fitur semantik pertama kali terjadi, di mana *Multi-Head Self-Attention* menganalisis hubungan logis antar-kata di dalam kalimat tersebut untuk membangun representasi makna yang utuh. *Output* mentah dari *BERT Encoder* masih berupa kumpulan matriks vektor per-token.

Puncak pembentukan informasi semantik kalimat berada pada bagian *Pooling Layer*, lapisan ini bertugas melebur dan merata-ratakan seluruh vektor kata yang dihasilkan oleh *BERT Encoder* menjadi satu vektor tunggal berdimensi tetap, yang dikenal sebagai *Sentence Embedding*. Vektor tunggal ini merupakan bentuk representasi ruang semantik yang merangkum keseluruhan isi dan makna konseptual dari kalimat terkait (Devlin *et al.*, 2019; Reimers & Gurevych, 2019).

2.2.3 Cosine Similarity

Cosine similarity merupakan salah satu metode yang digunakan untuk mengukur tingkat kemiripan antara dua vektor dalam ruang multidimensi. Inti dari

metode ini adalah melihat sudut antara dua vektor. Semakin kecil sudut antara dua vektor maka semakin besar pula tingkat kesamaannya (Chamidah *et al.,* 2022).

Dalam *text processing Cosine similarity* digunakan setelah teks diubah menjadi representasi vektor, baik melalui metode klasik seperti *TF-IDF* maupun metode *embedding* seperti *BERT* atau *SBERT*. Dua teks yang memiliki kemiripan makna akan menghasilkan vektor dengan arah yang relatif sama, sehingga nilai *Cosine similarity* di antara keduanya juga akan tinggi. Sebaliknya jika makna antar teks berbeda jauh, maka arah vektor juga akan berbeda dan nilai *Cosine similarity* yang diperoleh cenderung rendah (Hassan & Ahmed, 2023).

2.2.4 Threshold Similarity

Threshold merupakan nilai batas minimal atau maksimal yang harus dicapai untuk memicu suatu kondisi, keputusan, atau klasifikasi. Dalam konteks semantic similarity, penentuan nilai threshold sebaiknya tidak dipandang sekadar sebagai pencarian satu angka final yang mutlak, melainkan sebagai sebuah *trade-off* antara precision dan recall. Menetapkan threshold yang rendah akan meningkatkan sensitivitas sistem dalam menangkap kemiripan atau recall tinggi, namun berisiko memunculkan lebih banyak false positives. Sebaliknya, threshold yang tinggi akan menghasilkan aturan keputusan yang lebih ketat atau precision tinggi, namun berpotensi memunculkan false negatives.

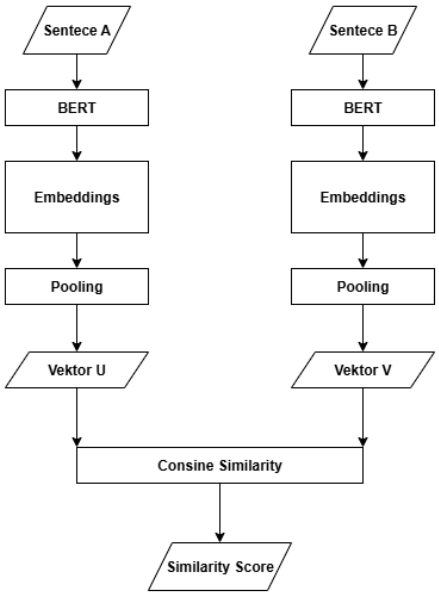
Sifat trade-off ini sejalan dengan pernyataan Reimers & Gurevych (2019), yang menjelaskan bahwa representasi vektor yang dihasilkan oleh model sentence embedding sangat dipengaruhi oleh karakteristik data dan arsitektur model yang digunakan. Oleh karena itu, nilai similarity yang optimal tidak bersifat universal,

melainkan harus dievaluasi secara objektif berdasarkan kebutuhan trade-off dari dataset serta tujuan spesifik sistem.

Sebagai contoh pendekatan empiris dalam mencari titik keseimbangan ini, penelitian Ghinassi *et al.,* (2023) menggunakan metode Grid Search pada proses optimasi model BiLSTM untuk mengeksplorasi dan menentukan batas keputusan kemiripan yang paling sesuai dengan data mereka. Di sisi lain, pada pendekatan semantic similarity berbasis transformer, penelitian terbaru oleh Holis *et al.,* (2025) dan Monir *et al.,* (2024) melakukan evaluasi komparatif terhadap lima nilai *threshold*. Hasil pengujian mereka menunjukkan bahwa model seperti Sentence-BERT maupun MiniLM menghasilkan keseimbangan performa yang paling baik pada threshold sekitar 0.5, menjadikannya titik trade-off yang paling ideal untuk karakteristik dataset yang mereka gunakan.

2.2.5 Perhitungan Semantic Similarity

Penggunaan *SBERT* dalam menghasilkan representasi kalimat pada beberapa penelitian dipadukan dengan *Cosine similarity* untuk menghitung tingkat kesamaan antar kalimat (Chamidah *et al.,* 2022). SBERT bukan sekadar menghasilkan representasi kalimat melainkan menyediakan dasar untuk memproyeksikan teks pendek ke dalam ruang vektor multidimensi, di mana kalimat yang bermakna serupa akan dipetakan pada posisi yang saling berdekatan, sementara kalimat yang berbeda makna akan berada jauh (Reimers & Gurevych, 2019). Pengukuran kesamaan semantik pada level makro ini kemudian diselesaikan dengan menghitung kedekatan atau jarak antar vektor kalimat tersebut menggunakan *Cosine Similarity* (Putra *et al.,* 2025).



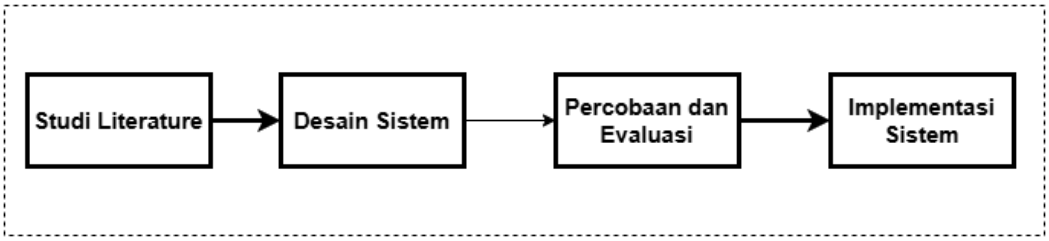
Gambar 2. 3 SBERT + Cosine Similarity

Pada Gambar 2.3 alur integrasi ini berjalan secara berurutan dimulai dari tahap *feature extraction* di mana teks *query* inputan diproses oleh BERT untuk menghasilkan *sentence embedding* berupa *dense vector*. Selanjutnya, vektor yang telah terbentuk dari masing-masing kalimat dibandingkan menggunakan rumus *Cosine Similarity* untuk mengukur sudut dan kedekatan antar vektor tersebut. Melalui pendekatan ini, sistem dapat bekerja penuh pada level semantik dan tidak lagi terbatas pada kecocokan kata secara leksikal.

BAB III
METODOLOGI PENELITIAN

3.1 Desain Penelitian

Tujuan utama dari penelitian ini adalah untuk merancang, mengimplementasikan, dan mengevaluasi sebuah sistem yang mampu mengukur tingkat kesamaan semantik antar judul skripsi. Penelitian ini memanfaatkan model *Sentence-BERT* untuk mengekstraksi makna dari teks dan *Cosine Similarity* untuk mengukur kedekatan makna tersebut. Desain penelitian disusun secara sistematis dan sekuensial untuk memastikan bahwa setiap tahapan, mulai dari studi literatur hingga implementasi sistem, dilaksanakan secara valid dan reliabel. Alur penelitian secara lengkap diilustrasikan pada Gambar 3.1.



Gambar 3. 1 Desain Penelitian

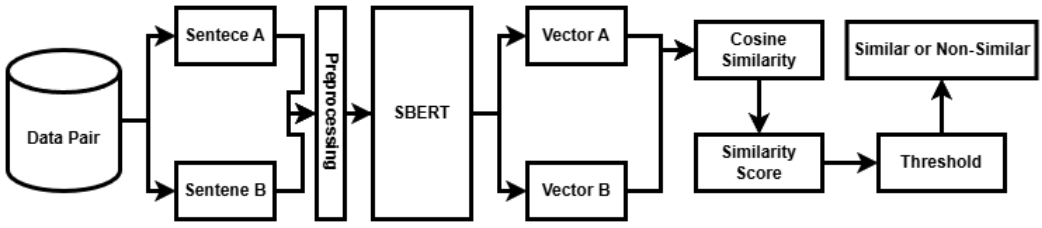
Penelitian ini diawali dengan studi literatur mendalam mengenai arsitektur SBERT (Reimers & Gurevych, 2019), *Cosine Similarity* (Hassan & Ahmed, 2023), dan metodologi evaluasi, yang dilanjutkan ke tahap pengumpulan dan pemrosesan data terhadap 1.887 judul skripsi. Pada tahap awal ini, data pasangan judul dibentuk menggunakan *pairwise combination* dengan memanfaatkan sifat *upper triangular*

matrix dan diseimbangkan melalui kombinasi TF-IDF serta stratified under-sampling guna menghasilkan pelabelan ground truth yang valid dan efisien.

Pada alur sistem SBERT mengekstraksi fitur judul skripsi menjadi *dense vector* dan mengukur kedekatan semantiknya via *Cosine Similarity*. Melalui percobaan dan evaluasi yang ketat menggunakan *K-Fold cross-validation* dan *Grid Search*, model akan diuji secara iteratif untuk menemukan nilai peforma pipeline dan mencari *threshold* paling optimal berdasarkan Akurasi, Presisi, Recall, dan F1-Score, yang hasil akhirnya bisa langsung diintegrasikan ke dalam desain dan implementasi sistem cek *similarity*.

3.2 Desain Sistem

Kualitas data adalah fondasi utama dalam penelitian berbasis text yang menentukan validitas model (Zha *et al.,* 2023). Tahapan ini berfokus pada bagaimana data mentah diperoleh, diproses, divalidasi keabsahannya, hingga dibagi menjadi himpunan data *ground truth* untuk kebutuhan penilaian akurasi *feature extraction* SBERT terhadap data.



Gambar 3. 2 Desain Sistem

Pada Gambar 3.2 sistem *similarity* teks ini dirancang dengan alur sekuensial yang bermula dari input berupa *Data Pair*, di mana himpunan pasangan kalimat ini sebelumnya dibentuk dari 1.887 judul skripsi dan telah melewati tahap penyaringan

strata kemiripan berbasis TF-IDF serta validasi *ground truth* oleh pakar linguistik dan teknologi informasi. *Data Pair* tersebut dipisah menjadi dua aliran, yakni Sentence A dan Sentence B, untuk diproses secara paralel ke dalam model SBERT. Di dalam arsitektur SBERT, setiap kalimat akan mengalami proses *preprocessing* seperti *casefolding* dan pemecahan token, lalu dikonversi menjadi matriks vektor kontekstual yang pada akhirnya dirangkum menggunakan teknik Mean Pooling untuk menghasilkan representasi semantik kalimat secara utuh dalam bentuk Vector A dan Vector B (Reimers & Gurevych, 2019).

Kedua vektor tersebut selanjutnya dievaluasi menggunakan *Cosine Similarity* yang membandingkan sudut antarvektor untuk menghasilkan sebuah *Similarity Score*. Pada tahapan akhir, skor kemiripan tersebut dikomparasikan terhadap sebuah *Threshold* yang titik potong optimalnya telah dikalibrasi menggunakan metode *Grid Search* melalui mekanisme *K-Fold Cross Validation* sehingga sistem dapat memetakan skor tersebut menjadi sebuah keputusan klasifikasi biner final, yaitu *Similar or Non-Similar*.

3.2.1 Data Pair

Kualitas data adalah fondasi utama dalam penelitian berbasis text yang menentukan validitas model (Zha *et al.,* 2023). Tahapan ini berfokus pada bagaimana data mentah diperoleh, diproses, divalidasi keabsahannya, hingga dibagi menjadi himpunan data *ground truth* untuk kebutuhan penilaian akurasi *feature extraction* SBERT terhadap data.

A. Data Awal

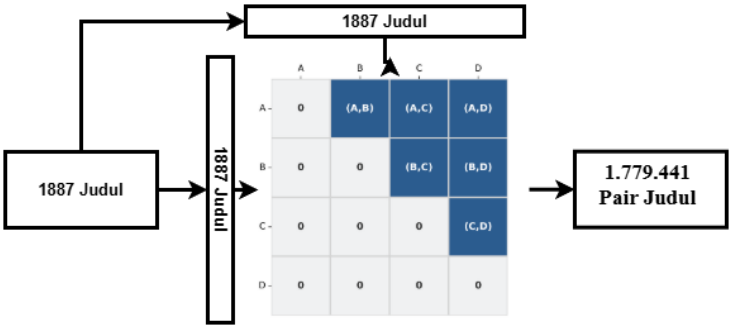
Dataset yang digunakan adalah kumpulan judul skripsi mahasiswa Program Studi Teknik Informatika yang diperoleh dari repositori digital Perpustakaan atau *e-thesis* UIN Maulana Malik Ibrahim Malang. Total data yang berhasil dikumpulkan adalah 1887 judul skripsi yang terbit khusus pada rentang tahun 2008-2026. Karakteristik dataset ini merupakan teks berbahasa Indonesia atau campuran yang memuat istilah-istilah teknis spesifik di bidang ilmu komputer dan informatika. Contoh judul terdahulu dari dataset ini dapat dilihat pada Tabel 3.1.

Tabel 3. 1 Judul Skripsi Pada Dataset

Judul
Klasifikasi kebakaran hutan berbasis citra menggunakan *Convolutional Neural Network*
Klasifikasi *Acute Lymphoblastic Leukemia (ALL)* menggunakan *Multilayer Perceptron (MLP)*
Sistem kontrol speaker murottal Al-Qur'an berbasis *Internet of Things* menggunakan algoritma *Aho-Corasick*

B. Proses Pembentukan Pasangan Kalimat

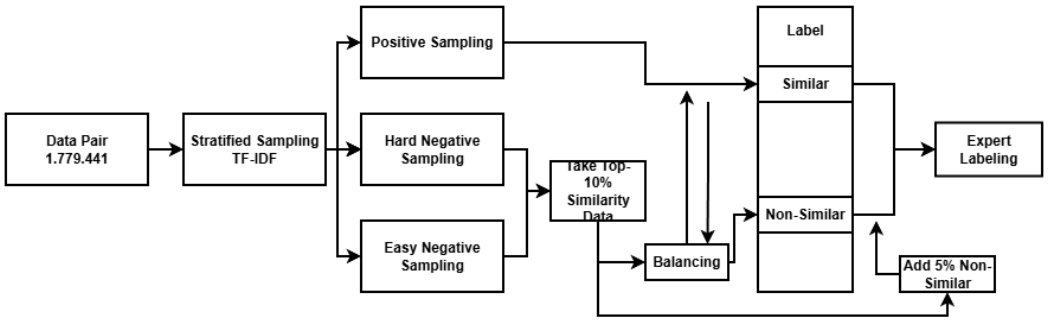
Tahap awal pada Gambar 3.3 dimulai dengan mengeksekusi *pairwise combination* dengan memanfaatkan sifat *upper triangular matrix* terhadap 1.887 judul skripsi bersih guna membangun pasangan data yang siap diperbandingkan. Untuk menghindari penghitungan dua kali pada pasangan node atau artikel sitasi, ilmuwan hanya mengambil bagian segitiga atasnya saja (lijun Yang *et al.*, 2019).



Gambar 3. 3 Pairwise Combination + *upper triangular*

Pada Rumus 3.1 jika kita memiliki total populasi judul unik yang disimbolkan sebagai N, maka total pasangan unik yang dapat terbentuk yang disimbolkan sebagai P ditentukan dengan cara mengalikan nilai N dengan hasil pengurangan N oleh angka satu $N - 1$, yang kemudian seluruh hasil perkalian tersebut dibagi dengan angka dua. Sesuai dengan visualisasi pada Gambar 3.2, perhitungan kombinasi menghasilkan populasi sebanyak 1.779.441 pasang judul.

$$P = \frac{N(N-1)}{2} \tag{3.1}$$



Gambar 3. 4 Alur Sebelum Labeling Expert

Untuk mengatasi fenomena *Severe Class Imbalance* dimana kemungkinan 98% data menumpuk di skor rendah yang berpotensi memicu *majority class bias*

atau kecenderungan model menebak salah satu kelas (He & Garcia, 2009), seluruh populasi pasangan ini disaring menggunakan skor *Cosine Similarity* berbasis *TF-IDF* ke dalam tiga strata (Zhang *et al.*, 2019).

Strata Tinggi dengan $0{,}70 \leq \text{skor} \leq 1{,}00$ yang merupakan kandidat kuat kelas Mirip atau disebut kelas positif, Strata Sedang atau *hard negative* dengan $0{,}40 \leq \text{skor} < 0{,}70$ dan Strata Rendah atau *easy negative* dengan $\text{skor} < 0{,}40$ sebagai representasi kelas Tidak Mirip (Karpukhin *et al.*, 2020; MetricGate, 2026). Dari tiga strata hasil *Stratified Under-sampling* seluruh pasang judul yang lolos penyaringan lalu divalidasi oleh *domain expert*. Untuk rentang skor tiap strata dapat dilihat pada Tabel 3.2.

