KLASIFIKASI *AUTHOR NAME DISAMBIGUATION* MENGGUNAKAN ALGORITMA VARIATIONAL AUTOENCODER DAN DEEP NEURAL NETWORK

TESIS

HAMID RAHMAN
ENTERPRISE SOFTWARE ENGINEERING
24242002P

PROGRAM STUDI TEKNIK INFORMATIKA – S2
PROGRAM PASCASARJANA
UNIVERSITAS BINA DARMA
PALEMBANG
TAHUN 2026

KLASIFIKASI *AUTHOR NAME DISAMBIGUATION* MENGGUNAKAN ALGORITMA VARIATIONAL AUTOENCODER DAN DEEP NEURAL NETWORK

Tesis ini diajukan sebagai salah satu syarat
untuk memperoleh gelar

MAGISTER KOMPUTER

HAMID RAHMAN
ENTERPRISE SOFTWARE ENGINEERING
24242002P

PROGRAM STUDI TEKNIK INFORMATIKA – S2
PROGRAM PASCASARJANA
UNIVERSITAS BINA DARMA
PALEMBANG
2026

Halaman Pengesahan Pembimbing Tesis

Judul Tesis: **KLASIFIKASI *AUTHOR NAME DISAMBIGUATION* MENGGUNAKAN ALGORITMA VARIATIONAL AUTOENCODER DAN DEEP NEURAL NETWORK**

Oleh HAMID RAHMAN, NIM 24242002P, Tesis ini telah disetujui dan disahkan oleh Pembimbing Program Studi Teknik Informatika – S2 konsentrasi ENTERPRISE SOFTWARE ENGINEERING, Program Pascasarjana Universitas Bina Darma pada 26 Februari 2026 dan telah dinyatakan LULUS.

Palembang, 26 Februari 2026
Mengetahui,
Program Studi Teknik Informatika – S2
Universitas Bina Darma
Ketua,

Dr. Usman Ependi, S.Kom., M.Kom.

Pembimbing,

Dr. Usman Ependi, S.Kom., M.Kom.

Halaman Pengesahan Penguji Tesis

Judul Tesis: **KLASIFIKASI *AUTHOR NAME DISAMBIGUATION* MENGGUNAKAN ALGORITMA VARIATIONAL AUTOENCODER DAN DEEP NEURAL NETWORK**

Oleh HAMID RAHMAN , NIM 24242002P, Tesis ini telah disetujui dan disahkan oleh Tim Penguji Program Studi Teknik Informatika – S2 konsentrasi ENTERPRISE SOFTWARE ENGINEERING, Program Pascasarjana Universitas Bina Darma pada 26 Februari 2026 dan telah dinyatakan LULUS.

Palembang, 26 Februari 2026
Mengetahui,
Program Pascasarjana
Universitas Bina Darma
Direktur,

PROGRAM PASCASARJANA

Prof. Dr. Ir. Achmad Syarifudin, M.Sc.

Penguji I,

Dr. Usman Ependi, M.Kom.

Penguji II,

Prof. Dr. Edi Surya Negara, M.Kom.

Penguji III,

Dr. A. Haidar Mirza, S.T., M.Kom.

SURAT PERNYATAAN

Saya yang bertanda tangan dibawah ini:

Nama : HAMID RAHMAN
NIM : 24242002P

Dengan ini menyatakan bahwa:

1. Karya tulis Saya (Tesis, Skripsi, Tugas Akhir) ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik baik (Magister, Sarjana, dan Ahli Madya) di Universitas Bina Darma;
2. Karya tulis ini murni gagasan, rumusan dan penelitian Saya sendiri dengan arahan tim pembimbing;
3. Dalam karya tulis ini tidak terdapat karya atau pendapat yang telah ditulis atau dipublikasikan orang lain, kecuali secara tertulis dengan jelas dikutip dengan mencantumkan nama pengarang dan memasukkan ke dalam daftar pustaka;
4. Karena yakin dengan keasliaan karya tulis ini, Saya menyatakan bersedia Tesis/Skripsi/Tugas Akhir, yang Saya hasilkan di unggah ke internet;
5. Surat Pernyataan ini Saya tulis dengan sungguh-sungguh dan apabila terdapat penyimpangan atau ketidakbenaran dalam pernyataan ini, maka Saya bersedia menerima sanksi dengan aturan yang berlaku di perguruan tinggi ini.

Demikian Surat Pernyataan ini saya buat agar dapat dipergunakan sebagaimana mestinya.

Palembang, 26 Februari 2026
Yang Membuat Pernyataan,

HAMID RAHMAN
NIM: 24242002P

ABSTRAK

Ambiguitas Nama Penulis atau *Author Name Disambiguation* (AND) merupakan tantangan fundamental yang memengaruhi integritas dan kualitas data dalam ekosistem perpustakaan digital. Pendekatan pencocokan berpasangan (*pairwise matching*) digunakan dalam menyelesaikan masalah tersebut, tetapi muncul kendala signifikan berupa ketidakseimbangan data (*imbalanced data*) yang ekstrem, di mana jumlah pasangan penulis berbeda (kelas mayoritas) sangat mendominasi pasangan penulis sama (kelas minoritas). Kondisi ini mengakibatkan model klasifikasi menjadi bias dan gagal mengidentifikasi penulis yang sebenarnya. Penelitian ini bertujuan mengembangkan solusi untuk mengatasi ketidakseimbangan data tersebut menggunakan algoritma *Variational Autoencoder* (VAE) sebagai teknik *generative oversampling*, serta mengintegrasikannya dengan *Deep Neural Network* (DNN) untuk proses klasifikasi. Metodologi penelitian memanfaatkan dataset DBLP yang melalui tahap pra-pemrosesan serta ekstraksi fitur kesamaan leksikal dan jarak temporal. Tiga skenario pengujian dirancang untuk membandingkan kinerja model: *baseline* (tanpa penanganan), metode VAE, dan metode SMOTE sebagai pembanding . Hasil eksperimen menunjukkan bahwa model *baseline* mengalami kegagalan total dalam mendeteksi kelas minoritas dengan nilai *F1-Score* 0.00. Sebaliknya, penerapan algoritma VAE terbukti paling efektif dengan mencapai *F1-Score* agregat sebesar 0.9997, sedikit lebih unggul dibandingkan metode SMOTE yang mencapai 0.9992. Penelitian ini menyimpulkan bahwa integrasi VAE dan DNN mampu menghasilkan data sintetis berkualitas tinggi yang menyeimbangkan distribusi kelas, sehingga secara signifikan meningkatkan akurasi dan sensitivitas sistem dalam menyelesaikan masalah disambiguasi nama penulis.

Kata Kunci: Ambiguitas Nama Penulis, *Imbalanced Data*, *Variational Autoencoder*, *Deep Neural Network*, *Oversampling*.

ABSTRACT

Author Name Disambiguation (AND) is a fundamental challenge affecting data integrity and quality within digital library ecosystems. In addressing this issue using a pairwise matching approach, a significant obstacle arises in the form of extreme data imbalance, where the number of different author pairs (majority class) vastly dominates the same author pairs (minority class). This condition causes classification models to become biased and fail to identify the actual authors. This study aims to develop a solution to overcome this data imbalance using the Variational Autoencoder (VAE) algorithm as a generative oversampling technique, integrating it with a Deep Neural Network (DNN) for the classification process. The research methodology utilizes the DBLP dataset, which undergoes pre-processing and feature extraction for lexical similarity and temporal distance. Three testing scenarios are designed to compare model performance: baseline (without data handling), the VAE method, and the SMOTE method as a benchmark. Experimental results indicate that the baseline model completely failed to detect the minority class with an F1-Score of 0.00. Conversely, the application of the VAE algorithm proved to be the most effective, achieving an aggregate F1-Score of 0.9997, slightly outperforming the SMOTE method which reached 0.9992. This study concludes that the integration of VAE and DNN successfully generates high-quality synthetic data that balances class distribution, thereby significantly improving the accuracy and sensitivity of the system in resolving author name disambiguation problems.

Kata Kunci: *Author Name Disambiguation, Imbalanced Data, Variational Autoencoder, Deep Neural Network, Oversampling.*

MOTTO DAN PERSEMBAHAN

"We cannot change what is done, so we must insist upon the truth and be loyal to what matters; in the end, we stand firm in our integrity and our effort."

Segala puji dan syukur penulis panjatkan ke hadirat Allah Swt. atas segala limpahan rahmat, hidayah, dan karunia-Nya. Berkat kemudahan dari-Nya, yang terwujud melalui dukungan berbagai pihak, penulis akhirnya dapat menyelesaikan penelitian Tesis ini.

Melalui kerja keras dan doa yang tak terputus, dengan segenap kerendahan hati dan rasa syukur, penulis mempersembahkan karya sederhana ini secara khusus kepada:

1. Kedua Orang Tua Tercinta, Ayahanda dan Ibunda. Yang telah mendidik dengan penuh kasih sayang, mengiringi dengan doa yang tak pernah putus, serta menjadi sumber kekuatan tak ternilai dalam setiap langkah penulis.
2. Keluarga dan Adik Penulis**.** Atas segenap hati, dukungan tulus, dan semangat yang senantiasa diberikan.
3. Para Rekan-rekan Seperjuangan. Terima kasih telah menjadi mitra diskusi dan saling memotivasi penulis dalam menyelesaikan pendidikan magister ini.

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Tuhan Yang Maha Esa, karena atas rahmat dan karunia-Nya, penulis dapat menyelesaikan penyusunan tesis yang berjudul **"Klasifikasi *Author Name Disambiguation* Menggunakan Algoritma *Variational Autoencoder* dan *Deep Neural Network*"**. Tesis ini disusun sebagai salah satu syarat untuk menyelesaikan Program Studi Magister Teknik Informatika di Universitas Bina Darma Palembang. Penulis menyadari bahwa kelancaran proses ini tidak terlepas dari bimbingan, arahan, dan dukungan dari berbagai pihak. Oleh karena itu, dengan segala kerendahan hati, penulis ingin menyampaikan ucapan terima kasih yang tulus kepada:

1. Ibu Prof. Dr. Sunda Ariana, M.Pd.,M.M. Selaku Rektor Universitas Bina Darma Palembang
2. Bapak Prof. Dr. Ir. Achmad Syarifudin, M.Sc. Selaku Direktur Pascasarja Universitas Bina Darma Palembang
3. Bapak Dr. Usman Ependi, S.Kom., M.Kom., selaku Dosen Pembimbing dan Ketua Program Studi Magister Teknik Informatika, yang telah meluangkan waktu, tenaga, dan pikiran untuk memberikan bimbingan, arahan, serta motivasi kepada penulis dalam penulisan tesis ini.
4. Bapak Prof. Dr. Edi Surya Negara Harahap, M.Kom. selaku penguji yang telah memberikan bimbingan serta masukan berharga dalam proses penyelesaian tesis ini.
5. Bapak Dr. A Haidar Mirza, S.T.,M.Kom. selaku penguji yang telah memberikan motivasi dan arahan dalam penulisan tesis ini.
6. Pihak Sekretariat Pascasarjana Universitas Bina Darma Palembang yang telah memberikan bimbingan pelayanan dengan baik.

Palembang, 26 Februari 2026

Hamid Rahman

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

DAFTAR LAMPIRAN

BAB I
PENDAHULUAN

1.1. Latar Belakang

Di era digital saat ini, perpustakaan digital (*digital library*) telah menjadi pilar penting dalam ekosistem akademik. Keunggulannya, seperti kapasitas penyimpanan yang masif dengan biaya rendah serta kemudahan aksesibilitas data kapan pun dan di mana pun, telah mendorong pertumbuhannya secara eksponensial (Weiss, 2016). Repositori – repositori besar seperti DBLP, Google Scholar, PubMed, dan CiteSeer tidak hanya berfungsi sebagai gudang literatur, tetapi juga menyediakan layanan fungsional yang memfasilitasi penemuan ilmiah (Ferreira dkk., 2020). Namun, di balik manfaat tersebut, perpustakaan digital juga menghadapi tantangan signifikan terkait integritas dan kualitas data. Berbagai jenis kesalahan sering terjadi, mulai dari kesalahan tipografis, hasil pemindaian (*scanning*) yang tidak sempurna, hingga inkonsistensi metadata akibat format kutipan yang berbeda (Ferreira dkk., 2012). Di antara berbagai tantangan ini, salah satu masalah paling krusial dan persisten adalah ambiguitas nama penulis.

