# PENERAPAN *SENTENCE-BERT* DALAM PENGEMBANGAN SISTEM INFORMASI MONITORING TEMA PENELITIAN DI UIN AR-RANIRY BANDA ACEH

**TUGAS AKHIR**

**Diajukan oleh:**
**RIFKA NURAINI**
**NIM. 190705032**
**Mahasiswa Fakultas Sains dan Teknologi**
**Program Studi Teknologi Informasi**




**PROGRAM STUDI TEKNOLOGI INFORMASI**
**FAKULTAS SAINS DAN TEKNOLOGI**
**UNIVERSITAS ISLAM NEGERI AR-RANIRY**
**BANDA ACEH**
**2026 M/1447 H**

# LEMBAR PERSETUJUAN

# PENERAPAN *SENTENCE-BERT* DALAM PENGEMBANGAN SISTEM INFORMASI MONITORING TEMA PENELITIAN DI UIN AR-RANIRY BANDA ACEH

**TUGAS AKHIR**

Diajukan Kepada Fakultas Sains dan Teknologi
Universitas Islam Negeri (UIN) Ar-Raniry Banda Aceh
Sebagai Salah Satu Beban Studi Memperoleh Gelar Sarjana
pada Prodi Teknologi Informasi


Oleh:
**RIFKA NURAINI**
**NIM. 190705032**
**Mahasiswa Fakultas Sains dan Teknologi**
**Program Studi Teknologi Informasi**


Disetujui untuk Dimunaqasyahkan Oleh:

| Pembimbing I, | Pembimbing II, |
|---|---|
| **Khairan AR, M.Kom** | **Malahayati, M.T** |
| **NIP. 198607042014031001** | **NIP. 198301272015032003** |

Mengetahui,
Ketua Program Studi Teknologi Informasi

**Malahayati, M.T**
**NIP. 198301272015032003**

جامعة الرانري
AR-RANIRY

## LEMBAR PENGESAHAN

# PENERAPAN *SENTENCE-BERT* DALAM PENGEMBANGAN SISTEM INFORMASI MONITORING TEMA PENELITIAN DI UIN AR-RANIRY BANDA ACEH

**TUGAS AKHIR**

Telah Diuji Oleh Panitia Ujian Munaqasyah Tugas Akhir
Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh dan Dinyatakan
Lulus Serta Diterima Sebagai Salah Satu Beban Studi Program Sarjana (S-1)
Dalam Prodi Teknologi Informasi

Pada Hari/Tanggal: Senin, 14 April 2026
25 Syawal 1447 H

di Darussalam, Banda Aceh
Panitia Ujian Munaqasyah Tugas Akhir

| Ketua, | Sekretaris, |
|---|---|
| **Khairan AR, M.Kom** | **Malahayati, M.T** |
| **NIP. 198607042014031001** | **NIP. 198301272015032003** |
| Penguji I, | Penguji II, |
| **Dr. Hendri Ahmadian, S.Si., M.I.M** | **Ridha Ilahi, M.T** |
| **NIP. 198301042014031002** | **NIP. 19790532014031001** |

Mengetahui,
Dekan Fakultas Sains dan Teknologi
UIN Ar-Raniry Banda Aceh

**Prof. Dr. Ir. Muhammad Dirhamsyah, M.T., IPU**
**NIP. 196210021988111001**

# LEMBAR PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini:

| | |
|---|---|
| Nama | : Rifka Nuraini |
| NIM | : 190705032 |
| Program Studi | : Teknologi Informasi |
| Fakultas | : Sains dan Teknologi |
| Judul Tugas Akhir | : PENERAPAN *SENTENCE-BERT* DALAM PENGEMBANGAN SISTEM INFORMASI MONITORING TEMA PENELITIAN DI UIN AR-RANIRY BANDA ACEH |

Dengan ini menyatakan bahwa dalam penulisan tugas akhir ini, saya:

1. Tidak menggunakan ide orang lain tanpa mampu mengembangkan dan mempertanggungjawabkan;
2. Tidak melakukan plagiasi terhadap naskah orang lain;
3. Tidak menggunakan karya orang lain tanpa menyebutkan sumber asli atau tanpa izin pemilik karya;
4. Tidak memanipulasi dan memalsukan data;
5. Mengerjakan sendiri karya ini dan mampu mempertanggung jawab atas karya ini;

Bila kemudian hari ini ada tuntutan dari pihak lain atas karya saya, dan telah melalui pembuktian yang dapat mempertanggungjawabkan dan ternyata memang ditemukan bukti bahwa saya telah melanggar pernyataan ini, maka saya siap dikenakan sanksi berdasarkan aturan yang berlaku di Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh. Demikian pernyataan ini saya buat dengan sesungguhnya dan tanpa paksaan dari pihak manapun.

Banda Aceh, 16 Februari 2026

Yang menyatakan




(Rifka Nuraini)

# ABSTRAK


| | |
|---|---|
| Nama | : Rifka Nuraini |
| NIM | : 190705032 |
| Program Studi | : Teknologi Informasi |
| Judul | : PENERAPAN *SENTENCE-BERT* DALAM PENGEMBANGAN SISTEM INFORMASI MONITORING TEMA PENELITIAN DI UIN AR-RANIRY BANDA ACEH |
| Tanggal Sidang | : 23 Februari 2026 |
| Jumlah Halaman | : 81 Halaman |
| Pembimbing I | : Khairan AR, M.Kom. |
| Pembimbing II | : Malahayati, M.T. |



Monitoring tema penelitian di UIN Ar-Raniry masih dilakukan secara manual sehingga proses pengecekan kesamaan judul membutuhkan waktu lama dan berpotensi terjadinya duplikasi tema. Oleh karena itu, penelitian ini bertujuan mengembangkan sistem informasi monitoring tema penelitian dengan menerapkan metode *Sentence*-BERT *Multilingual* berbasis *deep learning* untuk menganalisis tingkat kemiripan judul penelitian secara otomatis. Model *Sentence*-BERT *Multilingual* digunakan untuk menghasilkan representasi vektor kalimat (*Sentence embedding*) yang mampu menangkap makna semantik teks. Selanjutnya, tingkat kemiripan dihitung menggunakan metode *Cosine Similarity* berdasarkan hasil *embedding* yang diperoleh. Sistem dikembangkan berbasis web dengan menggunakan data judul penelitian mahasiswa sebagai bahan pengujian. Hasil penelitian menunjukkan bahwa penerapan metode deep learning melalui *Sentence*-BERT *Multilingual* mampu memberikan pengukuran kemiripan yang lebih akurat secara semantik dibandingkan metode pencocokan teks konvensional. Sistem yang dikembangkan dapat membantu proses monitoring tema penelitian menjadi lebih efektif, cepat, dan sistematis. Dengan adanya sistem ini, diharapkan dapat mendukung proses pengelolaan data penelitian di lingkungan akademik serta membantu mahasiswa dan pihak program studi dalam meminimalkan terjadinya kesamaan tema penelitian.



Kata Kunci: *Deep Learning, Sentence-BERT Multilingual,* Sistem Informasi, Monitoring Tema Penelitian

# *ABSTRACT*


| | |
|---|---|
| *Name* | : Rifka Nuraini |
| *Student ID* | : 190705032 |
| *Department* | *: Information Technology* |
| *Title* | *: Implementation of Sentence-BERT in the Development of a Research Theme Monitoring Information System at UIN Ar-Raniry Banda Aceh* |
| *Date* | *: 23th of February, 2026* |
| *Number of pages* | *:* 81 *Pages* |
| *Supervisor I* | : Khairan AR, M.Kom. |
| *Supervisor II* | : Malahayati, M.T. |



*Research theme monitoring at UIN Ar-Raniry is still conducted manually, making the process of checking title similarity time-consuming and prone to duplication of research topics. Therefore, this study aims to develop a research theme monitoring information system by applying the Multilingual Sentence-BERT method based on deep learning to automatically analyze the similarity level of research titles. The Multilingual Sentence-BERT model is used to generate sentence vector representations (sentence embeddings) that can capture the semantic meaning of text. Furthermore, the similarity level is calculated using the Cosine Similarity method based on the obtained embeddings. The system is developed as a web-based application using student research title data as testing material. The results show that the application of deep learning through Multilingual Sentence-BERT is able to provide more semantically accurate similarity measurements compared to conventional text matching methods. The developed system can assist the research theme monitoring process to become more effective, faster, and systematic. With this system, it is expected to support the management of research data in academic environments and help students and study programs minimize the occurrence of similar research topics.*



***Keywords:*** *Deep Learning, Multilingual Sentence-BERT, Information System, Research Theme Monitoring*

# KATA PENGANTAR

Puji syukur penulis panjatkan atas kehadirat Allah SWT yang telah memberikan rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan tugas akhir ini yang berjudul "**PENERAPAN *SENTENCE-BERT* DALAM PENGEMBANGAN SISTEM INFORMASI MONITORING TEMA PENELITIAN DI UIN AR-RANIRY** ". Shalawat beserta salam semoga tersampaikan kepada Rasulullah SAW berserta keluarga dan sahabat beliau sekalian yang telah memperjuangkan umat Islam kepada jalan kebenaran dengan dibekali ilmu yang bermanfaat untuk dunia dan akhirat.

Penyusunan tugas akhir ini diajukan sebagai salah satu syarat untuk menyelesaikan tugas akhir pada Program Studi Teknologi Informasi, Fakultas Sains danTeknologi, UIN Ar-Raniry. Dalam penulisan tugas akhir ini, penulis dengan segala kerendahan hati ingin mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Ibunda Cut Rosnita dan Ayahanda Alm. Nurun Saputra serta keluarga tercinta yang selalu memberikan doa dan dukungan yang tiada hentinya.
2. Ibu Malahayati, M.T. selaku Ketua Prodi Teknologi Informasi dan juga Pembimbing Tugas Akhir, yang telah memberikan bimbingan dan arahan dalam menyelesaikan tugas akhir ini.
3. Bapak Khairan AR, M.Kom sebagai Pembimbing Tugas Akhir yang senantiasa memberikan arahan dan motivasi.
4. Ibu Cut Ida Rahmadiana, S.Si selaku Staff Prodi Teknologi Informasi, yang senantiasa membantu penulis dalam pemberkasan administrasi.
5. Bapak Prof. Dr. Ir. M. Dirhamsyah, M.T., IPU selaku Dekan Fakultas Sains dan Teknologi UIN Ar-Raniry.
6. Bapak dan Ibu dosen Prodi Teknologi Informasi yang telah membekali penulis dengan ilmu pengetahuan di bidang Teknologi Informasi.
7. Sahabat dan teman-teman yang selalu memberikan dukungan moral untuk menyelesaikan penyusunan tugas akhir ini.

8. Pihak-pihak terkait lainnya yang membantu penulis dalam penyusunan tugas akhir ini.

Penulis menyadari sepenuhnya bahwa dalam penyusunan tugas akhir ini tidak cukup dikategorikan sempurna, untuk itu penulis dengan segala kerendahan hati menerima saran dan kritikan guna menyempurnakan penyusunan tugas akhir ini. Akhir kata, semoga tugas akhir ini dapat bermanfaat bagi penulis dan pembaca dan semoga dicatat sebagai sebuah amal kebaikan oleh Allah SWT.

Amin Ya Rabbal A'lamin.

Banda Aceh, 16 Februari 2026
Penulis

(Rifka Nuraini)

# DAFTAR ISI

LEMBAR PERSETUJUAN ..... ii
LEMBAR PENGESAHAN ..... iii
LEMBAR PERNYATAAN KEASLIAN ..... iii
ABSTRAK ..... iv
*ABSTRACT* ..... ivi
KATA PENGANTAR ..... viii
DAFTAR ISI ..... ix
DAFTAR GAMBAR ..... xii
DAFTAR TABEL ..... xiii
BAB I PENDAHULUAN ..... 1
1.1 Latar Belakang ..... 1
1.2 Rumusan Masalah ..... 3
1.3 Tujuan Penelitian ..... 3
1.4 Manfaat Penelitian ..... 3
1.5 Batasan Masalah ..... 4
BAB II TINJAUAN PUSTAKA ..... 5
2.1 Sistem Informasi Monitoring Penelitian ..... 5
2.2 *Deep Learning* ..... 5
2.3 *Natural Language Processing* (NLP) ..... 8
2.4 *Sentence Embedding* ..... 9
2.5 *Sentence-BERT Multilingual (SBERT)* ..... 10
2.6 *Cosine Similarity* ..... 13
2.7 *Python* ..... 14

2.8 *Framework Streamlit* ..... 15
2.11 Penelitian Terdahulu ..... 16
BAB III METODOLOGI PENELITIAN ..... 19
3.1 Tahapan Penelitian ..... 19
3.2 Lokasi dan Waktu Penelitian ..... 19
3.3 Identifikasi Kebutuhan ..... 20
3.4 Pengumpulan data ..... 22
3.4 Pengolahan Data (*Preprocessing*) ..... 23
3.5 Perancangan Alur Sistem ..... 25
3.6 Implementasi model *Sentence-Bert Multilingual* ..... 26
3.7 Implementasi Sistem dengan *Framework Streamlit* ..... 29
3.8 Pengujian dan Evaluasi Sistem ..... 30
BAB IV HASIL DAN PEMBAHASAN ..... 32
4.1 Hasil Pengolahan Data ..... 32
4.2 Implementasi Model *Machine learning* ..... 35
4.3 Implementasi Sistem Menggunakan *Framework Streamlit* ..... 39
4.4 Pengujian dan Evaluasi Sistem ..... 58
BAB V KESIMPULAN DAN SARAN ..... 58
5.1 Kesimpulan ..... 58
5.2 Saran ..... 59
DAFTAR PUSTAKA ..... 60

# DAFTAR GAMBAR

Gambar 2.1 *Machine Learning* ..... 6
Gambar 3.1 Tahapan Penelitian ..... 19
Gambar 3.2 Alur Sistem ..... 25
Gambar 4.1 *Import Library* ..... 40
Gambar 4.2 *Load* Data *Excel* ..... 41
Gambar 4.3 Model SBERT ..... 41
Gambar 4.4 Input Pengguna ..... 42
Gambar 4.5 Perhitungan *Similarity* ..... 43
Gambar 4.6 Visualisasi Hasil *Similarity* ..... 44
Gambar 4.7 *Download* Hasil ..... 44
Gambar 4.8 Tampilan Awal Sistem ..... 45
Gambar 4.9 Hasil Pencarian dan Perhitungan *Similarity* ..... 47
Gambar 4.10 Visualisasi Perbandingan Skor Similarity ..... 49
Gambar 4.11 Pengaturan Jumlah Hasil ..... 50
Gambar 4.12 Hasil Pencarian Berdasarkan Alfabet ..... 52
Gambar 4.13 Hasil Pencarian berdasarkan tahun ..... 53
Gambar 4.14 Hasil Pencarian berdasarkan *Similarity* ..... 54
Gambar 4.15 Grafik Perbandingan Dinamis ..... 55
Gambar 4.16 Hasil Download CSV ..... 56
Gambar 4.17 Hasil Download *Excel* ..... 57

جامعة الرانري
AR-RANIRY

# DAFTAR TABEL

Tabel 2.1 Penelitian Terdahulu ..... 17

Tabel 4.1 *Vektor Embedding* Judul Penelitian ..... 33

Tabel 4.2 Hasil *Similarity* Antar Judul ..... 34

Tabel 4.3 Hasil Setelah Penerapan *Threshold* ..... 36

Tabel 4.4 Hasil Perhitungan *Cosine Similarity* ..... 38

Tabel 4.5 Hasil Pengujian *BlackBox Testing* ..... 58

Tabel 4.6 Hasil Pengujian Waktu Respon Sistem ..... 59

Tabel 4.7 Peforma Model Sistem ..... 61

جامعة الرانري
AR-RANIRY

# BAB I
# PENDAHULUAN

## 1.1 Latar Belakang

Dalam era digital saat ini, teknologi informasi memegang peranan penting dalam mendukung berbagai aktivitas di sektor pendidikan, termasuk pengelolaan data akademik dan penelitian. Universitas Islam Negeri (UIN) Ar-Raniry Banda Aceh sebagai institusi pendidikan tinggi terus mengalami peningkatan jumlah mahasiswa dan dosen, sehingga kebutuhan terhadap sistem informasi yang efektif, akurat, dan mudah diakses semakin meningkat. Salah satu kebutuhan tersebut adalah sistem monitoring tema penelitian yang mampu menyediakan informasi secara real-time serta mendukung proses pengambilan keputusan berbasis data.

Proses pemantauan tema penelitian selama ini masih menghadapi sejumlah kendala terkait pengelolaan dan aksesibilitas data. Setiap usulan penelitian dosen memang direkap dan diverifikasi oleh Lembaga Penelitian dan Pengabdian kepada Masyarakat (LP2M) UIN Ar-Raniry sebagai dasar penyusunan Surat Keputusan (SK) pendanaan. Akan tetapi berdasarkan survei awal peneliti ke LP2M, Surat Keputusan (SK) tersebut diterbitkan dalam bentuk file PDF setiap tahun dan diunggah ke website resmi UIN Ar-Raniry tanpa dilengkapi sistem indeks maupun fitur pencarian otomatis. Format penyimpanan seperti ini mengharuskan proses penelusuran judul penelitian dilakukan secara manual. Selain itu, siste m monitoring penelitian yang tersedia saat ini juga belum dilengkapi dengan mekanisme otomatis untuk menganalisis tingkat kemiripan judul penelitian. Proses pengecekan masih dilakukan dengan membaca dokumen satu per satu, sehingga menjadi kurang efisien seiring bertambahnya jumlah data penelitian setiap tahunnya. Kondisi ini menyebabkan pencarian informasi, pengecekan kesamaan judul, serta validasi tema penelitian membutuhkan waktu lebih lama dan rawan ketidaktepatan. Akibatnya, potensi terjadinya duplikasi judul atau kesamaan tema penelitian antar dosen menjadi lebih sulit terdeteksi secara cepat dan efisien.

Perkembangan teknologi *Deep learning* memberikan peluang besar untuk meningkatkan kualitas sistem informasi monitoring penelitian. Dalam penelitian ini, *Deep learning* diterapkan secara spesifik untuk melakukan analisis kemiripan judul penelitian dengan menggunakan Model *Sentence-BERT* (*Bidirectional Encoder Representations from Transformers*), yaitu *Paraphrase Multilingual* yang digunakan untuk memahami konteks dan makna dalam judul penelitian secara lebih mendalam. Model ini menghasilkan representasi teks (*embedding*) yang kemudian dianalisis menggunakan metode *Cosine Similarity* untuk mengukur tingkat kedekatan makna antar judul. Sistem yang dikembangkan tidak melakukan analisis tren atau rekomendasi berbasis pola data, melainkan fokus pada fungsi utama yaitu deteksi kemiripan judul penelitian secara otomatis (Diana et al., 2023).

Penelitian ini memanfaatkan data judul penelitian dari LP2M UIN Ar-Raniry Banda Aceh dengan rentang tahun 2017 hingga 2024, yang diolah dan dianalisis menggunakan model *Machine learning* tersebut. Sistem monitoring tersebut akan ditampilkan dalam bentuk aplikasi web menggunakan *framework Streamlit*, karena kemampuannya menghasilkan antarmuka interaktif secara cepat, sederhana, dan mudah diintegrasikan dengan proses komputasi *Deep learning* (Mudding, 2024). Melalui sistem informasi monitoring ini, pengguna dapat memperoleh informasi tentang tingkat kemiripan judul, identitas peneliti, fakultas dan program studi, tahun penelitian, serta klaster penelitian secara langsung dan terpusat.

Dengan diterapkannya sistem informasi monitoring berbasis *Deep learning* ini, proses monitoring tema penelitian di lingkungan civitas akademika UIN Ar-Raniry Banda Aceh diharapkan menjadi lebih efektif, efisien, dan akurat. Inovasi ini juga mendukung visi UIN Ar-Raniry dalam mewujudkan institusi yang unggul dalam penelitian berbasis teknologi serta memberikan manfaat yang lebih luas bagi kebutuhan akademik maupun masyarakat.