Tabel 3. 2 Rentang Tiap Strata

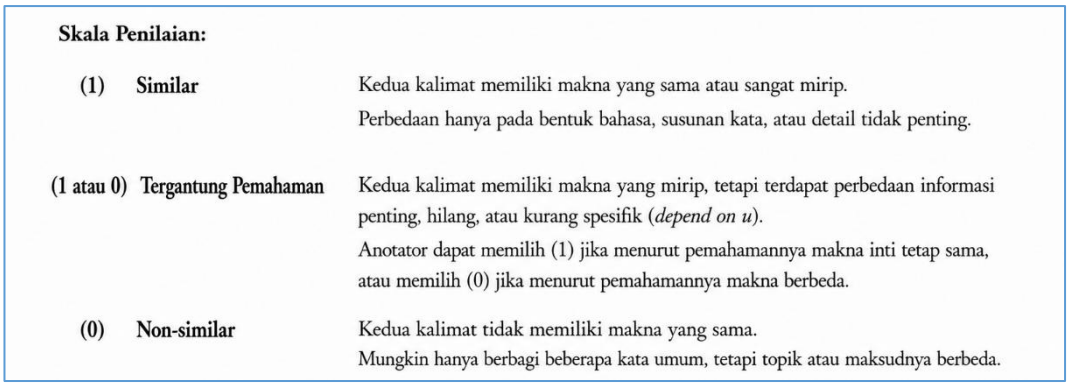
Strata	**Rentang Skor**
Tinggi / Similar / Positif	$0{,}70 \leq \text{skor} \leq 1{,}00$
Sedang / Hard Non-Similar / Hard Negative	$0{,}40 \leq \text{skor} < 0{,}70$
Rendah / Easy Non-Similar / Easy Negative	$\text{skor} < 0{,}40$

Untuk memastikan kualitas data tetap terjaga dan menghindari *False Negatives*, diterapkan *cutoff threshold* sebesar 10% dari total populasi strata. Pembatasan proporsi ini mengadaptasi teknik *Positive-Aware Mining* yang diusulkan oleh Moreira *et al.*, (2025), di mana pembatasan ruang sampel pada persentase teratas terbukti mampu meningkatkan akurasi *embedding quality* dibandingkan penggunaan sampel negatif yang tidak terkurasi.

Setelah menerapkan rasio *balance* 1:1, dialokasikan tambahan 5% data *negative* ekstra dengan menerapkan teknik *sample size padding,* untuk mengantisipasi adanya *mislabeling* atau penyusutan data selama proses validasi oleh *expert* tanpa merusak data yang *balance* (Israel, 1992).

C. Validasi Ground Truth oleh Expert

Proses validasi ground truth dilakukan melalui tinjauan *expert* oleh praktisi yang memiliki kualifikasi akademik di bidang Teknologi Informasi dan Linguistik Bahasa Inggris. Validasi dilakukan secara sistematis dengan mengacu pada rubrik penilaian pada gambar 3.5 yang mencakup ketepatan identifikasi metode atau teknik pada informatika dan bentuk bahasa implisit.



Skala Penilaian:		
(1)	**Similar**	Kedua kalimat memiliki makna yang sama atau sangat mirip. Perbedaan hanya pada bentuk bahasa, susunan kata, atau detail tidak penting.
(1 atau 0)	**Tergantung Pemahaman**	Kedua kalimat memiliki makna yang mirip, tetapi terdapat perbedaan informasi penting, hilang, atau kurang spesifik (*depend on u*). Anotator dapat memilih (1) jika menurut pemahamannya makna inti tetap sama, atau memilih (0) jika menurut pemahamannya makna berbeda.
(0)	**Non-similar**	Kedua kalimat tidak memiliki makna yang sama. Mungkin hanya berbagi beberapa kata umum, tetapi topik atau maksudnya berbeda.

Gambar 3. 5 Rubik Labeling

Proses ini melibatkan dua pakar yang melakukan verifikasi manual terhadap seluruh label untuk memastikan kesesuaian antara label sistem dengan nuansa kontekstual teks Bahasa campuran. Kolaborasi lintas disiplin ini dipilih untuk menjamin bahwa dataset yang dihasilkan memiliki tingkat presisi tinggi dan memenuhi standar kualitas yang ketat, sehingga dapat diandalkan sebagai *ground truth* yang valid dalam tahap pemodelan NLP selanjutnya.

3.2.2 Preprocessing

A. Case Folding

Proses case folding digunakan agar rumus tersebut mencakup fungsi mengubah huruf kapital sekaligus menghapus angka atau tanda baca menggunakan fungsi transformasi karakter. Pada Rumus 3.2, jika kita memiliki teks asli T, proses case folding dibentuk dengan cara mengambil setiap token atau karakter (w_i) yang merupakan elemen dari ($\in$) T.Namun, syaratnya (|) token w_i tersebut harus ditransformasikan menjadi elemen dari $\in$ alfabet huruf kecil ($[a-z]$). Melalui fungsi transformasi ini, setiap karakter huruf kapital ($[A-Z]$) akan dikonversi menjadi huruf kecil melalui penambahan nilai ASCII sebesar 32, sedangkan pada rumus 3.3 karakter yang bukan bagian dari alfabet seperti angka, tanda baca, dan simbol lainnya akan diubah menjadi himpunan kosong ($\emptyset$) atau dihapus secara otomatis dari teks hasil akhir (T').

$$T' = \{f(w_i) \mid w_i \in T\} \tag{3.2}$$

$$f(w_i) = \begin{cases} w_i + 32, & \text{jika } w_i \in [A-Z] \\ w_i, & \text{jika } w_i \in [a-z] \\ \emptyset, & \text{jika } w_i \notin [a-zA-Z] \end{cases} \tag{3.3}$$

3.2.3 SBERT

SBERT merupakan model yang digunakan untuk menghasilkan representasi vektor kalimat atau disebut *sentence embedding* sehingga dapat mengukur kemiripan semantik antarjudul skripsi. Pada penelitian ini, SBERT digunakan sebagai *pretrained sentence encoder* tanpa dilakukan *fine-tuning*, sehingga model hanya berperan sebagai *feature extractor* untuk menghasilkan

representasi vektor dari setiap judul. Proses pembentukan representasi tersebut meliputi tiga tahapan, yaitu tokenisasi, encoding, dan pooling.

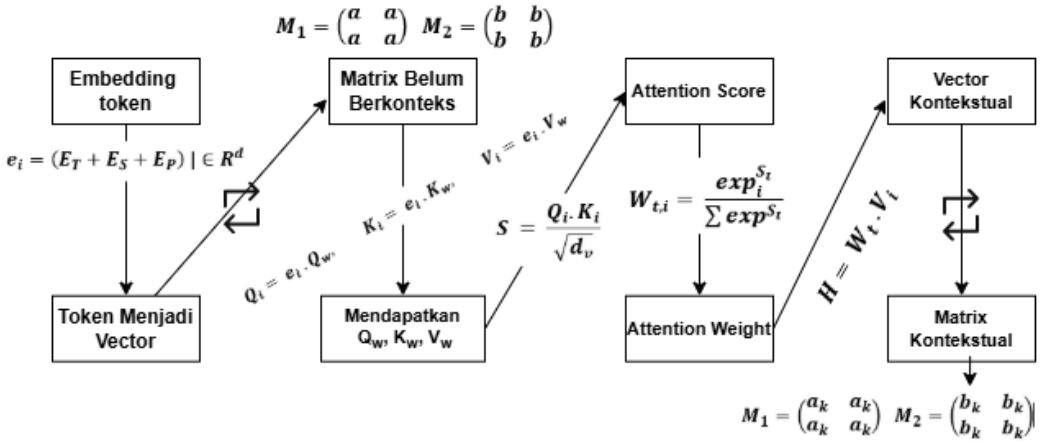
A. Tokenisasi

Langkah tokenisasi yaitu pemecahan kalimat menjadi unit-unit yang lebih kecil. Dijelaskan dalam rumus 3.4, proses memecah kalimat *input* S menjadi sebuah urutan token $(t_1, t_2, \dots, t_m)$, di mana m Adalah total token. Setiap token harus dikenali dan ada dalam kosakata $(\in) V$.

$$Tokenize(S) = (t_1, t_2, \dots, t_m), \quad t_i \in V \tag{3.4}$$

B. Encoding

Encoding mengkonversi token menjadi vektor dengan rumus 3.5, di mana fungsi lapisan *embedding* E bekerja memetakan setiap token t_i menjadi vektor e_i yang memiliki dimensi *d*. Vektor e_i kemudian diproses menggunakan rumus 3.5 dengan $Transformer$ bekerja mengambil e_i dan C yaitu konteks dari semua kalimat, untuk menghasilkan h_i atau vector kontekstual.



Gambar 3. 6 Encoding SBERT Bekerja

Proses encoding berfungsi untuk mengkonversi token menjadi vektor representasi. Pada tahapan ini, fungsi lapisan embedding E bekerja untuk memetakan setiap token t_i menjadi vektor e_i yang memiliki dimensi d.

$$e_i = E(t_i), \quad e_i \in R^d \tag{3.5}$$

Pada rumus 3.6 vektor $e_i$yang dihasilkan kemudian diproses lebih lanjut menggunakan arsitektur Transformer. Lapisan Transformer ini bekerja dengan mengambil vektor token e_i beserta C, yaitu representasi konteks dari keseluruhan kalimat, untuk menghasilkan h_i yang merupakan vektor kontekstual.

$$h_i = Transformer(e_i, C) \tag{3.6}$$

Hasil encoding mengubah token menjadi vektor dengan ukuran dimensi yang sesuai dengan model SBERT. Vektor-vektor ini kemudian diproses melalui mekanisme *self-attention* untuk menghasilkan vektor berkonteks yang lebih kaya akan makna gramatikal dan semantik. Vektor-vektor berkonteks dari setiap token ini selanjutnya akan ditumpuk menjadi sebuah matriks berkonteks untuk satu kalimat utuh.

Transformasi matriks kalimat menjadi vektor berkonteks dilakukan melalui proses *self-attention*. Langkah pertama adalah menentukan *Query Weight* (W_Q), *Key Weight* (W_K), dan *Value Weight* (W_V) yang dimensinya mengikuti panjang vektor dari model. Pada rumus 3.7 Berdasarkan matriks input X, sistem akan menghitung *Vektor Query, Vector Key,* dan *Vector Value* untuk setiap token dengan rumus operasi perkalian matriks *dot product*.

$$Q = X \cdot W_Q, \qquad K = X \cdot W_K, \qquad V = X \cdot W_V \tag{3.7}$$

Selanjutnya, sistem menentukan *Attention Score Matrix* (S). Matriks skor ini didapatkan melalui rumus 3.8 dengan operasi dot product antara vektor Q dan transposisi dari vektor K, yang kemudian hasilnya dibagi dengan akar dimensi vektor ($\sqrt{d}$) agar nilai varians tetap stabil.

$$S = \frac{Q \cdot K^T}{\sqrt{d}} \tag{3.8}$$

Skor yang telah didapatkan kemudian diubah menjadi probabilitas *Weight Attention* (A) menggunakan fungsi *softmax* seperti rumus 3.9. Fungsi ini bekerja dengan cara mengeksponensialkan setiap nilai di dalam matriks skor, lalu membaginya dengan hasil normalisasi.

$$A = \text{softmax}(S) = \frac{\exp(S_i)}{\sum_j \exp(S_j)} \tag{3.9}$$

Langkah terakhir untuk menemukan representasi vektor kontekstual akhir bagi setiap token adalah rumus 3.10 dengan melakukan *dot product* antara matriks *Weight Attention* (A) yang telah dinormalisasi dengan matriks *Value* (V).

$$h_i = A \cdot V \tag{3.10}$$

C. Pooling

Langkah *pooling* yaitu penggabungan semua vektor token kontekstual menjadi satu vektor tunggal yang mewakili keseluruhan kalimat. Dalam penelitian ini digunakan *Mean Pooling* sesuai rumus 3.11. Rumus ini bekerja untuk menghasilkan v, atau *sentence embedding* tunggal. Cara kerjanya adalah dengan menjumlahkan ($\sum_{i=1}^{m}$) semua representasi kontekstual h_i, dari token pertama ($i =$

1) hingga token terakhir (m). Hasil penjumlahan total ini kemudian diambil rata-ratanya melalui operasi $\frac{1}{m}$.

$$v = \frac{1}{m}\sum_{i=1}^{m} h_i \tag{3.11}$$

3.2.4 Cosine Similarity

Setelah dua judul misalnya $v_1\ dan\ v_2$ diubah menjadi *sentence embedding* tunggal *Cosine Similarity* digunakan untuk mengukur tingkat kemiripan di antara keduanya menggunakan rumus 3.12. Rumus ini bekerja dalam dua bagian. Bagian pembilang operator · atau *dot* melakukan operasi *dot product* $v_1 \cdot v_2$, yang bekerja dengan mengalikan elemen-elemen yang bersesuaian dari kedua vektor lalu menjumlahkan hasilnya. Bagian penyebut operator $|\dots|$ mengukur panjang dari setiap vektor. Dengan membagi *dot product* dengan perkalian *magnitude*nya, rumus ini secara efektif mengukur sudut di antara dua vektor, sehingga menghasilkan skor antara 0 sampai 1 terlepas dari panjang kalimat aslinya.

$$\mathrm{Cos}(v_1, v_2) = \frac{v_1 \cdot v_2}{|v_1|\,|v_2|} \tag{3.12}$$

3.2.5 Threshold

Nilai *similarity* yang optimal tidak bersifat universal karena sangat dipengaruhi oleh karakteristik data (Reimers & Gurevych, 2019), sehingga penelitian ini melakukan optimasi batas keputusan menggunakan metode *Grid Search*. Di dalam proses *cross-validation*, nilai *threshold* ini berfungsi sebagai *cut-off point* untuk mengubah skor kontinu dari *Cosine Similarity* menjadi keputusan

klasifikasi biner yang tegas, yaitu mirip atau tidak mirip. Pengujian ini dieksekusi melalui dua skema sekaligus, yakni *Nested K-Fold CV* yang bertujuan mengevaluasi performa keseluruhan *pipeline* secara objektif bebas dari bias kebocoran data, serta *Flat K-Fold CV* yang difokuskan untuk menetapkan satu nilai *threshold* final untuk aplikasi.

Dalam *Grid Search* penelitian ini memilih mekanisme pencarian sistematis berbasis interval (Ghinassi *et al.*, 2023) dan pencarian sistematis berbasis titik (Holis *et al.*, 2025; Monir *et al.*, 2024). Pada setiap iterasi *CV*, sistem akan menggunakan titik-titik threshold tersebut secara bergantian untuk memprediksi label pasangan judul, lalu membandingkannya langsung dengan data *ground truth* menggunakan *Confusion Matrix* untuk memutuskan batas akhir yang paling optimal. Rincian perbandingan cara dan penentuan titik *threshold* tersebut dirangkum pada Tabel 3.3.

Tabel 3. 3 Grid Search Threshold Terdahulu

Sumber	Sistem	Metode	Threshold
(Ghinassi *et al.*, 2023)	BiLSTM	Interval 0.05 per threshold	0.05 – 0.95
(Holis *et al.*, 2025)	SBERT	Titik per threshold	0.5, 0.6, 0.7, 0.8, 0.9

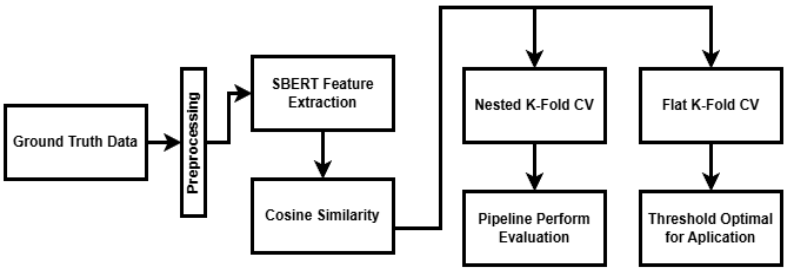
3.2.6 K-Fold Cross Validation

Untuk menjamin keandalan sistem dalam melakukan *similarity* semantik dan menghindari galat dalam estimasi akurasi, proses pengujian serta penalaan parameter tidak dilakukan pada *train-test split* tunggal. Sebagai gantinya, penelitian

ini menerapkan metode *K-Fold Cross-Validation* yang dibagi menjadi dua skema dengan peran yang sepenuhnya terpisah, *Nested K-Fold CV* untuk penilaian performa arsitektur dan *Flat K-Fold CV* untuk parameter / *threshold* pada aplikasi.

A. Alur Evaluasi

Alur evaluasi pada penelitian ini menggambarkan tahapan pemrosesan data secara sistematis yang dimulai dari data *ground truth*, melalui tahap *preprocessing*, ekstraksi fitur semantik, hingga proses evaluasi paralel. Sistem ini menggunakan dua pendekatan validasi, yaitu *Nested K-Fold CV* untuk evaluasi performa dan Flat *K-Fold CV* untuk penentuan *threshold* aplikasi. Rangkaian alur ini diilustrasikan pada Gambar 3.7.