Disambiguasi Nama Penulis atau *Author Name Disambiguation* (AND) adalah proses untuk menentukan secara akurat apakah sekumpulan nama penulis dalam berbagai publikasi merujuk pada individu yang sama atau berbeda (Lewis dkk., 2006; Zhu dkk., 2014). Masalah ini muncul karena berbagai faktor, seperti adanya penulis dengan nama yang identik atau mirip, variasi penulisan nama (misalnya, penggunaan inisial), perubahan nama pribadi, kesalahan pengetikan, hingga transliterasi nama dari aksara non-Romawi (McKay dkk., 2010; Shoaib dkk., 2020). Secara spesifik, ambiguitas ini terwujud dalam dua bentuk utama (Hussain & Asghar, 2017; Shen dkk., 2017) yaitu: Sinonim (*Synonyms*) satu penulis yang sama muncul dengan beberapa variasi nama yang berbeda di berbagai publikasi dan Homonim (*Homonyms*) beberapa penulis yang berbeda memiliki nama yang sama persis. kegagalan dalam mengatasi masalah AND dapat menyebabkan atribusi karya yang salah dan analisis sitasi yang tidak akurat.

Berbagai solusi telah diusulkan untuk mengatasi masalah AND, yang secara umum dapat dikategorikan sebagai pendekatan *supervised learning*, *unsupervised learning*, dan berbasis graph. Di antara berbagai metode tersebut, teknik pencocokan berpasangan (*pairwise matching*) telah menjadi salah satu pendekatan yang paling dominan dan efektif. Pendekatan ini bekerja dengan membandingkan dua entri publikasi secara langsung untuk menentukan apakah keduanya merujuk pada individu penulis yang sama. (K. Kim dkk., 2019) menggunakan pendekatan *pairwise matching* dengan memasangkan publikasi yang terkait untuk menentukan penulis yang sama atau tidak menggunakan fitur *structure-based Term Frequency-Inverse Document Frequency* (TF-IDF) yang selanjutnya dilakukan pencarian kesamaan antar atribut menggunakan *cosine similarity*. Penelitian ini menggunakan algoritma *Support Vector Machine* (SVM)*, Random Forest* (RF)*, Gradient Boost Tree* (GBT)*,* dan *Deep Neural Network* (DNN) dalam melakukan klasifikasi AND. Hasil dari penelitian ini mendapatkan nilai MAP sebesar 0.8929 pada algoritma GBT, 0.8907 pada algoritma RF, 0.8878 pada algoritma SVM, dan 0.8771 pada algoritma DNN. (Boukhers & Asundi, 2023) menggunakan teknik *Char2Vec* untuk membandingkan dua referensi dengan memanfaatkan representasi vektor dari fitur seperti *co-authors,* judul, dan sumber publikasi. Penggunaan *embedding character Char2Vec* digunakan untuk mencari representasi vektor dari fitur. Penelitian ini juga menggunakan algoritma *neural network* dalam melakukan klasifikasi. (Ebrahimi dkk., 2023) menggunakan teknik *pairwise* dalam menemukan kecocokan antar penulis dengan menentukan prioritas dan bobot dari kriteria data yang kemudian dilakukan klasifikasi menggunakan algoritma *Analytical Hierarchy Process* dan *Fuzzy Delphi Method*.

Meskipun pendekatan *pairwise matching* terbukti efektif, metode ini secara inheren menciptakan sebuah tantangan baru yang signifikan, yaitu ketidakseimbangan data (*imbalanced data*). Sebagai representasi empiris dari masalah tersebut, penelitian ini menggunakan *dataset* berlabel turunan DBLP (*DBLP-derived labeled data*) yang dikembangkan oleh (J. J. Kim & Kim, 2018). *Dataset* ini dipilih karena secara eksplisit mendemonstrasikan permasalahan ketimpangan data pada skenario pelatihan sistem *author name disambiguation*, di

mana implementasi pencocokan berpasangan menghasilkan rasio yang sangat asimetris antara jumlah pasangan penulis yang berbeda (kelas mayoritas) berbanding pasangan penulis yang sama (kelas minoritas). Ketidakseimbangan ekstrem ini dapat menyebabkan model klasifikasi menjadi bias, cenderung memprediksi kelas mayoritas, dan pada akhirnya gagal mengenali kasus positif yang langka namun krusial (J. J. Kim & Kim, 2018; Manzoor dkk., 2022). Karena banyak penelitian sebelumnya tidak secara eksplisit menangani masalah kelas tidak seimbang ini secara mendalam pada tahap pra-pemrosesan, kondisi ini menciptakan celah penelitian (*research gap*) yang jelas untuk dieksplorasi.

Untuk mengisi celah tersebut, penelitian ini mengusulkan sebuah kerangka kerja yang secara spesifik dirancang untuk mengatasi masalah data tidak seimbang dalam konteks klasifikasi AND pada himpunan data DBLP. Solusi yang diajukan menerapkan algoritma *Variational Autoencoder* (VAE), sebuah model generatif *deep learning*, sebagai teknik *oversampling* tingkat lanjut. VAE dipilih karena kemampuannya yang unggul dalam mempelajari distribusi fitur tersembunyi (*latent space*) dari kelas minoritas. Dengan melatih model VAE secara eksklusif pada himpunan data kelas minoritas, algoritma ini mampu membangkitkan sampel-sampel data sintetis baru yang bervariasi dan berkualitas tinggi, sehingga menyeimbangkan proporsi kelas sebelum tahap klasifikasi (Ai dkk., 2023; Utyamishev & Partin-Vaisband, 2019).

Dengan demikian, penelitian ini mengusulkan sebuah alur kerja dua tahap: pertama, menyeimbangkan dataset berpasangan menggunakan VAE, dan kedua, melatih sebuah *Deep Neural Network* (DNN) pada dataset yang telah seimbang tersebut untuk melakukan klasifikasi akhir. Pendekatan hibrida ini diharapkan dapat menghasilkan model klasifikasi AND yang lebih kuat, akurat, dan tidak bias, sehingga memberikan solusi yang lebih optimal dibandingkan metode-metode sebelumnya yang tidak secara khusus menangani tantangan data tidak seimbang.

1.2. Identifikasi Masalah

Berdasarkan latar belakang yang telah dijelaskan, identifikasi masalah dari penelitian ini adalah :

1) Ambiguitas Nama Penulis

   Kesalahan dalam membedakan penulis dikarenakan mempunyai banyak penamaan cara yang sama dapat mengakibatkan kesalahan dalam atribusi publikasi. Kesalahan ambiguitas nama penulis mempunyai dua bentuk yaitu: sinonim dan homonim. Sinomin menjadi masalah karena penulis yang sama dapat tercatat dengan nama yang berbeda akibat transliterasi, perubahan nama, atau variasi gaya tulisan. Sedangkan, homonim menjadi masalah karena penulis yang berbeda tetapi memiliki nama yang sama sehingga dapat menyebabkan kebingungan dalam hal pengelompokkan publikasi.

2) Ketidakseimbangan Data

   Ketidakseimbangan data (*imbalanced data*) menjadi salah satu tantangan utama dalam melakukan klasifikasi *author name disambiguation*. Tahapan *Pre-Processing* data sebelum melakukan klasifikasi pada *author name disambiguation* sering menghasilkan data yang tidak seimbang antara kasus positif dan negatif sehingga dapat membuat sistem klasifikasi tidak akurat dan efisien, risiko *overfitting* pada kelas mayoritas, dan kesulitan dalam mendeteksi kasus ambiguitas yang relevan.

1.3. Perumusan Masalah

Berdasarkan latar belakang masalah yang telah diuraikan, rumusan masalah dalam penelitian ini adalah sebagai berikut:

1) Bagaimana masalah ketidakseimbangan data memengaruhi efektivitas klasifikasi *author name disambiguation*?
2) Bagaimana cara mengatasi masalah data yang tidak seimbang (*imbalanced data*) guna meningkatkan akurasi identifikasi penulis yang ambigu?
3) Bagaimana mengembangkan arsitektur model klasifikasi menggunakan algoritma *deep learning* untuk menyelesaikan masalah *author name disambiguation*?
4) Sejauh mana integrasi algoritma *variational autoencoder* (VAE) dengan *deep neural network* (DNN) dapat meningkatkan performa klasifikasi dalam menyelesaikan masalah AND ?

1.4. Tujuan Penelitian

Tujuan umum penelitian ini adalah menghasilkan sebuah sistem untuk menangani data yang tidak seimbang pada tahap pemrosesan menggunakan algoritma VAE, serta untuk melakukan klasifikasi *author name disambiguation* dengan algoritma DNN. Guna mencapai tujuan umum tersebut, penelitian ini memiliki rincian tujuan sebagai berikut:

1) Mengembangkan solusi untuk mengatasi masalah *imbalanced data* dalam *author name disambiguation* (AND) guna meningkatkan akurasi identifikasi penulis yang ambigu.
2) Menerapkan algoritma *variational autoencoder* (VAE) untuk menghasilkan data sintetis dari kelas minoritas, sehingga dapat menyeimbangkan dataset pada masalah AND.
3) Mengintegrasikan algoritma *variational autoencoder* (VAE) dengan *deep neural network* (DNN) untuk meningkatkan performa klasifikasi dalam menyelesaikan masalah AND.
4) Mengukur kinerja dari model klasifikasi yang telah dirancang menggunakan *metric performance* dengan parameter *accuracy, precision, recall, specificity* dan *f-measure.*

1.5. Kebaruan (*Novelty*)

Kebaruan dalam penelitian ini terletak pada penggunaan algoritma *Variational Autoencoder* (VAE) dalam menyelesaikan masalah ketidakseimbangan data dalam proses klasifikasi *author name disambiguation* (AND. Pendekatan ini menawarkan solusi inovatif untuk mengatasi tantangan ketidakseimbangan data (*imbalanced data*) dengan memanfaatkan kemampuan VAE untuk menghasilkan data sintetis bagi kelas minoritas sekaligus menangkap struktur laten dari dataset. Selain itu, penelitian ini mengintegrasikan VAE dengan Deep Neural Network (DNN) untuk memanfaatkan kekuatan VAE dalam menyeimbangkan data dan kemampuan DNN dalam mengenali pola kompleks, sehingga meningkatkan performa klasifikasi secara keseluruhan. Kombinasi teknik generatif dan

diskriminatif yang diusulkan ini memberikan kontribusi baru yang efektif dalam menyelesaikan tantangan AND, baik dalam aspek teknis maupun implementasinya.

1.6. Manfaat Penelitian

Manfaat utama penelitian ini adalah menyediakan landasan teoretis dan praktis untuk penerapan algoritma VAE dalam menangani data tidak seimbang serta algoritma DNN dalam klasifikasi disambiguasi nama penulis. Secara lebih rinci, manfaat penelitian ini dijabarkan sebagai berikut:

1) Memberikan kontribusi penelitian di dalam bidang ambiguasi nama penulis.
2) Dapat menjadi salah satu solusi dalam mengatasi masalah data yang tidak seimbang dengan menggunakan dengan menggunakan algoritma VAE serta algoritma DNN dalam melakukan klasifikasi *author name disambiguation.*
3) Hasil dari penelitian ini dapat menjadi referensi untuk penelitian selanjutnya yang terkait.

1.7. Pembatasan Masalah

Batasan masalah dalam penelitian ini adalah sebagai berikut:

1) Penelitian ini berfokus pada identifikasi penulis (*author*) dalam konteks masalah disambiguasi nama penulis (*Author Name Disambiguation*).
2) Proses pra-pemrosesan data menggunakan teknik pencocokan berpasangan *(pairwise matching),* kombinasi*,* dan pengukuran kesamaan data *(similarity).*
3) Sumber data yang digunakan adalah dataset bibliografi DBLP Labeled Data (J. J. Kim & Kim, 2018) yang telah melalui tahap pembersihan, dengan atribut yang dimanfaatkan meliputi: *author name, author id, paper id, author, year, venue, title, NaN*.
4) Validasi kinerja model akan dievaluasi menggunakan metrik performa yang terdiri atas parameter *accuracy, precision, recall, specificity,* dan *F-measure*.

1.8. Sistematika Penulisan

Sistematika penulisan ini bertujuan untuk memberikan gambaran yang jelas mengenai struktur dan isi setiap bab dalam penelitian. Adapun rincian sistematika tersebut adalah sebagai berikut:

BAB I PENDAHULUAN

Bab ini menguraikan dasar pemikiran penelitian yang mencakup latar belakang, identifikasi dan perumusan masalah, tujuan, serta manfaat yang diharapkan. Selain itu, bab ini juga menjelaskan kebaruan (novelty) yang ditawarkan, batasan masalah untuk menjaga fokus penelitian, dan diakhiri dengan sistematika penulisan yang menjadi panduan struktur laporan.

BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI

Bab ini menyajikan dua bagian utama. Bagian pertama adalah tinjauan pustaka yang merangkum penelitian-penelitian terdahulu yang relevan dengan topik disambiguasi nama penulis. Bagian kedua memaparkan landasan teori yang kokoh mengenai konsep dan teknologi yang digunakan, seperti *machine learning*, *deep learning*, *Variational Autoencoder* (VAE), *Deep Neural Network* (DNN), hingga metode evaluasi model.

BAB III METODOLOGI PENELITIAN

Bab ini menjelaskan secara rinci mengenai rancangan dan langkah-langkah pelaksanaan penelitian. Pembahasan mencakup sumber data yang digunakan, prosedur penelitian yang sistematis mulai dari persiapan data, pra-pemrosesan, ekstraksi fitur, penanganan data tidak seimbang menggunakan VAE, hingga proses klasifikasi dengan DNN.

BAB IV HASIL DAN ANALISA

Bab ini memaparkan seluruh hasil dari eksperimen yang telah dilakukan. Hasil tersebut mencakup kinerja model pada data latih dan data uji yang kemudian dianalisis secara mendalam. Pada bab ini, validitas dan performa sistem yang dibangun akan diukur dan dibahas menggunakan berbagai metrik evaluasi untuk menguji hipotesis penelitian.

BAB V KESIMPULAN

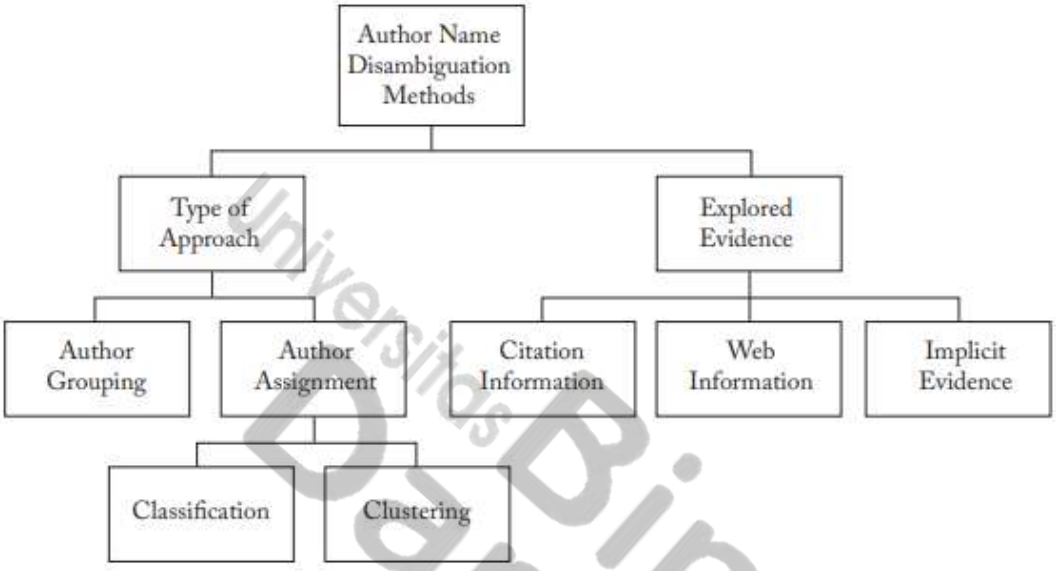
Bab ini berisi rangkuman dari keseluruhan hasil penelitian dan analisis. Peneliti akan menarik kesimpulan yang menjawab rumusan masalah dan tujuan penelitian. Selain itu, bab ini juga akan mengemukakan keterbatasan yang dihadapi selama penelitian serta memberikan saran untuk pengembangan penelitian selanjutnya di masa mendatang.

BAB II
TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1. Tinjauan Pustaka

Author Name Disambiguation (AND) adalah proses krusial untuk mengidentifikasi dan membedakan secara akurat apakah sekumpulan entri nama penulis dalam berbagai publikasi merujuk pada individu yang sama atau berbeda. Masalah ini merupakan tantangan fundamental dalam pengelolaan perpustakaan digital, karena ambiguitas nama dapat secara signifikan mengurangi kualitas dan reliabilitas informasi yang disajikan (Tran dkk., 2014). Kegagalan dalam mengatasi masalah AND berdampak langsung pada fungsionalitas inti repositori bibliografi, seperti akurasi hasil pencarian, penelusuran sitasi, dan sistem rekomendasi literatur (Ferreira dkk., 2020). Ambiguitas ini sendiri disebabkan oleh berbagai faktor, termasuk variasi penulisan nama, penggunaan inisial, kesalahan tipografis, perubahan nama pribadi, hingga transliterasi nama dari aksara non-Romawi (Gomide dkk., 2017; McKay dkk., 2010; Shoaib dkk., 2020). Secara spesifik, masalah ini terwujud dalam dua bentuk utama: Sinonim (*Synonyms*), di mana satu penulis yang sama memiliki beberapa variasi nama, dan Homonim (*Homonyms*), di mana beberapa penulis berbeda memiliki nama yang sama persis (Hussain & Asghar, 2018b; Shen dkk., 2017).

Untuk mengatasi masalah ini, secara umum terdapat dua pendekatan besar yang diusulkan dalam literatur (Ferreira dkk., 2010; Zhao dkk., 2017). Pendekatan pertama adalah Pengelompokan Penulis (*Author Grouping*) yang bersifat *unsupervised*, di mana tujuannya adalah untuk mengelompokkan berbagai entri publikasi yang diyakini milik penulis yang sama ke dalam satu klaster menggunakan algoritma *clustering*. Pendekatan kedua adalah Penugasan Penulis (*Author Assignment*) yang bersifat *supervised* dan membingkai AND sebagai masalah klasifikasi. Pendekatan kedua ini seringkali mengandalkan teknik pencocokan berpasangan (*pairwise matching*), di mana dua entri dibandingkan untuk diklasifikasikan sebagai "Sama" atau "Berbeda".



Gambar 2.1 Taxonomy Author Name Disambiguation Method

Berbagai penelitian telah mengeksplorasi beragam metode untuk disambiguasi nama penulis, yang dapat diklasifikasikan ke dalam beberapa pendekatan utama. Dalam kerangka kerja berbasis kemiripan (*similarity*) dan perbandingan berpasangan (*pairwise*), teknik-teknik pembelajaran mendalam (*deep learning*) menunjukkan efektivitas yang signifikan. Penelitian oleh (Boukhers & Asundi, 2023; Yamani, Nurmaini, Firdaus, dkk., 2020)mengaplikasikan *Neural Network* untuk tugas klasifikasi. Secara spesifik, (Firdaus dkk., 2022) menerapkan arsitektur *Capsule Network* dan memperoleh tingkat akurasi yang tinggi. Selain itu, (Yamani, Nurmaini, & Rini, 2020) juga melakukan pengujian terhadap metode *ensemble*, yaitu *Isolation Forest*. Pendekatan lain yang berbeda diusulkan oleh (Ebrahimi dkk., 2023) yang menggunakan metode pengambilan keputusan multikriteria, seperti *Analytical Hierarchy Process (AHP)* dan *Fuzzy Delphi*.

Pada pendekatan tanpa pengawasan (*unsupervised*), metode pengelompokan (*clustering*) merupakan salah satu teknik yang umum digunakan. Penelitian yang dilakukan oleh (Fan & Li, 2021; Zhou dkk., 2024) sama-sama mengimplementasikan *Hierarchical Agglomerative Clustering (HAC)* untuk mengelompokkan entitas penulis. Seiring dengan perkembangan penelitian, metode berbasis graf yang lebih modern mulai diaplikasikan untuk menangkap hubungan struktural yang kompleks. Sebagai contoh, (Chen dkk., 2021) menggunakan *Graph*

Convolutional Network (GCN), sedangkan (Ma & Xia, 2023) memanfaatkan *Graph Autoencoder*. Terakhir, studi komparatif oleh (D'Angelo & van Eck, 2020) turut melengkapi lanskap riset dengan mengevaluasi metode CvE dan DGA, yang secara keseluruhan menunjukkan keragaman teknik untuk mengatasi permasalahan disambiguasi nama penulis. Tabel 2.1 menyajikan rangkuman beberapa teknik klasifikasi *author name disambiguation* yang diteliti dalam beberapa tahun terakhir.

Tabel 2.1 Rangkuman Penelitian Terkait Author Name Disambiguation

No	Peneliti	Metode	Implementasi	Hasil
1	(D'Angelo & van Eck, 2020)	CvE Method DGA Method	Pairwise - Similarity	97.6%
2	(Yamani, Nurmaini, & Rini, 2020)	Isolation Forest	Pairwise - Similarity	99.5%
3	(Yamani, Nurmaini, Firdaus, dkk., 2020)	Deep Neural Network	Pairwise - Similarity	98.0%
4	(Fan & Li, 2021)	HAC	Similarity	92.58%
5	(Chen dkk., 2021)	GCN	Similarity	95.38%
6	(Firdaus dkk., 2022)	Capsule Network	Pairwise	99.80%
7	(Ma & Xia, 2023)	Graph Autoencoder	Similarity	93.0%
8	(Boukhers & Asundi, 2023)	Neural Network	Pairwise - Similarity	96,1%
9	(Ebrahimi dkk., 2023)	AHP Fuzzy Delphi Method Triangular Fuzzy Numbers	Pairwise	99.0%
10	(Zhou dkk., 2024)	HAC	Pairwise	83.57%

2.2. Landasan Teori

2.2.1. *Machine Learning*

Machine Learning (ML) adalah tipe dari *artificial intelligence* (AI) yang memungkinkan aplikasi perangkat lunak menjadi lebih akurat dalam memprediksi hasil. *Machine Learning* merupakan studi yang menerapkan algoritma pada sistem komputer untuk dapat menyelesaikan task tertentu tanpa instruksi secara eksplisit (Shalev-Shwartz & Ben-David, 2013). Prinsip kerjanya adalah dengan membangun sebuah model matematis melalui proses pelatihan (*training*), yang kemudian digunakan untuk mengenali pola. Dua paradigma utamanya mencakup *Supervised Learning* (Pembelajaran Terarah), yang memanfaatkan data berlabel untuk melakukan tugas prediksi seperti klasifikasi dan regresi, serta *Unsupervised Learning* (Pembelajaran Tak Terarah), yang bertugas menemukan struktur tersembunyi seperti pengelompokan (*clustering*) pada data tak berlabel (Jordan & Mitchell, 2015), Selain kedua metode tersebut, terdapat pula paradigma *Reinforcement Learning* (Pembelajaran Penguatan), di mana sebuah agen komputasi belajar untuk mengambil serangkaian tindakan dalam suatu lingkungan demi memaksimalkan imbalan kumulatif (Hastie dkk., 2009).

Meskipun setiap paradigma mempunyai pendekatan yang berbeda, inti dari ketiga paradigma dapat menciptakan model yang mampu menerapkan pengetahuan yang diperoleh dari data latih untuk membuat prediksi atau keputusan yang akurat terhadap data baru yang belum pernah ditemui sebelumnya. Kemampuan generalisasi inilah yang menjadikan ML sebagai teknologi transformatif di berbagai bidang, mulai dari analisis bisnis hingga penelitian ilmiah.

2.2.2. *Deep Learning*

Deep Learning (DL) atau Pembelajaran Mendalam adalah sub-bidang dari *Machine Learning* yang arsitekturnya didasarkan pada Jaringan Saraf Tiruan (*Artificial Neural Networks*) dengan banyak lapisan (multi-lapisan). Meskipun konsep jaringan saraf sudah ada sejak lama, titik balik yang mempopulerkan *Deep Learning* modern terjadi sekitar tahun 2006 melalui karya perintisan Geoffrey Hinton dan rekan-rekannya mengenai *Deep Belief Networks (Hinton dkk., 2006)*.

Sebelum dikenal luas sebagai *Deep Learning*, pendekatan ini sering disebut sebagai pembelajaran hierarkis (*hierarchical learning*), yang kemampuannya adalah mempelajari berbagai level abstraksi dan representasi data secara otomatis, sangat efektif untuk memahami data kompleks seperti gambar, suara, dan teks (Deng & Yu, 2013).

Istilah "deep" (mendalam) dalam *Deep Learning* memiliki beberapa makna yang saling melengkapi. Interpretasi paling umum merujuk pada arsitektur jaringan saraf yang memiliki banyak lapisan tersembunyi (*hidden layers*), tidak hanya satu atau dua. Namun, makna yang lebih mendasar dan berdampak besar adalah kemampuan model untuk melakukan rekayasa fitur secara otomatis (*automatic feature engineering*). Artinya, model dapat belajar dan mengekstraksi fitur-fitur relevan langsung dari data mentah tanpa memerlukan intervensi manusia. Keberhasilan fenomenal DL dalam beberapa tahun terakhir didorong oleh dua faktor utama: ketersediaan data dalam skala masif (*big data*) dan kemajuan pesat dalam perangkat keras komputasi paralel, terutama *Graphics Processing Units* (GPU), yang membuat proses pelatihan model yang besar menjadi jauh lebih efisien (Lecun dkk., 2015).