## 1.2 Rumusan Masalah

Bagaimana menerapkan metode Deep Learning menggunakan model *Sentence-BERT Multilingual* untuk menganalisis dan mendeteksi tingkat kemiripan judul penelitian secara otomatis dalam mendukung proses monitoring tema penelitian di UIN Ar-Raniry Banda Aceh?

## 1.3 Tujuan Penelitian

Tujuan dari penelitian ini yaitu menerapkan metode *Deep learning* dengan menggunakan model *Sentence-BERT Multilingual* untuk mengembangkan sistem yang mampu menganalisis dan mendeteksi tingkat kemiripan judul penelitian secara otomatis. Melalui penerapan model tersebut, sistem yang dihasilkan diharapkan dapat mendukung proses monitoring tema penelitian di UIN Ar-Raniry Banda Aceh dengan meningkatkan efisiensi dan akurasi pengelolaan data, serta meminimalkan potensi terjadinya duplikasi judul penelitian.

## 1.4 Manfaat Penelitian

Bedasarkan tujuan penelitian ini, maka manfaat yang didapatkan dari penelitian ini sebagai berikut :

1. Memberikan solusi dalam pengembangan sistem monitoring yang lebih efisien, terintegrasi, dan dapat mengolah data secara otomatis dengan menggunakan *Deep learning*.
2. Dapat memberikan kontribusi dalam menciptakan sistem, menganalisis pola dan tingkat persamaan dalam tema data penelitian.
3. Sebagai referensi untuk penelitian lebih lanjut terkait penerapan teknologi *Deep learning* dalam bidang akademik dan administrasi di institusi pendidikan.

## 1.5 Batasan Masalah

Batasan masalah pada penelitian ini di antaranya sebagai berikut :

1. Data yang digunakan dalam penelitian ini adalah judul penelitian yang diperoleh dari LP2M UIN Ar-Raniry Banda Aceh pada rentang tahun 2017 hingga 2024 dengan total sebanyak 2.769 judul penelitian
2. Analisis kemiripan tema hanya dilakukan berdasarkan teks judul, tanpa mempertimbangkan abstrak atau konten penelitian secara keseluruhan.
3. Model *Deep learning* yang digunakan adalah model *Sentence-BERT Multilingual* yang kemudian dihitung tingkat kemiripannya menggunakan metode *Cosine Similarity*.
4. Sistem informasi akan ditampilkan menggunakan *framework Streamlit* untuk keperluan visualisasi dan interaksi pengguna.
5. Output sistem berfokus pada pemantauan tema penelitian, pendeteksian kemiripan judul, dan penyajian informasi statistik penelitian.

# BAB II
# TINJAUAN PUSTAKA

## 2.1 Sistem Informasi Monitoring Penelitian

Sistem informasi monitoring digunakan di banyak bidang penting dan telah menjadi salah satu teknologi terpenting di dunia. Sistem Informasi Monotoring adalah sistem yang digunakan untuk memproses, mengumpulkan dan menganalisis data secara real time utnuk memantau aktivitas tertentu yang memungkinkan pengambilan keputusan yang lebih baik dan lebih cepat

Perancangan sistem informasi merupakan proses pengembangan perangkat lunak yang dirancang untuk memungkinkan pengguna mengakses data dan informasi dengan cepat. Sistem ini menggantikan metode penyimpanan data yang biasanya dilakukan secara manual menggunakan dokumen fisik seperti arsip kertas dan buku catatan. Melalui pemanfaatan teknologi komputer, data disimpan dalam database digital yang sistematis dan mudah diakses kapan saja.

Proses digitalisasi ini tidak hanya mempercepat pengambilan informasi tetapi juga meminimalkan resiko kehilangan data karena kerusakan fisik atau kesalahan manusia. Sistem informasi juga memungkinkan pengelompokan data yang lebih baik dengan kemampuan pencadangan otomatis. enkripsi data untuk keamanan dan pengaturan akses terkontrol berdasarkan peran pengguna. Hal ini memungkinkan operasional agar meningkatkan produktivitas dan akurasi data (Mahmmod et al., 2024).

## 2.2 *Deep Learning*

Deep Learning merupakan salah satu cabang dari machine learning yang menggunakan jaringan saraf tiruan (*Artificial Neural Network*) dengan banyak lapisan untuk mempelajari representasi data secara otomatis. Pendekatan ini memungkinkan sistem komputer untuk mengenali pola kompleks dalam data melalui proses pembelajaran berlapis yang dilakukan secara berulang menggunakan dataset dalam jumlah besar. Berbeda dengan metode *machine*

*learning* tradisional yang biasanya memerlukan proses ekstraksi fitur secara manual, *Deep Learning* mampu melakukan proses ekstraksi fitur secara otomatis melalui lapisan-lapisan jaringan yang saling terhubung (Aggarwal, 2023).

Pada umumnya, model *Deep Learning* terdiri dari beberapa komponen utama yaitu *input layer, hidden layer,* dan *output layer*. Lapisan *input* berfungsi untuk menerima data yang akan diproses oleh model. Selanjutnya data tersebut akan diproses melalui beberapa *hidden layer* yang berfungsi untuk mengekstraksi pola dan fitur dari data. Setiap lapisan memiliki sejumlah neuron yang saling terhubung melalui bobot (*weights*) dan bias yang akan disesuaikan selama proses pelatihan model. Proses ini memungkinkan model untuk mempelajari hubungan yang kompleks antara data input dan output sehingga menghasilkan prediksi yang lebih akurat (Courville, 2022).

Proses pembelajaran dalam *Deep Learning* biasanya dilakukan melalui tahap *training, validation,* dan *testing*. Pada tahap training, model dilatih menggunakan dataset untuk mempelajari pola yang terdapat dalam data. Selama proses ini, model akan menyesuaikan parameter seperti bobot dan bias menggunakan algoritma optimasi tertentu untuk meminimalkan kesalahan prediksi atau loss. Tahap validation digunakan untuk mengevaluasi performa model selama proses pelatihan agar model tidak mengalami overfitting. Sedangkan tahap testing digunakan untuk menguji kemampuan model dalam melakukan prediksi terhadap data yang belum pernah dilihat sebelumnya (Martin, 2023).

Perkembangan *Deep Learning* telah memberikan kontribusi besar dalam berbagai bidang teknologi. Salah satu bidang yang banyak memanfaatkan teknologi ini adalah pengolahan bahasa alami (*Natural Language Processing*/NLP). NLP merupakan bidang dalam kecerdasan buatan yang berfokus pada kemampuan komputer untuk memahami, menganalisis, dan memproses bahasa manusia dalam bentuk teks maupun suara. Dengan memanfaatkan *Deep Learning*, sistem NLP mampu memahami konteks kata dalam suatu kalimat serta hubungan semantik antar kata sehingga menghasilkan pemahaman teks yang lebih akurat (Mewati Ayu, et.al, 2024).

Seiring perkembangan teknologi, berbagai arsitektur *Deep Learning* telah dikembangkan untuk meningkatkan kemampuan pemrosesan bahasa alami. Salah satu arsitektur yang paling berpengaruh adalah *Transformer*, yang diperkenalkan untuk mengatasi keterbatasan model sebelumnya seperti *Recurrent Neural Network* (RNN) dan *Long Short-Term Memory* (LSTM). Arsitektur *Transformer* menggunakan mekanisme attention yang memungkinkan model untuk memperhatikan hubungan antara setiap kata dalam suatu kalimat secara lebih efektif. Dengan mekanisme ini, model dapat memahami konteks kata secara lebih baik serta memproses data secara paralel sehingga meningkatkan efisiensi dan akurasi dalam pemrosesan teks (Afina Lina dkk, 2025).

Perkembangan model berbasis *Transformer* kemudian melahirkan berbagai model bahasa modern yang memiliki kemampuan pemahaman bahasa yang lebih baik. Salah satu model yang cukup populer adalah BERT (*Bidirectional Encoder Representations from Transformers*) yang mampu memahami konteks kata dalam kalimat secara dua arah. Model ini kemudian menjadi dasar bagi berbagai pengembangan model lain yang digunakan dalam berbagai tugas NLP seperti klasifikasi teks, pencarian informasi, analisis sentimen, serta pengukuran kemiripan teks. Dalam penelitian ini, pendekatan *Deep Learning* digunakan untuk mendukung proses analisis kemiripan teks pada judul penelitian. Dengan memanfaatkan model berbasis *Deep Learning*, sistem dapat menghasilkan representasi teks dalam bentuk vektor yang merepresentasikan makna semantik dari suatu kalimat. Representasi ini memungkinkan sistem untuk membandingkan kemiripan antar judul penelitian secara lebih efektif dibandingkan dengan metode tradisional berbasis frekuensi kata.

Salah satu model *Deep Learning* yang digunakan dalam penelitian ini adalah *Sentence-BERT* (SBERT). Model ini digunakan untuk menghasilkan *embedding* kalimat dari judul penelitian sehingga memungkinkan sistem menghitung tingkat kemiripan antar judul penelitian secara semantik menggunakan metode *Cosine Similarity*. Dengan demikian, teknologi *Deep Learning* dalam penelitian ini berperan penting dalam mendukung pengembangan sistem informasi monitoring tema penelitian di UIN Ar-Raniry Banda Aceh.

## 2.3 *Natural Language Processing* (NLP)

*Natural Language Processing* (NLP) merupakan bidang dalam kecerdasan buatan yang berfokus pada bagaimana komputer dapat memahami, mengolah, dan menghasilkan bahasa manusia secara otomatis. Dalam perkembangannya, NLP modern sangat erat dengan *Deep learning* yang mampu mempelajari pola linguistik dari data dalam skala besar. Alih-alih menggunakan aturan manual seperti pada NLP tradisional, model NLP masa kini memanfaatkan pembelajaran statistik untuk mempelajari hubungan kata, konteks kalimat, hingga makna semantik secara menyeluruh.

Proses NLP melibatkan serangkaian tahapan pemrosesan teks. Tahapan awal biasanya mencakup *text cleaning*, seperti menghilangkan karakter yang tidak diperlukan, menyamakan bentuk huruf, serta menghilangkan elemen yang tidak relevan. Setelah itu, teks diproses melalui teknik seperti tokenisasi, yaitu proses memecah kalimat menjadi unit-unit kata atau token, dan normalisasi, seperti menghapus tanda baca atau menyederhanakan bentuk kata agar seragam. Tahapan ini sangat penting karena teks mentah memiliki banyak variasi yang dapat mengganggu proses pembelajaran model jika tidak disiapkan dengan benar (Alamanda, R et al., 2021)

Pada era *machine learning*, model NLP semakin canggih melalui penggunaan arsitektur Transformer, yang diperkenalkan pertama kali oleh Vaswani et al. (2017). Transformer memungkinkan model membaca seluruh konteks kalimat secara paralel menggunakan mekanisme *self-attention*, sehingga mampu menangkap makna secara lebih detail. Salah satu model Transformer paling berpengaruh adalah BERT (Bidirectional Encoder Representations from Transformers). BERT belajar membaca kalimat dari dua arah (kiri dan kanan) secara bersamaan sehingga memiliki pemahaman konteks yang jauh lebih kuat dibandingkan model sebelumnya seperti LSTM atau word2vec.

Dalam penelitian ini, NLP digunakan untuk memproses data judul penelitian sebelum dilakukan ekstraksi fitur menggunakan model *Machine learning*. Proses seperti pembersihan teks, tokenisasi, dan normalisasi diperlukan

agar model BERT dapat memahami struktur linguistik judul penelitian secara optimal. NLP menjadi fondasi utama sebelum judul penelitian dikonversi menjadi representasi numerik melalui teknik *Sentence embedding*. Dengan penerapan NLP modern yang berbasis deep learning, sistem mampu menganalisis kemiripan judul lebih akurat dan konsisten.

## 2.4 *Sentence Embedding*

*Sentence embedding* adalah metode representasi teks yang mengubah satu kalimat utuh menjadi vektor numerik berdimensi tinggi. Vektor ini mewakili makna semantik dan konteks kalimat, sehingga memungkinkan komputer untuk membandingkan kemiripan antara satu kalimat dengan kalimat lainnya secara matematis. Teknik ini berkembang dari keterbatasan word *embedding* tradisional seperti Word2Vec atau GloVe yang hanya mampu menangkap makna kata secara individual, namun tidak dapat memahami makna keseluruhan kalimat (Fathuddin et al., 2025)

Model *Sentence embedding* modern didasarkan pada arsitektur Transformer, khususnya BERT (Bidirectional Encoder Representations from Transformers). Berbeda dengan model sebelumnya yang membaca teks secara searah, BERT memproses kalimat secara bidirectional, yaitu memahami konteks kata berdasarkan kata sebelum dan sesudahnya. Hal ini membuat BERT lebih efektif dalam memahami makna kalimat yang kompleks, terutama yang mengandung istilah teknis atau multiinterpretasi (Ansyori et al., 2025)

Dalam penelitian ini digunakan model *Paraphrase Multilingual*, yaitu model ringan berbasis BERT yang dirancang untuk menghasilkan *embedding* kalimat dengan kualitas tinggi namun tetap efisien. Model ini mendukung berbagai bahasa, termasuk bahasa Indonesia, sehingga cocok digunakan untuk analisis judul penelitian yang bersifat campuran atau *Multilingual*. *Embedding* yang dihasilkan merepresentasikan makna kalimat ke dalam bentuk vektor yang memiliki kemiripan numerik apabila makna kalimat tersebut mirip.

Penggunaan *Sentence embedding* sangat penting dalam mendeteksi kemiripan judul penelitian. Dua judul yang memiliki makna dekat, meskipun

menggunakan kata yang berbeda, tetap menghasilkan vektor yang berdekatan. Dengan demikian, teknik ini mampu mengatasi variasi bahasa dan struktur kalimat, yang sering menjadi tantangan dalam pencarian manual. Oleh karena itu, *Sentence embedding* merupakan komponen inti dalam sistem analisis kemiripan yang dikembangkan.

## 2.5 *Sentence-BERT Multilingual (SBERT)*

BERT (*Bidirectional Encoder Representations from Transformers*) merupakan model bahasa berbasis arsitektur Transformer yang dikembangkan oleh peneliti Google pada tahun 2018 dan telah memberikan kontribusi besar dalam perkembangan bidang *Natural Language Processing* (NLP). Model ini dirancang untuk melakukan proses *pre-training* representasi teks secara mendalam dengan pendekatan *bidirectional*, yaitu dengan mempertimbangkan konteks kata dari dua arah sekaligus, baik dari kiri ke kanan maupun dari kanan ke kiri dalam suatu kalimat. Pendekatan ini memungkinkan model untuk memahami makna kata secara lebih kontekstual dibandingkan dengan model sebelumnya yang hanya memproses teks secara satu arah (Devlin et al., 2021).

Salah satu keunggulan utama BERT adalah kemampuannya untuk melakukan *fine-tuning* pada berbagai tugas NLP dengan menambahkan satu lapisan *output* tambahan. Dengan pendekatan ini, model yang telah melalui proses *pre-training* dapat digunakan kembali untuk berbagai tugas seperti klasifikasi teks, analisis sentimen, pengenalan entitas bernama (*Named Entity Recognition/NER*), serta sistem tanya jawab (*question answering*) tanpa harus melatih model dari awal. Hal ini menjadikan BERT sebagai model yang fleksibel dan efisien dalam pengembangan berbagai aplikasi berbasis NLP. Arsitektur BERT-Base memiliki sekitar 110 juta parameter yang dapat dipelajari selama proses pelatihan. Model ini terdiri dari 12 lapisan Transformer, dengan setiap lapisan memiliki 768 *hidden size* serta 12 *attention heads* yang berfungsi untuk menangkap hubungan antar kata dalam suatu kalimat. Dengan arsitektur yang cukup dalam dan jumlah parameter yang besar, BERT mampu menghasilkan representasi teks yang lebih kontekstual dan bermakna sehingga dapat

meningkatkan performa berbagai tugas pemrosesan bahasa alami (Mewati Ayu dkk, 2024).

Meskipun memiliki banyak keunggulan, BERT juga memiliki beberapa keterbatasan, terutama terkait dengan kebutuhan komputasi yang cukup besar. Proses pelatihan maupun *fine-tuning* model BERT membutuhkan sumber daya komputasi yang tinggi, seperti penggunaan GPU atau TPU. Selain itu, model ini juga memerlukan dataset dalam jumlah besar agar dapat menghasilkan performa yang optimal (Hanum et al., 2024).

Untuk mengatasi keterbatasan BERT dalam menghitung kemiripan antar kalimat secara efisien, dikembangkan model *Sentence*-BERT (SBERT) yang diperkenalkan oleh Reimers dan Gurevych (2019). SBERT merupakan pengembangan dari BERT yang dirancang untuk menghasilkan *Sentence embedding*, yaitu representasi numerik dari sebuah kalimat dalam bentuk vektor. Pada model BERT standar, perbandingan dua kalimat harus dilakukan dengan memasukkan kedua kalimat tersebut secara bersamaan ke dalam model, sehingga prosesnya menjadi kurang efisien jika digunakan pada dataset yang besar.

SBERT mengatasi permasalahan tersebut dengan menggunakan pendekatan *Siamese Network* atau *Triplet Network*, yang memungkinkan setiap kalimat diproses secara terpisah untuk menghasilkan vektor *embedding*. Dengan pendekatan ini, setiap kalimat dapat diubah menjadi representasi vektor dalam ruang multidimensi yang merepresentasikan makna semantik dari kalimat tersebut. Selanjutnya, tingkat kemiripan antar kalimat dapat dihitung menggunakan metode seperti *Cosine Similarity* untuk mengetahui seberapa dekat makna antara dua kalimat (Reimers, 2020).

Keunggulan utama SBERT adalah kemampuannya menghasilkan *embedding* kalimat secara cepat dan efisien, sehingga sangat cocok digunakan pada tugas-tugas seperti *semantic search, clustering* dokumen, serta pengukuran kemiripan teks. Dengan representasi vektor yang dihasilkan, sistem dapat membandingkan makna antar kalimat secara lebih akurat dibandingkan metode tradisional berbasis frekuensi kata (Husni Thamrin dkk, 2026).

*Preprocessing* teks merupakan tahap awal dalam pengolahan data sebelum data teks diproses menggunakan model SBERT. Tahap ini merupakan langkah penting dalam pengolahan data teks yang bertujuan untuk membersihkan dan menyiapkan data agar lebih mudah dipahami oleh model. *Preprocessing* membantu mengurangi *noise* pada data serta memastikan bahwa teks yang diproses memiliki format yang lebih konsisten. Beberapa tahapan dalam *preprocessing* teks meliputi:

1. *Case Folding* (CF)

   Mengubah seluruh teks menjadi huruf kecil untuk menghindari perbedaan makna antara kata yang ditulis dengan huruf kapital dan huruf kecil.
2. *Filtering* (F)

   Menghapus elemen yang tidak relevan seperti tanda baca, angka, atau simbol yang tidak diperlukan dalam proses analisis teks.
3. *Stemming* (S)

   Mengubah kata ke bentuk dasarnya sehingga variasi kata yang memiliki makna sama dapat diproses sebagai satu bentuk kata dasar.
4. *Tokenizing* (T)

   Memecah kalimat menjadi unit-unit kata atau token yang lebih kecil sehingga memudahkan proses analisis dan pemrosesan oleh model NLP (Adine Nayla et al., 2023).

Setelah proses preprocessing dilakukan, data teks yang telah dibersihkan kemudian dapat diproses menggunakan model *Sentence*-BERT (SBERT). Dalam penelitian ini, model SBERT digunakan untuk mengubah judul penelitian menjadi representasi vektor *embedding* yang merepresentasikan makna semantik dari setiap judul. Selanjutnya, sistem akan menghitung tingkat kemiripan antar judul penelitian menggunakan metode *Cosine Similarity*. Hasil perhitungan tersebut digunakan untuk membantu proses monitoring tema penelitian serta mengidentifikasi potensi kemiripan atau duplikasi judul penelitian di lingkungan UIN Ar-Raniry Banda Aceh.