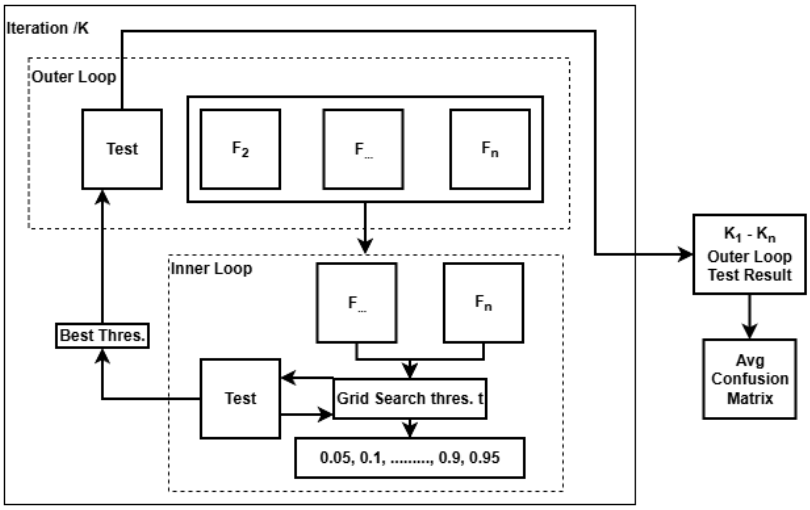
Gambar 3. 7 Alur Sistem

Berdasarkan Gambar 3.7, setelah skor kemiripan diperoleh, alur sistem bercabang menjadi dua jalur pemrosesan yang memiliki tujuan berbeda (Krstajic *et al.*, 2014). Jalur pertama berfokus pada evaluasi performa *pipeline* secara keseluruhan menggunakan metode *Nested K-Fold CV*. Pendekatan *nested cross-validation* ini secara ketat memisahkan proses optimasi parameter dari proses pengujian, sehingga menghasilkan *Pipeline Perform Evaluation* yang objektif dan bebas bias (Cawley & Talbot, 2010). Sementara itu, jalur kedua memproses

similarity score menggunakan metode *Flat K-Fold CV* yang difokuskan khusus untuk pencarian *parameter / threshold*. Tujuan utama dari jalur kedua ini adalah untuk menghasilkan *Threshold Optimal for Application*, yaitu nilai yang paling optimal untuk diimplementasikan ke sistem *similarity* (Holis *et al.*, 2025).

B. Nested K-Fold Cross Validation

Skema *Nested K-Fold CV* digunakan untuk melakukan evaluasi performa menyeluruh terhadap *pipeline* system (Cawley & Talbot, 2010; Krstajic *et al.*, 2014). Fungsi utama dari skema ini adalah menghasilkan metrik performa yang bersifat objektif, jujur, dan mencerminkan kemampuan generalisasi sistem yang sebenarnya ketika dihadapkan pada data judul baru di masa mendatang.



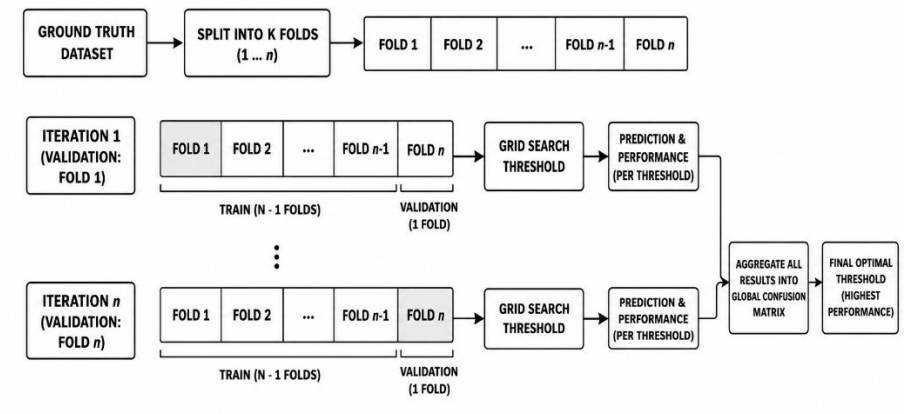
Gambar 3. 8 Nested K-Fold CV

Pada gambar 3.8 *nested* bekerja dengan memisahkan proses melalui dua tingkatan *loop*. Dataset *ground truth* terlebih dahulu dibagi menjadi *K fold* pada *outer loop*, di mana pada setiap iterasi satu fold digunakan sebagai test set murni,

sedangkan *K−1 fold* lainnya menjadi *training set*. Pada *inner loop*, *training set* tersebut dibagi kembali menjadi beberapa *fold* untuk menjalankan *Grid Search* terhadap berbagai nilai *similarity threshold*. *Threshold* yang memberikan performa terbaik pada *inner loop* kemudian dipilih dan digunakan untuk mengevaluasi test set murni pada *outer loop*, sehingga proses evaluasi menjadi lebih objektif dan terhindar dari *overfitting*. Dengan mekanisme ini data yang digunakan untuk mencari *threshold* tidak akan pernah bertemu dengan data yang digunakan untuk menguji performa akhir.

C. Flat K-Fold Cross Validation

Skema *Flat K-Fold CV* digunakan secara khusus untuk keperluan pemilihan *hyperparameter* (Cawley & Talbot, 2010; Krstajic *et al.*, 2014). Berbeda dengan *Nested CV* yang bertujuan untuk menguji keandalan teoretis sistem, *Flat CV* digunakan untuk implementasi aplikasi. Pada Gambar 3.9 skema ini bekerja dengan mengeksekusi satu lapisan siklus tunggal di seluruh dataset *ground truth*.



Gambar 3. 9 Flat K-Fold CV

Seluruh data *ground truth* langsung dibagi menjadi *K fold*, kemudian pada setiap iterasi sistem dilatih menggunakan *K−1 fold* dan divalidasi pada satu fold yang tersisa sambil menjalankan *Grid Search* untuk mengevaluasi beberapa nilai *similarity threshold*. Seluruh hasil prediksi dari setiap iterasi kemudian diakumulasikan ke dalam satu *Confusion Matrix global* untuk menghitung performa masing-masing *threshold*, sehingga diperoleh *threshold* optimal final dengan kinerja terbaik pada keseluruhan dataset. Nilai *threshold* tersebut selanjutnya digunakan sebagai parameter tetap dalam implementasi sistem.

3.3 Pengujian dan Evaluasi Sistem

3.3.1 Pengujian Sistem

Tahapan pengujian sistem pada penelitian ini dirancang untuk mengevaluasi keandalan pipeline menggunakan model SBERT *paraphrase-multilingual-mpnet-base-v2* dimensi 768. Perlu dicatat bahwa penelitian ini tidak melakukan proses *fine-tuning* terhadap model SBERT. Model tersebut secara eksklusif hanya digunakan sebagai *pretrained feature extractor* yang bertugas menghasilkan representasi vektor dari teks judul skripsi secara langsung.

Untuk memastikan hasil evaluasi bersifat objektif dan terhindar dari bias kebocoran data, pengujian utama mengimplementasikan skema *Nested Cross-Validation* dengan variasi 5-Fold dan 10-Fold guna mengamati pengaruh porsi data latih. Pada tahap optimalisasi, Usulan akan mengikuti rentang dan step terbaik dari sknario lain lalu ditambahkan *preprocessing stopword removal*. Kemudian dikomparasikan dengan hasil dua scenario lain yang tidak menggunakan *stopword removal* mengikuti penelitian terdahulu dan bertindak sebagai *baseline*. Rincian

batas pencarian threshold untuk masing-masing metode tersebut dirangkum pada Tabel 3.4.

Tabel 3. 4 Skenario Uji Pipeline

No.	Skenario	Rentang	Step
1.	Grid Search 5-Fold dari (Ghinassi *et al.*, 2023)	0.05-0.95	0.05
2.	Grid Search 10-Fold dari (Ghinassi *et al.*, 2023)	0.05-0.95	0.05
3.	Grid Search 5-Fold dari (Holis *et al.*, 2025)	0.5-0.9	0.1
4.	Grid Search 10-Fold dari (Holis *et al.*, 2025)	0.5-0.9	0.1

Berdasarkan Tabel 3.4 di atas, ketiga konfigurasi rentang *threshold* tersebut masing-masing akan diujikan pada lingkungan 5-Fold dan 10-Fold *Nested CV*. Kombinasi ini secara total akan menghasilkan 4 skenario pengujian evaluasi performa, dengan membandingkan efisiensi rentang dan kerapatan step metode Ghinassa serta Holis. Output dari 4 skenario ini berupa metrik rata-rata *Accuracy, Precision, Recall, dan F1-Score*.

Setelah pipeline terbaik didapatkan dari pengujian *Nested CV*, sistem akan memasuki tahap implementasi akhir untuk mengekstrak satu nilai *threshold*. Proses pencarian ini menggunakan skema *Flat Cross-Validation* sebagaimana Tabel 3.5.

Tabel 3. 5 Skenario Threshold Search

Input Konfigurasi	Skema Pengujian	Evaluasi Keputusan	Output Akhir
Analisis Fold dan Metode scenario sebelumnya	Flat K-Fold CV	Menguji skor F1-Score / Accuracy tertinggi dari seluruh titik *Grid Search*	1 Nilai Threshold Mutlak untuk Aplikasi

3.3.2 Evaluasi Sistem

Tahapan evaluasi sistem dilakukan untuk mengukur tingkat keberhasilan dan keandalan model dalam mendeteksi kemiripan judul. Evaluasi ini didasarkan pada perbandingan antara hasil prediksi kemiripan dari sistem dengan nilai *ground truth*. Hasil perbandingan tersebut kemudian dipetakan ke dalam komponen *Confusion Matrix*, yang meliputi *True Positive*, *True Negative*, *False Positive* dan *False Negative*. Berdasarkan keempat komponen dasar tersebut, performa sistem akan dihitung dan dianalisis menggunakan empat parameter pengukuran utama, yaitu Akurasi, Presisi, *Recall*, dan *F1-Score*.

A. Akurasi

Akurasi digunakan sebagai metrik utama untuk memberikan gambaran umum kinerja model dalam menebak keseluruhan. Pada rumus 3.13 metode akurasi ini bekerja dengan menjumlahkan semua prediksi yang benar, yaitu *TP dan TN*, lalu membagi (operasi $\frac{x}{y}$) jumlah tersebut dengan total keseluruhan data (*TP*+*TN*+*FP*+*FN*). Metrik ini menunjukkan seberapa sering model membuat keputusan yang tepat secara keseluruhan.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.13}$$

B. Presisi

Presisi mengukur tingkat ketepatan model ketika memprediksi sebuah judul masuk ke dalam label. Metrik ini berfokus pada minimalisasi *False Positive*. Pada rumus 3.13 metode presisi bekerja dengan mengambil jumlah *True Positive (TP*),

yaitu prediksi mirip yang memang benar mirip, lalu membagi (operasi $\frac{x}{y}$) jumlah tersebut dengan total semua data yang diprediksi sebagai mirip oleh sistem (*TP* + *FP*). Presisi yang tinggi menunjukkan bahwa jika sistem menandai judul sebagai mirip, maka prediksi tersebut kemungkinan benar.

$$\text{Presisi} = \frac{TP}{TP + FP} \tag{3.13}$$

C. Recall

Recall mengukur kemampuan model untuk menemukan kembali semua judul skripsi yang sebenarnya memang berasal label. Metrik ini berfokus pada minimalisasi *False Negative*. Rumus 3.14 untuk recall bekerja dengan mengambil jumlah *True Positive*, lalu membagi (operasi $\frac{x}{y}$) jumlah tersebut dengan total semua data yang seharusnya berlabel mirip (*TP* + *FN*). Recall yang tinggi sangat penting untuk memastikan sistem tidak melewatkan atau gagal mendeteksi judul-judul yang semestinya mirip.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3.14}$$

D. F-1 Score

F1-Score adalah nilai rata-rata harmonis yang menyeimbangkan antara Presisi dan Recall. Metrik ini sangat berguna karena terjadi *imbalance* antara jumlah data judul pada satu topik dengan topik lainnya di dalam dataset. Pada rumus 3.15 *F1-Score* bekerja dengan mengalikan 2 (operasi $2 \times ..$)dengan hasil pembagian (operasi $\frac{x}{y}$) antara perkalian *Presisi* dan *Recall* ($Presisi \times Recall$)

dengan penjumlahan *Presisi* dan *Recall* ($Presisi\ +\ Recall$). Nilai *F1-Score* yang tinggi menunjukkan model memiliki keseimbangan antara *Presisi* dan *Recall*.

$$F1\text{-}Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{3.15}$$

E. Specificity

Specificity digunakan untuk mengukur sejauh mana sistem atau model mampu mengidentifikasi kelas negative secara benar. nilai specificity dihitung dengan rumus 3.16 membagi jumlah prediksi benar pada kelas negatif TN dengan total keseluruhan data yang aslinya bernilai negatif. Total data yang aslinya negatif ini merupakan hasil penjumlahan antara TN dan kesalahan sistem dalam mendeteksi kelas positif FP. Metrik ini memastikan bahwa sistem tidak terlalu sensitif sehingga memunculkan banyak alarm palsu.

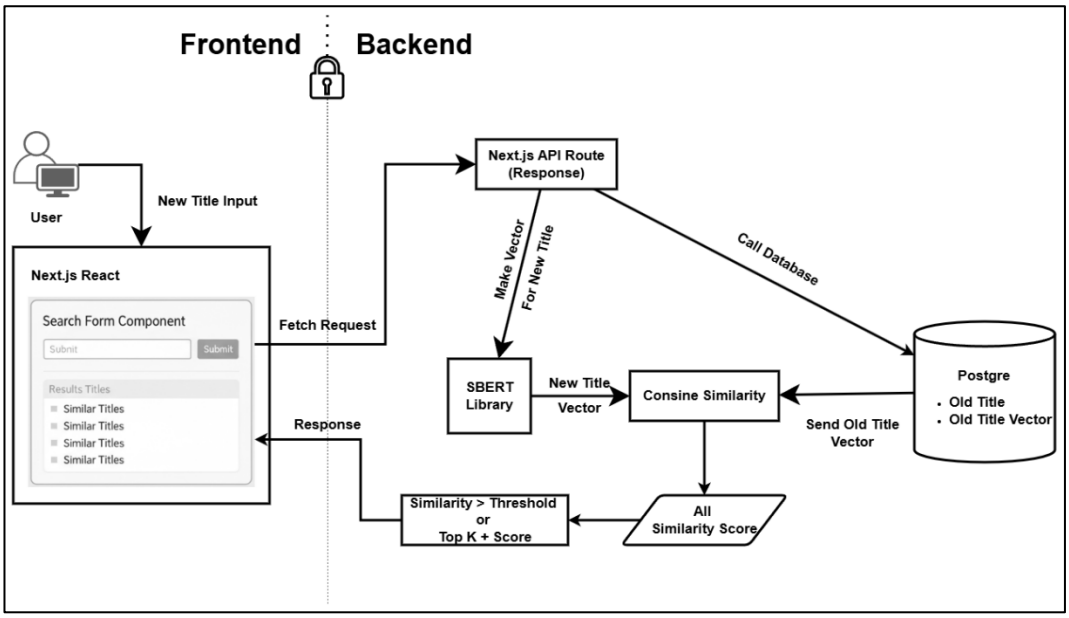
$$\text{Specificity} = \frac{TN}{TN + FP} \tag{3.16}$$

3.4 Implementasi Sistem

Tahap implementasi dimana desain konseptual dan alur kerja algoritma diterjemahkan ke dalam bahasa pemrograman. Aplikasi ini dikembangkan menggunakan arsitektur modern berbasis web, di mana interaksi dengan pengguna dikelola oleh sisi *frontend*, sementara tugas komputasi diserahkan ke sisi server.

Untuk *Frontend* digunakan dalam interaksi pengguna dan presentasi hasil dikembangkan menggunakan *React* di dalam *Next.js,* dapat dilihat di gambar 3.10 *frontend* ini akan mencakup *form input* untuk judul baru dan tampilan daftar judul yang memiliki kemiripan semantik. Komponen *frontend* ini akan mengirimkan permintaan ke *backend* ketika pengguna mengajukan judul baru.

Backend menangani semua logika pemrosesan data dan komputasi yang berat. Pada gambar 3.10 *backend* dibangun menggunakan *API Routes* di *Next.js*, bagian ini akan menerima judul baru dari *frontend,* memprosesnya dengan *SBERT*, mengelola koneksi dan *query* ke *PostgreSQL* untuk mengambil vektor *embedding*, melakukan perhitungan *Cosine Similarity* dan mengirimkan hasil *output* ke *frontend* berupa kalimat yang *similar* atau *top k* dengan *score* tertinggi.



Gambar 3. 10 Alur Dalam Sistem

Sistem deteksi kemiripan judul skripsi ini diimplementasikan dalam bentuk aplikasi web menggunakan framework Next.js dengan bahasa pemrograman JavaScript/React. Aplikasi ini dirancang untuk memudahkan program studi maupun mahasiswa dalam melakukan validasi orisinalitas judul skripsi dengan antarmuka yang modern, cepat, dan interaktif. Berikut adalah hasil implementasi dari setiap komponen aplikasi web yang telah dikembangkan.

3.4.1 Arsitektur Sistem

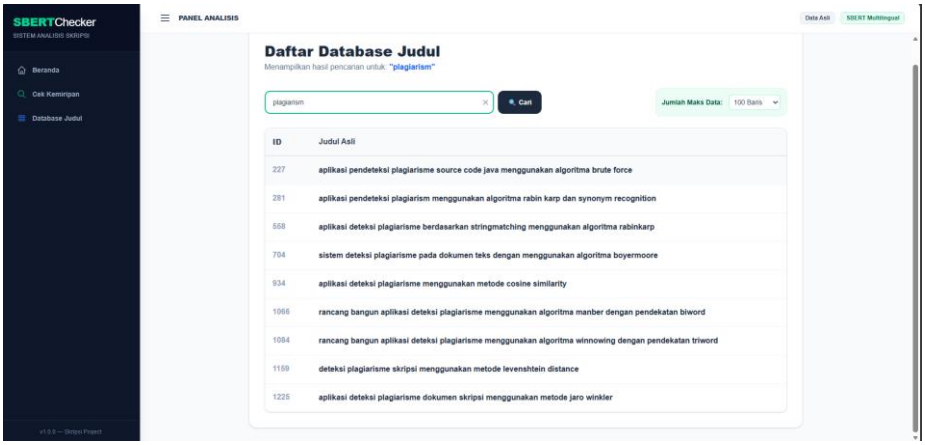
Web dibangun dengan arsitektur modern berbasis *serverless* yang terdiri dari beberapa lapisan utama. Pada lapisan *interface* atau *Frontend Layer*, sistem menggunakan Next.js sebagai *framework* utama yang dipadukan dengan *Tailwind CSS* untuk membangun tampilan yang responsif. Desain pada lapisan ini dibuat menggunakan pendekatan *Sidebar-Layout* untuk memberikan kesan *dashboard* profesional yang mudah dinavigasi.