1) *Variational Autoencoder*

Autoencoder (AE) adalah salah satu jenis arsitektur jaringan saraf dalam pembelajaran tak terarah (*unsupervised learning*) yang tujuan utamanya adalah mempelajari representasi data yang efisien. AE melatih dirinya sendiri untuk memampatkan atau menyandikan (*encode*) data masukan ke dalam representasi berdimensi lebih rendah, lalu merekonstruksi atau mendekode (*decode*) kembali data masukan asli seakurat mungkin dari representasi yang telah dimampatkan tersebut. Berkat kemampuannya ini, AE banyak digunakan untuk tugas-tugas seperti kompresi data, penghilangan derau (*denoising*) pada gambar, dan deteksi anomali.

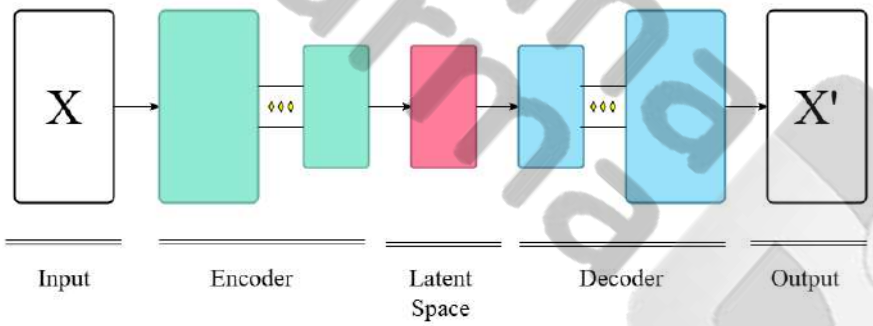
Meskipun terdapat banyak variasi, setiap arsitektur *autoencoder* pada dasarnya terdiri dari tiga komponen utama (Pinheiro Cinelli dkk., 2021) :

a. *Encoder* : Bagian ini berfungsi untuk memampatkan data masukan (x) dan memetakannya ke dalam sebuah ruang berdimensi lebih rendah yang

disebut ruang laten (*latent space*). Hasil dari penyandi adalah sebuah vektor tunggal (z) yang merupakan representasi terkompresi dari data asli.

b. *Latent Space* : Ini adalah representasi data masukan dalam bentuk yang paling padat dan terkompresi. Pada *autoencoder* standar, ruang laten ini bersifat deterministik, artinya setiap masukan akan dipetakan ke satu titik (vektor) yang statis di dalam ruang ini.

c. *Decoder* : Bagian ini bertugas untuk merekonstruksi data asli dari vektor laten (z). Arsitekturnya merupakan cerminan dari penyandi, yang secara bertahap "membongkar" representasi terkompresi hingga kembali ke dimensi data masukan semula.



Gambar 2.2 Arsitektur Autoencoder

Variational Autoencoder (VAE) adalah pengembangan dari arsitektur *autoencoder* standar yang menjadikannya sebuah model generatif yang kuat. Diperkenalkan pertama kali oleh Diederik P. Kingma dan Max Welling pada tahun 2013, VAE tidak hanya mampu merekonstruksi data, tetapi juga menghasilkan data baru yang belum pernah ada sebelumnya namun tetap mirip dengan data pelatihan (Kingma & Welling, 2013). Perbedaan fundamental antara AE dan VAE terletak pada cara mereka menangani ruang laten. Jika AE standar memetakan masukan ke satu vektor tunggal yang statis, penyandi pada VAE memetakan masukan ke sebuah distribusi probabilitas (biasanya distribusi Gaussian) yang diwakili oleh dua vektor: vektor rata-rata (*mean*) dan vektor simpangan baku (*standard deviation*). Pendekode kemudian mengambil sampel acak dari distribusi ini untuk merekonstruksi data (Doersch, 2021). Pendekatan probabilistik inilah yang memungkinkan VAE untuk menghasilkan variasi data baru dengan menjelajahi

titik-titik di sekitar ruang laten. Untuk memungkinkan proses optimasi model (seperti *backpropagation*) meskipun ada langkah pengambilan sampel acak, VAE menggunakan teknik cerdas yang disebut "trik reparameterisasi" (*reparameterization trick*). Teknik ini secara efektif memisahkan proses acak dari jaringan utama, sehingga gradien tetap dapat mengalir. Hasilnya, VAE memiliki dua kemampuan sekaligus: sebagai pengode otomatis yang andal untuk merekonstruksi data, dan sebagai model generatif untuk menciptakan data baru yang bervariasi (An & Cho, 2015).

2) *Deep Neural Network*

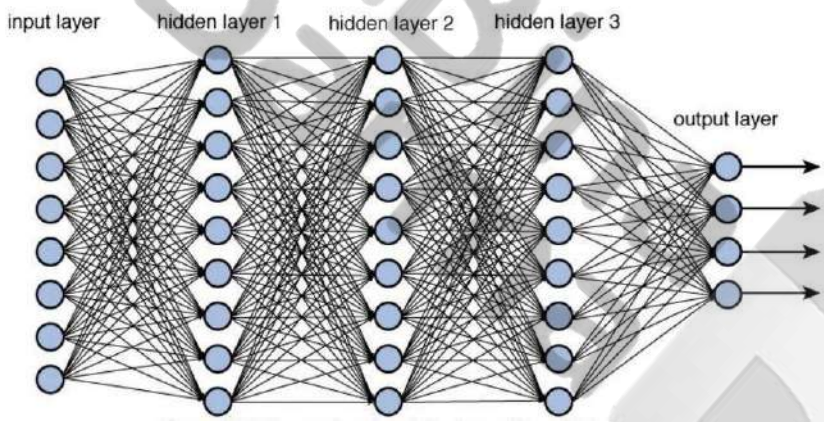
Deep Neural Network (DNN) adalah kelas dari Jaringan Saraf Tiruan (*Artificial Neural Network*) yang dicirikan oleh adanya banyak lapisan tersembunyi (*hidden layers*) di antara lapisan masukan dan keluaran. Istilah "deep" (mendalam) merujuk secara spesifik pada kedalaman arsitektur ini. Kedalaman tersebut memungkinkan DNN untuk melakukan pembelajaran fitur secara hierarkis (*hierarchical feature learning*), di mana lapisan-lapisan awal belajar mengenali fitur-fitur sederhana, dan lapisan yang lebih dalam menggabungkan fitur-fitur tersebut untuk membentuk representasi data yang jauh lebih kompleks dan abstrak. Kemampuan inilah yang membuat DNN sangat efektif dalam menangani tugas-tugas rumit yang tidak dapat diselesaikan oleh jaringan saraf "dangkal" (*shallow networks*) (Goodfellow, 2014; Sze dkk., 2017).

Secara struktural, sebuah DNN terdiri dari tiga jenis lapisan utama:

a) Lapisan Masukan (*Input Layer*): Merupakan gerbang awal tempat data mentah dimasukkan ke dalam jaringan. Setiap neuron di lapisan ini merepresentasikan satu fitur dari data masukan. Misalnya, untuk gambar berukuran 28x28 piksel, lapisan masukan akan memiliki 784 neuron.
b) Lapisan Tersembunyi (*Hidden Layers*): Ini adalah inti dari DNN, yang terdiri dari dua atau lebih lapisan. Pada arsitektur DNN yang paling umum, lapisan-lapisan ini bersifat *dense* atau *fully connected*, artinya setiap neuron terhubung ke semua neuron di lapisan sebelumnya. Konfigurasi padat ini

memungkinkan jaringan untuk memodelkan hubungan non-linear yang kompleks antar fitur.

c) Lapisan Keluaran (*Output Layer*): Merupakan lapisan terakhir yang menghasilkan prediksi akhir dari model. Jumlah neuron dan fungsi aktivasi pada lapisan ini sangat bergantung pada jenis tugas, misalnya satu neuron dengan fungsi Sigmoid untuk klasifikasi biner, atau beberapa neuron dengan fungsi Softmax untuk klasifikasi multikelas.



Gambar 2.3 Arsitektur DNN *(Altin Karataş & Biberci, 2023)*

2.2.3. *Similarity*

Dalam bidang *Natural Language Processing* (NLP) dan *Information Retrieval*, ukuran kemiripan (*similarity measure*) adalah sebuah metrik fundamental yang digunakan untuk menguantifikasi seberapa mirip atau berbedanya dua buah objek data. Ketika diterapkan pada data teks, tujuannya adalah untuk menghasilkan sebuah nilai numerik yang merepresentasikan tingkat kemiripan antara dua string, kalimat, atau dokumen. Proses ini merupakan langkah krusial dalam berbagai aplikasi machine learning, seperti deteksi plagiarisme, pengelompokan dokumen (*document clustering*), sistem rekomendasi, dan mesin pencari (Manning dkk., 2009).

Terdapat beragam teknik untuk menghitung kemiripan, yang dapat dikategorikan berdasarkan pendekatannya:

a) Berbasis Himpunan (*Set-based*): Mengukur tumpang tindih antara elemen-elemen dalam dua himpunan, seperti *Jaccard Similarity* dan *Dice Similarity*.
b) Berbasis Vektor (*Vector-based*): Mengubah teks menjadi representasi vektor dan mengukur kemiripan dalam ruang vektor, seperti *Cosine Similarity*.
c) Berbasis Jarak Edit (*Edit-based*): Menghitung jumlah operasi minimum (sisip, hapus, ganti) yang diperlukan untuk mengubah satu string menjadi string lainnya, seperti *Levenshtein Distance.*

Jaccard Similarity, yang juga dikenal sebagai Indeks Jaccard atau Koefisien Jaccard, adalah metrik berbasis himpunan yang ditemukan oleh ahli botani Paul Jaccard. Metrik ini mengukur kemiripan antara dua himpunan data dengan cara menghitung rasio antara ukuran irisan (elemen yang sama) dan ukuran gabungan (total elemen unik) dari kedua himpunan tersebut. Nilai yang dihasilkan berkisar antara 0 (tidak ada elemen yang sama) hingga 1 (kedua himpunan identik) (Yudhana dkk., 2018). Secara matematis, formula *Jaccard Similarity* untuk dua himpunan, A dan B, adalah sebagai berikut:

$$J(A,B)=\frac{|A\cap B|}{|A\cup B|} \quad (2.1)$$

Keterangan :

J = *Jaccad Similarity*
|A ∩ B| = Jumlah data yang anggotanya berasal dari A yang juga menjadi anggota B.
|A ∪ B| = Jumlah data yang anggotanya berasal dari A atau B maupun keduanya.

2.2.4. *Imbalanced Data*

Data tidak seimbang (*imbalanced data*) adalah kondisi yang umum terjadi dalam pengumpulan data, di mana distribusi kelas dalam dataset sangat tidak merata. Secara spesifik, ini merujuk pada situasi di mana jumlah sampel (instans) pada satu kelas jauh lebih sedikit dibandingkan dengan kelas lainnya. Kelas dengan jumlah sampel yang sedikit ini disebut kelas minoritas (*minority class*), sementara

kelas dengan jumlah sampel yang melimpah disebut kelas mayoritas (*majority class*) (Fernández dkk., 2018; He & Garcia, 2009).

Keberadaan data yang tidak seimbang menjadi tantangan serius dalam proses klasifikasi pada *machine learning*. Mayoritas algoritma klasifikasi standar dirancang dengan asumsi distribusi kelas yang seimbang dan bertujuan untuk memaksimalkan akurasi secara keseluruhan. Akibatnya, ketika dihadapkan pada data tidak seimbang, model akan cenderung bias terhadap kelas mayoritas. Dampak utamanya adalah model dapat mencapai akurasi yang tinggi hanya dengan selalu memprediksi kelas mayoritas, sementara performanya dalam mengidentifikasi kelas minoritas sangat buruk. Fenomena ini dapat menyebabkan kesalahan klasifikasi (*misclassification*) yang fatal pada kelas minoritas, yang sering kali justru merupakan kelas yang paling penting untuk dideteksi. Sebagai contoh, model deteksi penipuan yang gagal mengidentifikasi transaksi penipuan akan kehilangan tujuan utamanya, meskipun akurasi keseluruhannya mungkin terlihat tinggi.