## 2.6 *Cosine Similarity*

*Cosine Similarity* merupakan metode perhitungan kemiripan antara dua vektor yang diukur berdasarkan nilai cosinus sudut yang terbentuk di antara kedua vektor tersebut. Nilai *Cosine Similarity* berada pada rentang –1 hingga 1. Namun, pada kasus representasi teks berbasis *embedding*, nilai yang dihasilkan umumnya berada pada rentang 0 hingga 1 karena vektor yang dihasilkan biasanya memiliki nilai positif. Semakin mendekati nilai 1, maka tingkat kemiripan antara dua kalimat atau teks secara semantik semakin tinggi (Sihombing et al., 2024).

Secara matematis, *Cosine Similarity* dihitung dengan membagi hasil perkalian dot *product* antara dua vektor dengan hasil kali magnitudo masing-masing vektor. Secara matematis rumus *Cosine Similarity* dapat dituliskan sebagai berikut:

$$\text{Similarity}(A, B) = \frac{A \cdot B}{\| A \| \times \| B \|}$$

Keterangan:

A = Vektor pertama

B = Vektor kedua

$A \cdot B$ = Hasil perkalian dot *product* antara vektor A dan B

||A|| = Magnitudo atau panjang vektor A

||B|| = Magnitudo atau panjang vektor B

Metode ini sangat efektif digunakan untuk membandingkan teks karena *Cosine Similarity* tidak bergantung pada panjang kalimat. Dua kalimat yang memiliki panjang berbeda tetap dapat dihitung tingkat kemiripannya selama kedua kalimat tersebut direpresentasikan dalam bentuk *embedding* yang sama. Dengan demikian, *Cosine Similarity* mampu mengukur kesamaan makna antar teks secara lebih akurat dibandingkan metode yang hanya bergantung pada pencocokan kata secara langsung (Mewati Ayub dkk, 2024).

Dalam penelitian ini, *Cosine Similarity* digunakan untuk mengukur tingkat kedekatan makna antara judul penelitian yang telah diubah menjadi *embedding* oleh model *Sentence*-BERT (SBERT). Sebagai contoh, judul

“Analisis Sistem Informasi Akademik” dan “Evaluasi Sistem Informasi Kampus” mungkin terlihat berbeda secara literal, namun secara semantik memiliki keterkaitan makna yang cukup dekat. *Cosine Similarity* dapat mendeteksi kedekatan tersebut secara efisien dibandingkan metode pencocokan kata kunci (*keyword matching*).

Metode *Cosine Similarity* dipilih karena memiliki perhitungan yang relatif sederhana, stabil, serta memiliki performa yang baik dalam mengukur kemiripan semantik antar teks. Nilai similarity yang dihasilkan juga dapat digunakan untuk menentukan ambang batas (*threshold*) tertentu sebagai indikator apakah dua judul penelitian dianggap memiliki kemiripan yang tinggi atau tidak. Dengan demikian, *Cosine Similarity* berperan sebagai mekanisme utama dalam mendeteksi potensi duplikasi atau kesamaan tema penelitian secara otomatis dalam sistem monitoring tema penelitian.

## 2.7 *Python*

Python adalah bahasa pemograman dengan tingkat tinggi yang diciptakan oleh Guido Van Rossum dan diperkenalkan pada tahun 1991. Saat ini, Python juga menjadi salah satu bahasa yang sangat diminati. Python juga adalah bahasa pemograman multifungsi yang dimanfaatkan untuk *Machine learning* dan Deep Learning. Python dipilih peneliti ini karena memiliki sintaks yang sederhana. Selain itu, juga dilengkapi dengan banyak perpustakaan dan didukung oleh komunitas yang besar, karena python adalah open source. Untuk menulis kode sumber Pyhton bisa memanfaatkan IDE seperti VS code, Sublime Text, PyCharm, atau juga dapat menggunakan IDE berbasis web seperti, Jupyter Notebook , dan Google Colab. (Alfarizi et al., 2023).

Programming yang memanfatkan interpreter untuk mengeksekusi kode programnya. Interpreter ini mampu langsung menerjemahkan kode. Python juga bisa digunakan diberbagai sistem operasi seperti Windows, Linux, dan lainnya. Python mengadopsi cara pemograman yang berasal dari berbagai bahasa lain seperti, bahasa C, pemograman objek seperti java, dan pemograman berbasis objek seperti java , dan bahasa fungsional seperti Lisp (Nazar, 2024).

## 2.8 *Framework Streamlit*

*Streamlit* merupakan *framework* berbasis Python yang digunakan untuk membangun aplikasi web interaktif dengan cepat dan mudah, terutama untuk keperluan pengembangan aplikasi berbasis data, kecerdasan buatan, dan sistem pakar. Menurut Travis (2021), *Streamlit* memungkinkan pengembang membuat antarmuka pengguna (*user interface*) tanpa perlu mempelajari bahasa pemrograman web seperti HTML, CSS, atau JavaScript. *Framework* ini sangat populer di kalangan peneliti dan pengembang karena kemampuannya menampilkan hasil komputasi dan model AI secara real-time melalui tampilan web yang sederhana namun responsif.

Keunggulan utama *Streamlit* terletak pada kesederhanaan sintaksnya, integrasi langsung dengan pustaka Python seperti Pandas, NumPy, Scikit-learn, dan TensorFlow, serta kemampuannya menampilkan komponen interaktif seperti tombol, *dropdown*, grafik, dan formulir input dengan mudah. Selain itu, aplikasi yang dibangun dengan *Streamlit* dapat dijalankan secara lokal atau dihosting secara daring, sehingga pengguna dapat mengaksesnya kapan saja melalui browser (Isna, R. H. S., & Ardiansyah, T. (2024).

Dalam penelitian ini, *Streamlit* digunakan untuk mengimplementasikan sistem pakar diagnosis kerusakan laptop berbasis metode *Forward Chaining*. *Framework* ini memudahkan peneliti dalam menampilkan antarmuka berbentuk formulir gejala yang dapat dipilih oleh pengguna, kemudian menampilkan hasil diagnosis beserta solusi yang sesuai. Dengan menggunakan *Streamlit* , sistem pakar dapat dijalankan secara efisien tanpa memerlukan proses instalasi yang rumit, serta memberikan pengalaman interaktif yang ramah bagi pengguna umum maupun teknisi pemula.

## 2.9 *Visual Studio Code (*Kode Editor*)*

Visual Studio Code merupakan editor kode yang ringan namun kuat yang di jalankan di desktop dan tersedia untuk Windows, macOS dan Linux. Muncul dengan dukungan bawaan untuk JavaScript, TypeScript dan Node.js dan

memiliki ekstensi yang banyak untuk bahasa lain seperti C ++, C #, Java, Python, PHP, dan lain sebagainya.

## 2.10 *Black Box Testing*

Black Box Testing atau yang sering disebut dengan sebutan pengujian fungsional adalah metode pengujian perangkat lunak yang digunakan untuk menguji perangkat lunak tanpa perlu diketahui struktur internal kode program. Dalam pengujian ini, penguji mengetahui apa yang harus dilakukan oleh program tetapi tidak memiliki pengetahuan tentang bagaimana melakukannya.

## 2.11 Penelitian Terdahulu

Penelitian yang terkait pada peneletian ini dibutuhkan beberapa referensi dan perbandingan literatur yang berkaitan dengan tema penelitian guna terhindar dari duplikasi dan *plagiarisme.* Penelitian terdahulu bertujuan sebagai bahan perbandingan dalam penelitian ini. Berikut ini penelitian terdahulu yang penulis gunakan sebagai referensi :

جامعة الرانري
AR-RANIRY

Table 2.1 Studi Literatur

| No | Peneliti (Tahun) | Judul Penelitian | Hasil Penelitian | Persamaan Perbedaan |
|---|---|---|---|---|
| 1. | (Sandiwarno, 2024) | Penerapan *Machine learning* untuk prediksi Bencana Banjir | Hasil analis, eksperimen, dan pengujian terhadap curah hujan dan bencana banjir menunjukan bahwa model yang digunakan yaitu *Deep Neural Investigation Network (*DNIN) lebih unggul dibandingkan dengan metode *Convolutional Neural Network* (CNN). Model DNIN yang digunakan dapat dijadikan alternatif dalam memprediksi banjir | • Pesamaan dengan peneliti ini yaitu menggunakan penerapan *Deep learning*<br>• Perbedaan dalam penelitian ini terletak pada model yang digunakan yaitu DNIN dan CNN sedangkan peneliti menggunakan model *Sentence*-BERT (SBERT) |
| 2 | (Mewati Ayub dkk, 2024) | Penerapan *Sentence* BERT Untuk Similaritas Kompetensi Pekerjaan dan Mata Kuliah | Penerapan *Sentence* BERT untuk mengukur tingkat kemiripan antara kompetensi pekerjaan dan mata kuliah secara semantik sehingga dapat membantu dalam mengidentifikasi kesesuaian kompetensi dengan materi pembelajaran. | • Persamaan dengan penelitian terletak pada penggunaan model *Sentence*-BERT untuk menghitung kemiripan teks (*similarity*).<br>• Perbedaan penelitian ini terletak pada Objek |

| | | | | |
|---|---|---|---|---|
| | | | | Penelitian. Sedangkan peneliti mengukur kemiripan judul penelitian |
| 3 | (Firmansyah, 2024) | Penggunaan Deep Lerning dalam peramalan bisnis dan prediksi pasar | Penerapan *Deep learning* dalam peramalam bisnis dan prediksi pasar memberikan manfaat besar bagi perusahaan dalam meningkatkan pengambilan keputusan , dengan memanfaatkan data historis dan algoritma *Deep learning* yang tepat, perusahaan dapat meramalkan permintaan pelanggan, penjualan, dan tren pasar dengan lebih akurat. | • Persamaan dengan penelitian ini terkait dengan penerapan *Deep learning*<br>• Perbedaan penelitian ini terletak pada fokus penelitian. Penelitian terdahulu untuk prediksi pasar Sedangkan penelitian ini untuk mengukur kemiripan judul penelitian |
| 4. | (Hanum et al., 2024) | Analisis kinerja model klasifikasi teks bert dalam | Evaluasi menunjukan model bert memiliki akurasi sebesar 77% pada data yang validasi dalam mengelompokan berita hoaks, menunjukan kinerja model *Machine learning* | • Persamaan dengan penelitian ini terkait dengan menggunakan model bert<br>• Perbedaan penelitian ini |

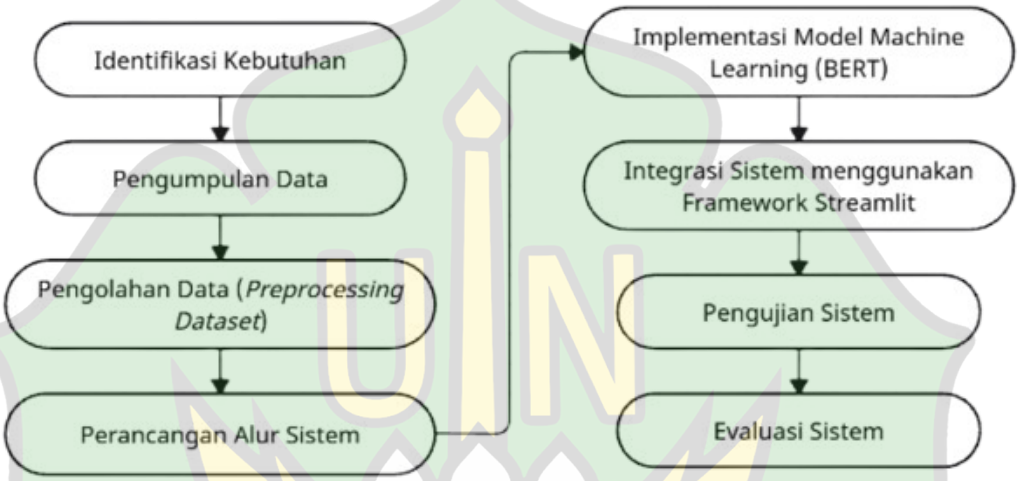
| | | | | |
|---|---|---|---|---|
| | | mendeteksi berita hoaks | dalam mengelompokan berita hoaks. | terletak pada tujuannya. Sedangkan peneliti menggunakan S-BER untuk mengukur kemiripan judul penelitian. |
| 5 | (Afina Lina dkk, 2025) | Penerapan *Sentence*-Bert dan Cosine Similarity untuk Pencarian Semantik Dokumen Skripsi dalam Format PDF | penerapan *Sentence*-BERT dan Cosine Similarity dalam penelitian ini meningkatkan akurasi pencarian semantik dokumen skripsi dalam format PDF sehingga pengguna dapat menemukan dokumen yang relevan dengan lebih efektif. | • Pesamaan dengan penelitian ini yaitu sama-sama menggunakan *Sentence*-BERT dan *Cosine Similarity*.<br>• Perbedaan penelitian ini terletak pada tujuan sistem. Penelitian terdahulu untuk pencarian dokumen skripsi, penelitian ini untuk mendeteksi kemiripan judul penelitian. |

جامعة الرانري
AR-RANIRY

# BAB III
# METODOLOGI PENELITIAN

## 3.1 Tahapan Penelitian

Tahapan penelitian dapat dilihat pada gambar 3.1 dibawah ini.




Gambar 3. 1 Tahapan Penelitian

## 3.2 Lokasi dan Waktu Penelitian

Penelitian ini dilaksanakan pada unit Lembaga Penelitian dan Pengabdian kepada Masyarakat (LP2M) UIN Ar-Raniry Banda Aceh sebagai sumber utama data judul penelitian. Proses pengumpulan data dilakukan melalui website resmi LP2M serta dokumen SK penelitian yang tersedia secara publik.

Adapun waktu pelaksanaan penelitian berlangsung mulai dari Juli 2024 hingga Januari 2025, meliputi tahap pengumpulan data, pengolahan dataset, perancangan sistem, implementasi model *machine learning*, hingga pengujian sistem.

## 3.3 Identifikasi Kebutuhan

### 3.3.1 Kebutuhan Data

Kebutuhan data dalam penelitian ini berfokus pada penyediaan informasi penelitian yang lengkap dan terstruktur sebagai dasar analisis *Machine learning*. Data diperoleh secara resmi dari LP2M UIN Ar-Raniry Banda Aceh, mencakup judul penelitian, nama dan jabatan peneliti, nomor registrasi, fakultas, program studi, klaster, serta tahun penelitian dengan rentang waktu 2017–2024. Dataset ini kemudian diolah kembali oleh peneliti melalui proses pembersihan data seperti penyesuaian format, penghapusan duplikasi, dan perapian kolom agar siap digunakan dalam proses *embedding* menggunakan model BERT. Ketersediaan data yang valid dan konsisten sangat penting untuk memastikan hasil perhitungan kemiripan judul penelitian dapat dilakukan secara akurat dan mendukung tujuan sistem monitoring yang dikembangkan.

### 3.3.2 Kebutuhan Fungsional

Kebutuhan fungsional merupakan kebutuhan yang berkaitan langsung dengan fungsi yang harus disediakan oleh sistem. Adapun kebutuhan fungsional pada sistem ini meliputi:

1. Upload Dataset Penelitian, Sistem harus menyediakan fitur untuk mengunggah file dataset penelitian dalam format *Excel* (.xlsx) yang berisi judul penelitian, nama peneliti, tahun penelitian, dan informasi pendukung lainnya.
2. Pencarian dan Deteksi Kemiripan Judul Menggunakan *Machine learning*, Sistem harus mampu menerima input judul atau tema penelitian dari pengguna, lalu menghitung tingkat kemiripan (*Similarity*) dengan menggunakan model BERT dan *Cosine Similarity*.
3. Penentuan *Threshold* Kemiripan, Sistem harus menyediakan slider *threshold* untuk menentukan batas minimal nilai *Similarity* yang dianggap relevan.

4. Visualisasi Hasil Pencarian, Sistem mampu menampilkan hasil pencarian dalam bentuk tabel yang sudah diberi highlight berdasarkan skor *Similarity*.
5. Visualisasi Grafik, Sistem harus menyediakan grafik batang (bar chart) untuk membandingkan skor *Similarity* antar judul.
6. Penggabungan Data Peneliti & Jabatan, Sistem mampu menggabungkan data peneliti dan jabatan menjadi satu baris untuk setiap judul penelitian.
7. Download Hasil Pencarian, Sistem menyediakan fitur untuk mengunduh hasil pencocokan *Similarity* dalam format CSV maupun *Excel*.
8. Penyajian Informasi Metadata Penelitian, Sistem menampilkan informasi seperti fakultas, prodi, tahun penelitian, klaster penelitian, dan nomor registrasi.

### 3.3.3 Kebutuhan Perangkat Lunak (Software)

Beberapa perangkat lunak yang di butuhkan di dalam penelitian ini adalah sebagai berikut :

a. Python sebagai bahasa pemrograman utama.
b. Pustaka Pandas untuk pengolahan dan manipulasi data.
c. Pustaka PyTorch sebagai backend model BERT.
d. *Sentence*Transformer untuk memuat model *paraphrase-Multilingual.*
e. *Framework Streamlit* untuk pembuatan antarmuka sistem.
f. Altair untuk visualisasi grafik.
g. Openpyxl untuk membaca dan menulis file *Excel*.
h. IO Bytes (BytesIO) untuk proses ekspor file.
i. Fitur cache *Streamlit* (st.cache_data dan st.cache_resource) untuk optimasi performa.

### 3.3.4 Kebutuhan Perangkat Keras (Hardware)

Beberapa perangkat keras yang di butuhkan di dalam penelitian ini adalah sebuah Laptop dengan minimal spesifikasi sebagai berikut :

- Prosesor : 2,4 GHz Dual-Core.
- RAM : 8 GB.
- SSD : 256 GB.

### 3.4 Pengumpulan data

Pengumpulan Data pada penelitian ini bersumber dari dokumen resmi yang diterbitkan oleh Lembaga Penelitian dan Pengabdian kepada Masyarakat (LP2M) UIN Ar-Raniry Banda Aceh. Data tersebut berupa Surat Keputusan (SK) pendanaan penelitian dosen yang diterbitkan setiap tahun dalam bentuk file PDF. Rentang data yang digunakan dalam penelitian ini mencakup tahun 2017 hingga 2024, yang diakses melalui website resmi UIN Ar-Raniry pada menu publikasi dokumen LP2M.

Setiap SK berisi daftar judul penelitian dosen beserta nama peneliti, fakultas, program studi, dan tahun pelaksanaan penelitian. Seluruh data tersebut kemudian diekstraksi dan dikompilasi menjadi sebuah dataset yang digunakan sebagai dasar pemrosesan teks, pembuatan *embedding*, serta perhitungan tingkat kemiripan judul penelitian menggunakan model *Machine learning*. Proses pengumpulan data dilakukan melalui metode utama yaitu Studi dokumen yang dilakukan dengan cara mengakses dan mengunduh file PDF SK pendanaan penelitian dari website resmi UIN Ar-Raniry. Setiap file SK kemudian diperiksa secara manual untuk memastikan kelengkapan data. Setelah itu, konten PDF diekstraksi menggunakan perangkat lunak pengolah teks dan dikonversi ke dalam format tabel agar dapat diolah menjadi dataset. Tahap ini mencakup:

- Mengidentifikasi SK penelitian per tahun
- Mengekstraksi judul penelitian, nama peneliti, fakultas, prodi, dan tahun
- Menyusun data mentah menjadi dataset terstruktur
- Mengubah dataset tersebut ke dalam format *Excel*

Studi dokumen ini menjadi metode utama karena seluruh informasi penelitian terdokumentasi dalam bentuk PDF tanpa sistem indeks pencarian otomatis.

## 3.4 Pengolahan Data (*Preprocessing*)

Pengolahan data atau *Preprocessing* merupakan tahap penting untuk memastikan bahwa dataset judul penelitian yang digunakan dalam proses pemodelan *Machine learning* berada dalam kondisi bersih, terstruktur, dan siap untuk dianalisis. Pada penelitian ini, proses pengolahan dilakukan melalui beberapa tahapan, mulai dari ekstraksi data dari dokumen sumber hingga pembentukan dataset akhir yang konsisten. Adapun langkah-langkah pengolahan data adalah sebagai berikut:

### 3.4.1 Ekstraksi Data PDF

Data awal berupa dokumen PDF Surat Keputusan (SK) hasil penelitian dari LP2M UIN Ar-Raniry tahun 2017–2024. Proses ekstraksi dilakukan secara manual maupun semi-otomatis menggunakan bantuan perangkat lunak pembaca PDF dan tool ekstraksi teks. Pada tahap ini, seluruh informasi penting seperti judul penelitian, nama peneliti, fakultas, dan tahun SK diambil dari dokumen asli untuk kemudian di ubah dan disesuaikan ke dalam format *Excel*. Tantangan yang sering muncul meliputi perbedaan format antar dokumen, pemisahan baris yang tidak seragam, serta karakter teks yang tidak selalu terdeteksi dengan baik, sehingga memerlukan verifikasi dan perbaikan manual.