Untuk *Database Layer* sistem memanfaatkan *Supabase* yang berbasis *PostgreSQL*. Komponen ini berperan sebagai *Database as a Service* yang bertugas menyimpan ribuan repositori judul skripsi dan vectornya secara *real-time* dan aman.

Sementara itu, pada lapisan *Model & Processing Layer* aplikasi mengintegrasikan perangkat lunak *@xenova/transformers*. Perangkat lunak ini berfungsi untuk menjalankan model *Multilingual Sentence-BERT* berdimensi 768 secara langsung di dalam aplikasi. Model tersebut diproses menggunakan teknologi *WebAssembly* di dalam lingkungan *serverless Vercel*, sehingga komputasi untuk mencari nilai *semantic similarity* dapat berjalan tanpa memerlukan infrastruktur server *GPU* yang terpisah.

3.4.2 Halaman Database

Halaman Eksplorasi *Database J*udul merupakan komponen manajemen data yang memungkinkan pengguna untuk melihat, memfilter, dan mencari keseluruhan data judul skripsi yang telah terintegrasi di dalam *Supabase*. Halaman Database dapat dilihat pada Gambar 3.11.



Gambar 3. 11 Database Judul

Sistem menyediakan kolom pencarian yang terhubung langsung ke *database* menggunakan fungsi filter bawaan dari *Supabase*. Fitur ini dirancang agar tidak peka terhadap *case-insensitive*, sehingga memungkinkan pengguna mencari kata kunci judul spesifik di seluruh database secara akurat, meskipun terdapat perbedaan penulisan huruf besar dan kecil. Terdapat juga mekanisme indikator visual berupa animasi memuat yang aktif saat sistem mengambil data dari *server*.

Untuk menjaga performa aplikasi agar tidak terbebani saat memuat ribuan data sekaligus, sistem dilengkapi dengan kontrol pembatasan data. Pengguna dapat menggunakan menu *dropdown* untuk mengatur jumlah baris yang ditampilkan pada tabel. Pilihan yang disediakan mencakup penayangan 50, 100, 1.000, hingga batas maksimal 2.000 baris data terbaru secara dinamis. Cuplikan kode utama yang menjalankan logika pengelolaan data pada halaman database pada Tabel 3.6.

Tabel 3. 6 Potongan Code Halaman Database

```
// app/database/page.jsx

import { useState, useEffect } from 'react';

import { supabase } from '@/lib/supabase';
```

```
// ... (import komponen lain)
export default function DatabasePage() {
  const [katakunci, setKatakunci] = useState('');
  const [limitData, setLimitData] = useState(50);
  const [dataJudul, setDataJudul] = useState([]);
  // ... (inisialisasi state loading dan error)
  useEffect(() => {
    async function fetchData() {
      // ... (mengaktifkan status loading)
      const { data, error } = await supabase
        .from('titles')
        .select('*')
        .ilike('judul_asli', `%${katakunci}%`)
        .limit(limitData);
      if (data) setDataJudul(data);
      // ... (penanganan error dan menonaktifkan
status loading)  }
    fetchData();
  }, [katakunci, limitData]);
  return (
    // ... (struktur UI komponen web)
```

```
    <input onChange={(e) =>
setKatakunci(e.target.value)} placeholder="Cari
judul..." />
    <select onChange={(e) =>
setLimitData(e.target.value)}>
       <option value={50}>50 Baris</option>
       // ... (opsi limit 100, 1000, 2000)
    </select>
    // ... (render hasil dataJudul ke dalam bentuk
tabel)
  );}
```

Kode pada tabel 3.6 merupakan bagian inti dari antarmuka halaman database, di mana penjabaran fungsi spesifik dari masing-masing potongan sintaks tersebut dapat dilihat pada Tabel 3.7 di bawah ini.

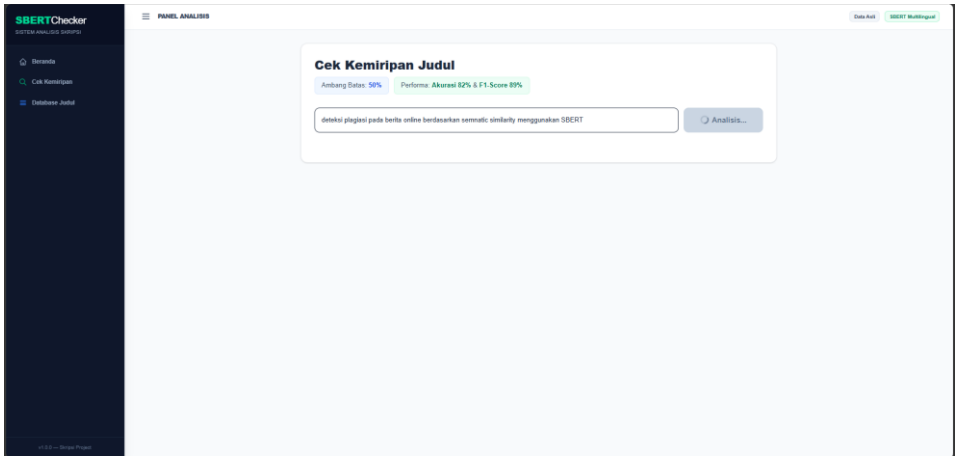
Tabel 3. 7 Fungsi Potongan Code di halaman database

File	Potongan Kode	Fungsi Kode
daftar/page.jsx	`onChange={(e) => setKatakunci(e.target.value)}`	Menangkap ketikan pengguna pada kolom pencarian secara real-time untuk dijadikan parameter filter pencarian.
daftar/page.jsx	`<select onChange={(e) => setLimitData(e.target.value)}>`	Menyediakan menu dropdown untuk mengatur batas maksimal jumlah data yang ditarik secara dinamis.
daftar/page.jsx	`.select('*').ilike('judul_asli', %${katakunci}%).limit(limitData)`	Fungsi query Supabase. ilike digunakan agar pencarian mengabaikan

File	Potongan Kode	Fungsi Kode
		huruf kapital dan limit membatasi beban jaringan.
daftar/page.jsx	`if (isLoading) return <LoadingSpinner />`	Menampilkan indikator visual berupa animasi loading saat sistem masih menunggu balikan data dari server Supabase.

3.4.3 Halaman Cek Similarity

Halaman ini merupakan fitur utama dari sistem, di mana pengguna dapat menginputkan rancangan judul skripsi baru untuk dikomparasikan secara semantik dengan data judul lama di database menggunakan model SBERT. Halaman Cek Similarity dapat dilihat pada Gambar 3.12.



Gambar 3. 12 Cek Similarity

Cuplikan kode utama pada berkas API route yang memproses perhitungan di balik layar saat pengguna melakukan pengecekan pada tabel 3.8.

Tabel 3. 8 Perhitungan Backend

```
// app/api/cek/route.js
import { NextResponse } from 'next/server';
import { pipeline } from '@xenova/transformers';
```

```
import { supabase } from '@/lib/supabase';


    // Load Model SBERT Multilingual
    if (!modelInstance) {
       modelInstance = await pipeline('feature-
extraction', 'Xenova/paraphrase-multilingual-mpnet-
base-v2');
    }

    // Ekstraksi teks menjadi vektor embedding
    const output = await
modelInstance(teksUntukDiencode, { pooling: 'mean',
normalize: true });
    const embedding = Array.from(output.data);

    // Perhitungan Cosine Similarity di Database
menggunakan RPC
    const { data, error } = await
supabase.rpc('match_titles_dynamic', {
      query_embedding: embedding,
      // ... (parameter konfigurasi RPC lainnya)
    });

    // ... (penanganan error eksekusi)
    return NextResponse.json({ success: true, data });
  } catch (err) {
    // ... (penanganan server error)
  }
}
```

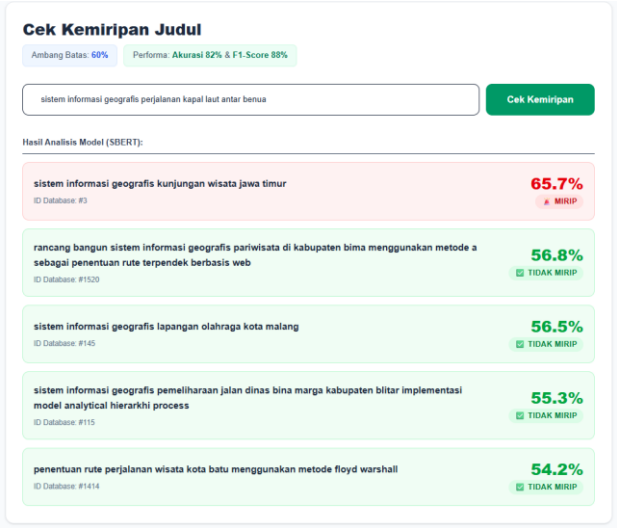
Blok kode di tabel 3. menunjukkan proses komputasi cerdas yang terjadi di backend saat pengguna melakukan pengecekan kemiripan. Rincian implementasi fungsional dari proses ekstraksi dan pelabelan tersebut dijabarkan lebih lanjut pada Tabel 3.9.

Tabel 3. 9 Fungsi Backend

File	Potongan Kode	Fungsi Kode
Check/route.js	`const modelInstance = await pipeline('feature-`	Memuat model machine learning SBERT ke dalam memori untuk

File	Potongan Kode	Fungsi Kode
	`extraction', 'Xenova/...')`	mempersiapkan proses konversi teks.
Check/route.js	`const output = await modelInstance(teksUntukDiencode, { ... })`	Mengekstraksi teks judul masukan pengguna menjadi format representasi vektor angka berdimensi 768.
Check/route.js	`supabase.rpc('match_titles_dynamic', { query_embedding: embedding })`	Memanggil fungsi kustom (RPC) di dalam server Supabase untuk menjalankan perhitungan matematis Cosine Similarity.

Pada bagian atas halaman, sistem secara transparan menampilkan metrik performa model berupa nilai Threshold sebesar 50%, serta nilai evaluasi model yang mencakup Akurasi sebesar 82% dan F1-Score sebesar 89%. Untuk memulai proses deteksi, pengguna hanya perlu memasukkan rancangan judul pada kolom teks yang disediakan lalu menekan tombol cek kemiripan seperti Gambar 3.13.



Gambar 3. 13 Hasil Cek Similarity

Pada Gambar 4.9 setelah pengguna menginputkan judul dan menekan tombol cek, sistem akan memproses ekstraksi fitur teks dan menghitung nilai Cosine Similarity. Hasil deteksi tersebut kemudian dimunculkan dalam bentuk kartu informasi yang interaktif. Setiap kartu menampilkan judul dari database pembanding, nomor ID database, serta persentase tingkat kemiripan.

Sistem memberikan indikator visual yang jelas berdasarkan persentase kemiripan yang dihasilkan. Apabila persentase kemiripan menyentuh angka ambang batas 50% atau lebih, sistem akan menampilkan label berwarna merah bata bertuliskan Mirip. Label ini tidak serta-merta menjatuhkan vonis plagiasi, melainkan berfungsi sebagai peringatan awal bahwa terdapat kesamaan konteks bahasan yang terlalu dekat dengan penelitian terdahulu. Sebaliknya, apabila nilai persentase berada di bawah ambang batas tersebut, sistem akan menampilkan label berwarna hijau bertuliskan Tidak Mirip. Indikator ini menandakan bahwa rancangan judul tersebut secara semantik memiliki kebaruan ide atau arah penelitian yang berbeda dari data yang sudah ada. Blok kode interface di tabel 3.10 mengatur interaksi langsung dengan pengguna serta logika visualisasi hasil komparasi yang dikembalikan oleh server.

Tabel 3. 10 Code Frontend

```
// app/cek/page.jsx

import { useState } from 'react';

// ... (import komponen UI dan ikon lainnya)

export default function CekSimilarityPage() {

  const [judulInput, setJudulInput] = useState('');
```

```
  const [hasilCek, setHasilCek] = useState([]);
  // ... (deklarasi state untuk status loading)
  const handleCekKemiripan = async () => {
    // ... (mengaktifkan animasi loading)
    // Mengirim rancangan judul ke API backend
    const response = await fetch('/api/cek', {
      method: 'POST',
      headers: { 'Content-Type': 'application/json' },
      body: JSON.stringify({ newTitle: judulInput })
    });
    const result = await response.json();
    if (result.success) {
      setHasilCek(result.data);  //  Menyimpan  hasil
respons dari server
    }
    // ... (menonaktifkan loading dan penanganan error)
  };
  return (
    // ... (struktur layout bagian atas halaman)
    <input               onChange={(e)               =>
setJudulInput(e.target.value)}  placeholder="Masukkan
rancangan judul skripsi..." />
```

```
    <button onClick={handleCekKemiripan}>Cek
Kemiripan</button>
    // ... (pembuatan daftar kartu hasil deteksi)
    {hasilCek.map((item, index) => {
      // Mengubah nilai probabilitas (0-1) menjadi
format persentase
      const persentase = (item.similarity *
100).toFixed(2);
      // Logika pelabelan dan pewarnaan berdasarkan
threshold 60%
      const status = persentase >= 60 ? "Mirip" :
"Tidak Mirip";
      const warnaLabel = persentase >= 60 ? "bg-red-
500" : "bg-green-500";
      return (
        <div key={index} className="kartu-hasil-
evaluasi">
          <p                          className="teks-
judul">{item.judul_asli}</p>
          // ... (elemen UI lainnya)
          <span          className={`indikator-status
${warnaLabel} text-white`}>
            {status} ({persentase}%)
```

```
            </span>

        </div>       );     })});}
```

Rincian mengenai fungsi pemanggilan data dan pengkondisian visual pada antarmuka tersebut dijabarkan pada Tabel 3.11.

Tabel 3. 11 Fungsi Frontend

File	Potongan Kode	Fungsi Kode
page.jsx	`fetch('/api/cek', { method: 'POST', body: ... })`	Mengirimkan teks rancangan judul yang diketik oleh pengguna ke berkas API route untuk diproses oleh model SBERT.
page.jsx	`const persentase = (item.similarity * 100).toFixed(2)`	Mengonversi nilai mentah Cosine Similarity menjadi format persentase yang mudah dipahami dengan pembulatan dua angka di belakang koma.
page.jsx	`const status = persentase >= 50 ? "Mirip" : "Tidak Mirip"`	Logika penentuan status secara dinamis berdasarkan nilai *threshold* sebesar 50% yang telah ditetapkan.
page.jsx	`const warnaLabel = persentase >= 50 ? "bg-red-500" : "bg-green-500"`	Logika pewarnaan frontend yang merender latar belakang warna merah bata jika indikasi mirip terpenuhi, dan hijau jika sebaliknya.
page.jsx	`<span className={indikator-status ${warnaLabel}}>`	Menyisipkan hasil evaluasi pewarnaan ke dalam properti kelas HTML agar tampilan langsung berubah secara real-time sesuai hasil deteksi.

BAB IV
HASIL DAN PEMBAHASAN

4.1 Hasil

4.1.1 Persiapan data

Tahap pengumpulan data berhasil menghimpun sebanyak 1.887 dokumen judul skripsi unik mahasiswa Program Studi Teknik Informatika UIN Maulana Malik Ibrahim Malang kurun waktu tahun 2008 sampai 2026. Teks judul skripsi yang telah dikumpulkan tersebut selanjutnya masuk ke dalam tahap pemrosesan data, di mana proses *pairwise combination* membentuk populasi pasangan kombinasi judul yang masif yaitu sebanyak 1.779.441 pasang data. Adapun perbedaan total judul terdapat pada tabel 4.1 dibawah.

Tabel 4. 1 Total Dataset

Dataset	**Bentuk**	**Jumlah**
Awal	Kalimat Tunggal	1887
pairwise combination	Pasangan Kalimat	1.779.441

Untuk mengatasi penumpukan data pada rentang skor kemiripan yang rendah, seluruh pasangan judul diproses lebih lanjut menggunakan penyaringan terstratifikasi *Cosine Similarity* dan *TF-IDF*. Data dikeolompokkan ke dalam similar, hard non-similar, dan easy non-similar. Kategori hard negative dan easy negative dibentuk dengan mengambil pasangan yang memiliki nilai *similarity* 10% tertinggi dari kelompok *non-similar*, sehingga diperoleh contoh kasus yang lebih menantang dan lebih representatif untuk proses evaluasi. Selanjutnya, kedua

kategori digabungkan ke label *non-similar*. Sebaran awal pasangan data dengan label *similar* dan *non-similar* dapat dilihat pada Tabel 4.2.

Tabel 4. 2 Jumlah dan Rasio Data

Keterangan	**Perbandingan Data Awal**		**Ratio & Proporsi True**		**Total Data**
	Positive	Negative	Ratio	Proporsi	
Frist Pair Data	333	1.779.108	1:5.342	0,0187%	1.779.441
First Undersampling (Take ngative that top-10% similar)	333	8.901	1:26	3,6%	9.234
Undersampling 2 (Balance)	333	333	1:1	50%	666
Upsampling (+ 5% negative)	333	367	1:1,1	47,57%	700

Rasio dari *Similar* dan *Non-Similar* ini masih 1:26 bisa dianggap *moderate imbalance*, agar lebih *balance* dilakukan *Under Sampling* lagi dengan menyamakan rasio tiap kelas, tetapi dialokasikan juga tambahan 5% data *negative* ekstra dengan menerapkan teknik *sample size padding*, untuk mengantisipasi adanya *mislabeling* atau penyusutan data selama proses validasi oleh *expert.* Didapatkan rasio *mild imbalance* di 1:1.1, setelahnya data diberi label validasi oleh *expert annotator*.