Untuk mengatasi dampak negatif dari data tidak seimbang, berbagai teknik telah dikembangkan. Secara umum, pendekatan ini dapat dikelompokkan menjadi dua kategori utama (He & Garcia, 2009):

1) Pendekatan pada Level Data (*Data-Level Approaches*): Teknik ini berfokus pada penyeimbangan kembali distribusi kelas dalam data pelatihan melalui metode *resampling*.
    A. *Oversampling*: Menambah jumlah sampel pada kelas minoritas. Beberapa teknik populer meliputi:
        a) SMOTE (*Synthetic Minority Over-sampling Technique*): Menciptakan sampel sintetis baru dengan melakukan interpolasi antara sampel minoritas yang berdekatan
        b) *Generative Oversampling* (contoh: VAE): Menggunakan model generatif seperti *Variational Autoencoder* (VAE) yang dilatih pada kelas minoritas untuk menghasilkan sampel sintetis baru yang berkualitas tinggi dan beragam. Pendekatan ini mampu menangkap distribusi data yang kompleks dengan lebih baik (Ai dkk., 2023; Sun dkk., 2023).

B. *Undersampling*: Mengurangi jumlah sampel pada kelas mayoritas secara acak atau terinformasi untuk menyeimbangkannya dengan kelas minoritas.

2) Pendekatan pada Level Algoritma (*Algorithmic-Level Approaches*): Teknik ini memodifikasi algoritma pembelajaran itu sendiri agar lebih sensitif terhadap kelas minoritas. Contohnya adalah pembelajaran peka-biaya (*cost-sensitive learning*), di mana model diberikan "penalti" atau biaya yang lebih tinggi jika salah mengklasifikasikan sampel dari kelas minoritas.

2.2.5. *Text Normalization*

Text Normalization atau normalisasi teks merupakan suatu proses fundamental dan tahapan krusial dalam bidang Pengolahan Bahasa Alami atau *Natural Language Processing* (NLP). Proses ini secara teoretis bertujuan untuk mengubah teks yang memiliki banyak variasi ke dalam sebuah bentuk kanonis atau bentuk standar yang seragam. Secara konseptual, normalisasi teks terletak pada kemampuannya untuk mengurangi kompleksitas dan dimensi data secara signifikan. Dalam konteks pemodelan statistik dan pembelajaran mesin, setiap kata unik dalam korpus data dianggap sebagai satu fitur terpisah. (Manning dkk., 2009) dalam buku fundamentalnya menyatakan bahwa variasi bentuk kata secara langsung memperbesar ukuran kosakata (*vocabulary*) yang harus ditangani oleh sebuah sistem. Pembesaran ruang fitur ini tidak hanya meningkatkan beban komputasi tetapi juga dapat menyebabkan masalah *data sparsity*, di mana banyak fitur memiliki frekuensi kemunculan yang sangat rendah. Oleh karena itu, normalisasi teks memainkan peran vital dalam memetakan berbagai variasi kata ke representasi tunggal, yang pada gilirannya mengurangi dimensi ruang fitur dan membantu model untuk dapat melakukan generalisasi dengan lebih baik dari data yang ada.

Praktik normalisasi teks sendiri melibatkan serangkaian teknik yang dapat diaplikasikan secara modular. Tahapan yang paling umum dan mendasar mencakup *case folding*, yaitu proses konversi seluruh karakter alfabet menjadi format huruf kecil (*lowercase*). Proses ini kemudian seringkali diikuti oleh pembersihan tanda

baca (*punctuation removal*), penghapusan angka, serta eliminasi spasi berlebih (*whitespace removal*). Tahapan yang lebih lanjut dalam normalisasi menyentuh aspek morfologi bahasa. (Anjali & Jivani, 2011) menjelaskan dua teknik morfologi yang populer, yaitu *stemming* dan *lemmatization*. Teknik *stemming* memotong akhir kata atau sufiks berdasarkan serangkaian aturan heuristik untuk mendapatkan bentuk dasar kata, meskipun hasilnya terkadang bukan merupakan kata yang valid. Sebaliknya, teknik *lemmatization* menggunakan analisis leksikal dan kamus bahasa untuk mengubah kata ke bentuk dasarnya yang valid (lemma), sehingga prosesnya lebih akurat secara linguistik dibandingkan *stemming*.

Penerapan normalisasi teks secara komprehensif telah terbukti memberikan dampak positif yang signifikan pada kinerja berbagai aplikasi NLP. Sebuah studi oleh (Haddi dkk., 2013) menunjukkan bahwa langkah-langkah pra-pemrosesan, termasuk normalisasi, secara signifikan meningkatkan akurasi pada tugas analisis sentimen. Dalam sistem pencarian informasi (*information retrieval*), normalisasi memastikan bahwa kueri dapat mengambil dokumen yang mengandung frasa dari kata, sehingga meningkatkan relevansi hasil pencarian. Dengan demikian, normalisasi teks tidak dapat dianggap sebagai sekadar langkah teknis, melainkan sebuah fondasi teoretis yang kuat yang memungkinkan mesin untuk memahami dan memproses bahasa manusia secara lebih efektif dan mendekati pemahaman semantik yang sebenarnya.

2.2.6. *Min-Max Scaller*

Min-Max Scaler merupakan metode pra-pemrosesan untuk melakukan transformasi fitur dengan mengubah skala setiap fitur secara individual ke dalam rentang nilai tertentu (Tan dkk., 2018). Pra-pemrosesan ini bertujuan agar rentang nilai setiap sampel pada suatu fitur tidak terlalu lebar. Proses penskalaan ini dilakukan dengan mengurangi setiap nilai sampel dengan nilai minimum pada fiturnya, lalu hasilnya dibagi dengan selisih antara nilai maksimum dan nilai minimum dari fitur tersebut (Nasution dkk., 2019). Rumus matematis untuk *Min-Max Scaler* ditunjukkan pada Persamaan (2.2), dengan X yang merepresentasikan nilai sampel.

$$X^i = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (2.2)$$

Dimana :

X^i=Nilai hasil normalisasi

X=Nilai data aktual

X_{min}=Nilai minimum dari data aktual

X_{max} =Nilai maksimum dari data aktual

2.2.7. *Confusion Matrix*

Confusion Matrix, atau matriks kebingungan, adalah sebuah tabel yang digunakan secara luas untuk memvisualisasikan dan menganalisis kinerja sebuah model klasifikasi. Matriks ini menyajikan perbandingan terperinci antara kelas aktual (kebenaran dasar atau *ground truth*) dengan kelas yang diprediksi oleh model. Elemen pada diagonal utama menunjukkan jumlah prediksi yang benar, sedangkan elemen di luar diagonal utama menunjukkan kesalahan yang dibuat oleh model (Luque dkk., 2019; Tan dkk., 2018). Untuk kasus klasifikasi biner (dengan kelas Positif dan Negatif), *confusion matrix* memiliki struktur *matrix* 2x2 sebagai berikut

Tabel 2.2 Matrix 2x2 Confusion Matrix

	Prediksi: Positif	Prediksi: Negatif
Aktual: Positif	TP (True Positive)	FN (False Negative)
Aktual: Negatif	FP (False Positive)	TN (True Negative)

Keempat kuadran dalam matriks ini didefinisikan sebagai:

1) True Positive (TP): Jumlah sampel positif yang benar diprediksi sebagai positif. (Contoh: Pasien sakit diprediksi sakit).
2) True Negative (TN): Jumlah sampel negatif yang benar diprediksi sebagai negatif. (Contoh: Orang sehat diprediksi sehat).

3) False Positive (FP) / *Type I Error*: Jumlah sampel negatif yang salah diprediksi sebagai positif. (Contoh: Orang sehat diprediksi sakit; sebuah "alarm palsu").
4) False Negative (FN) / *Type II Error*: Jumlah sampel positif yang salah diprediksi sebagai negatif. (Contoh: Pasien sakit diprediksi sehat; sebuah kasus yang "terlewatkan").

Dari keempat komponen dasar tersebut, dapat diturunkan ke berbagai metrik untuk mengevaluasi kinerja dari model (Fawcett, 2006).

a) *Accuracy*

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{2.3}$$

Parameter akurasi pada Persamaan 2.3 digunakan untuk mengukur proporsi prediksi yang benar (baik positif maupun negatif) di antara total jumlah sampel.

b) Presisi

$$\text{Presisi} = \frac{TP}{FP+TP} \tag{2.4}$$

Precision (presisi), sebagaimana ditunjukkan pada Persamaan 2.4, merupakan metrik yang mengukur akurasi dari keseluruhan prediksi positif yang dihasilkan model. Nilai *precision* yang tinggi mengindikasikan bahwa prediksi positif yang dibuat oleh model memiliki probabilitas kebenaran yang tinggi.

c) Recall

$$Recall = \frac{TP}{TP+FN} \tag{2.5}$$

Parameter *recall* pada Persamaan 2.5 mengukur kemampuan model dalam mengidentifikasi kembali seluruh sampel positif yang ada. Metrik ini juga dikenal dengan sebutan *True Positive Rate* (TPR).

d) Specificity

$$Specificity = \frac{TN}{TN+FP} \tag{2.6}$$

Parameter *specificity* pada Persamaan 2.6 digunakan untuk mengukur kemampuan model dalam mengidentifikasi sampel negatif secara tepat. Metrik ini juga dikenal sebagai *True Negative Rate* (TNR).

e) F-1 Score

$$F\text{-}Measure = \frac{2 \text{ x Presisi x Recall}}{\text{Presisi+Recall}} \quad (2.7)$$

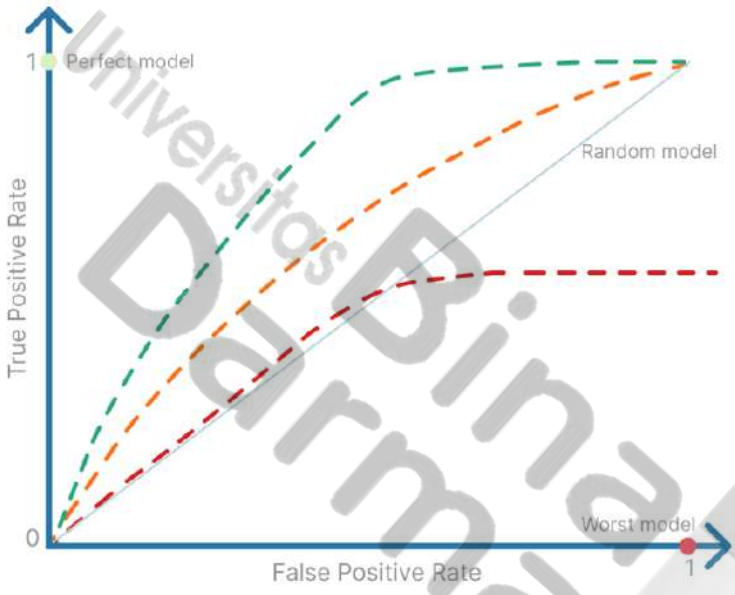
Parameter F1-Score pada Persamaan 2.7 merupakan nilai rata-rata harmonik (*harmonic mean*) dari Presisi dan Recall. F1-Score memberikan skor tunggal yang menyeimbangkan keduanya dan sangat berguna ketika kepentingan antara FP dan FN seimbang. Setelah mengetahui nilai akurasi, *error rate*, presisi, dan *recall*, penulis menggunakan kurva ROC untuk memvisualisasikan akurasi metode yang digunakan.

2.2.8. *Receiver Operating Characteristic Curve* (ROC)

Kurva *Receiver Operating Characteristic* (ROC) adalah sebuah plot grafis yang berfungsi untuk mengilustrasikan dan mengevaluasi kemampuan diagnostik dari sebuah model klasifikasi biner. Tujuannya adalah untuk memvisualisasikan *trade-off* (pertukaran) antara tingkat keberhasilan dalam mendeteksi kelas positif versus tingkat alarm palsu yang dihasilkan. Kurva ini dibuat dengan memetakan *True Positive Rate* (TPR) atau Sensitivitas pada sumbu Y, terhadap *False Positive Rate* (FPR) pada sumbu X (Fawcett, 2006). Sebuah kurva ROC tidak hanya menampilkan satu titik performa, melainkan serangkaian performa model di berbagai tingkatan. Ini dibuat dengan cara menggeser ambang batas (*threshold*) klasifikasi secara kontinu. Untuk setiap nilai ambang batas, pasangan nilai (FPR, TPR) yang baru dihitung dan diplot sebagai satu titik. Ketika semua titik ini dihubungkan, terbentuklah sebuah kurva yang menunjukkan spektrum kinerja model. Dalam plot ROC:

1) Garis Diagonal (y=x): Merepresentasikan kinerja model yang setara dengan tebakan acak.