### 3.4.2 Pembersihan Data (Data *Cleaning*)

Tahap ini dilakukan untuk memastikan kualitas data sehingga dapat diproses oleh model *Machine learning* secara optimal. Proses pembersihan mencakup:

- Menghapus duplikasi data, baik duplikasi judul penelitian yang identik maupun data ganda yang muncul akibat proses ekstraksi.
- Menyesuaikan format penulisan, seperti konsistensi huruf kapital, pemisahan kolom, penulisan nama fakultas, tahun, dan struktur data lain yang tidak seragam.
- Normalisasi teks, termasuk menghilangkan tanda baca yang tidak diperlukan, memperbaiki kesalahan pengetikan, menghapus karakter

aneh hasil OCR, serta melakukan lowercasing untuk persiapan analisis teks.

- Validasi kelengkapan data, seperti memastikan setiap entri memiliki judul, tahun, dan metadata yang relevan.

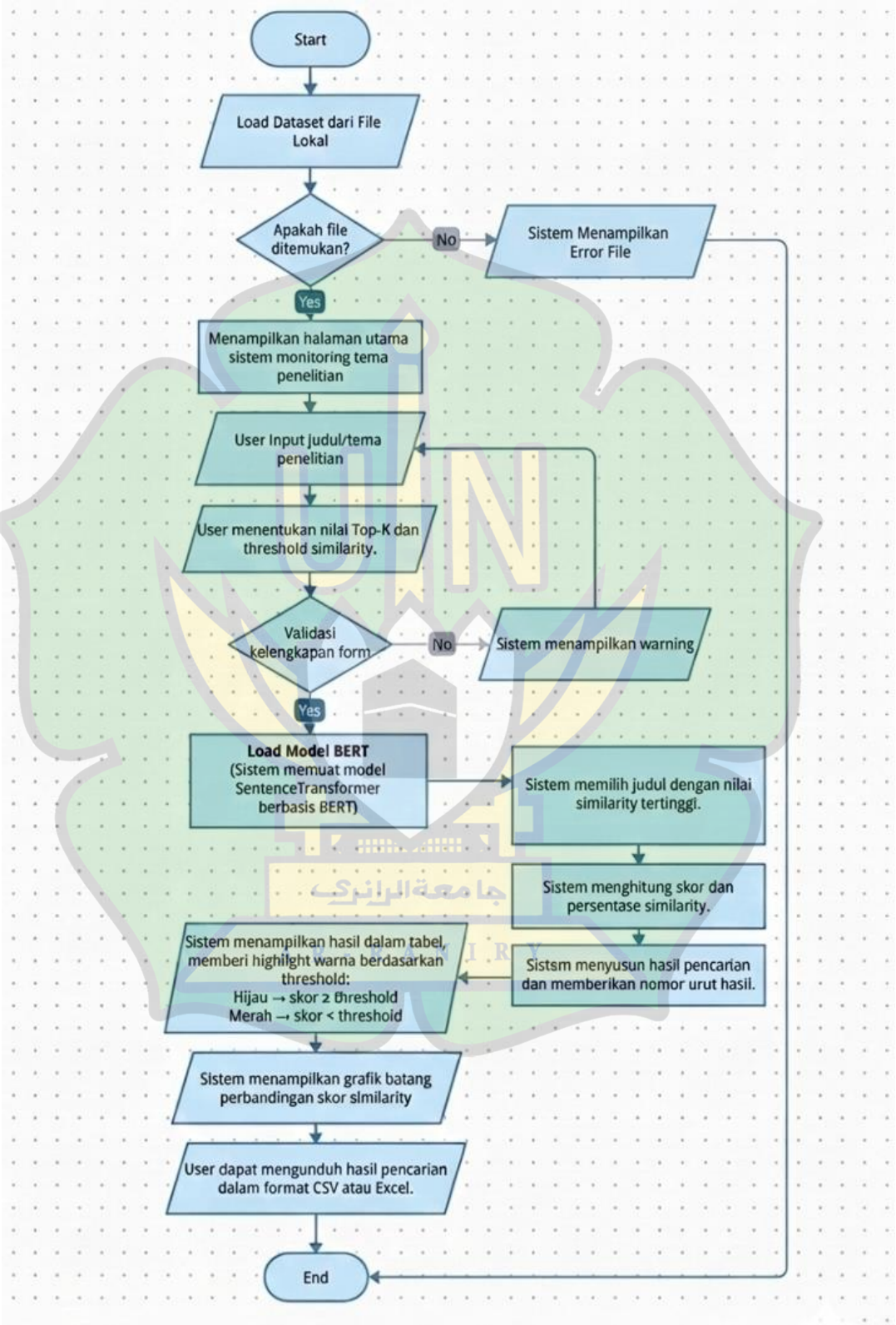
### 3.4.3 Penentuan Kolom Dataset Akhir

Setelah data bersih, tahap selanjutnya adalah menyusun struktur dataset final yang akan digunakan pada proses pemodelan. Kolom dataset yang disiapkan meliputi:

- Judul Penelitian
- Nama Peneliti
- Fakultas/Program Studi
- Tahun Penelitian / Tahun SK
- Klaster atau Kategori

Dataset final ini kemudian diekspor ke format *Excel* dan menjadi dasar untuk proses *embedding*, analisis kesamaan teks, serta pengujian model dalam mendeteksi tingkat kemiripan judul penelitian.

## 3.5 Perancangan Alur Sistem




Gambar 3. 2 Alur Sistem

Alur sistem pada *Sistem Informasi Monitoring Tema Penelitian Dosen UIN Ar-Raniry menggunakan Machine learning dengan Model BERT* diawali dengan proses pemuatan dataset penelitian dosen yang tersimpan dalam file *Excel* lokal. Sistem terlebih dahulu melakukan pengecekan terhadap keberadaan file sebagai sumber data. Apabila file tidak ditemukan, sistem akan menghentikan proses dan menampilkan pesan kesalahan. Jika file tersedia, sistem menampilkan halaman utama sistem informasi monitoring tema penelitian dosen UIN Ar-Raniry. Pengguna kemudian memasukkan tema penelitian yang ingin dianalisis serta menentukan parameter pencarian berupa nilai Top-K dan *threshold Similarity*. Sistem selanjutnya melakukan validasi input, dan apabila input tidak lengkap, sistem akan menampilkan pesan peringatan kepada pengguna.

Setelah input dinyatakan valid, sistem memuat model *SentenceTransformer* berbasis BERT sebagai metode *machine learning* yang digunakan. Sistem menghitung tingkat kemiripan (*Cosine Similarity*) antara tema penelitian yang dimasukkan pengguna dan judul penelitian dosen yang tersimpan dalam dataset. Berdasarkan hasil perhitungan tersebut, sistem memilih judul penelitian dengan nilai *Similarity* tertinggi, menghitung skor dan persentase kemiripan, serta menyusun hasil pencarian secara terurut. Hasil monitoring ditampilkan dalam bentuk tabel dengan penanda warna berdasarkan *threshold Similarity* dan divisualisasikan dalam grafik batang untuk memudahkan analisis. Pada tahap akhir, sistem menyediakan fitur pengunduhan hasil monitoring dalam format CSV atau *Excel* sebagai bahan dokumentasi dan evaluasi penelitian dosen.

## 3.6 Implementasi model *Sentence-Bert Multilingual*

Implementasi model *Deep learning* dalam penelitian ini berfokus pada pemanfaatan model *Sentence embedding* berbasis BERT untuk merepresentasikan judul penelitian ke dalam bentuk vektor numerik, yang kemudian dianalisis tingkat kemiripannya menggunakan *Cosine Similarity*.

Tahapan implementasi meliputi pemilihan model, proses pembentukan *embedding*, perhitungan kesamaan antar judul, serta penerapan *threshold* untuk menentukan batas relevansi hasil.

### 3.6.1 *Paraphrase Multilingual*

Model yang digunakan pada penelitian ini adalah *Paraphrase-Multilingual*-L12-v2, yaitu model *Sentence embedding* ringan yang dikembangkan oleh *Sentence-Transformers*. Model tersebut merupakan penyederhanaan arsitektur BERT yang memiliki jumlah parameter lebih kecil namun tetap mampu mempertahankan performa tinggi dalam tugas pemahaman teks. Keunggulan model ini adalah kemampuannya dalam memahami berbagai bahasa, termasuk bahasa Indonesia, sehingga cocok digunakan untuk memproses judul penelitian yang ditulis dalam ragam bahasa dan struktur kalimat yang bermacam-macam. Model tersebut juga menghasilkan representasi vektor berdimensi tetap untuk setiap kalimat, yang kemudian digunakan sebagai dasar perhitungan kesamaan semantik antar judul. Model *Paraphrase-Multilingual* dipilih berdasarkan beberapa pertimbangan:

1. Akurasi tinggi pada tugas *semantic Similarity* dan *paraphrase detection*.
2. Ukuran relatif kecil sehingga lebih efisien dalam proses komputasi dan lebih cepat digunakan untuk inferensi.
3. Mendukung banyak bahasa, termasuk bahasa Indonesia, tanpa perlu pelatihan ulang (*fine-tuning*).
4. Telah teruji pada domain teks pendek, seperti judul penelitian, yang membutuhkan pemahaman semantik yang mendalam.

Proses menghasilkan *Sentence embedding* dilakukan melalui beberapa langkah:

1. Judul penelitian yang sudah dibersihkan dimasukkan ke dalam model *Paraphrase-Multilingual*.
2. Model mengubah setiap judul menjadi representasi vektor berdimensi tetap (biasanya 384 atau 768 dimensi tergantung variannya).

3. Vektor-vektor ini menyimpan informasi makna sehingga judul yang mirip secara semantik akan memiliki vektor yang lebih dekat satu sama lain.
4. *Embedding* yang dihasilkan disimpan dan digunakan untuk proses analisis kesamaan menggunakan *Cosine Similarity*.

### 3.6.2 *Cosine Similarity*

Dalam penelitian ini, perhitungan tingkat kemiripan antar judul penelitian dilakukan menggunakan metode *Cosine Similarity*. Metode ini digunakan untuk mengukur seberapa dekat hubungan makna antara dua vektor *embedding* yang dihasilkan oleh model BERT. *Cosine Similarity* bekerja dengan membandingkan arah dari dua vektor, bukan besarnya nilai vektor tersebut, sehingga sangat cocok digunakan untuk data berbasis teks seperti judul penelitian. Perhitungan dilakukan secara otomatis melalui fungsi bawaan pada *Library machine learning*, sehingga proses pengukuran *Similarity* dapat dilakukan dengan cepat dan akurat tanpa memerlukan perhitungan matematis secara manual. Hasil nilai *Similarity* kemudian digunakan untuk mengidentifikasi judul-judul yang memiliki kemiripan makna, sehingga mendukung sistem dalam mendeteksi potensi duplikasi judul penelitian secara lebih efektif.

### 3.6.3 *Threshold Similarity*

*Threshold Similarity* adalah batas nilai minimum *Cosine Similarity* yang digunakan untuk menentukan apakah dua judul dianggap mirip secara semantik. Nilai *threshold* ditentukan berdasarkan eksperimen awal dan karakteristik data judul penelitian. *Threshold* berfungsi untuk menyaring hasil perbandingan. Hanya judul dengan nilai *Similarity* di atas *threshold* yang dianggap relevan dan ditampilkan kepada pengguna. Jika nilai *Similarity* berada di bawah *threshold*, judul tersebut dianggap tidak memiliki kesamaan yang cukup signifikan. Pemilihan *threshold* yang tepat sangat mempengaruhi kualitas rekomendasi:

- *Threshold* terlalu rendah → banyak judul tidak relevan ikut masuk (false positive).
- *Threshold* terlalu tinggi → judul relevan bisa terlewat (false negative).
- *Threshold* optimal menghasilkan keseimbangan antara recall dan precision sehingga sistem dapat memberikan rekomendasi kesamaan judul dengan lebih akurat dan dapat dipercaya.

Nilai *threshold* yang baik biasanya diuji melalui beberapa percobaan terhadap sampel data.

## 3.7 Implementasi Sistem dengan *Framework Streamlit*

Implementasi sistem monitoring kemiripan judul penelitian pada penelitian ini dilakukan menggunakan *framework Streamlit* . *Streamlit* dipilih karena mampu menyediakan antarmuka aplikasi berbasis web yang interaktif, sederhana, dan mudah dikembangkan tanpa memerlukan konfigurasi kompleks. *Framework* ini sangat mendukung proses integrasi antara komponen antarmuka pengguna (UI) dengan komputasi *machine learning* yang dilakukan di backend, sehingga memungkinkan model BERT dan perhitungan *Cosine Similarity* dijalankan langsung dalam aplikasi.

Pada tahap implementasi, struktur aplikasi dibangun dengan memisahkan beberapa komponen utama, yaitu pemuatan dataset, pemanggilan model *Sentence embedding* berbasis BERT, perhitungan *Cosine Similarity*, serta penampilan hasil analisis. Pengguna dapat memasukkan judul penelitian melalui form input pada antarmuka *Streamlit* , kemudian sistem secara otomatis memproses judul tersebut menjadi vektor *embedding*, melakukan pencocokan dengan seluruh data pada dataset, dan menampilkan daftar judul yang memiliki tingkat kemiripan tertinggi.

*Streamlit* juga digunakan untuk menampilkan informasi tambahan seperti nama peneliti, fakultas, program studi, tahun penelitian, dan klaster penelitian agar hasil analisis tidak hanya fokus pada *Similarity*, tetapi juga memberikan konteks yang relevan bagi pengguna. Selain itu, komponen visual

seperti tabel interaktif digunakan untuk memudahkan pengguna dalam membaca dan membandingkan hasil kemiripan.

Dengan memanfaatkan *Streamlit* , sistem dapat dijalankan secara lokal maupun dipublikasikan secara online tanpa konfigurasi server yang rumit. *Framework* ini mendukung proses pengembangan sistem monitoring yang cepat, fleksibel, dan mudah dikelola, sehingga sesuai dengan kebutuhan penelitian ini yang mengutamakan integrasi *machine learning* ke dalam sebuah aplikasi antarmuka yang mudah diakses oleh civitas akademika.

## 3.8 Pengujian dan Evaluasi Sistem

Pengujian dan evaluasi sistem dilakukan untuk memastikan bahwa aplikasi monitoring kemiripan judul penelitian dapat berjalan sesuai dengan kebutuhan dan menghasilkan keluaran yang akurat. Pada penelitian ini, pengujian dilakukan menggunakan metode Blackbox Testing, yaitu pengujian yang berfokus pada fungsi sistem tanpa melihat proses internal atau struktur kode. Pendekatan ini digunakan untuk mengevaluasi apakah setiap fitur sistem telah bekerja sebagaimana mestinya berdasarkan input dan output yang dihasilkan.

Pengujian dilakukan pada beberapa aspek utama, yaitu:

1. Pengujian Fungsionalitas Aplikasi (Blackbox Testing), Pengujian ini meliputi pemeriksaan seluruh fungsi antarmuka *Streamlit* , seperti proses input judul, pemuatan dataset, pemanggilan model *embedding*, perhitungan *Similarity*, serta penampilan hasil dalam bentuk tabel. Setiap fitur diuji dengan berbagai skenario input untuk memastikan aplikasi memberikan respon yang benar dan tidak mengalami error.
2. Pengujian Hasil Kemiripan Judul (*Similarity* Result Validation), Pengujian ini dilakukan dengan memasukkan beberapa judul penelitian yang sudah diketahui tingkat kemiripannya. Hasil *Similarity* kemudian dibandingkan dengan ekspektasi berdasarkan penilaian manual. Tujuan pengujian ini adalah menilai apakah sistem mampu memberikan hasil yang logis dan sesuai dengan kedekatan makna antar judul.

3. Evaluasi *Threshold Similarity*, Pengujian ini dilakukan untuk menentukan nilai *threshold* yang paling sesuai agar sistem hanya menampilkan hasil kemiripan yang relevan. Beberapa nilai *threshold* dicoba untuk melihat dampaknya terhadap jumlah dan kualitas hasil *Similarity* yang ditampilkan.
4. Evaluasi Performa Sistem Secara Umum, Meskipun tidak dilakukan pengukuran performa model secara kompleks, evaluasi ringan dilakukan untuk memastikan aplikasi dapat memproses judul dengan waktu respons yang wajar. Pengamatan ini mencakup waktu pemuatan model, waktu perhitungan *Similarity*, serta waktu rendering antarmuka *Streamlit* selama pengujian.

Melalui serangkaian pengujian tersebut, sistem dapat dinilai dari aspek fungsionalitas, akurasi hasil *Similarity*, serta kenyamanan pengguna. Hasil evaluasi ini menjadi dasar untuk memastikan bahwa aplikasi telah bekerja sesuai tujuan, yaitu mendeteksi kemiripan judul penelitian secara otomatis, akurat, dan efisien.

جامعة الرانري
AR-RANIRY

# BAB IV
# HASIL DAN PEMBAHASAN

## 4.1 Hasil Pengolahan Data

Tahap pengolahan data pada penelitian ini menghasilkan sebuah dataset judul penelitian yang siap digunakan dalam proses analisis kemiripan tema penelitian. Data bersumber dari dokumen Surat Keputusan (SK) penelitian LP2M UIN Ar-Raniry Banda Aceh periode tahun 2017–2024 yang tersedia dalam bentuk file PDF. Dokumen-dokumen tersebut diekstraksi dan diolah menjadi format data terstruktur melalui tahapan pembersihan dan normalisasi. Dataset akhir terdiri dari ratusan judul penelitian yang telah diseleksi dan dibersihkan dari data duplikasi serta ketidakkonsistenan format. Setiap data penelitian memiliki beberapa atribut utama, yaitu judul penelitian, nama peneliti, fakultas, program studi, tahun penelitian, dan klaster penelitian. Dataset inilah yang selanjutnya digunakan sebagai basis data dalam sistem monitoring tema penelitian.

### 4.1.1 Hasil *Embedding* Judul Penelitian

Proses utama dalam implementasi model *machine learning* adalah pembentukan *Sentence embedding* menggunakan model BERT *Paraphrase Multilingual*. Model ini berfungsi untuk mengubah judul penelitian dalam bentuk teks menjadi representasi numerik berupa vektor *embedding*.

Pada tahap ini, setiap judul penelitian yang telah dipreprocessing diproses oleh encoder BERT untuk menghasilkan vektor berdimensi tetap. Vektor ini merepresentasikan makna keseluruhan judul penelitian, bukan sekadar frekuensi kata atau kemunculan istilah tertentu. Dengan demikian, judul yang memiliki susunan kata berbeda namun konteks yang serupa dapat menghasilkan *embedding* yang saling berdekatan dalam ruang vektor.

Proses *embedding* dilakukan terhadap dua jenis data, yaitu judul penelitian dalam dataset LP2M dan judul yang dimasukkan oleh pengguna. *Embedding* dataset disiapkan terlebih dahulu dan digunakan kembali pada saat sistem dijalankan, sehingga proses pencarian kemiripan dapat dilakukan dengan lebih efisien.

### 4.1.2 Hasil Vektor / Matriks *Embedding*

Setelah data siap, setiap judul penelitian diproses menggunakan model *Sentence embedding* berbasis BERT (*Paraphrase Multilingual*) untuk menghasilkan representasi numerik. Proses ini bertujuan mengubah teks judul penelitian menjadi vektor numerik yang dapat dipahami dan dibandingkan oleh sistem *machine learning*.