4.1.2 Validasi Ground Truth oleh Expert

Dengan *expert* yang melabeli *similar* atau *non-similar* pada data pair yang mengikuti rubik pada gambar 3.4, diperoleh distribusi kelas dengan rasio 1:2,9 dataset pada Tabel 4.3.

Tabel 4. 3 Total Ground Truth

Keterangan	**Data *Pair***
Similar Pair	521
Non-Similar Pair	179
Total	**700**

4.1.3 Skenario Rentang Ghinassi

A. Skenario 1 Rentang Ghinassi 5-Fold

Skenario ini merupakan pengujian *pipeline SBERT* menggunakan parameter rentang *threshold* berdasarkan penelitian Ghinassi, yaitu pada rentang 0.05 - 0.95 dengan ukuran step 0.05. Evaluasi ini dijalankan menggunakan skema *5-Fold Nested Cross-Validation* untuk menguji ketahanan model secara objektif. Hasil pengujian untuk setiap *outer fold* beserta *unbiased score* disajikan pada Tabel 4.4.

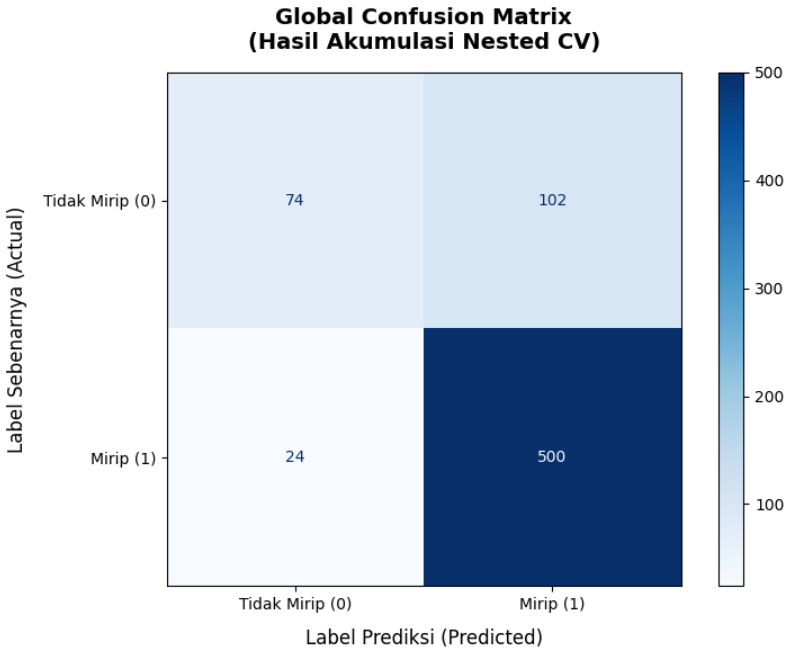
Tabel 4. 4 CV Ghinassi 5-Fold

Fold	T (inner Fold)	Acc. (%)	Prec. (%)	Rec. (%)	Specif. (%)	F1-Score (%)
Fold 1	0.50	82.14	83.74	95.37	37.50	89.18
Fold 2	0.50	79.29	77.60	98.98	33.33	87.00
Fold 3	0.60	85.71	89.66	92.86	57.14	91.23
Fold 4	0.50	82.14	80.92	100.00	26.47	89.45
Fold 5	0.60	80.71	84.11	90.00	57.50	86.96
Global Avg.	-	**82.00**	**83.20**	**95.44**	**42.39**	**88.76**

Berdasarkan Tabel 4.4, proses optimasi pada inner loop secara dinamis memilih nilai threshold terbaik pada angka 0.50 dan 0.60. Secara keseluruhan, performa rata-rata unbiased dari model SBERT pada skenario ini mencatatkan nilai Accuracy sebesar 82.00% dan F1-Score yang tinggi sebesar 88.76%.

Tingginya nilai Recall global yang menyentuh angka 95.44% bahkan mencapai 100% pada Fold 4 menunjukkan bahwa model pada rentang parameter ini sangat sensitif dan agresif dalam menjaring pasangan judul yang memiliki kemiripan semantik. Namun, nilai Precision yang sedikit lebih rendah di angka 83.20% mengindikasikan adanya sejumlah pasangan judul false positive.

Untuk melihat lebih detail distribusi tebakan model yang menghasilkan metrik performa tersebut, akumulasi dari seluruh hasil prediksi pada outer fold digambarkan melalui grafik Global Confusion Matrix pada Gambar 4.1.



Gambar 4. 1 Confusio Matrix Ghinassi 5-Fold

Berdasarkan grafik Global Confusion Matrix pada Gambar 4.1, terlihat jelas rincian sebaran dari 700 total data uji yang dievaluasi sepanjang 5 outer fold. Model berhasil memprediksi kelas Mirip True Positive dengan sangat baik sebanyak 500 pasangan dokumen, dan hanya meleset mengenali kemiripan pada 24 pasangan dokumen *False Negative*. Rasio True Positive yang sangat mendominasi inilah yang menjadi faktor pendorong tingginya nilai Recall 95.44%.

Kelas Tidak Mirip model berhasil menebak dengan benar sebanyak 74 pasangan dokumen *True Negative*. Namun, terdapat 102 pasangan dokumen yang sebenarnya Tidak Mirip tetapi diprediksi sebagai Mirip oleh model *False Positive*.

Angka False Positive yang cukup signifikan ini merupakan konsekuensi dari terpilihnya threshold yang relatif rendah 0.50 dan 0.60, sehingga membuat model cenderung melonggarkan batas kemiripan dan memicu turunnya nilai Precision ke angka 83.20%. Secara umum, model SBERT pada skenario ini sudah bekerja dengan sangat baik (F1-Score 88.76%), namun memiliki *bias* untuk lebih mudah mengklasifikasikan dokumen ke dalam kelas Mirip.

B. Skenario 2 Rentang Ghinassi 10-Fold

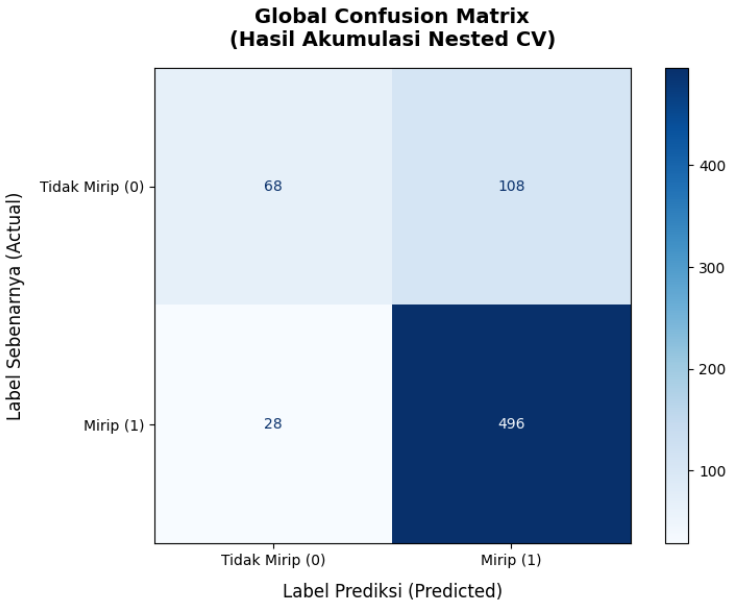
Skenario 2 memperluas pengujian parameter rentang threshold Ghinassi dengan rentang 0.05 hingga 0.95, step 0.05 ke dalam skema *10-Fold Nested Cross-Validation*. Pengujian ini bertujuan untuk melihat dampak pembagian data yang lebih rapat terhadap performa dan pencarian *threshold* optimal model SBERT. Hasil pengujian untuk ke-10 *outer fold* beserta rata-rata global disajikan Tabel 4.5.

Tabel 4. 5 CV Ghinassi 10-Fold

Fold	T (inner Fold)	Acc. (%)	Prec. (%)	Rec. (%)	Specif. (%)	F1-Score (%)
Fold 1	0.50	78.57	80.65	94.34	29.41	86.96
Fold 2	0.50	85.71	86.89	96.36	46.67	91.38
Fold 3	0.50	74.29	71.43	100.00	28	83.33
Fold 4	0.60	80.00	88.24	84.91	64.71	86.54
Fold 5	0.50	84.29	82.54	100.00	38.89	90.43
Fold 6	0.60	84.29	91.53	90.00	50	90.76
Fold 7	0.50	81.43	79.69	100.00	31.58	88.70
Fold 8	0.50	82.86	82.09	100.00	20	90.16
Fold 9	0.50	74.29	73.02	97.87	26.09	83.64
Fold 10	0.60	80.00	88.24	84.91	64,71	86.54
Global Avg.	-	**80.57**	**82.43**	**94.84**	**40.00**	**87.84**

Berdasarkan Tabel 4.5, mekanisme inner loop mengunci nilai threshold terbaik pada angka 0.50 di sebagian besar lipatan data serta beberapa di angka 0.60. Secara akumulatif, skema 10-Fold ini mencatatkan nilai murni rata-rata global berupa Accuracy sebesar 80.57% dan F1-Score sebesar 87.84%. Pola metrik pada skenario ini memperlihatkan nilai Recall yang tetap dominan tinggi menyentuh 94.84%, bahkan mencapai tingkat sempurna 100% pada empat fold berbeda. Hal ini mengonfirmasi bahwa cakupan parameter tersebut efektif mendeteksi kemiripan semantik antarjudul skripsi, meskipun harus ditebus dengan nilai Precision sebesar 82.43% akibat kemunculan false positive.

Capaian performa di setiap subset data diilustrasikan secara lebih mendalam melalui distribusi prediksi model menggunakan grafik Global Confusion Matrix pada Gambar 4.2.



Gambar 4. 2 Confusion Matrix Ghinassi 10-Fold

Berdasarkan grafik Global Confusion Matrix pada Gambar 4.2, terlihat rincian sebaran dari 700 total data uji yang dievaluasi sepanjang 10 outer fold. Model SBERT berhasil memprediksi kelas Mirip secara tepat True Positive sebanyak 496 pasangan dokumen dan hanya meleset mengenali kemiripan False Negative pada 28 pasangan dokumen. Rasio True Positive yang dominan inilah yang mendorong tingginya capaian Recall 94.84%.

Sebaliknya untuk kelas Tidak Mirip, model berhasil menebak dengan benar True Negative pada 68 pasangan dokumen. Kendala utama terlihat pada 108 pasangan dokumen yang sebenarnya Tidak Mirip tetapi justru diprediksi sebagai Mirip oleh model False Positive. Tingkat False Positive yang cukup besar ini menegaskan kembali dampak dari pemilihan threshold rendah 0.50 yang membuat model terlalu agresif dalam menentukan kemiripan, sehingga pada akhirnya menekan angka Precision ke bawah.

4.1.4 Skenario Rentang Holis

A. Skenario 3 Rentang Holis 5-Fold

Skenario 3 mengimplementasikan parameter pencarian *threshold* berdasarkan penelitian Holis, yang membatasi pencarian pada rentang 0.5 hingga 0.9 dengan ukuran step 0.1. Pengujian ini dievaluasi menggunakan skema *5-Fold Nested Cross-Validation* dengan hasil pengujian untuk setiap *outer fold* beserta capaian rata-rata *unbiased score* disajikan pada Tabel 4.6 berikut.

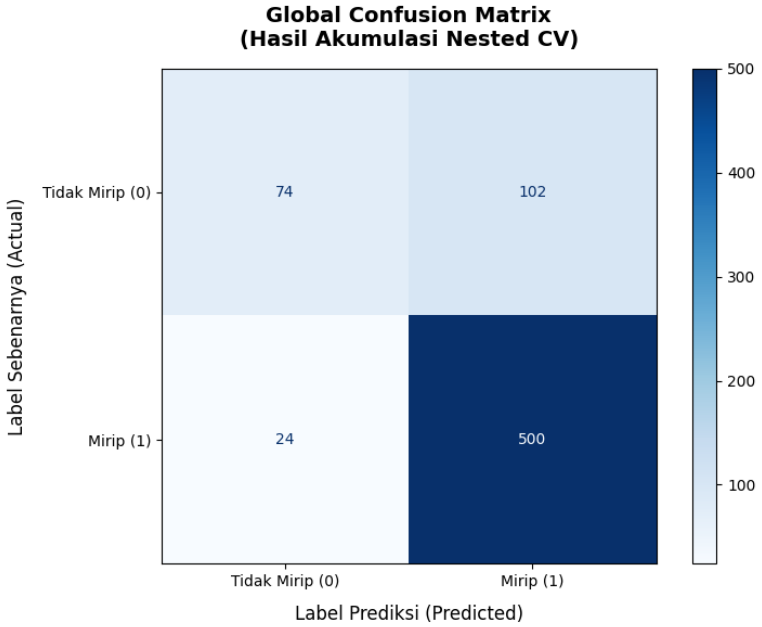
Tabel 4. 6 CV Holis 5-Fold

Fold	T (inner Fold)	Acc. (%)	Prec. (%)	Rec. (%)	Specif. (%)	F1-Score (%)
Fold 1	0.50	82.14	83.74	95.37	37.50	89.18
Fold 2	0.50	79.29	77.60	98.98	33.33	87.00

Fold	T (inner Fold)	Acc. (%)	Prec. (%)	Rec. (%)	Specif. (%)	F1-Score (%)
Fold 3	0.60	85.71	89.66	92.86	57.14	91.23
Fold 4	0.50	82.14	80.92	100.00	26.47	89.45
Fold 5	0.60	80.71	84.11	90.00	57.50	86.96
Global Avg.	-	**82.00**	**83.20**	**95.44**	**42.39**	**88.76**

Berdasarkan Tabel 4.6, proses optimasi inner loop secara dinamis memilih angka 0.50 dan 0.60 sebagai nilai threshold terbaik di kelima lipatan data. Secara keseluruhan, model SBERT pada skenario ini menghasilkan rata-rata metrik Accuracy sebesar 82.00% dan F1-Score sebesar 88.76%. Metrik Recall tetap mendominasi dengan capaian sangat tinggi yaitu 95.44%, yang berarti model ini sangat peka dalam meloloskan dokumen sebagai kelas Mirip. Kepekaan deteksi ini diiringi oleh nilai Precision yang berada di angka 83.20%.

Distribusi tebakan model secara keseluruhan yang menghasilkan metrik performa tersebut digambarkan melalui grafik Global Confusion Matrix pada Gambar 4.3.



Gambar 4. 3 Confusion Matrix Holis 5-Fold

Visualisasi pada Gambar 4.3 memperlihatkan rincian sebaran dari 700 total data uji yang dievaluasi di sepanjang lima outer fold. Model terbukti sangat tangguh dalam mendeteksi kemiripan dengan mencatatkan True Positive sebanyak 500 pasangan dokumen, dan hanya meleset menghasilkan False Negative pada 24 pasangan dokumen. Rasio True Positive yang mendominasi inilah yang mendongkrak kalkulasi metrik Recall hingga menyentuh 95.44 persen.

Model berhasil menebak benar True Negative sebanyak 74 pasangan dokumen. Terdapat 102 pasangan dokumen yang sebenarnya Tidak Mirip tetapi diprediksi sebagai Mirip sehingga diklasifikasikan sebagai False Positive. Tingginya kasus False Positive ini menegaskan bahwa meskipun model sangat agresif dalam mengenali kemiripan dokumen, akurasi tebakannya secara keseluruhan sedikit tertekan oleh kecenderungan model yang keliru menyimpulkan

dokumen berbeda sebagai dokumen yang mirip akibat penumpukan batas keputusan di area tengah.

B. Skenario 4 Rentang Holis 10-Fold

Skenario 4 merupakan perluasan pengujian parameter pencarian threshold dari Holis dengan rentang 0.5 hingga 0.9 ke dalam skema 10-Fold Nested Cross-Validation. Pengujian ini dilakukan untuk mengobservasi konsistensi parameter tersebut ketika dihadapkan pada pembagian data latih dan data uji yang lebih rapat. Hasil evaluasi metrik untuk seluruh outer fold beserta capaian rata-rata globalnya dirangkum pada Tabel 4.7.