2) Titik (0,1) (Pojok Kiri Atas): Merepresentasikan klasifikasi sempurna (100% sensitivitas, 100% spesifisitas). Semakin kurva "cembung" mendekati titik ini, semakin baik kinerja model.



Gambar 2.4 Ilustrasi ROC Curve

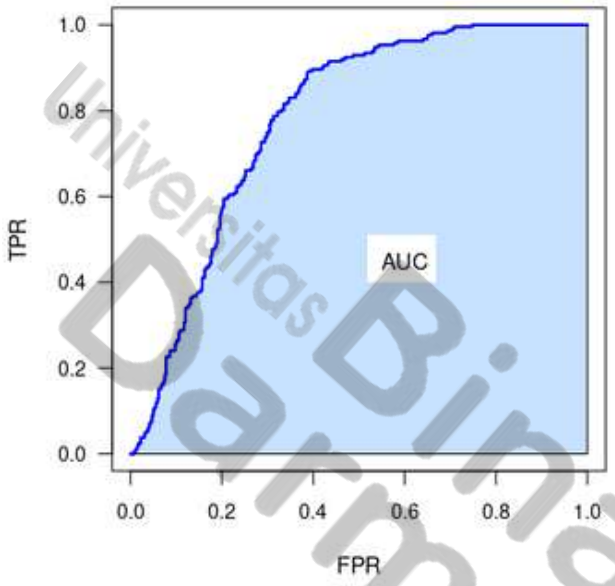
Untuk meringkas keseluruhan kinerja yang direpresentasikan oleh kurva ROC ke dalam satu angka tunggal, digunakan metrik *Area Under the Curve* (AUC). Sesuai namanya, AUC mengukur luas total area di bawah kurva ROC. Nilai AUC berkisar antara 0 dan 1, di mana nilai yang lebih tinggi menunjukkan performa model yang lebih baik dalam membedakan antara kelas positif dan negatif di semua ambang batas.

Interpretasi umum dari nilai AUC adalah sebagai berikut (Tan dkk., 2018) :

1) AUC = 1.0: Model sempurna.
2) $0.9 \leq$ AUC < 1.0: Kinerja sangat baik (*excellent*).
3) $0.8 \leq$ AUC < 0.9: Kinerja baik (*good*).
4) $0.7 \leq$ AUC < 0.8: Kinerja cukup (*fair*).
5) AUC = 0.5: Model tidak memiliki kemampuan diskriminatif (setara tebakan acak).
6) AUC < 0.5: Kinerja lebih buruk daripada tebakan acak.

Keunggulan utama AUC adalah metrik ini tidak bergantung pada ambang batas klasifikasi tertentu dan memberikan gambaran performa yang lebih andal,

terutama pada kasus dataset yang tidak seimbang (*imbalanced data*), di mana metrik akurasi bisa sangat menyesatkan.



Gambar 2.5 Ilustrasi AUC Curve

BAB III
METODOLOGI PENELITIAN

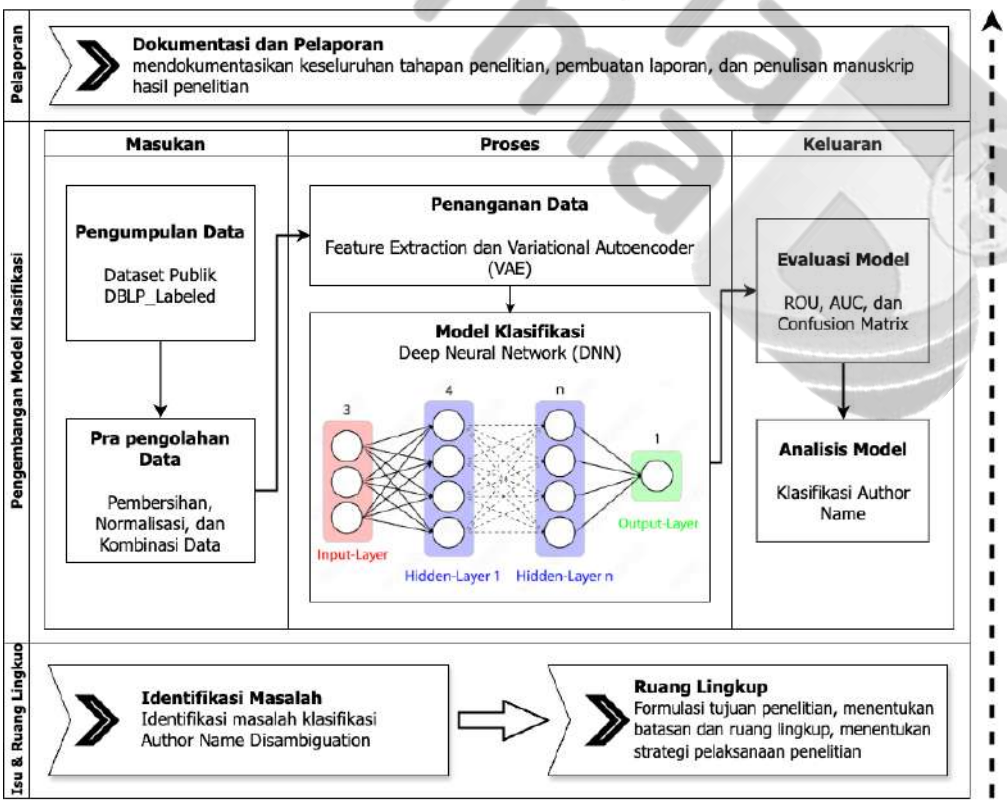
3.1. Waktu Dan Tempat Penelitian

Penelitian ini memanfaatkan data sekunder berupa dataset jejaring publikasi yang bersumber dari DBLP (*Digital Bibliography & Library Project*). Dataset ini merupakan hasil kurasi yang dipublikasikan oleh (J. Kim, 2018; J. J. Kim & Kim, 2018) pada tahun 2018. DBLP sendiri adalah sebuah repositori bibliografi daring yang sangat komprehensif dan menjadi rujukan utama untuk publikasi ilmiah di bidang ilmu komputer. Dataset yang digunakan merupakan hasil dari proses ekstraksi dan pra-pemrosesan yang dilakukan oleh para penulis tersebut. Di dalamnya terkandung atribut-atribut publikasi yang terstruktur, seperti judul artikel, nama-nama penulis, tahun terbit, serta nama jurnal atau konferensi tempat publikasi. Periode waktu yang dicakup oleh dataset ini merefleksikan rekaman data yang tersedia dalam basis data DBLP hingga saat ekstraksi dilakukan pada tahun 2018. Kelengkapan dan struktur data yang kaya ini menjadikannya sumber yang sangat relevan dan andal untuk dianalisis sesuai dengan tujuan penelitian ini.

3.2. Prosedur Penelitian

Penelitian ini mengusulkan sebuah kerangka kerja sistematis untuk mengatasi masalah disambiguasi nama penulis (*author name disambiguation*) menggunakan pendekatan *deep learning*. Tujuan utamanya adalah untuk mengembangkan dan mengevaluasi model klasifikasi yang mampu membedakan secara akurat apakah dua entri nama penulis merujuk pada individu yang sama atau berbeda. Kontribusi utama dari penelitian ini terletak pada penerapan *Variational Autoencoder* (VAE) sebagai teknik *generative oversampling* untuk menangani data tidak seimbang, serta melakukan analisis perbandingan antara arsitektur jaringan saraf dalam dan dangkal. Proses penelitian akan dilaksanakan melalui beberapa tahapan yang terstruktur dan saling berkelanjutan. Seluruh alur proses penelitian ini dirangkum secara visual pada Gambar 3.1.

Pelaksanaan penelitian ini dirancang melalui serangkaian tahapan yang sistematis dan terintegrasi, dimulai dari fondasi teoretis hingga evaluasi akhir. Proses diawali dengan studi literatur yang mendalam untuk mengidentifikasi ruang lingkup masalah disambiguasi nama penulis, yang kemudian dilanjutkan dengan pengumpulan dan persiapan data sekunder dari dataset DBLP. Setelah data siap, dilakukan tahap pra-pemrosesan dan ekstraksi fitur untuk menghasilkan variabel-variabel prediktif yang relevan, seperti tingkat kemiripan nama, kesamaan rekan penulis, dan afiliasi. Menyadari sifat dataset yang secara inheren tidak seimbang, penelitian ini menerapkan teknik *generative oversampling* menggunakan Variational Autoencoder (VAE) untuk menciptakan sampel sintetis bagi kelas minoritas, sehingga distribusi data pelatihan menjadi lebih seimbang.



Gambar 3.1 Prosedur Penelitian

Puncak dari metodologi ini adalah perancangan eksperimen komparatif untuk menguji hipotesis utama. Pada penelitian ini dilakukan dua kali percobaan dalam melakukan penanganan data yang tidak seimbang. Percobaan pertama, penanganan data yang tidak seimbang menggunakan algoritma VAE kemudian setelah data

seimbang dilakukan klasifikasi menggunakan model *Deep Neural Network* (VAE-DNN). Sedangkan percobaan kedua, algoritma *Synthetic Minority Over-sampling Technique* (SMOTE) sebagai penanganan data yang tidak seimbang juga akan digabungkan dengan model *Deep Neural Network*. Kinerja dari kedua model tersebut kemudian akan dievaluasi dan dianalisis secara komprehensif menggunakan metrik standar, termasuk Akurasi, Presisi, *Recall*, *F1-Score*, serta analisis kurva ROC dan nilai AUC. Terakhir, hasil perbandingan akan dianalisis secara mendalam untuk menarik kesimpulan mengenai efektivitas arsitektur *deep learning* untuk masalah ini, yang kemudian akan didokumentasikan secara rinci dalam laporan akhir penelitian.

3.2.1. Studi Pustaka

Tahapan studi pustaka dalam penelitian ini dilakukan secara sistematis untuk membangun landasan teoretis, mengidentifikasi perkembangan terkini (*state-of-the-art*), dan menemukan celah penelitian (*research gap*). Tinjauan mendalam difokuskan pada tiga pilar utama: metode-metode *Author Name Disambiguation* yang sudah ada, penerapan arsitektur *Deep Neural Network* (DNN) untuk tugas klasifikasi, serta penggunaan model generatif seperti *Variational Autoencoder* (VAE) untuk menangani data tidak seimbang. Dari sintesis literatur tersebut, teridentifikasi bahwa kombinasi VAE untuk penyeimbangan data dengan DNN untuk klasifikasi pada masalah ini masih menjadi area yang menjanjikan dan belum banyak dieksplorasi. Penelitian ini merujuk pada beberapa publikasi ilmiah sebagai acuan utama, yang meliputi karya (Ai dkk., 2023; J. J. Kim & Kim, 2018; Utyamishev & Partin-Vaisband, 2019).

3.2.2. Persiapan Data

Sumber data utama yang digunakan dalam penelitian ini adalah dataset publik bernama DBLP_Labeled_Data.txt yang diperkenalkan oleh (J. J. Kim & Kim, 2018). Dataset ini dipilih karena secara spesifik menyediakan data berlabel untuk tugas disambiguasi nama penulis, yang sangat sesuai dengan domain masalah penelitian ini. Lebih lanjut, sebuah karakteristik penting dari dataset ini adalah

adanya distribusi kelas yang tidak seimbang (*imbalanced*) secara alami, yang menjadikannya kasus uji yang ideal untuk mengevaluasi efektivitas metode penanganan data tidak seimbang yang diusulkan.

Secara mentah, dataset ini terdiri dari 5.018 baris rekaman publikasi. Atribut-atribut relevan yang akan dimanfaatkan dari dataset ini mencakup *author_name*, *author_id*, *paper id*, *author list*, *year*, *venue* dan *NaN*. Dalam konteks ini, atribut *author_id* akan berfungsi sebagai label kebenaran dasar (*ground truth*) untuk validasi dan evaluasi model. Deskripsi lebih rinci mengenai setiap atribut ditampilkan pada Tabel 3.1.