Sebagai contoh, judul penelitian berikut: "*Pengembangan Sistem Informasi Akademik Berbasis Web*", diproses oleh model dan menghasilkan sebuah vektor *embedding* berdimensi 384, yang merepresentasikan makna semantik judul tersebut secara menyeluruh. Proses ini dilakukan terhadap seluruh judul dalam dataset sehingga diperoleh kumpulan *embedding* yang merepresentasikan keseluruhan data penelitian. *Embedding* yang dihasilkan tidak merepresentasikan kata per kata, melainkan makna keseluruhan kalimat, sehingga judul dengan susunan kata berbeda tetapi memiliki konteks serupa akan menghasilkan *embedding* yang saling berdekatan.

Berikut contoh sebagian kecil nilai dari vektor *embedding* yang dihasilkan ditunjukkan pada tabel 4.1:

*Tabel 4.1 Contoh Representasi Vektor Embedding Judul Penelitian*

| Dimensi ke- | Nilai *Embedding* |
|---|---|
| 1 | 0.0123 |
| 2 | -0.0845 |
| 3 | 0.2317 |
| ... | ... |
| 384 | -0.0198 |

Nilai-nilai *embedding* tersebut merupakan representasi numerik dari makna judul penelitian. Seluruh judul penelitian dalam dataset diproses dengan cara yang sama sehingga membentuk sebuah matriks *embedding*, di mana setiap baris mewakili satu judul penelitian dan setiap kolom merepresentasikan dimensi *embedding*. Setiap judul penelitian direpresentasikan dalam bentuk vektor satu dimensi dengan panjang tetap. Kumpulan seluruh vektor *embedding* tersebut membentuk sebuah matriks *embedding* yang digunakan dalam proses perhitungan *Similarity*.

### 4.1.3 Hasil *Similarity* Antar Judul

Setelah *embedding* terbentuk, sistem menghitung tingkat kemiripan antara judul penelitian yang dimasukkan oleh pengguna dengan judul-judul yang terdapat dalam dataset menggunakan metode *Cosine Similarity*.

Sebagai contoh, pengguna memasukkan judul: "*Sistem Informasi Monitoring Penelitian Berbasis Web*", Judul tersebut kemudian dibandingkan dengan seluruh judul dalam dataset. Contoh hasil perhitungan *Similarity* terhadap beberapa judul penelitian ditunjukkan pada tabel 4.2 berikut.

*Tabel 4.2 Contoh Hasil Similarity Antar Judul*

| Judul Penelitian Dataset | Nilai *Similarity* |
|---|---|
| Analisis Sistem Informasi Akademik Berbasis Web | 0.89 |
| Sistem Informasi Manajemen Akademik | 0.76 |
| Aplikasi Monitoring Kegiatan Mahasiswa | 0.41 |

Berdasarkan hasil tersebut, dapat dilihat bahwa judul dengan kemiripan makna yang tinggi menghasilkan nilai *Similarity* mendekati 1, sedangkan judul yang kurang relevan memiliki nilai *Similarity* yang lebih rendah.

## 4.2 Implementasi Model *Machine learning*

Implementasi model *machine learning* dalam penelitian ini berfokus pada penerapan model *Sentence embedding* berbasis BERT untuk mendukung proses deteksi kemiripan judul penelitian secara otomatis. Model yang digunakan adalah *Paraphrase Multilingual* yang diimplementasikan menggunakan *Library Sentence-transformers*.

Model ini digunakan tanpa proses pelatihan ulang (fine-tuning) dan langsung dimanfaatkan untuk menghasilkan *embedding* judul penelitian. Setiap judul penelitian, baik yang berasal dari dataset maupun yang dimasukkan oleh pengguna, diubah menjadi vektor *embedding* berdimensi tetap sehingga dapat dibandingkan dalam ruang vektor yang sama.

### 4.2.1 Implementasi Model BERT (*Paraphrase Multilingual*)

Implementasi model BERT dilakukan secara terintegrasi dalam sistem berbasis web menggunakan *framework Streamlit* . Model diinisialisasi satu kali pada saat aplikasi dijalankan, kemudian digunakan untuk memproses seluruh permintaan pengguna.

Ketika pengguna memasukkan judul penelitian, sistem secara otomatis memanggil model untuk menghasilkan *embedding* judul tersebut. *Embedding* ini kemudian dibandingkan dengan *embedding* dataset yang telah tersedia. Proses ini berjalan secara real-time sehingga pengguna dapat langsung memperoleh hasil pencarian kemiripan judul tanpa harus menunggu proses yang lama.

Integrasi ini menunjukkan bahwa model *machine learning* tidak hanya digunakan sebagai alat analisis, tetapi juga sebagai bagian inti dari sistem informasi monitoring penelitian.

### 4.2.2 Implementasi *Threshold*

*Threshold* kemiripan digunakan sebagai nilai ambang untuk menentukan judul penelitian yang dianggap memiliki kemiripan signifikan. *Threshold* berperan sebagai parameter penting dalam menentukan kualitas hasil yang ditampilkan oleh sistem. Nilai *threshold* digunakan untuk menyaring hasil *Similarity* sehingga hanya judul

penelitian dengan tingkat kemiripan tertentu yang ditampilkan kepada pengguna.

Dalam implementasinya, nilai *threshold* dapat disesuaikan sesuai kebutuhan pengguna. Jika *threshold* ditetapkan rendah, sistem akan menampilkan lebih banyak judul yang memiliki kemiripan rendah hingga sedang. Sebaliknya, jika *threshold* ditetapkan tinggi, sistem hanya akan menampilkan judul dengan kemiripan yang sangat tinggi. Penggunaan *threshold* ini membantu pengguna dalam melakukan validasi judul penelitian dengan lebih fleksibel dan terarah, serta mengurangi potensi kesalahan dalam menafsirkan hasil *Similarity*.

Dalam sistem ini, *threshold* digunakan untuk menyaring hasil *Similarity* agar hanya judul yang memiliki tingkat kemiripan signifikan yang ditampilkan. Sebagai contoh, jika *threshold* ditetapkan sebesar 0.75, maka hanya judul dengan nilai *Similarity* ≥ 0.75 yang ditampilkan. Penggunaan *threshold* ini membantu mengurangi hasil yang kurang relevan dan memfokuskan sistem pada potensi duplikasi judul penelitian, seperti ditunjukkan pada tabel 4.3 berikut.

*Tabel 4.3 Contoh hasil jika threshold ditetapkan ≥ 0.75:*

| Judul Penelitian | *Similarity* | Status |
|---|---|---|
| Pengembangan Sistem Informasi Monitoring Tema Penelitian di Perguruan Tinggi | 0.91 | Memenuhi *Threshold* |
| Sistem Informasi Penelitian di Lingkungan Kampus | 0.79 | Memenuhi *Threshold* |
| Aplikasi Pengelolaan Data Akademik Mahasiswa | 0.44 | Tidak Memenuhi *Threshold* |

Judul dengan nilai *Similarity* di bawah *threshold* tidak ditampilkan, sehingga hasil yang diperoleh lebih fokus dan relevan. Penerapan *threshold* ini membantu pengguna untuk lebih fokus pada judul-judul yang berpotensi memiliki kesamaan tema secara signifikan.

### 4.2.2 Implementasi *Cosine Similarity*

*Cosine Similarity* digunakan dalam sistem ini untuk mengukur tingkat kedekatan makna antar judul penelitian berdasarkan vektor *embedding* yang dihasilkan oleh model *Sentence embedding* berbasis BERT. Setiap judul penelitian direpresentasikan dalam bentuk vektor numerik, sehingga perbandingan kemiripan dapat dilakukan secara matematis dalam ruang vektor multidimensi. Metode *Cosine Similarity* mengukur sudut antara dua vektor, sehingga fokus pada kesamaan arah vektor, bukan pada besar nilainya. Pendekatan ini memungkinkan sistem untuk menilai kesamaan makna judul penelitian secara lebih kontekstual.

Dalam penelitian ini, Penggunaan metode *Cosine Similarity* memungkinkan sistem mendeteksi kesamaan makna antar judul penelitian meskipun terdapat perbedaan susunan kata atau istilah yang digunakan. Dengan demikian, pendekatan ini dinilai lebih akurat dibandingkan metode pencocokan teks secara langsung (string matching) yang hanya mengandalkan kesamaan karakter atau kata.

Pada implementasi sistem, *Cosine Similarity* digunakan untuk membandingkan *embedding* judul yang dimasukkan oleh pengguna dengan *embedding* judul penelitian yang terdapat dalam dataset data penelitian UIN Ar-Raniry. Nilai *Similarity* yang dihasilkan berada pada rentang 0 hingga 1, di mana nilai yang lebih tinggi menunjukkan tingkat kemiripan makna yang semakin besar. Hasil perhitungan *Similarity* ini selanjutnya digunakan sebagai dasar pengurutan judul penelitian, sehingga judul dengan nilai kemiripan tertinggi akan ditampilkan pada posisi teratas sebagai hasil akhir sistem.

Sebagai contoh, pengguna memasukkan judul penelitian sebagai berikut: "*Sistem Informasi Monitoring Tema Penelitian Berbasis Web*", Sistem kemudian menghitung nilai *Cosine Similarity* antara judul input tersebut dengan judul-judul penelitian yang terdapat dalam dataset. Contoh hasil perhitungan *Similarity* ditunjukkan pada Tabel 4.4 di bawah ini.

*Tabel 4.4 Contoh Hasil Perhitungan Cosine Similarity*

| No | Judul Penelitian dalam Dataset | Nilai *Similarity* |
|---|---|---|
| 1 | Pengembangan Sistem Informasi Monitoring Penelitian di Perguruan Tinggi | 0.91 |
| 2 | Sistem Informasi Penelitian Berbasis Web | 0.83 |
| 3 | Analisis Sistem Informasi Akademik | 0.62 |
| 4 | Aplikasi Pengelolaan Data Akademik Mahasiswa | 0.44 |

Berdasarkan Tabel 4.4, terlihat bahwa judul penelitian yang memiliki konteks dan tujuan yang serupa dengan judul input menghasilkan nilai *Similarity* yang lebih tinggi. Sebaliknya, judul penelitian yang memiliki konteks berbeda menghasilkan nilai *Similarity* yang lebih rendah. Hasil ini menunjukkan bahwa *Cosine Similarity* mampu membedakan tingkat kemiripan makna judul penelitian secara bertahap dan proporsional.

Nilai *Cosine Similarity* yang dihasilkan digunakan sebagai dasar utama dalam proses penentuan dan pengurutan hasil pencarian judul penelitian. Dengan mekanisme ini, sistem dapat menyajikan hasil pencarian yang lebih relevan dan terfokus pada judul-judul yang memiliki potensi kesamaan tema secara signifikan. Pendekatan ini mendukung tujuan sistem informasi monitoring penelitian, yaitu membantu proses identifikasi kemiripan judul secara otomatis dan memudahkan proses validasi tema penelitian di lingkungan UIN Ar-Raniry Banda Aceh.

## 4.3 Implementasi Sistem Menggunakan *Framework Streamlit*

Sistem informasi monitoring tema penelitian pada penelitian ini diimplementasikan dalam bentuk aplikasi berbasis web menggunakan *framework Streamlit* . Aplikasi ini dirancang untuk memudahkan pengguna dalam melakukan pencarian dan analisis kemiripan judul penelitian secara interaktif dan real-time.

Pada halaman utama sistem, pengguna disajikan tampilan untuk mengunggah file dataset penelitian serta memasukkan judul penelitian yang ingin dianalisis. Setelah judul dimasukkan, sistem secara otomatis memproses teks dan menghitung tingkat kemiripan dengan judul-judul penelitian yang terdapat dalam dataset menggunakan model *machine learning* yang telah diimplementasikan.

Hasil perhitungan kemiripan judul ditampilkan dalam bentuk tabel yang memuat informasi judul penelitian, nama peneliti, fakultas, program studi, tahun penelitian, serta nilai *Similarity*. Tabel hasil ditampilkan secara terurut berdasarkan nilai kemiripan tertinggi sehingga memudahkan pengguna dalam mengidentifikasi judul yang paling relevan.

Selain tabel, sistem juga menampilkan visualisasi pendukung berupa grafik untuk menunjukkan perbandingan nilai *Similarity* antar judul penelitian. Sistem menyediakan pengaturan *threshold* kemiripan yang dapat disesuaikan oleh pengguna untuk menyaring hasil pencarian sesuai dengan kebutuhan analisis.

Melalui implementasi ini, *Streamlit* berperan sebagai antarmuka yang menghubungkan proses analisis *machine learning* dengan pengguna, sehingga sistem dapat digunakan secara mudah dan efisien dalam mendukung monitoring tema penelitian. Berikut penjelasan kode program dan tampilan antarmuka sistem dapat dilihat di bawah ini.

### 4.3.1 Kode Program

a. Import *Library*

```
import pandas as pd
import torch
from sentence_transformers import SentenceTransformer, util
import streamlit as st
import altair as alt
from io import BytesIO
import os
```

Gambar 4.1 Import Library

Pada bagian awal program dilakukan proses import *Library* yang ditunjukkan pada Gambar 4.1. Digunakan untuk mendukung pengembangan sistem. *Library* yang digunakan dalam penelitian ini antara lain:

- Pandas digunakan untuk membaca dan mengolah data dalam format tabel seperti file Excel.
- Torch digunakan untuk melakukan operasi tensor serta mendukung proses komputasi pada model *deep learning*.
- *Sentence_transformers* digunakan untuk memuat model *Sentence-*BERT serta menghitung nilai kemiripan menggunakan fungsi *cosine similarity*.
- Streamlit digunakan untuk membangun antarmuka sistem berbasis web secara interaktif.
- Altair digunakan untuk menampilkan visualisasi data dalam bentuk grafik.
- BytesIO digunakan untuk mengelola proses ekspor data ke dalam format file.
- Os digunakan untuk memeriksa keberadaan file pada sistem.

b. *Load* data *Excel*

```
# LOAD FILE
# ========================================
file_path = "data_test.xlsx"

if not os.path.exists(file_path):
    st.error(f"❌ File '{file_path}' tidak ditemukan.")
    st.stop()

df_unique, judul_col, tahun_col = load_data(file_path)
model = load_model()

judul_list = df_unique[judul_col].astype(str).tolist()
judul_embeddings = model.encode(judul_list, convert_to_tensor=True)
```

Gambar 4.2 *Load* Data *Excel*

Fungsi *load*_data() pada gambar 4.2 di atas digunakan untuk membaca data judul penelitian dari file Excel. Data kemudian diproses untuk memastikan bahwa kolom yang berisi judul penelitian dapat dikenali oleh sistem.

Pada tahap ini dilakukan beberapa proses, yaitu:

1. Membaca file Excel menggunakan **pandas**.
2. Membersihkan nama kolom dari spasi yang tidak diperlukan.
3. Mendeteksi kolom yang mengandung kata **"judul"** sebagai kolom utama data penelitian.
4. Mendeteksi kolom **tahun penelitian** apabila tersedia.
5. Menghapus data duplikat untuk memastikan setiap judul hanya muncul satu kali.

Hasil dari proses ini berupa dataset yang telah dibersihkan dan siap digunakan dalam proses analisis kemiripan.

c. Model *Sentence*-BERT

```
# LOAD MODEL
# ====================================
@st.cache_resource
def load_model():
    return SentenceTransformer("sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2")
```

Gambar 4.3 Model SBERT

Fungsi *load*_model() pada gambar 4.3 di atas digunakan untuk memuat model *Sentence*-BERT yang akan digunakan dalam proses *embedding* teks. Model yang digunakan dalam penelitian ini adalah *paraphrase-Multilingual-MiniLM*-L12-v2, yang merupakan model *Multilingual* sehingga dapat memahami berbagai bahasa termasuk bahasa Indonesia. Decorator st.*cache_resource* digunakan untuk menyimpan model dalam memori sehingga model tidak perlu dimuat ulang setiap kali aplikasi dijalankan.

d. Proses input pengguna

```
# INPUT
# ========================================
query = st.text_input("🔍 Masukkan tema penelitian yang ingin dicari:")
top_k = st.slider("Jumlah hasil yang ditampilkan:", 1, 20, 5)
threshold = st.slider("Threshold similarity:", 0.0, 1.0, 0.50, 0.05)
```

Gambar 4.4 Input Pengguna

Pada gambar 4.4 di atas Sistem menyediakan beberapa komponen input yang dapat digunakan oleh pengguna untuk melakukan pencarian tema penelitian. Input yang tersedia meliputi:

1. Text Input

   Digunakan untuk memasukkan kata kunci atau tema penelitian yang ingin dicari.
2. Slider Jumlah Hasil

   Digunakan untuk menentukan jumlah hasil pencarian yang akan ditampilkan.
3. Slider Threshold Similarity

   Digunakan untuk menentukan batas minimal nilai similarity yang dianggap relevan.

e. Proses Perhitungan *Similarity*

```
if st.button("🔍 Cari"):
    if query.strip() == "":
        st.warning("Masukkan query dulu.")
    else:
        query_emb = model.encode(query, convert_to_tensor=True)
        cos_scores = util.cos_sim(query_emb, judul_embeddings)[0]
        top_results = torch.topk(cos_scores, k=top_k)

        hasil = []
        for score, idx in zip(top_results[0], top_results[1]):
            pers_percent = int(round(float(score) * 100))
            row = {"Judul Penelitian": judul_list[idx]}

            if tahun_col:
                row["Tahun"] = df_unique.iloc[int(idx)][tahun_col]

            row.update({
                "Skor Similarity": float(score),
                "Persentase Similarity": f"{pers_percent}%"
            })

            hasil.append(row)

        df_hasil = pd.DataFrame(hasil)
        df_hasil.insert(0, "No.", range(1, len(df_hasil) + 1))
```

(variable) df_unique: Series[Any]

Gambar 4.5 Perhitungan *Similarity*

Pada gambar 4.5 di atas, ketika pengguna menekan tombol Cari, sistem akan melakukan beberapa tahapan pemrosesan, yaitu:

1. Mengubah query yang dimasukkan pengguna menjadi *embedding* vektor menggunakan model *Sentence*-BERT.
2. Menghitung nilai *Cosine Similarity* antara query dengan seluruh judul penelitian dalam dataset.
3. Mengambil nilai *similarity* tertinggi menggunakan fungsi torch.topk().
4. Mengubah nilai *similarity* menjadi persentase agar lebih mudah dipahami oleh pengguna.

f. Visualisasi Hasil Similarity

```
chart = alt.Chart(df_hasil).mark_bar().encode(
    x=alt.X("Skor Similarity:Q", scale=alt.Scale(domain=[0, 1]), title="Skor (0-1)"),
    y=alt.Y("Judul Penelitian:N", sort="-x"),
    color=alt.Color(
        "Status:N",
        scale=alt.Scale(
            domain=["Memenuhi Threshold", "Di Bawah Threshold"],
            range=["green", "red"]
        ),
        legend=alt.Legend(title="Status Similarity")
    ),
    tooltip=tooltip_cols
).properties(
    width="container",
    height=400,
    title="Perbandingan Skor Similarity"
)

st.altair_chart(chart, use_container_width=True)
```

Gambar 4.6 Visualisasi Hasil *Similarity*

Pada gambar 4.6 di atas, untuk mempermudah interpretasi hasil, sistem menampilkan grafik perbandingan similarity menggunakan *Library* Altair. Grafik ditampilkan dalam bentuk diagram batang (*bar chart*) yang menunjukkan nilai *similarity* setiap judul penelitian terhadap query yang dimasukkan pengguna.