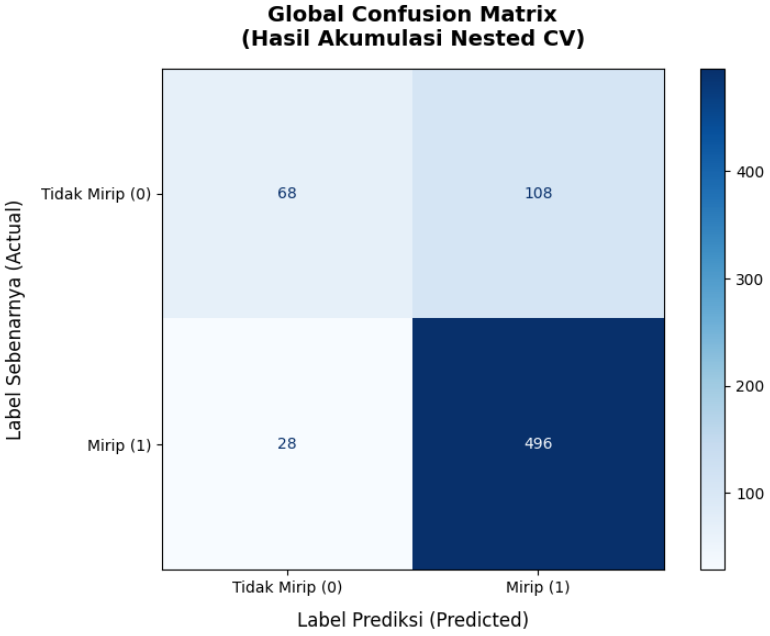
Tabel 4. 7 CV Holis 10-Fold

Fold	T (inner Fold)	Acc. (%)	Prec. (%)	Rec. (%)	Specif. (%)	F1-Score (%)
Fold 1	0.50	78.57	80.65	94.34	29.41	86.96
Fold 2	0.50	85.71	86.89	96.36	46.67	91.38
Fold 3	0.50	74.29	71.43	100.00	28	83.33
Fold 4	0.60	80.00	88.24	84.91	64.71	86.54
Fold 5	0.50	84.29	82.54	100.00	38.89	90.43
Fold 6	0.60	84.29	91.53	90.00	50	90.76
Fold 7	0.50	81.43	79.69	100.00	31.58	88.70
Fold 8	0.50	82.86	82.09	100.00	20	90.16
Fold 9	0.50	74.29	73.02	97.87	26.09	83.64
Fold 10	0.60	80.00	88.24	84.91	64,71	86.54
Global Avg.	-	**80.57**	**82.43**	**94.84**	**40.00**	**87.84**

Pada tabel 4.7 secara konsisten menetapkan angka 0.50 dan 0.60 sebagai nilai threshold terbaik di seluruh sepuluh lipatan data. Secara akumulatif, performa rata-rata unbiased score mencatatkan Accuracy sebesar 80.57% dan F1-Score sebesar 87.84%. Angka Recall kembali mendominasi dengan capaian yang sangat

tinggi yaitu 94.84%, yang menandakan model berhasil mendeteksi hampir seluruh pasangan judul skripsi yang memiliki kemiripan. Akan tetapi, tingginya angka tersebut berdampak langsung pada nilai Precision yang tertahan di angka 82.43%.

Distribusi performa di kesepuluh lipatan data tersebut divisualisasikan lebih rinci melalui pemetaan kelas prediksi pada Gambar 4.4 untuk melihat tingkat kestabilan model.



Gambar 4. 4 Confusion Matrix Holis 10-Fold

Visualisasi pada Gambar 4.4 mengonfirmasi pola distribusi tebakan model yang selaras dengan skenario sebelumnya. Dari total 700 data uji yang dievaluasi sepanjang sepuluh outer fold, model secara agresif menebak kelas Mirip dan sukses menghasilkan True Positive sebanyak 496 pasangan dokumen. Kesalahan prediksi berupa False Negative tergolong rendah, yakni hanya terjadi pada 28 pasangan dokumen. Tingkat True Positive yang dominan inilah yang mempertahankan metrik Recall di level 94.84%.

Di sisi lain, untuk data yang sebenarnya masuk dalam kelas Tidak Mirip, model hanya mampu mengidentifikasi True Negative sebanyak 68 pasangan dokumen. Sisanya, terdapat 108 pasangan dokumen yang keliru diprediksi sebagai Mirip sehingga menyumbang angka False Positive yang cukup besar. Pola visual ini menegaskan bahwa konfigurasi rentang pencarian dan ukuran step ini membuat model cenderung kehilangan kemampuan diskriminasinya terhadap dokumen yang berdekatan secara semantik namun sebenarnya tidak serupa, sehingga akurasi keseluruhan tetap dibayangi oleh tingginya kasus salah tebak positif.

4.1.5 Skenario Penentuan *Threshold* Aplikasi

Tahap akhir dari pengujian ini adalah menetapkan satu nilai threshold mutlak yang akan diimplementasikan pada sistem aplikasi. Ekstraksi nilai ini dilakukan melalui metode Grid Search pada keseluruhan 700 pasangan data, menyisir rentang skor cosine similarity 0.05 hingga 0.95 dengan ukuran step 0.05.

Berdasarkan Grid Search performa model mengalami fluktuasi seiring dengan peningkatan nilai threshold. Keseimbangan performa tertinggi berdasarkan titik puncak metrik F1-Score ditemukan pada threshold 0.50. Rincian pergerakan metrik pada beberapa titik threshold pengujian disajikan pada Tabel 4.8.

Tabel 4. 8 Grid Search Threshold

Threshold	Acc. (%)	Prec. (%)	Rec. (%)	F1-Score (%)
0.05	74.86	74.86	100.00	85.62
0.10	74.86	74.86	100.00	85.62
...	...	...	...	...
0.45	79.86	79.24	99.05	88.04
0.50	**82.00**	**81.59**	**98.09**	**89.08**
0.55	81.86	83.36	94.66	88.65
...	...	...	...	...
0.85	55.00	99.53	40.08	57.14

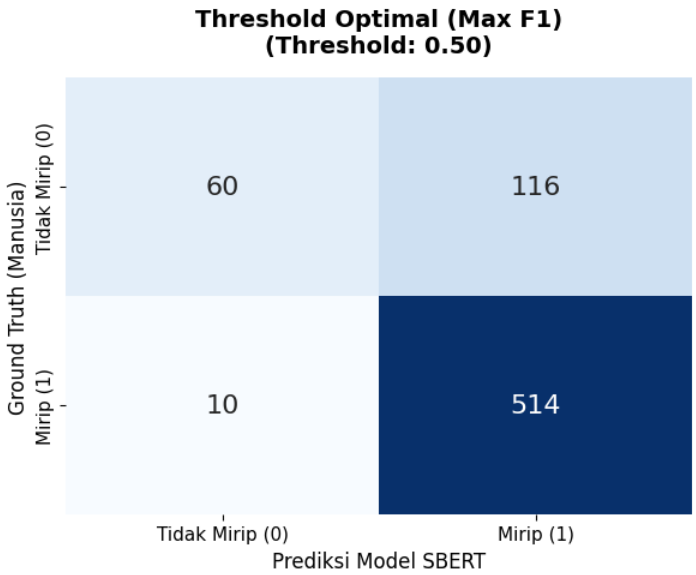
Threshold	Acc. (%)	Prec. (%)	Rec. (%)	F1-Score (%)
0.90	42.14	100.00	22.71	37.01
0.95	30.43	100.00	07.06	13.19

Untuk memastikan stabilitas model pada titik optimal tersebut, dilakukan evaluasi 5-Fold Cross-Validation secara spesifik pada threshold 0.50 yang dijabarkan pada Tabel 4.9. Hasil pengujian menunjukkan bahwa model mampu mempertahankan konsistensinya di setiap lipatan data, dengan rata-rata Accuracy mencapai 82.00% dan F1-Score sebesar 89.03%.

Tabel 4. 9 Flat 10-Fold Thres. 0.5

Fold	Acc. (%)	Prec. (%)	Rec. (%)	F1-Score (%)
Fold-1	82.14	83.74	95.37	89.18
Fold-2	79.29	77.60	98.98	87.00
Fold-3	86.43	86.05	99.11	92.12
Fold-4	82.14	80.92	100.00	89.45
Fold-5	80.00	79.51	97.00	87.39
Avg.	**82.00**	**81.56**	**98.09**	**89.03**

Kinerja klasifikasi pada *threshold* final 0.5 divisualisasikan melalui *Confusion Matrix* yang membandingkan prediksi model SBERT dengan label *ground truth* pada gambar 4.5.



Gambar 4. 5 Confusion Matrix Thres. 0.5

Berdasarkan *confusion matrix* pada Gambar 4.5, distribusi deteksi kemiripan dari total 700 pasangan judul menunjukkan bahwa sistem berhasil memprediksi secara tepat 514 pasangan judul sebagai dokumen True Positive dan 60 pasangan judul sebagai dokumen True Negative. Di sisi komputasi galat, sistem mencatatkan 116 False Positive, di mana judul diklasifikasikan mirip padahal kenyataannya berbeda, serta 10 False Negative, di mana sistem terlewat atau gagal mendeteksi kemiripan yang sebenarnya ada.

Meskipun kemunculan angka False Positive relatif menonjol, sangat minimnya jumlah False Negative menegaskan bahwa sistem memiliki kepekaan yang sangat tinggi agar tidak ada kemiripan judul yang terlewat. Oleh karena itu, nilai threshold 0.50 merupakan batas keputusan yang mampu memaksimalkan daya deteksi sekaligus menjadi titik paling optimal untuk diterapkan pada aplikasi pendeteksi kemiripan judul skripsi yang sesungguhnya.

4.2 Pembahasan

4.2.1 Pembahasan Skenario 1 - 4

Setelah melakukan pengujian pada Skenario 1, Skenario 2, Skenario 3, dan Skenario 4 menggunakan dua skema evaluasi yang berbeda yakni 5-Fold dan 10-Fold, hasil rata-rata global unbiased score dari keempat pengujian tersebut dikompilasi untuk membandingkan karakteristik performa masing-masing rentang threshold. Rekapitulasi perbandingan performa global disajikan pada Tabel 4.10.

Tabel 4. 10 Hasil Semua Skenario

Skenario	**Skema Pengujian**	**Acc. (%)**	**Prec. (%)**	**Rec. (%)**	**Specif. (%)**	**F1-Score (%)**
1 (Ghinassi)	5-Fold	82.00	83.20	95.44	42.39	88.76
2 (Ghinassi)	10-Fold	80.57	82.43	94.84	40.00	87.84
3 (Holis)	5-Fold	82.00	83.20	95.44	42.39	88.76
4 (Holis)	10-Fold	80.57	82.43	94.84	40.00	87.84

Berdasarkan Tabel 4.10, evaluasi komparatif menunjukkan bahwa skema 5-Fold secara konsisten memberikan performa yang lebih baik dibandingkan skema 10-Fold. Peningkatan jumlah fold data ke 10-Fold justru memperlihatkan sedikit penurunan performa Accuracy secara global dari 82.00% menjadi 80.57%, yang juga diikuti oleh penurunan pada metrik Precision, Recall, dan Specificity. Selain itu, hasil pengujian memperlihatkan bahwa metode pencarian Ghinassi dan Holis menghasilkan angka performa yang identik. Hal ini menandakan bahwa pada data mentah yang belum dibersihkan, algoritma cenderung mengekstraksi nilai threshold di titik yang sama terlepas dari ukuran step pencarian yang digunakan.

Meskipun pengujian 5-Fold mencatatkan F1-Score tertinggi mencapai 88.76%, kenaikan F1-Score ini murni terkatrol oleh nilai Recall yang terlampau ekstrem mencapai 95.44% akibat sistem yang meloloskan hampir semua dokumen sebagai kelas Mirip. Agresivitas ini berdampak pada kemampuan model dalam membedakan dokumen yang sebenarnya berbeda, dibuktikan dengan sangat rendahnya nilai Specificity yang hanya menyentuh 42.39%. Tingkat spesifisitas yang minim ini mengindikasikan tingginya False Positive. Rincian sampel pasangan judul yang menyumbang galat tersebut disajikan pada Tabel 4.11.

Tabel 4. 11 False Negative dari Baseline

Judul A	**Judul B**	**Label GT**	**Label SBERT**
sistem informasi mahasiswa asing dan reminder izin tinggal pada bagian kemahasiswaan universitas islam negeri maulana malik ibrahim malang	implementasi simple additive weighting pada penilaian kelayakan akreditasi program studi universitas islam negeri maulana malik ibrahim malang	0	1
rancang bangun aplikasi online untuk ujian masuk jalur reguler mandiri di universitas islam negeri uin maulana malik ibrahim malang	rancang bangun sistem informasi monitoring dan evaluasi hafalan alquran program beasiswa santri berprestasi pbsb berbasis web pada universitas islam negeri uin maulana malik ibrahim malang dengan metode extreme programming xp	0	1
decision support systems untuk mengukur tingkat adaptasi santri terhadap mahad sunan ampel alali msaa universitas islam negeri uin maulana malik ibrahim malang dengan metode fuzzy sugeno	monitoring perkembangan prestasi talim al quran mahasantri mahad sunan ampel alali malang menggunakan metode fuzzy cmeans	0	1
perancangan dan pembuatan pembangkit jadwal perkuliahan otomatis di fakultas sains dan teknologi universitas islam negeri maulana malik ibrahim malang	audit teknologi informasi di fakultas sains dan teknologi universitas islam negeri maulana malik ibrahim dengan menggunakan framework cobit 41	0	1

Berdasarkan Tabel 4.11, teridentifikasi bahwa kelemahan utama pada model baseline bermuara pada terjadinya inflasi kemiripan. Model SBERT menangkap kesamaan leksikal yang luar biasa tinggi pada deretan identitas institusi dan kalimat yang sudah disingkat. Meskipun kedua dokumen secara aktual membahas ranah studi yang tidak relevan satu sama lain, eksistensi frasa institusi yang seragam dan panjang ini mendominasi perhitungan vektor.

Merujuk pada keseluruhan analisis komparatif dan evaluasi galat tersebut, dapat disimpulkan bahwa arsitektur *pipeline baseline* memiliki kelemahan mendasar akibat adanya kata-kata noise. Oleh karena itu, penerapan pembersihan teks melalui *stopword removal* dapat diimplenetasikan untuk kedepannya. Dengan membuang atribut institusional, kata hubung dan kata yang memiliki semantic yang bisa diabaikan, representasi dimensi vektor akan dipaksa untuk berfokus murni pada objek, permasalahan, dan metodologi dari setiap judul skripsi.

4.2.2 Perbandingan Dengan Metode lain

Setelah mendapatkan performa optimal dari skenario usulan yang menggunakan model SBERT dengan mempertahankan teks asli, tahap selanjutnya adalah mengkomparasikan kinerja arsitektur tersebut dengan beberapa metode pengekstraksi fitur teks lainnya. Pengujian komparatif ini melibatkan pendekatan pencocokan berbasis leksikal dan probabilistik tradisional yaitu Bag-of-Words (BoW), TF-IDF, dan algoritma BM25 (Normalized), serta pendekatan berbasis model semantik lain yaitu Text2Vec. Seluruh metode dievaluasi menggunakan data uji yang sama untuk melihat perbandingan efektivitasnya secara objektif. Rincian hasil komparasi dari seluruh metode disajikan pada Tabel 4.12.

Tabel 4. 12 Komparasi Metode Lain

Metode	**Acc. (%)**	**Prec. (%)**	**Rec. (%)**	**Specif. (%)**	**F1-Score (%)**
Bag-of-Words	91.86	92.76	96.80	78.38	94.64
TF-IDF	89.71	93.38	92.67	81.43	93.01
BM25 (Normalized)	81.86	85.37	91.36	54.58	88.22
SBERT - Usulan	80.86	84.71	91.21	51.71	87.64
Text2Vec	74.14	75.12	97.74	5.74	84.90

Berdasarkan data pada Tabel 4.12, pendekatan leksikal tradisional mencatatkan angka performa yang secara statistik masih mengungguli metode lainnya. Bag-of-Words meraih posisi tertinggi dengan capaian Accuracy 91.86% dan F1-Score 94.64%, disusul oleh TF-IDF dengan F1-Score 93.01%. Tingginya performa metode leksikal ini dipicu oleh karakteristik dataset judul skripsi yang cenderung memiliki banyak irisan kata kunci eksak apabila membahas persoalan yang sama. Karena metode leksikal sangat bergantung pada kecocokan string secara harfiah, model-model ini mampu menebak dokumen secara akurat, terbukti dari tingkat Precision dan Specificity yang sangat tinggi pada TF-IDF yang mencapai 81.43% ketika mendapati dua dokumen yang kosakatanya tidak saling beririsan.

Secara lebih mendalam, superioritas angka pada pendekatan leksikal ini wajib dianalisis melalui hulu pembentukan data. Eksperimen pada tahapan awal konstruksi dataset pencarian dan penentuan pasangan kandidat untuk kelas dokumen positif dilakukan dengan memanfaatkan bantuan metode TF-IDF. Implikasinya, dataset yang terbentuk secara inheren sudah terfilter dan didominasi oleh pasangan judul yang memiliki karakteristik tumpang tindih kata kunci secara eksplisit. Karakteristik data yang kaya akan kesamaan string matching inilah yang

menjadi alasan utama mengapa BoW, TF-IDF, dan bahkan BM25 mendulang skor evaluasi yang lebih tinggi. Format data uji sangat selaras dengan cara kerja matematis ketiga metode tersebut.

Meskipun skenario usulan SBERT berada di bawah bayang-bayang bias leksikal dataset tersebut, capaian F1-Score sebesar 87.64% dan Recall yang tetap tinggi di angka 91.21% membuktikan ketangguhan arsitektur berbasis transformer ini. SBERT tidak sekadar bergantung pada kecocokan string mentah, melainkan mengekstrak representasi vektor berdasarkan kedekatan makna kontekstual secara utuh. Dalam implementasi aplikasi di dunia nyata, ketergantungan penuh pada metode string matching seperti BoW dan TF-IDF sangat rentan, karena sistem akan mudah dikelabui oleh teknik parafrase atau penggunaan sinonim kata oleh mahasiswa. Oleh karena itu, SBERT tetap hadir sebagai solusi konseptual yang jauh lebih andal untuk tahap produksi karena kemampuannya mendeteksi kesamaan makna di balik redaksi kalimat yang berbeda.

Lebih lanjut, jika dikomparasikan dengan sesama model representasi semantik lainnya, SBERT membuktikan stabilitas yang jauh lebih baik. Algoritma Text2Vec meskipun memiliki tingkat Recall yang tinggi, mengalami anjlok yang sangat drastis pada tingkat Specificity hingga menyentuh angka 5.74%. Hal ini menandakan bahwa Text2Vec menghasilkan tingkat false positive yang sangat fatal, di mana model menganggap hampir semua judul memiliki kemiripan. Keseluruhan perbandingan ini menegaskan bahwa penggunaan model SBERT dengan mempertahankan struktur kalimat asli merupakan langkah yang tepat untuk

menghasilkan pemahaman semantik yang kuat sekaligus menjaga keseimbangan deteksi klasifikasi yang wajar.