Tabel 3.1 Deskripsi DBLP Dataset

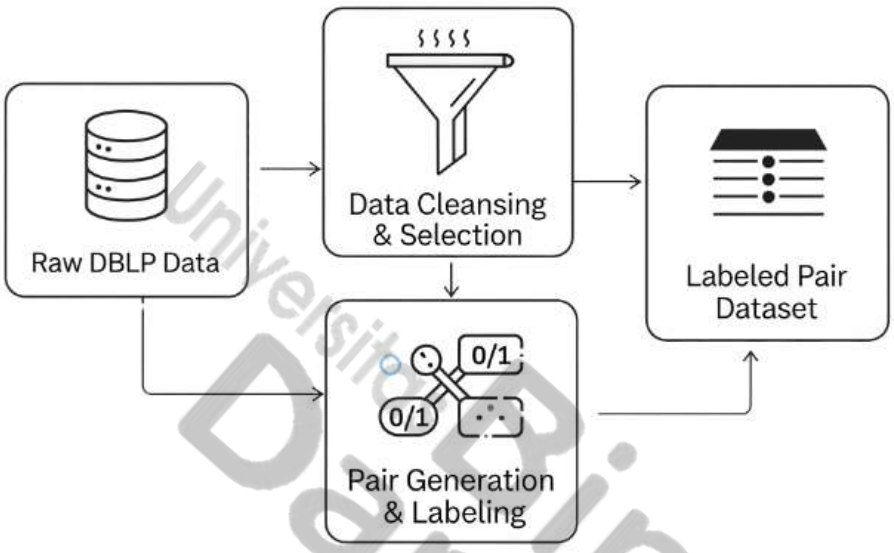
	author name	author id	paper id	author	year	venue	title	NaN
0	ajay k. gupta	agupta-10		ajay k. gupta|kurt maly|irwin b. levinstein	1996	ifip world conference on it tools	pme privileg mangag enforc system secure	NaN
1	ajay k. gupta	agupta-10		ajay k. gupta|kurt maly|hussein m. abdel-wahab	1997	webnet	coprocess java base environ collabor process	NaN
2	ajay k. gupta	agupta-10		ajay k. gupta|kurt maly|bjorn	1996	wetice	privileg manag enforc system distribut	NaN
3	ajay k. gupta	agupta-10		ajay k. gupta|hussein m. abdel-wahab|kurt	1996	wetice	softwar architectur interprocess commun iri	NaN
	…	…	...	…	…	…	…	…

	author name	author id	paper id	author	year	venue	title	NaN
5015	yan chen	ychen-9		yan chen|randy h. katz|john kubiatowicz	2002	iptps	dynam replica placement scalabl content	NaN
5016	yan chen	ychen-9		yan chen|john kubiatowicz|david bindel|steven	2000	asplos	oceanstor architectur global scale persist storag	NaN
5017	yan chen	ychen-9		yan chen|randy h. katz|john kubiatowicz	2002	pervasive	scan dynam scalabl effici content distribut	NaN
5018	yan chen	ychen-9		yan chen|bhaskaran raman|sharad agarwal	2002	pervasive	sahara model servic composit multipl provid	NaN

3.2.3. Pra-Pemrosesan Data

Tahap pra-pemrosesan data bertujuan untuk mengubah data mentah dari dataset DBLP menjadi sebuah himpunan data (*dataset*) yang bersih, terstruktur, dan siap untuk tahap ekstraksi fitur.

Proses ini melibatkan dua langkah utama yaitu: *Data Cleansing and Selection* dan *Pair Generation and Labeling.* Proses *data cleansing and selection* dibagi menjadi beberapa tahapan tambahan seperti penghapusan kolom yang tidak relevan dan *text cleaning.* Sedangkan proses *pair generation and labeling* dilakukan untuk mentransformasi dataset menjadi berpasangan (*pairwise*) dan membuat label dari data yang telah menjadi pasangan tersebut.



Gambar 3.2 Pra-Pemrosesan Data

1) *Data Cleansing and Selection*

Tahap awal dalam pra-pengolahan adalah mempersiapkan data mentah dengan melakukan seleksi atribut dan pembersihan konten. Proses ini krusial untuk memastikan kualitas dan konsistensi data sebelum masuk ke tahap selanjutnya, mengingat kinerja model *Deep Learning* sangat sensitif terhadap *noise* atau data yang tidak relevan. Langkah ini dilandasi oleh prinsip bahwa kualitas luaran model sangat bergantung pada kualitas data masukan. Tahapan ini dibagi menjadi dua kegiatan utama sebagai berikut.

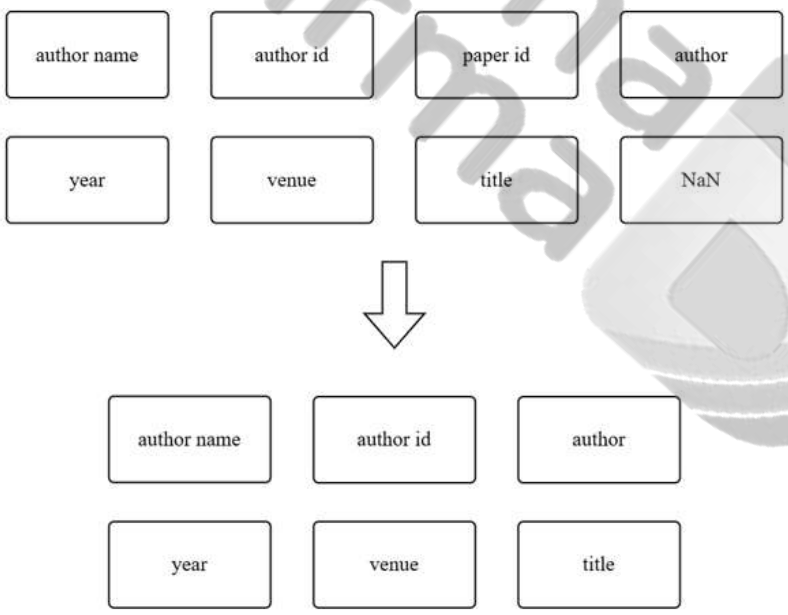
A. Penghapusan Kolom

Langkah awal dalam pra-pemrosesan data adalah seleksi atribut melalui penghapusan kolom. Proses ini bertujuan untuk menyederhanakan *dataset* dengan hanya mempertahankan kolom-kolom yang informatif untuk tugas klasifikasi. Eliminasi atribut yang tidak relevan atau *redundant* sangat penting untuk mengurangi dimensi data dan mencegah model mempelajari pola yang tidak bermakna.

Secara spesifik, dari total delapan kolom pada *dataset* mentah, dua kolom yang tidak relevan dihapus, sehingga menyisakan enam kolom inti. Proses ini divisualisasikan pada Gambar 3.3. Keenam kolom terpilih ini kemudian dipersiapkan untuk peran yang berbeda dalam pemodelan:

a) Fitur (*Features*): *author_name*, *author_list*, *year*, *venue*, dan *title*. Kelima atribut ini dipilih karena mengandung informasi semantik dan bibliografis yang esensial untuk membedakan identitas penulis satu dengan yang lainnya. Atribut-atribut ini akan menjadi *input* bagi model.

b) Label (*Target Variable*): *author_id*. Atribut ini bertindak sebagai penanda unik untuk setiap entitas penulis. Dalam konteks pelatihan model *supervised learning*, atribut ini berfungsi sebagai *ground truth* atau jawaban yang harus diprediksi oleh model untuk memverifikasi kebenaran klasifikasi.



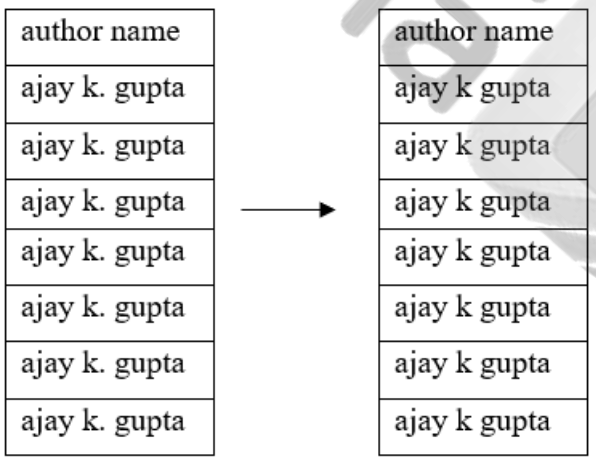
Gambar 3.3 Penghapusan Kolom

B. *Text Cleaning*

Setelah seleksi kolom, dilakukan pembersihan pada seluruh data tekstual untuk memastikan konsistensi dan kualitasnya. Proses ini bertujuan untuk menstandardisasi teks dengan menghilangkan elemen-elemen yang dapat mengganggu analisis atau menciptakan variasi semu. Standardisasi ini sangat krusial dalam pemrosesan teks untuk mengurangi dimensi ruang fitur (*vocabulary size*). Seperti yang ditunjukkan pada Gambar 3.4, langkah-langkah utamanya adalah:

a) Konversi Huruf Kecil (*Lowercasing*): Mengubah seluruh teks menjadi format huruf kecil. Langkah ini dilakukan untuk memastikan kata yang sama diperlakukan secara identik (misalnya, 'Gupta' dan 'gupta'). Tanpa konversi ini, model akan menganggap kedua kata tersebut sebagai dua entitas yang berbeda, yang dapat mengurangi akurasi pencocokan.

b) Penghapusan Tanda Baca (*Punctuation Removal*): Menghapus semua simbol tanda baca (seperti titik, koma, tanda hubung). Simbol-simbol ini sering kali dianggap sebagai *noise* dalam konteks pencocokan nama dan judul publikasi. Tujuannya adalah untuk menyederhanakan data, mengurangi variasi token yang tidak perlu, dan meningkatkan efisiensi komputasi dengan memfokuskan model pada kata-kata inti yang mengandung makna.



Gambar 3.4 *Text Cleaning*

2) *Pair Generation and Labeling*

Tahap selanjutnya adalah mentransformasi dataset dari format individual menjadi format pasangan (*pairwise*). Proses ini dilakukan dengan membangkitkan kombinasi seluruh kemungkinan pasangan baris data. Tujuannya adalah untuk menyiapkan data agar dapat dilakukan proses komparasi dan penghitungan kemiripan antara satu entri dengan entri lainnya, sebagaimana dijelaskan dalam rumus kombinasi pada Persamaan (3.1).

Penerapan metode kombinasi ini menghasilkan peningkatan volume data yang eksponensial. Setelah proses ini, jumlah data baru yang terbentuk adalah sebanyak 12.587.653 pasangan baris data. Volume data yang masif ini menyediakan ruang sampel yang cukup luas bagi model untuk mempelajari nuansa perbedaan dan kemiripan yang spesifik antarpenulis.

$$C\ (n,r) = \binom{n+r-1}{r} = \frac{(n+r-1)!}{r!(n-1)} \quad (3.1)$$

Keterangan :

C′ = Kombinasi dengan pengulangan

n = jumlah data yang bisa dipilih

r = jumlah data yang harus dipilih dalam setiap grup

! = faktorial

Hasil dari proses kombinasi dapat dilihat pada Tabel 3.2, di mana kolom *n* dan *r* merepresentasikan dua entitas penulis yang akan dibandingkan.

Tabel 3.2 Hasil Proses Kombinasi

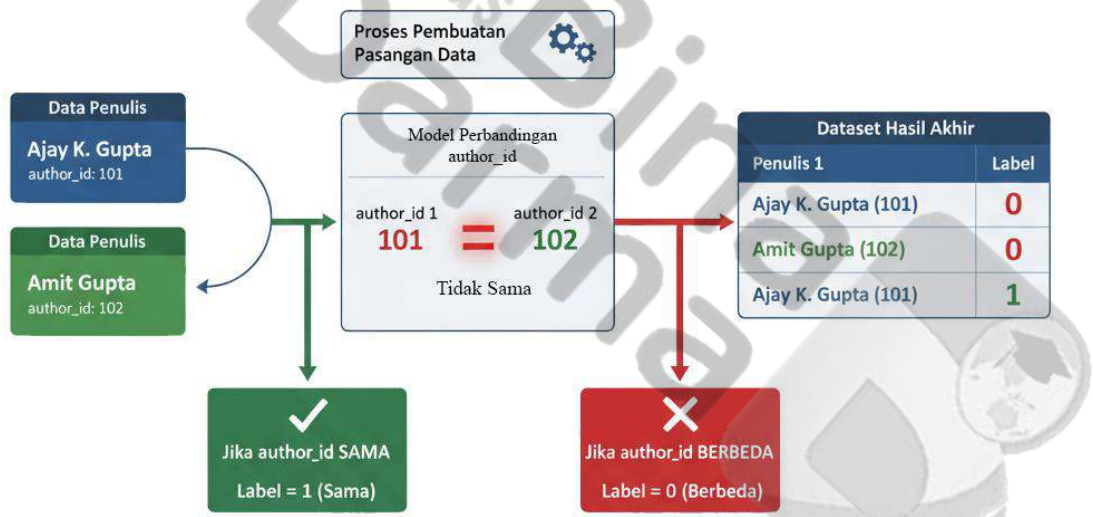
n	r
ajay k. gupta	amit gupta
ajay k. gupta	amit gupta
ajay k. gupta	arobinda gupta
ajay k. gupta	arobinda gupta
ajay k. gupta	alok gupta

Setelah setiap pasangan terbentuk, proses pelabelan dilakukan secara otomatis. Setiap pasangan diberi label kebenaran dasar (*ground truth*) dengan membandingkan nilai pada kolom author_id:

A. Label 1 (Sama): Diberikan jika *author_id* dari kedua data dalam pasangan tersebut identik, yang mengindikasikan bahwa kedua rekaman merujuk pada individu yang sama.