Selain itu, sistem juga memberikan indikator warna:

- Hijau → nilai similarity memenuhi *threshold*
- Merah → nilai similarity berada di bawah *threshold*

Visualisasi ini membantu pengguna dalam memahami hasil analisis kemiripan secara lebih intuitif.

g. Download Hasil

```
# DOWNLOAD
# ======================================
st.subheader("⬇️ Download Hasil")

csv = convert_df(df_hasil, to_excel=False)
excel = convert_df(df_hasil, to_excel=True)

st.download_button("Download sebagai CSV", csv, "hasil_similarity.csv")
st.download_button("Download sebagai Excel", excel, "hasil_similarity.xlsx")
```

Gambar 4.7 Download Hasil

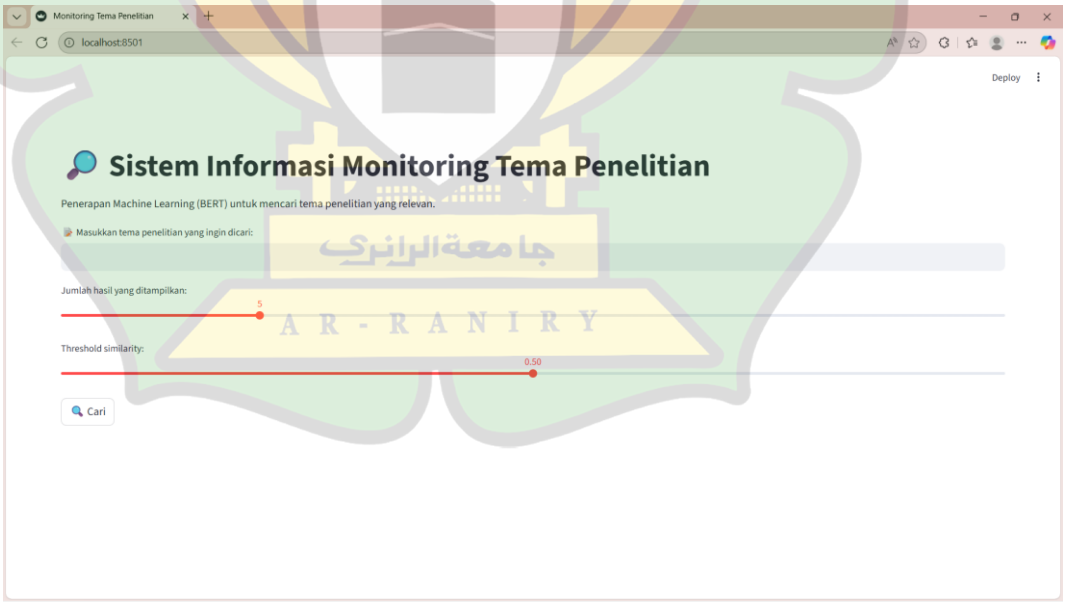
Pada gambar 4.7 di atas, Fungsi *convert*_df() digunakan untuk mengubah data hasil perhitungan *similarity* ke dalam format file yang dapat diunduh oleh pengguna. Sistem menyediakan dua format file yang dapat dipilih, yaitu CSV dan *Excel*. Jika pengguna memilih format *Excel*, sistem akan memanfaatkan *BytesIO* untuk membuat file sementara sebelum proses pengunduhan dilakukan.

Melalui fitur ini, pengguna dapat menyimpan hasil analisis similarity yang telah diperoleh sehingga data tersebut dapat digunakan kembali untuk keperluan dokumentasi, analisis lanjutan, maupun referensi dalam proses penelitian.

### 4.3.2 Tampilan Antarmuka Sistem

a. Tampilan Awal

Halaman awal ini berfungsi sebagai antarmuka utama bagi pengguna untuk melakukan pencarian dan analisis kemiripan tema penelitian. Dapat dilihat pada gambar 4.8 di bawah ini.




Gambar 4.8 Tampilan Awal Sistem Informasi Monitoring Tema Penelitian

Gambar 4.8 di atas menampilkan halaman awal Sistem Informasi Monitoring Tema Penelitian yang dikembangkan menggunakan *framework*

*Streamlit* . Pada bagian atas halaman ditampilkan judul sistem beserta deskripsi singkat yang menjelaskan bahwa sistem menerapkan model *machine learning* berbasis BERT untuk menemukan tema penelitian yang relevan. Pada bagian utama halaman, sistem menyediakan kolom input judul penelitian yang digunakan oleh pengguna untuk memasukkan judul atau tema penelitian yang ingin dianalisis.

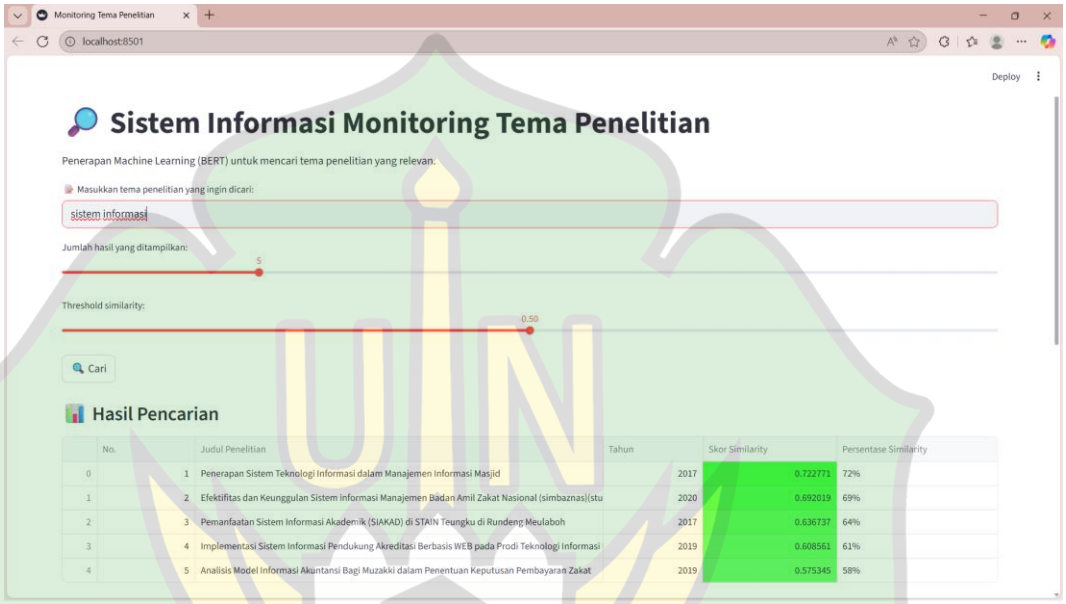
Judul yang dimasukkan akan diproses oleh sistem dan dibandingkan dengan judul-judul penelitian yang terdapat dalam dataset LP2M UIN Ar-Raniry. Input ini menjadi dasar dalam proses pembentukan *embedding* dan perhitungan kemiripan judul penelitian. Selain kolom input, halaman awal juga dilengkapi dengan slider pengaturan jumlah hasil yang ditampilkan. Slider ini memungkinkan pengguna menentukan jumlah judul penelitian dengan nilai kemiripan tertinggi yang ingin ditampilkan oleh sistem. Dengan adanya fitur ini, pengguna dapat menyesuaikan banyaknya hasil sesuai dengan kebutuhan analisis, baik untuk melihat hasil secara ringkas maupun untuk melakukan penelusuran yang lebih luas.

Selanjutnya, sistem menyediakan slider *threshold Similarity* yang berfungsi sebagai nilai ambang kemiripan judul penelitian. Pengguna dapat mengatur tingkat kemiripan minimum yang dianggap relevan, sehingga hanya judul penelitian dengan nilai *Similarity* di atas *threshold* yang akan ditampilkan. Pengaturan ini membantu pengguna memfokuskan hasil pencarian pada judul-judul yang memiliki kesamaan tema secara signifikan.

Setelah seluruh parameter ditentukan, pengguna dapat menekan tombol pencarian untuk menjalankan proses perhitungan kemiripan judul. Sistem kemudian memproses input secara real-time dan menampilkan hasil sesuai dengan jumlah dan *threshold* yang telah ditentukan. Secara keseluruhan, tampilan halaman awal dirancang sederhana, interaktif, dan mudah digunakan, sehingga mendukung proses monitoring dan validasi tema penelitian secara efektif.

b. Tampilan Hasil Pencarian

Halaman ini berfungsi untuk melihat hasil dari pencarian tema penelitian dari kata kunci yang diinput oleh pengguna. Dapat dilihat pada gambar 4.2 di bawah ini.




Gambar 4.9 Hasil Pencarian dan Perhitungan *Similarity* Judul Penelitian

Gambar 4.9 menampilkan hasil pencarian kemiripan judul penelitian setelah pengguna memasukkan kata kunci *"sistem informasi"* pada halaman utama sistem.

Setelah pengguna menekan tombol Cari, sistem akan melakukan proses analisis dengan tahapan sebagai berikut:

1. Tema penelitian yang diinput oleh pengguna dikonversi menjadi vektor numerik menggunakan model BERT *Paraphrase Multilingual*, yang mampu merepresentasikan makna semantik dari teks dalam berbagai bahasa.
2. Data judul penelitian yang ada dalam basis data sebelumnya juga telah direpresentasikan dalam bentuk *embedding* vektor.
3. Sistem menghitung tingkat kemiripan antara vektor input pengguna dan vektor judul penelitian menggunakan metode *Cosine Similarity*.

4. Hasil perhitungan kemudian diurutkan berdasarkan nilai *Similarity* tertinggi dan ditampilkan pada tabel hasil pencarian.

Judul-judul penelitian dalam tabel telah diurutkan secara *default* berdasarkan nilai *Similarity* tertinggi, sehingga judul dengan tingkat kemiripan makna paling tinggi terhadap judul input pengguna ditampilkan pada posisi teratas.

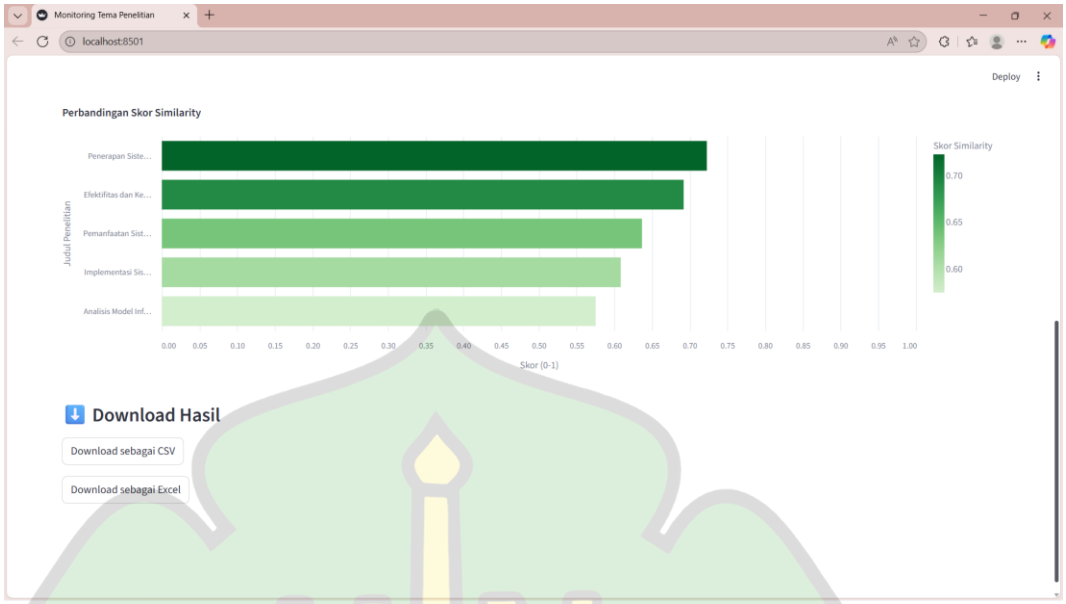
Pada kolom skor *Similarity*, sistem menampilkan nilai numerik hasil perhitungan *Cosine Similarity* dalam rentang 0 hingga 1. Nilai ini kemudian dikonversi ke dalam bentuk persentase pada kolom persentase *Similarity* untuk memudahkan pengguna dalam memahami tingkat kemiripan antar judul penelitian.

Tampilan warna hijau pada kolom *Similarity* digunakan sebagai indikator visual bahwa judul-judul tersebut telah memenuhi nilai *threshold* yang ditentukan sebelumnya.

Berdasarkan hasil yang ditampilkan, dapat dilihat bahwa judul-judul yang mengandung tema sistem informasi menghasilkan nilai *Similarity* yang relatif tinggi, sedangkan judul dengan konteks yang semakin berbeda menunjukkan nilai *Similarity* yang lebih rendah. Tampilan tabel tersebut dapat membantu pengguna dalam melakukan evaluasi awal terhadap potensi kesamaan atau duplikasi tema penelitian secara cepat dan terstruktur.

c. Tampilan Grafik Perbandingan dan Download Hasil

Halaman ini berfungsi untuk menampilkan visualisasi grafik perbandingan skor *Similarity* dari judul-judul penelitian yang memiliki tingkat kemiripan tertinggi terhadap judul input pengguna. Dapat dilihat pada gambar 4.10 di bawah ini.




Gambar 4.10 Visualisasi Perbandingan Skor *Similarity* dan Fitur Unduh Hasil

Gambar 4.10 di atas menampilkan Grafik yang disajikan dalam bentuk horizontal bar chart dan memperlihatkan perbandingan nilai *Cosine Similarity* masing-masing judul penelitian secara visual, sehingga memudahkan pengguna dalam melihat perbedaan tingkat kemiripan antar judul secara cepat dan intuitif.

Setiap batang pada grafik merepresentasikan satu judul penelitian, dengan panjang batang menunjukkan besarnya nilai *Similarity* dalam rentang 0 hingga 1. Warna gradasi hijau digunakan sebagai indikator visual tingkat kemiripan, di mana warna yang lebih gelap menunjukkan nilai *Similarity* yang lebih tinggi. Ketika pengguna mengarahkan kursor ke salah satu batang, sistem menampilkan informasi detail berupa judul penelitian, tahun penelitian, nilai skor *Similarity*, serta persentase kemiripan. Fitur ini membantu pengguna memahami konteks hasil pencarian secara lebih mendalam tanpa harus kembali ke tabel hasil.

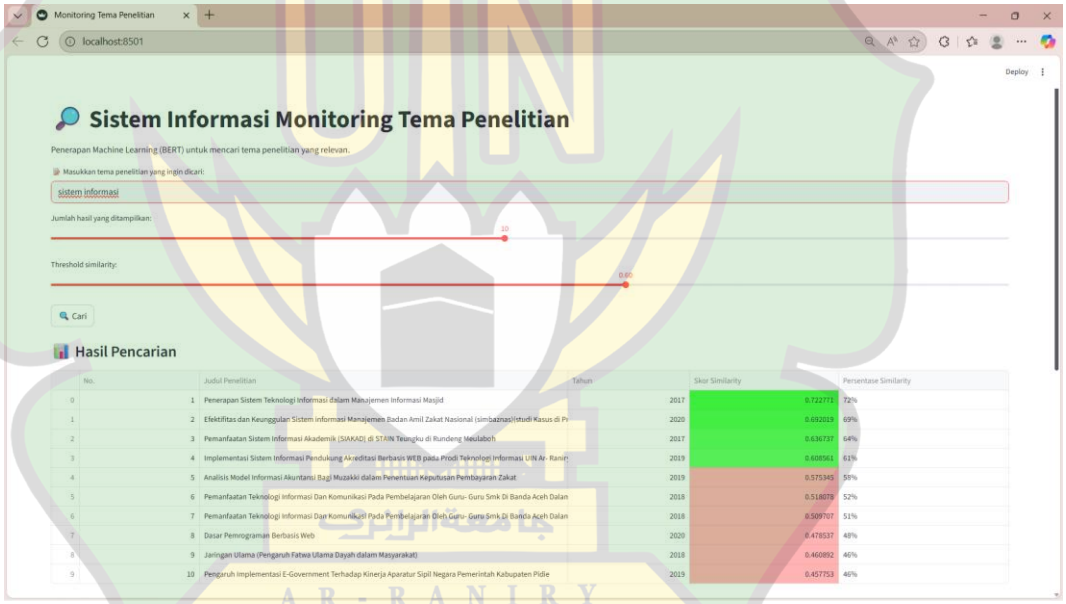
Selain visualisasi grafik, pada bagian bawah tampilan juga disediakan fitur *Download Hasil*. Sistem memungkinkan pengguna untuk mengunduh hasil pencarian dalam dua format, yaitu CSV dan *Excel*. Fitur ini memudahkan pengguna, khususnya pihak LP2M atau civitas akademika, untuk melakukan

dokumentasi, analisis lanjutan, atau pelaporan hasil monitoring tema penelitian di luar sistem.

Keberadaan grafik perbandingan dan fitur unduh hasil ini menunjukkan bahwa sistem tidak hanya menyajikan hasil *Similarity* dalam bentuk tabel, tetapi juga menyediakan visualisasi dan keluaran data yang mendukung proses evaluasi dan pengambilan keputusan secara lebih efektif.

d. Tampilan Hasil Pencarian Berdasarkan Jumlah

Tampilan hasil ini berfungsi untuk menampilkan visualisasi tabel yang dapat di atur jumlah hasilnya menurut parameter jumlah yang diinginkan oleh pengguna. Dapat dilihat pada gambar 4.11 di bawah ini.




Gambar 4.11 Pengaturan Jumlah Hasil dan *Threshold Similarity*

Gambar 4.11 di atas menunjukkan hasil pencarian judul penelitian setelah pengguna melakukan penyesuaian parameter pencarian pada sistem. Pada tampilan ini, pengguna mengatur jumlah hasil yang ditampilkan menjadi 10 judul penelitian serta menaikkan nilai *threshold Similarity* menjadi 0,60, berbeda dengan kondisi awal sistem yang secara default menampilkan 5 hasil dengan *threshold* 0,50.

Perubahan nilai *threshold Similarity* berpengaruh langsung terhadap hasil yang ditampilkan oleh sistem. Dengan *threshold* sebesar 0,60, sistem hanya

menampilkan judul penelitian yang memiliki tingkat kemiripan makna yang relatif tinggi terhadap judul input pengguna. Hal ini terlihat dari dominasi nilai skor *Similarity* yang berada pada kategori menengah hingga tinggi, sementara judul dengan nilai kemiripan rendah tidak lagi ditampilkan. Kondisi ini membantu pengguna untuk lebih fokus pada judul-judul yang berpotensi memiliki kesamaan tema secara signifikan.

Pengaturan slider jumlah hasil yang ditampilkan memberikan fleksibilitas kepada pengguna dalam menentukan banyaknya judul penelitian yang ingin dianalisis. Pada contoh ini, meskipun jumlah hasil diatur menjadi 10, sistem tetap melakukan penyaringan berdasarkan *threshold* yang telah ditentukan. Dengan demikian, hasil yang ditampilkan tidak hanya lebih banyak, tetapi juga tetap relevan dan terkontrol dari sisi kualitas kemiripan.

Tampilan tabel hasil pencarian menampilkan informasi penting berupa judul penelitian, tahun penelitian, skor *Similarity*, serta persentase kemiripan. Perbedaan warna pada kolom skor *Similarity* juga memberikan indikator visual yang memudahkan pengguna dalam mengidentifikasi judul dengan tingkat kemiripan tertinggi secara cepat. Implementasi pengaturan parameter ini menunjukkan bahwa sistem mampu menyesuaikan hasil pencarian secara dinamis sesuai dengan kebutuhan pengguna dalam proses monitoring dan validasi tema penelitian.

e. Tampilan Hasil Pencarian Diurutkan Berdasarkan Alfabet

Pengguna dapat mengatur hasil pencarian judul atau tema penelitian di urutkan berdasarkan alfabet. Dapat dilihat pada gambar 4.12 di bawah ini.