4.2.3 Pembahasan Similarity jika Berbeda Kata Sambung

Selain evaluasi terhadap error pada dokumen yang berbeda topik, analisis lebih lanjut dilakukan untuk melihat bagaimana model merespons kalimat yang secara semantik identik namun memiliki variasi dalam struktur kata sambung dan penambahan kata non-esensial. Perbandingan nilai representasi vektor untuk fenomena ini disajikan pada Tabel 4.13.

Tabel 4. 13 Nilai Vektor pada Variasi Struktur dan Kalimat Sambung

Judul	perang tank dengan menggunakan algoritma fuzzy sugeno sebagai pengatur perilaku npc	game perang tank dengan menggunakan algoritma fuzzy sugeno untuk mengatur perilaku npc
Vector	[**0.0186**,0.0517, -0.0049,0.0097,0.0274,…...]	[**0.0191**,0.0578, -0.0033,0.0341,0.0242,……]

Berdasarkan hasil komputasi pada Tabel 4.13, terdapat perbedaan struktur berupa penambahan kata benda di awal kalimat game serta perbedaan penggunaan klausa sambungan, yaitu frasa "sebagai pengatur" dibandingkan dengan "untuk mengatur". Meskipun terdapat variasi redaksional tersebut, model menghasilkan nilai kalkulasi vektor yang sangat berdekatan, yaitu pada angka 0.0186 dan 0.0191.

Hasil ini menunjukkan bahwa model sentence embedding memiliki stabilitas yang baik dalam mengenali kesamaan makna kontekstual secara global. Pergeseran nilai yang sangat tipis tersebut membuktikan bahwa perubahan kata sambung atau penambahan kalimat penyerta sedikit mengubah arah orientasi vektor

secara signifikan, walau model tetap mampu mempertahankan fokus pada substansi inti informasi yang sama, yaitu penerapan algoritma Fuzzy Sugeno untuk perilaku NPC pada tema perang tank.

4.2.4 Integrasi Nilai Keislaman

Jika pada awal penelitian perintah mengevaluasi rekam jejak masa lalu dalam Q.S. Al-Hasyr ayat 18 dijadikan sebagai landasan motivasi untuk membangun sistem verifikasi yang akurat, maka hasil evaluasi dalam penelitian ini merupakan wujud realisasinya. Model SBERT dapat mengenali dalam mencapai akurasi optimal telah mengubah konsep ketelitian evaluasi yang sebelumnya bersifat teoretis menjadi instrumen teknologi yang nyata. Realisasi ini sejalan dengan penegakan kebenaran dalam firman Allah SWT, Q.S. Al-Hujrat ayat 13:

يٰٓاَيُّهَا النَّاسُ اِنَّا خَلَقْنٰكُمْ مِّنْ ذَكَرٍ وَّاُنْثٰى وَجَعَلْنٰكُمْ شُعُوْبًا وَّقَبَاۤىِٕلَ لِتَعَارَفُوْاۗ اِنَّ
اَكْرَمَكُمْ عِنْدَ اللّٰهِ اَتْقٰىكُمْۗ اِنَّ اللّٰهَ عَلِيْمٌ خَبِيْرٌ ١٣

"*Wahai manusia! Sungguh, Kami telah menciptakan kamu dari seorang laki-laki dan seorang perempuan, kemudian Kami jadikan kamu berbangsa-bangsa dan bersuku-suku agar kamu saling mengenal. Sungguh, yang paling mulia di antara kamu di sisi Allah ialah orang yang paling bertakwa. Sungguh, Allah Maha Mengetahui, Mahateliti*." (Q.S. Al-Hujrat : 13)

Penafsiran dalam kitab Tafsir Al-Mishbah jilid 13 untuk ayat Al-Hujrat ayat 13 memberikan penjelasan yang komprehensif mengenai prinsip kemanusiaan dan keragaman universal. Seruan *Yā ayyuha an-nās* menunjukkan bahwa pesan ayat ini ditujukan kepada seluruh umat manusia tanpa memandang batas agama, ras,

ideologi, maupun batasan geografis. Al-Qur'an kemudian menegaskan kesetaraan biologis dan anti-diskriminasi melalui frasa *Khalaqnākum min żakarin wa unṡā*, yang berarti semua manusia setara karena berasal dari rahim dan sperma yang sama, sehingga tidak ada ruang bagi rasialisme atau supremasi kasta. Keragaman tersebut semakin dipertegas lewat *Syu'ūban wa qabā'ila* sebagai keniscayaan sosiologis, di mana Tuhan sengaja mendesain dunia dengan pluralitas suku, bangsa, budaya, dan bahasa. Tujuan dari desain keragaman ini bukanlah perpecahan, melainkan *Li-ta'ārafū*, yaitu interaksi positif untuk saling mengenal, bertukar manfaat, dan berkolaborasi. Pada puncaknya, ayat ini menetapkan *Inna akramakum 'indallāhi atqākum*, sebuah meritokrasi spiritual di mana satu-satunya standar kemuliaan hakiki yang diakui adalah ketakwaan yang tecermin lewat kebersihan hati dan keluhuran perilaku (Shihab, 2002).

Jika ditarik ke dalam ruang lingkup penelitian ini, keberagaman struktur kalimat dan variasi kata dalam sebuah dokumen teks merepresentasikan pluralitas *syu'ūban wa qabā'ila* yang secara kasat mata tampak berbeda. Namun, perintah untuk saling mengenali dan menemukan titik temu *li-ta'ārafū* diimplementasikan secara komputasional melalui pemodelan Sentence-BERT. Model SBERT bekerja dengan mengekstraksi konsep utama atau *semantic embedding* dari keberagaman teks tersebut guna mengenali persamaan esensi di baliknya, tanpa mendiskriminasi perbedaan leksikal atau susunan kata semata. Pada akhirnya, standar kemuliaan yang paling objektif *inna akramakum* direpresentasikan oleh tingginya skor kedekatan semantik yang dihasilkan oleh sistem. Pendekatan ini membuktikan bahwa teknologi pemrosesan bahasa alami mampu menembus batas-batas

perbedaan redaksional, mengenali persamaan makna kalimat secara universal, dan memetakan interaksi antar-teks secara akurat dan terukur.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Penelitian ini berhasil memetakan dua hasil utama, yaitu penentuan arsitektur pipeline terbaik melalui pengujian komparatif dan ekstraksi nilai threshold untuk implementasi pada sistem aplikasi. Berdasarkan hasil evaluasi terhadap empat skenario pengujian model Sentence-BERT, Skenario 1 dan 3 yang menggunakan metode rentang dan step Ghinassi dan Holis melalui skema 5-Fold Cross Validation menjadi arsitektur pipeline yang paling ideal. Skenario 1 dan 3 menghasilkan tingkat Accuracy sebesar 82.00%, Precision 83.20%, Recall 95.44%, F1-Score 88.76%, dan specificity 42,39%. Konfigurasi ini terbukti jauh lebih stabil dan objektif dibandingkan skenario metode Holis yang mengalami kebuntuan performa akibat ukuran step yang terlalu lebar. Skenario 1 terbukti mampu memberikan titik deteksi yang luwes dan kokoh tanpa mengorbankan kepresisian sistem akibat penumpukan prediksi positif yang keliru.

Pengujian Grid Search terhadap keseluruhan 700 pasangan data berhasil mengekstrak nilai threshold sebesar 0.50 sebagai decision boundary untuk aplikasi deteksi kemiripan judul skripsi. Pada titik potong 0.50 ini, sistem mencapai performa puncak dengan tingkat Accuracy sebesar 82.00% dan F1-Score sebesar 89.03%. Tingginya nilai keseimbangan performa ini tercermin dari kemampuan sistem dalam menekan kesalahan klasifikasi negatif secara optimal, di mana model sukses mengidentifikasi 514 kasus True Positive dan 60 kasus True Negative, serta

menyisakan 116 galat False Positive dan hanya 10 galat False Negative. Angka galat negatif yang sangat minim tersebut membuktikan bahwa model memiliki kepekaan yang sangat tinggi dalam mencegah lolosnya judul skripsi yang memiliki tingkat kemiripan semantik.

5.2 Saran

Berdasarkan hasil penelitian yang diperoleh menggunakan dataset rumpun keilmuan Teknik Informatika ini, terdapat beberapa rekomendasi penting untuk pengembangan sistem di masa mendatang. Pertama, peneliti menyarankan pengembangan berikutnya memperluas cakupan ekstraksi teks analisis dengan turut melibatkan bagian abstrak, rumusan masalah, atau batasan masalah. Untuk membedakan deretan penelitian yang menggunakan metode serupa tetapi sesungguhnya memiliki konteks persoalan berbeda, sehingga penilaian kebaruan menjadi jauh lebih akurat dan terhindar dari prediksi *False Positive*. Selanjutnya, disarankan pula untuk mengeksplorasi varian model *transformer* yang dilatih khusus menggunakan korpus bahasa Indonesia, seperti IndoBERT atau IndoNLI. Pendekatan *fine tuning* spesifik pada ranah akademik lokal akan sangat membantu model dalam menangkap struktur linguistik dengan presisi yang lebih tinggi.

Mengingat pelabelan *ground truth* dari *expert / gold standard* memiliki rasio 1:2,9 yang sangat didominasi oleh data positif, penelitian selanjutnya disarankan untuk menambah jumlah anotasi dari *expert* demi keseimbangan data. Alternatif lainnya adalah memperbanyak kelas negatif menggunakan data *silver standard* melalui komputasi NLP, dengan syarat mutlak bahwa data hasil

komputasi tersebut tidak boleh dilibatkan pada bagian validasi atau uji saat melakukan tahap pemisahan data latih dan data uji.

Terakhir, karena specificity masih rendah dan nilai false negative sangat tinggi disarankan untuk menambahkan preprocessing seperti stopword removal dan lainnya agar lebih fokus dalam semantic dan menghilangkan kata atau kalimat yang semanticnya dapat dihilangkan, seperti instansi, tempat, kata sambung, dan kalimat yang sebenarnya sudah disingkat dikalimat judul.

DAFTAR PUSTAKA

Boudaa, S., Boudaa, T., & El Haddadi, A. (2024). Semantic Textual Similarity: Overview and Comparative Study between Arabic and English. *Computacion y Sistemas*, *28*(3), 1209–1228. https://doi.org/10.13053/CyS-28-3-5035

Cahyawijaya, S., Winata, G. I., Wilie, B., Vincentio, K., Li, X., Kuncoro, A., Ruder, S., Lim, Z. Y., Bahar, S., Khodra, M. L., Purwarianti, A., & Fung, P. (2021). IndoNLU: Benchmark and Resources for Evaluating Indonesian Natural Language Understanding. *EMNLP 2021 - 2021 Conference on Empirical Methods in Natural Language Processing, Proceedings*, 8875–8898. https://doi.org/10.18653/v1/2021.emnlp-main.699

Cawley, G. C., & Talbot, N. L. C. (2010). On Over-fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation. *Journal of Machine Learning Research*, *11*, 2079–2107.

Chamidah, N., Santoni, M. M., Irmanda, H. N., Astriratma, R., & Yulnelly, Y. (2022). Penilaian Esai Pendek Otomatis Berdasarkan Similaritas Semantik dengan SBERT. *Techno.Com*, *21*(4), 732–740. https://doi.org/10.33633/tc.v21i4.6758

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, *1*, 4171–4186.

Ghinassi, I., Wang, L., Newell, C., & Purver, M. (2023). Comparing neural sentence encoders for topic segmentation across domains : not your typical text similarity task. *PeerJ*. https://doi.org/10.7717/peerj-cs.1593

Hassan, R. T., & Ahmed, N. S. (2023). Evaluating of Efficacy Semantic Similarity Methods for Comparison of Academic Thesis and Dissertation Texts. *Science Journal of University of Zakho*, *11*(3), 396–402. https://doi.org/10.25271/sjuoz.2023.11.3.1120

He, H., & Garcia, E. A. (2009). Learning from Imbalanced Data. *IEEE Transactions on Big Data*, *21*(9), 1263–1284.

Holis, R. M., Utomo, P. E. P., & Hutabarat, B. F. (2025). Semantic FAQ Chatbot Using SBERT (Sentence-BERT) and Cosine Similarity for Academic Services. *Brilliance*, *5*(2), 915–922. https://doi.org/10.47709/brilliance.v5i2.7027

Irawan, B. H., Simarangkir, M. S. H., & Erlinna, E. (2021). Deteksi Kemiripan

Judul Skripsi Menggunakan Algoritma Levenshtein Distance Pada Kampus Stmik Mic Cikarang. *Edutic - Scientific Journal of Informatics Education*, *7*(2), 143–149. https://doi.org/10.21107/edutic.v7i2.10051

Israel, G. D. (1992). Determining Sample Size. *IFAS Extension*, 1–5. https://doi.org/https://doi.org/10.32473/EDIS-PD006-1992

Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., & Yih, W. (2020). Dense Passage Retrieval for Open-Domain Question Answering. In B. Webber, T. Cohn, Y. He, & Y. Liu (Eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 6769–6781). Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.emnlp-main.550

Krstajic, D., Buturovic, L. J., Leahy, D. E., & Thomas, S. (2014). Cross-validation pitfalls when selecting and assessing regression and classification models. *Journal of Cheminformatics*, 1–15. http://www.jcheminf.com/content/6/1/10

lijun Yang, Han, L., & Liu, N. (2019). *A new approach to journal co-citation matrix construction based on the number of co-cited articles in journals*. https://repository.mmu.ac.uk/articles/journal_contribution/A_new_approach_to_journal_co-citation_matrix_construction_based_on_the_number_of_co-cited_articles_in_journals/32501931

Maulida, L. (2024). *PENCARIAN JUDUL DAN PENDETEKSI KEMIRIPAN ABSTRAK TUGAS AKHIR PRODI TEKNOLOGI INFORMASI MENGGUNAKAN COSINE SIMILARITY*. Politeknik Negeri Tanah Laut.

MetricGate. (2026). *Semantic Similarity Calculator*.

Monir, S. S., Lau, I., Yang, S., & Zhao, D. (2024). *VectorSearch : Enhancing Document Retrieval with Semantic Embeddings and Optimized Search*.

Moreira, G. D. S. P., Ak, R., Osmulski, R., Xu, M., Schifferer, B., & Oldridge, E. (2025). NV-Retriever : Improving text embedding models with effective hard-negative mining. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX)* (Vol. 1, Issue 1). Association for Computing Machinery.

Pilehvar, M. T., & Camacho-Collados, J. (2020). Embeddings in Natural Language Processing: Theory and Advances in Vector Representations of Meaning. *Synthesis Lectures on Human Language Technologies*, *13*(4), 1–175. https://doi.org/10.2200/S01057ED1V01Y202009HLT047

Putra, A. P., Singgih Putri, D. P., & Wiranatha, A. K. A. C. (2025). Scientific Paper Recommendation System: Application of Sentence Transformers and Cosine Similarity Using arXiv Data. *Journal of Applied Informatics and Computing*,

9(4), 1374–1382. https://doi.org/10.30871/jaic.v9i4.9766

Ranasinghe, T., Hettiarachchi, H., Orasan, C., & Mitkov, R. (2025). *MUSTS: MUltilingual Semantic Textual Similarity Benchmark*. *2*, 331–353. https://doi.org/10.18653/v1/2025.acl-short.27

Reilly, J., Shain, C., Borghesani, V., Kuhnke, P., Vigliocco, G., Peelle, J. E., Mahon, B. Z., Buxbaum, L. J., Majid, A., Brysbaert, M., Borghi, A. M., De Deyne, S., Dove, G., Papeo, L., Pexman, P. M., Poeppel, D., Lupyan, G., Boggio, P., Hickok, G., … Vinson, D. (2024). What we mean when we say semantic: Toward a multidisciplinary semantic glossary. In *Psychonomic Bulletin and Review* (Vol. 32, Issue 1). https://doi.org/10.3758/s13423-024-02556-7

Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using siamese BERT-networks. *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, 3982–3992. https://doi.org/10.18653/v1/d19-1410

Sanjaya, A., Fauzi, I., & Uddin, M. F. (2016). Pencegah plagiasi dengan deteksi kemiripan judul skripsi. *Nusantara Oof Engineering*, *3*(2), 7–11.

Shihab, M. Q. (2002). *Tafsir Al-Mishbah Pesan, Kesan dan Keserasian Al-Qur'an* (13th ed.). Lenera Hati. https://drive.google.com/file/d/17vTZL0-HZBQa6_f3umdKeFc-K0ZQ_TlU/view?pli=1

Shihab, M. Q. (2005). *Tafsir Al-Misbahbah Pesan, Kesan dan Keserasian Al-Qur'an Jilid 14* (14th ed.). Lentera Hati.