Hasil Pencarian

| No. | Judul Penelitian | Tahun | Skor Similarity | Persentase Similarity |
|---|---|---|---|---|
| 5 | Analisis Model Informasi Akuntansi Bagi Muzakki dalam Penentuan Keputusan Pembayaran Zakat | 2019 | 0.575345 | 58% |
| 8 | Dasar Pemrograman Berbasis Web | 2020 | 0.478537 | 48% |
| 2 | Efektifitas dan Keunggulan Sistem informasi Manajemen Badan Amil Zakat Nasional (simbaznas)(studi Kasus di Propinsi Aceh dan Nusa Tenggara Barat) | 2020 | 0.692019 | 69% |
| 4 | Implementasi Sistem Informasi Pendukung Akreditasi Berbasis WEB pada Prodi Teknologi Informasi UIN Ar- Raniry | 2019 | 0.608561 | 61% |
| 9 | Jaringan Ulama (Pengaruh Fatwa Ulama Dayah dalam Masyarakat) | 2018 | 0.460892 | 46% |
| 3 | Pemanfaatan Sistem Informasi Akademik (SIAKAD) di STAIN Teungku di Rundeng Meulaboh | 2017 | 0.636737 | 64% |
| 7 | Pemanfaatan Teknologi Informasi Dan Komunikasi Pada Pembelajaran Oleh Guru- Guru Smk Di Banda Aceh Dalam Rangka Implementasi Kurikulum 2013 | 2018 | 0.509707 | 51% |
| 6 | Pemanfaatan Teknologi Informasi Dan Komunikasi Pada Pembelajaran Oleh Guru- Guru Smk Di Banda Aceh Dalam Rangka Implementasi Kurikulum 2014 | 2018 | 0.518078 | 52% |
| 1 | Penerapan Sistem Teknologi Informasi dalam Manajemen Informasi Masjid | 2017 | 0.722771 | 72% |
| 10 | Pengaruh Implementasi E-Government Terhadap Kinerja Aparatur Sipil Negara Pemerintah Kabupaten Pidie | 2019 | 0.457753 | 46% |

Gambar 4.12 Tampilan Tabel Hasil Pencarian Berdasarkan Alfabet

Gambar 4.12 di atas menampilkan hasil pencarian judul penelitian dalam bentuk tabel yang diurutkan berdasarkan abjad judul penelitian. Pada tampilan ini, sistem menyajikan daftar judul penelitian yang memiliki tingkat kemiripan dengan judul input pengguna, namun urutan data tidak berdasarkan skor *Similarity*, melainkan berdasarkan urutan alfabet pada kolom judul penelitian.

Tabel hasil pencarian tetap menampilkan informasi utama yang dibutuhkan pengguna, yaitu nomor urut, judul penelitian, tahun penelitian, skor *Similarity*, serta persentase kemiripan. Meskipun urutan judul disusun secara alfabetis, nilai skor *Similarity* tetap dihitung menggunakan metode *Cosine Similarity* dan ditampilkan pada kolom tersendiri, sehingga pengguna masih dapat melihat tingkat kemiripan masing-masing judul terhadap judul input.

Pengurutan berdasarkan huruf ini berguna dalam konteks penelusuran data secara administratif, contohnya ketika pengguna ingin mencari atau mencocokkan judul tertentu secara manual berdasarkan nama judul. Dengan adanya tampilan ini, sistem tidak hanya berfungsi sebagai alat pencari kemiripan berbasis *machine learning*, tetapi juga sebagai sarana bantu pengelolaan data judul penelitian yang lebih terstruktur dan mudah dibaca.

Warna pada kolom skor *Similarity* tetap digunakan sebagai indikator visual untuk membedakan tingkat kemiripan antar judul penelitian, sehingga meskipun data diurutkan secara alfabetis, pengguna tetap dapat dengan cepat mengenali judul dengan tingkat kemiripan yang lebih tinggi. Hal ini menunjukkan bahwa sistem mampu menyajikan hasil pencarian dalam

berbagai bentuk tampilan tanpa menghilangkan informasi penting terkait analisis kemiripan judul penelitian.

f. Tampilan Hasil Pencarian Diurutkan Berdasarkan Tahun

Pengguna dapat mengatur hasil pencarian judul atau tema penelitian di urutkan berdasarkan tahun. Dapat dilihat pada gambar 4.13 di bawah ini.



Hasil Pencarian

| No. | Judul Penelitian | ↑ Tahun | Skor Similarity | Persentase Similarity |
|---|---|---|---|---|
| 1 | Penerapan Sistem Teknologi Informasi dalam Manajemen Informasi Masjid | 2017 | 0.722771 | 72% |
| 3 | Pemanfaatan Sistem Informasi Akademik (SIAKAD) di STAIN Teungku di Rundeng Meulaboh | 2017 | 0.636737 | 64% |
| 6 | Pemanfaatan Teknologi Informasi Dan Komunikasi Pada Pembelajaran Oleh Guru- Guru Smk Di Banda Aceh Dalam Rangka Implementasi Kurikulum 2014 | 2018 | 0.518078 | 52% |
| 7 | Pemanfaatan Teknologi Informasi Dan Komunikasi Pada Pembelajaran Oleh Guru- Guru Smk Di Banda Aceh Dalam Rangka Implementasi Kurikulum 2013 | 2018 | 0.509707 | 51% |
| 9 | Jaringan Ulama (Pengaruh Fatwa Ulama Dayah dalam Masyarakat) | 2018 | 0.460892 | 46% |
| 4 | Implementasi Sistem Informasi Pendukung Akreditasi Berbasis WEB pada Prodi Teknologi Informasi UIN Ar- Raniry | 2019 | 0.608561 | 61% |
| 5 | Analisis Model Informasi Akuntansi Bagi Muzakki dalam Penentuan Keputusan Pembayaran Zakat | 2019 | 0.575345 | 58% |
| 10 | Pengaruh Implementasi E-Government Terhadap Kinerja Aparatur Sipil Negara Pemerintah Kabupaten Pidie | 2019 | 0.457753 | 46% |
| 2 | Efektifitas dan Keunggulan Sistem informasi Manajemen Badan Amil Zakat Nasional (simbaznas)(studi Kasus di Propinsi Aceh dan Nusa Tenggara Barat) | 2020 | 0.692019 | 69% |
| 8 | Dasar Pemrograman Berbasis Web | 2020 | 0.478537 | 48% |

Gambar 4.13 Tampilan Tabel Hasil Pencarian berdasarkan tahun

Gambar 4.13 di atas menampilkan hasil pencarian judul penelitian dalam bentuk tabel yang diurutkan berdasarkan tahun penelitian. Pada tampilan ini, sistem menyajikan daftar judul penelitian yang memiliki tingkat kemiripan dengan judul input pengguna, namun urutan data tidak berdasarkan skor *Similarity*, melainkan berdasarkan tahun pelaksanaan penelitian.

Tabel hasil pencarian tetap menampilkan informasi utama yang dibutuhkan pengguna, yaitu nomor urut, judul penelitian, tahun penelitian, skor *Similarity*, serta persentase kemiripan. Meskipun urutan data disusun berdasarkan tahun, nilai skor *Similarity* tetap ditampilkan pada kolom tersendiri sebagai indikator tingkat kemiripan masing-masing judul penelitian terhadap judul input pengguna.

Pengurutan berdasarkan tahun ini berguna dalam proses monitoring penelitian, karena memungkinkan pengguna untuk melihat perkembangan dan sebaran tema penelitian secara kronologis. Dengan tampilan tersebut, pengguna dapat mengidentifikasi judul-judul penelitian yang relevan pada periode waktu tertentu.

Warna pada kolom skor *Similarity* tetap digunakan sebagai indikator visual untuk membedakan tingkat kemiripan antar judul penelitian. Hal ini membantu pengguna dalam mengenali judul dengan tingkat kemiripan yang lebih tinggi meskipun data tidak diurutkan berdasarkan nilai *Similarity*.

g. Tampilan Hasil Pencarian Diurutkan Berdasarkan *Similarity* Terendah

Pengguna dapat mengatur hasil pencarian judul atau tema penelitian di urutkan berdasarkan nilai *Similarity* terendah. Dapat dilihat pada gambar 4.14 di bawah ini.


Hasil Pencarian

| No. | Judul Penelitian | Tahun | ↑ Skor Similarity | Persentase Similarity |
|---|---|---|---|---|
| 10 | Pengaruh Implementasi E-Government Terhadap Kinerja Aparatur Sipil Negara Pemerintah Kabupaten Pidie | 2019 | 0.457753 | 46% |
| 9 | Jaringan Ulama (Pengaruh Fatwa Ulama Dayah dalam Masyarakat) | 2018 | 0.460892 | 46% |
| 8 | Dasar Pemrograman Berbasis Web | 2020 | 0.478537 | 48% |
| 7 | Pemanfaatan Teknologi Informasi Dan Komunikasi Pada Pembelajaran Oleh Guru- Guru Smk Di Banda Aceh Dalam Rangka Implementasi Kurikulum 2013 | 2018 | 0.509707 | 51% |
| 6 | Pemanfaatan Teknologi Informasi Dan Komunikasi Pada Pembelajaran Oleh Guru- Guru Smk Di Banda Aceh Dalam Rangka Implementasi Kurikulum 2014 | 2018 | 0.518078 | 52% |
| 5 | Analisis Model Informasi Akuntansi Bagi Muzakki dalam Penentuan Keputusan Pembayaran Zakat | 2019 | 0.575345 | 58% |
| 4 | Implementasi Sistem Informasi Pendukung Akreditasi Berbasis WEB pada Prodi Teknologi Informasi UIN Ar- Raniry | 2019 | 0.608561 | 61% |
| 3 | Pemanfaatan Sistem Informasi Akademik (SIAKAD) di STAIN Teungku di Rundeng Meulaboh | 2017 | 0.636737 | 64% |
| 2 | Efektifitas dan Keunggulan Sistem informasi Manajemen Badan Amil Zakat Nasional (simbaznas)(studi Kasus di Propinsi Aceh dan Nusa Tenggara Barat) | 2020 | 0.692019 | 69% |
| 1 | Penerapan Sistem Teknologi Informasi dalam Manajemen Informasi Masjid | 2017 | 0.722771 | 72% |

Gambar 4.14 Tampilan Tabel Hasil Pencarian berdasarkan *Similarity*

Gambar 4.14 di atas menampilkan hasil pencarian judul penelitian dalam bentuk tabel yang diurutkan berdasarkan skor *Similarity* terendah. Pada tampilan ini, sistem menyajikan daftar judul penelitian yang memiliki tingkat kemiripan dengan judul input pengguna, namun urutan data dimulai dari judul dengan nilai kemiripan paling rendah hingga yang paling tinggi.

Tabel hasil pencarian menampilkan informasi utama, yaitu nomor urut, judul penelitian, tahun penelitian, skor *Similarity*, dan persentase kemiripan. Pengurutan berdasarkan skor *Similarity* terendah memungkinkan pengguna untuk melihat judul-judul penelitian yang memiliki tingkat kemiripan paling kecil terhadap judul input, sehingga dapat digunakan sebagai pembanding terhadap judul dengan kemiripan tinggi.

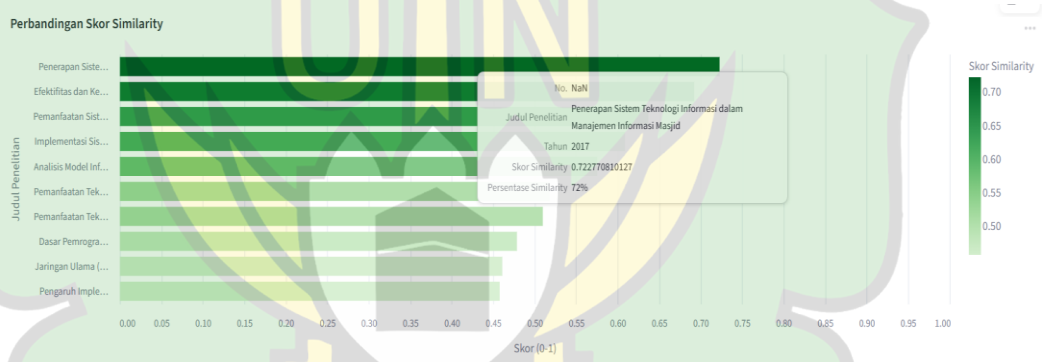
Nilai skor *Similarity* dan persentase kemiripan tetap ditampilkan pada kolom tersendiri sebagai indikator tingkat kemiripan masing-masing judul penelitian. Sistem juga menggunakan visualisasi warna pada kolom skor

*Similarity*, di mana warna merah menunjukkan tingkat kemiripan yang rendah dan warna hijau menunjukkan tingkat kemiripan yang lebih tinggi.

Dengan adanya tampilan hasil pencarian yang diurutkan berdasarkan skor *Similarity* terendah ini, sistem memberikan fleksibilitas kepada pengguna dalam meninjau hasil pencarian dari berbagai sudut pandang tanpa menghilangkan informasi penting terkait tingkat kemiripan judul penelitian.

h. Tampilan Grafik Perbandingan yang dinamis

Tampilan ini berfungsi untuk menampilkan visualisasi grafik perbandingan skor *Similarity* dari judul-judul penelitian yang menyesuaikan berdasarkan parameter jumlah hasil yang diinginkan oleh pengguna. Dapat dilihat pada gambar 4.15 di bawah ini.




Gambar 4.15 Tampilan Grafik Perbandingan Dinamis

Gambar 4.15 di atas memperlihatkan grafik hasil pencarian judul penelitian yang menyesuaikan dengan jumlah data yang dipilih oleh pengguna. Grafik ini digunakan sebagai pelengkap tabel hasil pencarian untuk memberikan gambaran umum terhadap sebaran skor *Similarity* dari judul-judul penelitian yang ditampilkan.

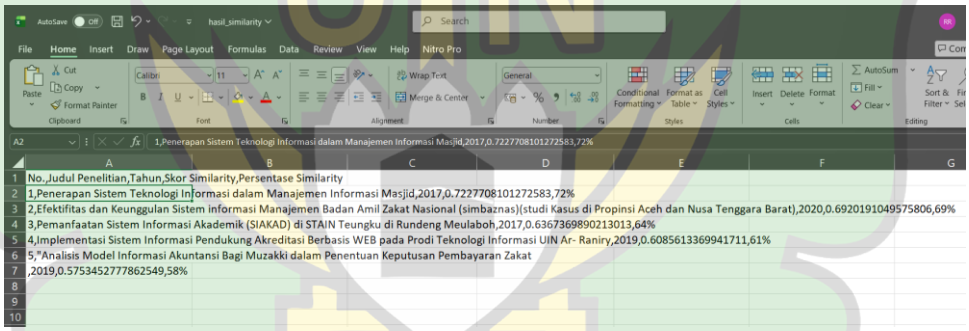
Perubahan jumlah hasil pencarian yang ditentukan pengguna secara langsung memengaruhi jumlah data pada grafik. Semakin banyak hasil yang dipilih, maka semakin banyak pula judul penelitian yang ditampilkan dalam grafik, dan sebaliknya. Hal ini menunjukkan bahwa sistem mampu menampilkan visualisasi data secara dinamis sesuai dengan kebutuhan pengguna.

Grafik menyajikan nilai skor *Similarity* pada setiap judul penelitian dalam bentuk batang, sehingga pengguna dapat membandingkan tingkat kemiripan secara visual tanpa harus membaca seluruh data pada tabel. Visualisasi warna tetap digunakan sebagai pembeda tingkat kemiripan, namun grafik ini lebih difokuskan sebagai ringkasan hasil pencarian dibandingkan sebagai alat analisis detail.

Dengan adanya grafik yang menyesuaikan jumlah hasil pencarian ini, sistem memberikan fleksibilitas kepada pengguna dalam meninjau hasil analisis kemiripan judul penelitian secara ringkas dan mudah dipahami.

i. Tampilan Hasil Download CSV

Tampilan ini menampilkan hasil download data pencarian tema/judul dalam bentuk format CSV. Dapat dilihat pada gambar 4.16 di bawah ini.




Gambar 4.16 Hasil Download CSV

Gambar 4.16 di atas menampilkan hasil unduhan data pencarian dalam format CSV yang dibuka menggunakan aplikasi spreadsheet. File ini berisi data hasil pencarian kemiripan judul penelitian yang dihasilkan oleh sistem setelah pengguna menekan tombol Download sebagai CSV.

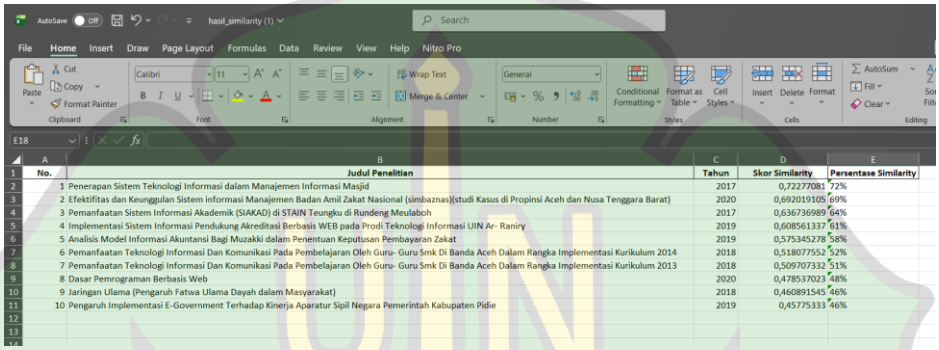
Data yang tersimpan dalam file CSV ditampilkan dalam bentuk tabel sederhana yang memuat informasi utama, yaitu judul penelitian, tahun penelitian, skor *Similarity*, dan persentase kemiripan. Setiap baris merepresentasikan satu judul penelitian yang sebelumnya ditampilkan pada tabel hasil pencarian di dalam sistem.

Dengan tersedianya fitur unduh dalam format CSV, pengguna dapat menyimpan hasil pencarian untuk keperluan dokumentasi, pengolahan data

lanjutan, atau pelaporan. Tampilan ini menunjukkan bahwa sistem tidak hanya menyajikan hasil analisis secara langsung, tetapi juga mendukung proses ekspor data secara praktis dan mudah digunakan.

j. Tampilan Hasil Download *Excel*

Tampilan ini menampilkan hasil download data pencarian tema/judul dalam bentuk format *Excel*. Dapat dilihat pada gambar 4.17 di bawah ini.



| No. | Judul Penelitian | Tahun | Skor Similarity | Persentase Similarity |
|---|---|---|---|---|
| 1 | Penerapan Sistem Teknologi Informasi dalam Manajemen Informasi Masjid | 2017 | 0,72277081 | 72% |
| 2 | Efektifitas dan Keunggulan Sistem informasi Manajemen Badan Amil Zakat Nasional (simbaznas)(studi Kasus di Propinsi Aceh dan Nusa Tenggara Barat) | 2020 | 0,692019105 | 69% |
| 3 | Pemanfaatan Sistem Informasi Akademik (SIAKAD) di STAIN Teungku di Rundeng Meulaboh | 2017 | 0,636736989 | 64% |
| 4 | Implementasi Sistem Informasi Pendukung Akreditasi Berbasis WEB pada Prodi Teknologi Informasi UIN Ar- Raniry | 2019 | 0,608561337 | 61% |
| 5 | Analisis Model Informasi Akuntansi Bagi Muzakki dalam Penentuan Keputusan Pembayaran Zakat | 2019 | 0,575345278 | 58% |
| 6 | Pemanfaatan Teknologi Informasi Dan Komunikasi Pada Pembelajaran Oleh Guru- Guru Smk Di Banda Aceh Dalam Rangka Implementasi Kurikulum 2014 | 2018 | 0,518077552 | 52% |
| 7 | Pemanfaatan Teknologi Informasi Dan Komunikasi Pada Pembelajaran Oleh Guru- Guru Smk Di Banda Aceh Dalam Rangka Implementasi Kurikulum 2013 | 2018 | 0,509707332 | 51% |
| 8 | Dasar Pemrograman Berbasis Web | 2020 | 0,478537023 | 48% |
| 9 | Jaringan Ulama (Pengaruh Fatwa Ulama Dayah dalam Masyarakat) | 2018 | 0,460891545 | 46% |
| 10 | Pengaruh Implementasi E-Government Terhadap Kinerja Aparatur Sipil Negara Pemerintah Kabupaten Pidie | 2019 | 0,45775333 | 46% |

Gambar 4.17 Hasil Download *Excel*

Gambar 4.17 di atas menampilkan hasil unduhan data pencarian dalam format *Excel* yang dibuka menggunakan aplikasi spreadsheet. File ini dihasilkan setelah pengguna menekan tombol Download sebagai *Excel* pada halaman hasil pencarian sistem.

Data hasil pencarian disajikan dalam bentuk tabel *Excel* yang memuat informasi utama, yaitu judul penelitian, tahun penelitian, skor *Similarity*, dan persentase kemiripan. Setiap baris pada tabel merepresentasikan satu judul penelitian yang sebelumnya ditampilkan pada sistem.