Sunandar, D., & Muiz, A. (2025). Thesis Title Similarity Detection System Using Levenshtein Distance and Cosine Similarity. *Bit-Tech*, *8*(1), 1131–1139. https://doi.org/10.32877/bt.v8i1.2864

Sunilkumar, P., & Shaji, A. P. (2019). A Survey on Semantic Similarity. *2019 6th IEEE International Conference on Advances in Computing, Communication and Control, ICAC3 2019*, *December 2019*. https://doi.org/10.1109/ICAC347590.2019.9036843

Zha, D., Bhat, Z. P., Lai, K. H., Yang, F., Jiang, Z., Zhong, S., & Hu, X. (2023). Data-centric Artificial Intelligence: A Survey. *ACM Computing Surveys*, *57*(5). https://doi.org/10.1145/3711118

Zhang, G., Bai, B., Liang, J., Bai, K., Chang, S., Yu, M., Zhu, C., & Zhao, T. (2019). Selection Bias Explorations and Debias Methods for Natural Language Sentence Matching Datasets. In A. Korhonen, D. Traum, & L. Màrquez (Eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 4418–4429). Association for Computational Linguistics. https://doi.org/10.18653/v1/P19-1435

LAMPIRAN

Contoh Perhitungan SBERT dan Cosine Similarity

C. Preprocessing

$$T' = \{w_i \in T \mid w_i \in [a-z]\} \quad (6.1)$$

Hasilnya adalah kalimat menggunakan huruf *lower case* tanpa ada *upper case* di tiap kalimat agar tidak membingungkan system contohnya dibawah:

$T' = w_i \in T1 =$ D, e, t, e, k, s, i, , K, e, m, i, r, i, p, a, n, , L, e, v, e, s, t, e, i, n, , D, i, s, t, a, n, c, e

$T' = w_i \in [a-z] =$ d, e, t, e, k, s, i, , k, e, m, i, r, i, p, a, n, , l, e, v, e, s, t, e, i, n, , d, i, s, t, a, n, c, e

T1 = Deteksi Kemiripan Jarak Levestein = deteksi kemiripan jarak levestein

$T2 = SistemPencarianJarakHamming = sistempencarianjarakhamming$

D. Tokenize

$$Tokenize(S) = (t_1, t_2, \dots, t_m), \quad t_i \in V \quad (6.2)$$

kalimat contoh yang sudah di *case folding* akan di*tokenize* menjadi token tiap kata atau sub-kata.

$Tokenize(S) =$ ([CLS], $deteksi, kemiripan, jarak, levestein,$ [SEP]) | $\boldsymbol{t_i \in V}$

$Tokenize(S) =$ ([CLS], $deteksi, ke, \#\#mirip, \#\#an, jarak, levestein,$ [SEP])

T1 = ([CLS], deteksi, ke, ##mirip, ##an, jarak, levestein, [SEP])

T2 = ([CLS], sistem, pen, ##cari, ##an, jarak, humming, [SEP])

E. Encoding

$$e_i = E(t_i), \quad e_i \in R^d \quad (6.3)$$

$$h_i = Transformer(e_i, C) \quad (6.4)$$

Sebagai contoh kita pakai panjang 4 dimensi dan menghasilkan sebagai berikut:

Tabel 6. 1 Banyak Token Tiap Kalimat

Kalimat	Token	Matriks (M)	Ukuran
T1	8	M1	8x4
T2	8	M2	8x4

$$e_{jarak} = E(t_{jarak}) \mid \in R^4$$

$$e_{jarak} = (E_{Token} + E_{Segment} + E_{Posisi}) \mid \in R^d$$

$$e_{jarak} = (E_{jarak} + E_A + E_6) \mid \in R^4$$

$$e_{jarak} = ((0.2 \quad -0.1 \quad 0.6 \quad -0.3) + (0.1 \quad 0.1 \quad -0.2 \quad 0.2) + (0.1 \quad 0.3 \quad -0.1 \quad 0.2))$$

$$e_{jarak} = (\mathbf{0.4} \quad \mathbf{0.3} \quad \mathbf{0.2} \quad \mathbf{0.1})$$

Matrix kalimat sebelum *self-attention*

$$M_1 = \begin{pmatrix} 0.1 & 0.1 & 0.1 & 0.1 \\ 0.5 & 0.6 & 0.07 & 0.8 \\ -0.1 & -0.2 & -0.3 & -0.4 \\ -0.5 & -0.6 & -0.7 & -0.8 \\ 0.9 & 0.8 & 0.7 & 0.6 \\ \mathbf{0.4} & \mathbf{0.3} & \mathbf{0.2} & \mathbf{0.1} \\ -0.9 & 0.8 & -0.7 & 0.6 \\ 0.1 & 0.1 & 0.1 & 0.1 \end{pmatrix} M_2 = \begin{pmatrix} 0.1 & 0.1 & 0.1 & 0.1 \\ 0.7 & -0.5 & 0.1 & -0.3 \\ -0.4 & 0.9 & -0.2 & 0.5 \\ 0.6 & 0.4 & -0.8 & -0.2 \\ 0.9 & 0.8 & 0.7 & 0.6 \\ 0.4 & 0.3 & 0.2 & 0.1 \\ -0.3 & -0.1 & 0.9 & 0.7 \\ 0.1 & 0.1 & 0.1 & 0.1 \end{pmatrix}$$

Sebagai contoh pada M_1 terdapat token "jarak" = (0.4, 0.3, 0.2, 0.1) akan dibuat menjadi *vector* berkonteks melalui proses *self-attention* berikut langkahnya :

a. Menentukan *Query Weight, Key Weight* dan *Value Weight* dengan dimensi 4x4 mengikuti panjang *vector* dari model.

$$Q_w = \begin{pmatrix} 0.1873 & 0.4754 & 0.3660 & 0.2993 \\ 0.0780 & 0.0780 & 0.0290 & 0.4331 \\ 0.3006 & 0.3540 & 0.0103 & 0.4850 \\ 0.4162 & 0.1062 & 0.0909 & 0.0917 \end{pmatrix} K_w = \begin{pmatrix} 0.1521 & 0.2624 & 0.2160 & 0.1456 \\ 0.3059 & 0.697 & 0.1461 & 0.1832 \\ 0.2280 & 0.3926 & 0.0998 & 0.2571 \\ 0.2962 & 0.0232 & 0.3038 & 0.0853 \end{pmatrix}$$

$$V_w = \begin{pmatrix} 0.0325 & 0.4744 & 0.4828 & 0.4042 \\ 0.1523 & 0.0488 & 0.3421 & 0.2201 \\ 0.0610 & 0.2476 & 0.0172 & 0.4547 \\ 0.1294 & 0.3313 & 0.1559 & 0.2600 \end{pmatrix}$$

b. Menentukan *Vector Query (Q), Vector Key (K)* dan *Vector Value (V)* dari token "jarak" dengan rumus, $Q_i = e_i \,.\, Q_w \; dan \; K_i = e_i \,.\, K_w \; serta \; V_i = e_i \,.\, V_w$.

$$Q_{jarak} = (0.2000 \quad 0.2950 \quad 0.1663 \quad 0.3558)$$

$$K_{jarak} = (0.2279 \quad 0.2067 \quad 0.1806 \quad 0.1731)$$

$$V_{jarak} = (0.0838 \quad 0.2871 \quad 0.3148 \quad 0.3446)$$

c. Menentukan *Attention Score Matrix* (S) dengan hasil *dot product vector* Q dan *vector* K lalu dibagi dengan dimensi *vector*, rumusnya $S = \frac{Q.K}{\sqrt{d_v}}$. Dibawah ini Adalah table Matrix *Attention Score* dari semua token :

$$S = \frac{(0.2000 \times 0.2279) + (0.2950 \times 0.2067) + (0.1663 \times 0.1806) + (0.3558 \times 0.1731)}{2}$$

$$S = \frac{(0.0456) + (0.0609) + (0.03003) + (0.0616)}{2} = \frac{0.19813}{2} \approx \mathbf{0.0991}$$

Tabel 6. 2 Hasil Matrix Attention Score

	[CLS]	**deteksi**	**ke**	**##mirip**	**##an**	**jarak**	**levestein**	**[SEP]**
[CLS]	0.0149	0.0967	-0.0371	-0.0967	0.1120	0.0375	-0.0227	0.0149
deteksi	0.0950	0.6168	-0.2368	-0.6168	0.7132	0.2382	-0.1348	0.0950
ke	-0.0354	-0.2300	0.0886	0.2300	-0.2652	-0.0883	0.0439	-0.0354
##mirip	-0.0950	-0.6168	0.2368	0.6168	-0.7132	-0.2382	0.1348	-0.0950
##an	0.1137	0.7368	-0.2820	-0.7368	0.8550	0.2865	-0.1834	0.1137
jarak	**0.0392**	**0.2534**	**-0.0967**	**-0.2534**	**0.2949**	**0.0991**	**-0.0697**	**0.0392**
levestein	-0.0407	-0.2606	0.0978	0.2606	-0.3092	-0.1057	0.1243	-0.0407
[SEP]	0.0149	0.0967	-0.0371	-0.0967	0.1120	0.0375	-0.0227	0.0149

d. Mendapatkan *Weight Attention* (A) dengan rumus $A = Softmax(S)$

Jika dijabarkan *softmax* dari S_{jarak} dilakukan dengan cara diekspontensialkan ($e^{S_{jarak}}$) lalu dinormalisasi ($\sum e^{S_{jarak}}$). Diakhir tiap nilai eskponesial dibagi dengan hasil normalisasinya didapat $A_{jarak,i} = \frac{e_i^{S_{jarak}}}{\sum e^{S_{jarak}}}$

$$S_{jarak} = (\mathbf{0.0392} \quad \mathbf{0.2534} \quad \mathbf{-0.0967} \quad \mathbf{-0.2534} \quad \mathbf{0.2949} \quad \mathbf{0.0991} \quad \mathbf{-0.0697} \quad \mathbf{0.0392})$$

$$e^{S_{jarak}} = (e^{\mathbf{0.0392}} \quad e^{\mathbf{0.2534}} \quad e^{\mathbf{-0.0967}} \quad e^{\mathbf{-0.2534}} \quad e^{\mathbf{0.2949}} \quad e^{\mathbf{0.0991}} \quad e^{\mathbf{-0.0697}} \quad e^{\mathbf{0.0392}})$$

$$e^{S_{jarak}} = (1.0400 \quad 1.2885 \quad 0.9078 \quad 0.7762 \quad 1.3430 \quad 1.1042 \quad 0.9327 \quad 1.0400)$$

$$\sum e^{S_{jarak}} = 1.0400 + 1.2885 + 0.9078 + 0.7762 + 1.3430 + 1.1042 + 0.9327 + 1.0400$$

$$\approx \mathbf{8.4}$$

$$A_{jarak,[CLS]} = \frac{1.0400}{8.4} \approx 0.1233 \rightarrow A_{jarak,[SEP]}$$

$$A_{jarak} \approx (0.1233 \quad 0.1528 \quad 0.1077 \quad 0.0921 \quad 0.1593 \quad 0.1309 \quad 0.1106 \quad 0.1233)$$

e. Untuk menemukan *vector* kontekstual untuk token "jarak" dilakukan dengan *vector* dari A_{jarak} *dot product* dengan *Matrix Value*, rumusnya $H = A_{jarak} . V$

Tabel 6. 3 Vector A dengan Matrix Value

	$\mathbf{A_{jarak}}$	**Matrix Value**
[CLS]	0.1233	0.0375 0.1102 0.0998 0.1339
deteksi	0.1528	0.2539 0.7048 0.5834 0.8604
ke	0.1077	-0.1038 -0.2640 -0.1842 -0.3248
##mirip	0.0921	-0.2539 -0.7048 -0.5834 -0.8604
##an	0.1593	0.2715 0.8381 0.8138 1.0141
jarak	0.1309	0.0838 0.2871 0.3148 0.3446
levestein	0.1106	0.1275 -0.3625 -0.0794 -0.3500
[SEP]	0.1233	0.0375 0.1102 0.0998 0.1339

Melakukan *dot product vector* A_{jarak} dengan *Matrix Value* yang akan menghasilkan vector kontekstual berdimensi 4, caranya dibawah ini :

$$h_1 = \begin{pmatrix} 0.1233 & 0.1528 & 0.1077 & 0.0921 & 0.1593 & 0.1309 & 0.1106 & 0.1233 \\ 0.0375 & 0.2539 & -0.1038 & -0.2539 & 0.2715 & 0.0838 & 0.1275 & 0.0375 \\ 0.0046 & 0.0388 & -0.0112 & -0.0234 & 0.0432 & 0.0110 & 0.0141 & 0.0046 \end{pmatrix} \times$$

$$h_1 = 0.0046 + 0.0388 + (-0.0112) + (-0.0234) + 0.0432 + 0.0110 + 0.0141 + 0.0046)$$

$$\approx \mathbf{0.0818}$$

Cara atas diteruskan degan 3 kolom lainnya, lalu akan menghasilkan :

$$h_{jarak} = (\mathbf{0.818} \quad \mathbf{0.1726} \quad \mathbf{0.2023} \quad \mathbf{0.2183})$$

Semua cara diatas dilakukan terus sampai didapatkan semua *vector* kontekstual dari tiap token. Hasilnya M_1 dan M_2 kontekstual dibawah ini :

$$M_1 = \begin{pmatrix} 0.0664 & 0.1213 & 0.1595 & 0.1567 \\ 0.1156 & 0.2841 & 0.2960 & 0.3522 \\ 0.0332 & 0.0159 & 0.0705 & 0.0297 \\ -0.0065 & -0.1060 & -0.0332 & -0.1177 \\ 0.1261 & 0.3209 & 0.3265 & 0.3962 \\ \mathbf{0.0818} & \mathbf{0.1726} & \mathbf{0.2023} & \mathbf{0.2183} \\ 0.0309 & 0.0015 & 0.0600 & 0.0132 \\ 0.0664 & 0.12137 & 0.1595 & 0.1567 \end{pmatrix} \quad M_2 = \begin{pmatrix} 0.0840 & 0.2455 & 0.2514 & 0.2871 \\ 0.0843 & 0.2492 & 0.2534 & 0.2913 \\ 0.0828 & 0.2395 & 0.2482 & 0.2798 \\ 0.0800 & 0.2359 & 0.2451 & 0.2736 \\ 0.1077 & 0.3134 & 0.3015 & 0.3764 \\ 0.0891 & 0.2605 & 0.2617 & 0.3069 \\ 0.0910 & 0.2628 & 0.2638 & 0.3109 \\ 0.0840 & 0.2455 & 0.2514 & 0.2871 \end{pmatrix}$$

F. Pooling

Dengan 2 matrix hasil *encoding* dilakukan proses *pooling* untuk mendapatkan *vector* representasi dari kalimat, contoh proses pooling :

$$M_1 = \frac{\begin{pmatrix} 0.0664 & 0.1213 & 0.1595 & 0.1567 \\ 0.1156 & 0.2841 & 0.2960 & 0.3522 \\ 0.0332 & 0.0159 & 0.0705 & 0.0297 \\ -0.0065 & -0.1060 & -0.0332 & -0.1177 \\ 0.1261 & 0.3209 & 0.3265 & 0.3962 \\ \mathbf{0.0818} & \mathbf{0.1726} & \mathbf{0.2023} & \mathbf{0.2183} \\ 0.0309 & 0.0015 & 0.0600 & 0.0132 \\ 0.0664 & 0.12137 & 0.1595 & 0.1567 \end{pmatrix}}{\frac{(0.5139 \quad 0.9326 \quad 1.2431 \quad 1.2093)}{8}} +$$

$$\boldsymbol{M_1} = (\mathbf{0.642} \quad \mathbf{0.1166} \quad \mathbf{0.1554} \quad \mathbf{0.1512})$$

$$M_2 = \frac{\begin{pmatrix} 0.0840 & 0.2455 & 0.2514 & 0.2871 \\ 0.0843 & 0.2492 & 0.2534 & 0.2913 \\ 0.0828 & 0.2395 & 0.2482 & 0.2798 \\ 0.0800 & 0.2359 & 0.2451 & 0.2736 \\ 0.1077 & 0.3134 & 0.3015 & 0.3764 \\ 0.0891 & 0.2605 & 0.2617 & 0.3069 \\ 0.0910 & 0.2628 & 0.2638 & 0.3109 \\ 0.0840 & 0.2455 & 0.2514 & 0.2871 \end{pmatrix}}{\frac{(0.8555 \quad 1.0376 \quad 1.4945 \quad 1.3392)}{8}} +$$

$$\boldsymbol{M_2} = (\mathbf{0.1069} \quad \mathbf{0.1297} \quad \mathbf{0.1868} \quad \mathbf{0.1674})$$

G. Cosine Similarity

Untuk cosine similarity hasil yang didapat perlu 2 vektor contoh di pooling. Cara perhitungannya ada dibawah ini:

Dot Product :

$$v_1 \cdot v_2 = \begin{pmatrix} 0.0642 & 0.1166 & 0.1554 & 0.1512 \\ 0.1069 & 0.1297 & 0.1868 & 0.1674 \end{pmatrix} \times \begin{matrix} \\ (0.006863 & 0.015122 & 0.029011 & 0.025305) \end{matrix}$$

$$v_1 \cdot v_2 = 0.006863 + 0.015122 + 0.029011 + 0.025305 \approx \mathbf{0.0763}$$

Magnitude :

$$\|v_1\|^2 = \sqrt{0.0642^2 + 0.1166^2 + 0.1554^2 + 0.1512^2}$$

$$\|v_1\|^2 \approx \sqrt{0.004122 + 0.013596 + 0.024149 + 0.022861}$$

$$\|v_1\|^2 \approx \sqrt{0.064728} \approx \mathbf{0.2544}$$

$$\|v_1\|^2 = \sqrt{0.1069^2 + 0.1297^2 + 0.1868^2 + 0.1674^2}$$

$$\|v_1\|^2 \approx \sqrt{0.011428 + 0.016822 + 0.034894 + 0.028023}$$

$$\|v_1\|^2 \approx \sqrt{0.091167} \approx \mathbf{0.3019}$$

$$\mathrm{CosS}(v_1, v_2) = \frac{v_1 \cdot v_2}{|v_1|\,|v_2|} = \frac{0.076301}{0.2544\,.\ 0.3019}$$

$$\mathrm{CosS}(v_1, v_2) = \frac{0.076301}{0.076899} \approx \mathbf{0.9922}$$