Penyajian data dalam format *Excel* memudahkan pengguna dalam melakukan pengolahan dan analisis lanjutan, seperti penyortiran, penyaringan, atau pengelompokan data sesuai kebutuhan. Tampilan ini menunjukkan bahwa sistem mendukung ekspor hasil pencarian ke dalam format tabel yang terstruktur dan mudah digunakan.

## 4.4 Pengujian dan Evaluasi Sistem

Pengujian dan evaluasi sistem dilakukan untuk mengetahui sejauh mana sistem informasi monitoring tema penelitian yang dikembangkan mampu berfungsi sesuai dengan kebutuhan serta tujuan penelitian. Pengujian difokuskan pada aspek fungsionalitas sistem, performa pemrosesan, kualitas hasil *Similarity*, serta efektivitas sistem dalam mendeteksi kemiripan judul penelitian secara otomatis. Evaluasi difokuskan pada pengujian fungsional, performa sistem dan model, kualitas hasil *Similarity*, serta dampak implementasi sistem terhadap proses monitoring tema penelitian di UIN Ar-Raniry Banda Aceh.

### 4.4.1 Pengujian Fungsional Sistem

Pengujian fungsional sistem dilakukan untuk memastikan seluruh fungsi utama sistem berjalan sesuai dengan kebutuhan pengguna. Metode yang digunakan adalah *blackbox testing*, yaitu pengujian berdasarkan input dan output tanpa memperhatikan struktur internal program.

Pengujian difokuskan pada proses input judul penelitian, pengaturan parameter pencarian, serta keluaran sistem berupa hasil *Similarity* dalam bentuk tabel dan grafik.

*Tabel 4.5 Hasil Pengujian Black Box testing*

| No | Fitur yang Diuji | Input | Output yang Diharapkan | Hasil |
|---|---|---|---|---|
| 1 | Input judul penelitian | Teks judul penelitian | Sistem menerima input | Berhasil |
| **2** | Slider jumlah hasil | Nilai 5–10 | Jumlah hasil sesuai input | Berhasil |
| **3** | Slider *threshold Similarity* | Nilai 0.50–0.90 | Hasil terfilter sesuai *threshold* | Berhasil |
| **4** | Tombol pencarian | Klik tombol Cari | Sistem menampilkan hasil *Similarity* | Berhasil |
| **5** | Tabel hasil pencarian | Judul + skor *Similarity* | Data tampil lengkap | Berhasil |

| | | | | |
|---|---|---|---|---|
| **6** | Grafik perbandingan *Similarity* | Hasil *Similarity* | Grafik ditampilkan | Berhasil |
| **7** | Download hasil | CSV / *Excel* | File berhasil diunduh | Berhasil |

Berdasarkan hasil pengujian, seluruh fungsi utama sistem dapat berjalan dengan baik dan sesuai dengan rancangan sistem yang telah dibuat.

### 4.4.2 Pengujian Performa Sistem

Pengujian performa sistem dilakukan untuk mengetahui kecepatan respon sistem dalam memproses permintaan pencarian judul penelitian. Pengujian dilakukan dengan mengamati waktu respon sistem sejak pengguna menekan tombol pencarian hingga hasil ditampilkan di layar.

Pengujian dilakukan pada beberapa skenario input dengan jumlah hasil dan nilai *threshold* yang berbeda.

*Tabel 4.6 Hasil Pengujian Waktu Respon Sistem*

| No | Jumlah Data Ditampilkan | *Threshold* | Waktu Respon (detik) |
|---|---|---|---|
| 1 | 5 hasil | 0.50 | ± 5.2 detik |
| 2 | 10 hasil | 0.50 | ± 10.4 detik |
| 3 | 10 hasil | 0.70 | ± 14.5 detik |

Hasil pengujian menunjukkan bahwa sistem mampu memberikan respon dalam waktu singkat dan relatif stabil. Hal ini didukung oleh mekanisme penyimpanan *embedding* dataset yang telah diproses sebelumnya, sehingga sistem tidak perlu melakukan perhitungan *embedding* ulang terhadap seluruh data.

### 4.4.3 Analisis Hasil *Similarity* dan Kesesuaiannya dengan Tujuan Penelitian

Analisis hasil *Similarity* dilakukan untuk mengevaluasi sejauh mana sistem mampu menampilkan judul penelitian yang relevan berdasarkan kemiripan makna dengan judul input pengguna. Pada tahap ini, sistem

menghasilkan daftar judul penelitian yang diurutkan berdasarkan nilai *Similarity* tertinggi menggunakan metode *Cosine Similarity* terhadap vektor *embedding* yang dihasilkan oleh model *Sentence embedding* berbasis BERT (*Paraphrase Multilingual*).

Berdasarkan hasil pengujian, judul penelitian yang memiliki konteks tema serupa dengan judul input cenderung memperoleh nilai *Similarity* yang lebih tinggi dan muncul pada peringkat teratas hasil pencarian. Hal ini menunjukkan bahwa representasi *embedding* yang dihasilkan model mampu menangkap makna semantik judul penelitian secara cukup baik, meskipun terdapat perbedaan susunan kata atau istilah yang digunakan. Sebagai contoh, judul penelitian yang sama-sama membahas sistem informasi atau monitoring penelitian tetap teridentifikasi memiliki kemiripan meskipun tidak menggunakan kata kunci yang identik.

Hasil *Similarity* yang ditampilkan sistem juga dipengaruhi oleh pengaturan nilai *threshold* yang ditentukan oleh pengguna. Dengan nilai *threshold* yang lebih tinggi, sistem hanya menampilkan judul penelitian dengan tingkat kemiripan yang signifikan, sehingga membantu pengguna dalam memfokuskan perhatian pada judul-judul yang berpotensi memiliki kesamaan tema. Sebaliknya, penggunaan *threshold* yang lebih rendah menghasilkan cakupan hasil yang lebih luas, meskipun terdapat beberapa judul dengan relevansi yang lebih rendah.

Secara keseluruhan, hasil *Similarity* yang dihasilkan sistem telah sesuai dengan tujuan penelitian, yaitu menyediakan mekanisme otomatis untuk mendeteksi kemiripan judul penelitian sebagai bagian dari proses monitoring tema penelitian di UIN Ar-Raniry Banda Aceh. Sistem ini mampu membantu proses validasi awal judul penelitian dan mengurangi ketergantungan pada proses pengecekan manual yang memerlukan waktu dan ketelitian tinggi.

### 4.4.4 Pengujian Performa Model *Machine learning*

Pengujian performa model *machine learning* difokuskan pada efisiensi pembentukan *embedding* dan proses perhitungan *Similarity*. Model *Paraphrase Multilingual* digunakan tanpa proses pelatihan ulang (*pre-trained* model), sehingga performa model bergantung pada efisiensi inferensi.

*Tabel 4.7 Performa Model dalam Proses Pencarian*

| Proses | Waktu Rata-rata |
|---|---|
| *Embedding* judul input | ± 7.3 detik |
| Perhitungan *Cosine Similarity* | ± 5.1 detik |
| Total proses ML | ± 12.4 detik |

Hasil ini menunjukkan bahwa model cukup ringan dan efisien untuk digunakan dalam sistem berbasis web. Penggunaan model tersebut dapat memberikan keseimbangan antara akurasi representasi makna dan kecepatan pemrosesan.

### 4.4.5 Evaluasi Sistem Secara Keseluruhan

Evaluasi sistem dilakukan untuk menilai efektivitas penerapan *machine learning* dalam mendukung proses monitoring tema penelitian, sekaligus mengidentifikasi kelebihan dan keterbatasan sistem yang dikembangkan. Evaluasi ini mencakup kualitas hasil *Similarity*, dampak implementasi sistem terhadap proses kerja, serta kesesuaian sistem dengan tujuan penelitian yang telah ditetapkan.

Berdasarkan hasil pengujian dan penggunaan sistem, dapat disimpulkan bahwa sistem informasi monitoring tema penelitian yang dikembangkan mampu meningkatkan efisiensi proses pencarian dan pengecekan kesamaan judul penelitian. Pengguna tidak lagi harus menelusuri dokumen Surat Keputusan (SK) penelitian dalam bentuk PDF secara manual, melainkan dapat langsung memperoleh hasil pencarian kemiripan judul secara cepat dan terstruktur melalui sistem berbasis web.

Namun demikian, sistem masih memiliki beberapa keterbatasan. Model *Paraphrase Multilingual* hanya menggunakan judul penelitian sebagai sumber informasi, sehingga dalam beberapa kasus ditemukan judul dengan nilai *Similarity* tinggi tetapi kurang relevan secara konteks (false positive), serta judul yang relevan namun memiliki nilai *Similarity* rendah (false negative). Kondisi ini dipengaruhi oleh keterbatasan informasi semantik yang terkandung dalam judul penelitian yang relatif singkat.

Meskipun terdapat keterbatasan tersebut, sistem ini tetap memberikan nilai tambah yang signifikan dibandingkan proses manual. Sistem mampu menyajikan hasil pencarian secara konsisten, objektif, dan dapat disesuaikan melalui parameter *threshold*. Selain itu, integrasi sistem dengan teknologi *machine learning* memungkinkan proses monitoring tema penelitian dilakukan secara lebih modern dan berbasis data.

Secara keseluruhan, penerapan model *Sentence embedding* berbasis BERT dalam sistem informasi monitoring tema penelitian telah memenuhi tujuan penelitian, yaitu menyediakan solusi pendukung untuk mendeteksi kemiripan judul penelitian secara otomatis. Sistem ini diharapkan dapat menjadi dasar pengembangan lebih lanjut, seperti penambahan data abstrak penelitian atau pengayaan fitur analisis di masa mendatang, guna meningkatkan akurasi dan kualitas hasil monitoring tema penelitian di lingkungan UIN Ar-Raniry Banda Aceh.

جامعة الرانري
AR-RANIRY

# BAB V
# KESIMPULAN DAN SARAN

## 5.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi, serta pengujian sistem yang telah dilakukan, dapat disimpulkan bahwa penerapan *machine learning* menggunakan model *Sentence embedding* berbasis BERT (*Paraphrase Multilingual*) berhasil diimplementasikan dalam pengembangan sistem informasi monitoring tema penelitian di UIN Ar-Raniry Banda Aceh. Sistem ini mampu mengubah judul penelitian menjadi representasi numerik (*embedding*) dan mengukur tingkat kemiripan makna antar judul secara otomatis menggunakan metode *Cosine Similarity*.

Hasil pengujian menunjukkan bahwa sistem dapat menampilkan judul penelitian yang memiliki kemiripan tema dengan judul input pengguna secara relevan dan terstruktur. Penggunaan parameter threshold *Similarity* memberikan fleksibilitas kepada pengguna dalam menyaring hasil pencarian sesuai dengan kebutuhan, sehingga membantu proses validasi judul penelitian secara lebih efisien. Dibandingkan dengan proses manual yang sebelumnya mengandalkan pencarian dokumen PDF, sistem ini mampu mempercepat proses monitoring serta mengurangi potensi terjadinya duplikasi judul penelitian.

Implementasi sistem berbasis web menggunakan *framework Streamlit* menghasilkan antarmuka yang mendukung pelaksanaan seluruh fitur sesuai fungsinya, sebagaimana dibuktikan melalui pengujian *blackbox testing*, serta mampu menampilkan visualisasi hasil Similarity dan menyediakan fitur unduh data. Secara keseluruhan, sistem yang dikembangkan telah memenuhi tujuan penelitian, yaitu menyediakan alat bantu berbasis teknologi *machine learning* untuk mendukung proses monitoring tema penelitian di lingkungan UIN Ar-Raniry Banda Aceh.

## 5.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, terdapat beberapa saran yang dapat dipertimbangkan untuk pengembangan sistem di masa mendatang, antara lain:

1. Sistem dapat dikembangkan lebih lanjut dengan menambahkan data abstrak atau kata kunci penelitian agar analisis kemiripan tidak hanya bergantung pada judul penelitian, sehingga akurasi hasil *Similarity* dapat ditingkatkan.
2. Model *machine learning* yang digunakan dapat dibandingkan dengan model lain atau dilakukan *fine-tuning* menggunakan data penelitian lokal untuk memperoleh hasil *embedding* yang lebih spesifik sesuai dengan konteks penelitian di UIN Ar-Raniry.
3. Fitur pengujian performa sistem dan kualitas hasil *Similarity* dapat dikembangkan lebih lanjut secara kuantitatif, misalnya dengan melibatkan penilaian pakar atau pengguna untuk mengukur tingkat relevansi hasil secara lebih objektif.
4. Sistem informasi monitoring tema penelitian ini dapat diintegrasikan secara langsung dengan sistem informasi LP2M atau basis data penelitian kampus agar proses pembaruan data dapat dilakukan secara otomatis dan berkelanjutan.

Dengan adanya pengembangan lanjutan tersebut, diharapkan sistem ini dapat memberikan manfaat yang lebih optimal dalam mendukung pengelolaan dan monitoring tema penelitian di lingkungan perguruan tinggi.

# DAFTAR PUSTAKA


Adi Saputra, R. (2024). Penerapan Teknik *Machine learning* Dalam Sistem Rekomendasi E-Commerce. *Duniadata.Org*, *1*(4), 1–15. Duniadata.org

Agiharta, K. F., Suteja, B. R., Ayub, M., & Program Studi Magister Ilmu Komputer, Universitas Kristen Maranatha. (2024). Application of *Sentence* BERT for job competency and course similarity. *Jurnal Teknik Informatika Dan Sistem Informasi*, *10*(3), 449–450. http://dx.doi.org/10.28932/jutisi.vXiX.X

Alfarizi, M. R. S., Al-farish, M. Z., Taufiqurrahman, M., Ardiansah, G., & Elgar, M. (2023). Penggunaan Python Sebagai Bahasa Pemrograman untuk *Machine learning* dan Deep Learning. *Karya Ilmiah Mahasiswa Bertauhid (KARIMAH TAUHID)*, *2*(1), 1–6.

Alamanda, R., Suhery, C., & Brianorman, Y. (2021). Aplikasi pendeteksi plagiat terhadap karya tulis berbasis web menggunakan natural language processing dan algoritma Knuth–Morris–Pratt. *Jurnal Coding, Sistem Komputer Untan, 4*(1), 33–44.

Ansyori, M. F., & Mujianto, A. H. (2025). *Penerapan Natural Language Processing (NLP) dengan metode Cosine Similarity pada sistem E-Monev untuk pencarian program pembangunan daerah*. SHIFT: Jurnal Teknologi dan Sistem Informasi, 5(2), 84–102. https://doi.org/10.24252/shift.v5i2.183

Asri, J. S., Amalia, F., Dzaki, M. T., Fikri, M., & Rianisa, A. (2025). Implementasi metode *Cosine Similarity* dalam sistem profiling dosen berbasis data bibliometrik untuk pemetaan kompetensi akademik. *Bulletin of Computer Science Research, 5*(6), 1351–1362. https://doi.org/10.47065/bulletincsr.v5i6.811

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*(Mlm), 4171–4186.

Dewi, R. C. (2024). Penerapan *Machine learning* Dalam Penjadwalan Tugas Pada Sistem Komputer Terdistribusi. *Jurnal Dunia Data*, *1*(4), 1–17. http://www.teknologipintar.org/index.php/duniadata/article/view/82

Diana, R., Warni, H., & Sutabri, T. (2023). Penggunaan Teknologi *Machine learning* Untuk Pelayanan Monitoring Kegiatan Belajar Mengajar Pada Smk Bina Sriwijaya Palembang. *JUTEKIN (Jurnal Teknik Informatika)*, *11*(1). https://doi.org/10.51530/jutekin.v11i1.709

Fatmasari, R., Septiani, R. K., Pinem, T. H., Fabiyanto, D., & Gata, W. (2024). Implementasi Algoritma BERT Pada Komentar Layanan Akademik dan Non Akademik Universitas Terbuka di Media Sosial. *Sains, Aplikasi, Komputasi Dan Teknologi Informasi*, *5*(2), 96. https://doi.org/10.30872/jsakti.v5i2.13915

Fathuddin, M. A. H., Mandyartha, E. P., & Nurlaili, A. L. (2025). Penerapan *Sentence*-BERT dan *Cosine Similarity* untuk pencarian semantik dokumen skripsi dalam format PDF. *Ranah Research Journal,* *8*(1), 322–336. https://doi.org/10.38035/rrj.v8i1

Firmansyah, D. (2024). Penggunaan *Machine learning* Dalam Peramalan Bisnis dan Prediksi Pasar. *Teknologipintar.Org*, *4*(3), 1–20. http://teknologipintar.org/index.php/teknologipintar/article/view/619

Hanum, A. R., Zetha, I. A., Fajrina, J. N., Wulandari, R. A., Putri, C., Andina, S. P., Yudistira, N., & Brawijaya, U. (2024). *MENDETEKSI BERITA HOAKS PERFORMANCE ANALYSIS OF THE BERT TEXT CLASSIFICATION ALGORITHM*. *11*(3), 537–546. https://doi.org/10.25126/jtiik938093

Mahmmod, B. M., Naser, M. A., Shanin Al-Sudani, A. H., Alsabah, M., Mohammed, H. J., Alaskar, H., Almarshad, F., Hussain, A., & Abdulhussain, S. H. (2024). Patient Monitoring System Based on Internet of Things: A Review and

Related Challenges with Open Research Issues. *IEEE Access*, *12*(September). https://doi.org/10.1109/ACCESS.2024.3455900

Mudding, A. A. (2024). Mengungkap Opini Publik: Pendekatan BERT-based-caused untuk Analisis Sentimen pada Komentar Film. *Journal of System and Computer Engineering (JSCE)*, *5*(1), 36–43. https://doi.org/10.61628/jsce.v5i1.1060

Nazar, R. (2024). Implementasi Pemrograman Python Menggunakan Google Colab. *Jurnal Informatika Dan Komputer (JIK)* , *15*(1), 50–56.

Pane, E. S., & Caroline, C. (2024). Optimalisasi evaluasi pelaksanaan pelatihan melalui analisis sentimen otomatis dengan model text classification. *Prosiding PITNAS Widyaiswara*, 1, 141–154.

Sandiwarno, S. (2024). Penerapan *Machine learning* Untuk Prediksi Bencana Banjir. *Jurnal Sistem Informasi Bisnis*, *14*(1), 62–76. https://doi.org/10.21456/vol14iss1pp62-76

Saputra, P. S., Budiawan, W., & Sriyanto, S. (2016). Perancangan Sistem Informasi Monitoring Status Pembayaran Apartemen Berbasis Web (Studi Kasus: Paltrow City, Semarang). *Industrial Engineering Online Journal*, *5*(1).

Sihombing, S. H., & Nasution, M. A. E. (2024). Implementasi metode *Cosine Similarity* untuk deteksi kemiripan data pada dokumen Word dan perbedaan isi dokumen Word. *JID (Jurnal Info Digit), 2*(2), 736–748.

Sitanggang, S. S., Defit, S., & Ramadhan, M. (2021). Analisis Optimasi Fungsi Pelatihan *Machine learning* Neural Network dalam Peramalan Kemiskinan. *Jurnal Edukasi Dan Penelitian Informatika (JEPIN)*, *7*(3), 359. https://doi.org/10.26418/jp.v7i3.50092

Wijoyo A, Saputra A, Ristanti S, Sya'ban S, Amalia M, & Febriansyah R. (2024). Pembelajaran *Machine learning*. *OKTAL (Jurnal Ilmu Komputer Dan Science)*, *3*(2), 375–380. https://journal.mediapublikasi.id/index.php/oktal/article/view/2305

Yanuar, R. A. A. (2024). *Jurnal Teknik Informatika, Vol. 16, No. 2, April 2024*. *16*(2), 1–7. https://ejurnal.ulbi.ac.id/index.php/informatika/article/view/3533

Wuryanto, Thamrin, H., Wantoro, J., & Magister Informatika, Fakultas Komunikasi dan Informatika Universitas Muhammadiyah Surakarta. (2026). Algoritma Deep Learning untuk Mengukur Tingkat Kesesuaian Rumusan dan Asesmen Capaian Pembelajaran Mata Kuliah. In *Jurnal PETISI: Vol. Vol. 07* (Issue No. 01) [Journal-article].

جامعة الرانري
AR-RANIRY