B. Label 0 (Berbeda): Diberikan jika *author_id* keduanya berbeda, yang menandakan bahwa kedua rekaman merujuk pada dua individu yang berbeda.

Hasil akhir dari tahap ini adalah *dataset* besar yang siap untuk proses ekstraksi fitur. Pada tahap selanjutnya, data mentah dalam pasangan ini akan dikonversi menjadi nilai numerik melalui perhitungan skor kemiripan (*similarity scores*), sehingga dapat diproses oleh algoritma pembelajaran mesin.



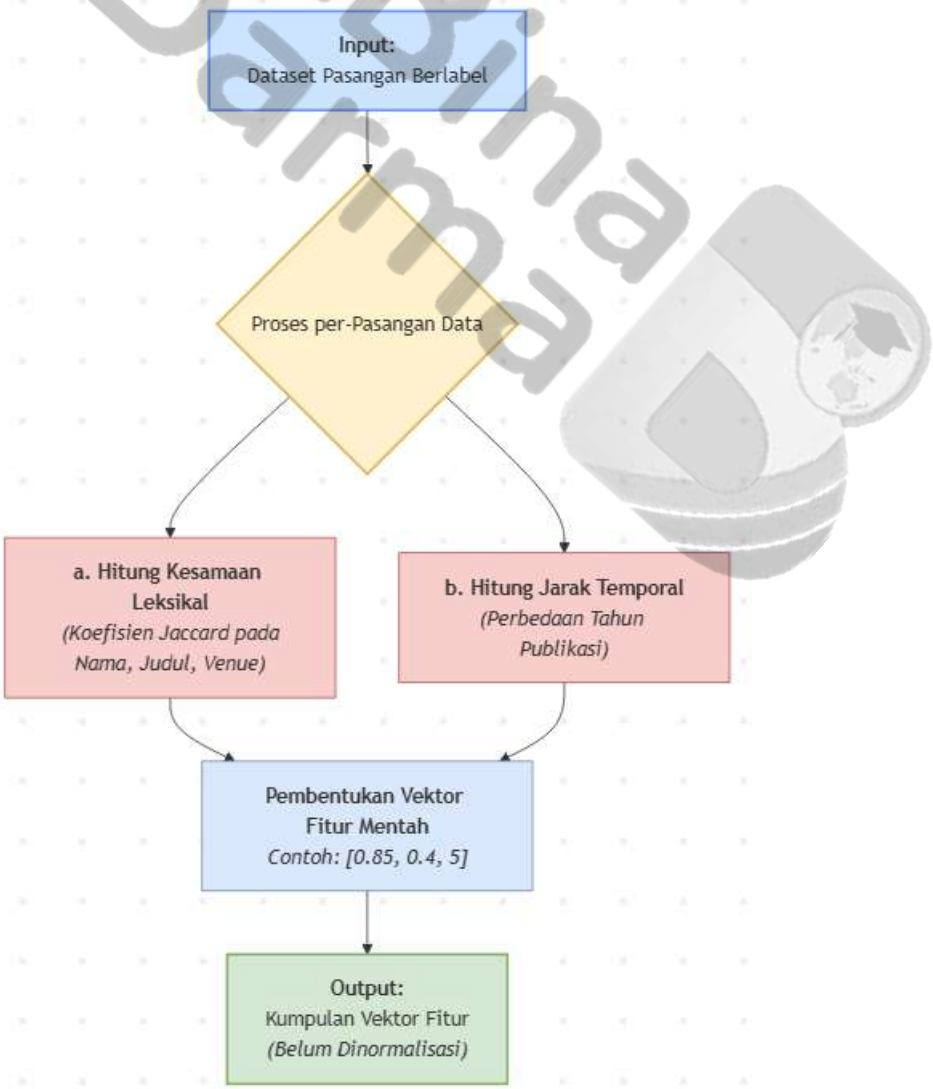
Gambar 3.5 Proses Pembuatan *Labeling* Pasangan Data

3.2.4. Ekstraksi Fitur

Tahap ekstraksi fitur bertujuan untuk menguantifikasi hubungan antara setiap pasangan data menjadi sebuah vektor fitur numerik. Representasi kuantitatif ini penting agar data dapat diproses dan dipelajari oleh model *machine learning*. Dalam penelitian ini, proses ekstraksi fitur melibatkan dua jenis pendekatan utama. Pertama, fitur jarak temporal yang dihitung dari selisih tahun publikasi (*year difference*). Kedua, serangkaian fitur kesamaan leksikal yang diukur menggunakan Koefisien Kesamaan Jaccard (*Jaccard Similarity Coefficient*). Metode Jaccard ini diaplikasikan untuk membandingkan berbagai atribut tekstual yang relevan dari setiap pasangan data, seperti nama penulis, judul publikasi, dan nama jurnal atau konferensi.

a. Fitur Kesamaan Leksikal *(Jaccard Similarity)*

Fitur ini bertujuan untuk mengukur tingkat kesamaan konten berbasis teks antara dua entri data dalam satu pasangan. Untuk mengukurnya, digunakan Koefisien Kesamaan Jaccard *(Jaccard Similarity Coefficient).* Metode ini bekerja dengan cara membandingkan ukuran himpunan irisan (elemen yang sama, A∩B) dengan ukuran himpunan gabungan (seluruh elemen unik, A∪B) dari dua set data. Secara matematis, rumus Jaccard dapat dilihat pada Persamaan (2.1) (Yudhana dkk., 2018).



Gambar 3.6 Diagram Alir Ekstraksi Fitur

Nilai yang dihasilkan berkisar antara 0 (tidak ada kesamaan) hingga 1 (identik sempurna). Dalam penelitian ini, Koefisien Jaccard diterapkan pada empat atribut tekstual berikut:

1) *author_name* (nama penulis utama)
2) *author* (keseluruhan daftar penulis dalam publikasi)
3) *venue* (nama jurnal atau konferensi)
4) *title* (judul publikasi)

Contoh hasil perhitungan kesamaan untuk beberapa pasangan data dapat dilihat pada Tabel 3.3.

Tabel 3.3 Hasil Penghitungan Kesamaan Data

author_name	author	venue	title
1	1	0	0
1	1	0	0.60
1	1	0	0.05
1	1	0	0
…	…	…	…
0.33	0.33	0	0
0.33	0.33	0	0
0.33	0.33	0	0
1	1	0	0.23

b. Fitur Jarak Temporal (*Year Difference*)

Fitur ini diekstraksi untuk mengukur jarak waktu (temporal) antara dua publikasi yang dibandingkan. Asumsi yang mendasari fitur ini adalah bahwa pasangan publikasi dengan kesamaan nama yang tinggi namun memiliki rentang waktu publikasi yang sangat jauh, cenderung merujuk pada individu yang berbeda. Oleh karena itu, fitur ini berfungsi sebagai penyeimbang krusial terhadap fitur kesamaan leksikal. perhitungan dilakukan dengan mencari selisih nilai absolut antara kolom tahun (*year*) dari kedua data. Rumusnya adalah sebagai berikut:

$$\text{YearDifference} = |\text{Year1} - \text{Year2}| \quad (3.2)$$

c. Normalisasi Fitur (*Feature Normalization*)

Nilai mentah yang dihasilkan dari ekstraksi fitur (khususnya *Year Difference*) memiliki rentang yang berbeda dengan fitur *Jaccard Similarity* (yang sudah berada di rentang 0-1). Agar semua fitur memberikan kontribusi yang seimbang selama pelatihan model, diperlukan proses normalisasi. Dalam penelitian ini, digunakan Normalisasi *Min-Max* yang mengubah skala setiap nilai fitur ke dalam rentang seragam antara 0 dan 1. Hasil dari normalisasi fitur *Year Difference* inilah yang ditampilkan pada Tabel 3.4.

Tabel 3.4 Proses Year Difference

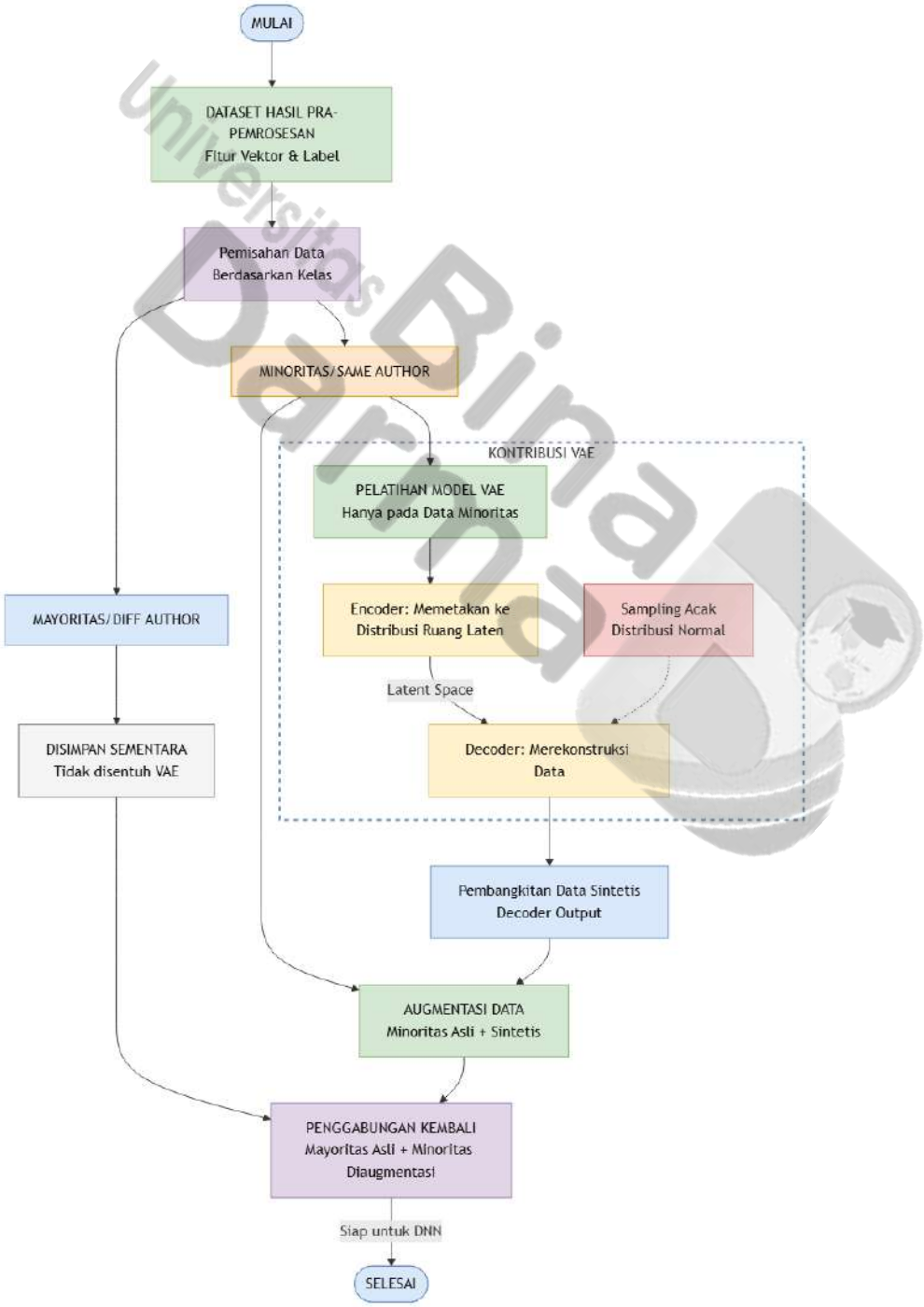
year
0.03
0
0
0.03 . . 0.03
0.21
0.07
0.14

3.2.5. Proses Penanganan Imbalanced Data

A. Penanganan Data dengan Algoritma Variational Autoencoder

Untuk mengatasi tantangan ketidakseimbangan kelas secara fundamental, penelitian ini mengadopsi *Variational Autoencoder* (VAE) sebagai pendekatan utama. Berbeda dengan metode *oversampling* konvensional yang hanya melakukan interpolasi antar titik data yang ada, VAE adalah model generatif yang mampu mempelajari distribusi statistik yang mendasari data kelas minoritas. Dengan pemahaman mendalam ini, VAE dapat menghasilkan sampel-sampel sintetis baru yang beragam dan realistis, yang secara efektif memperkaya representasi kelas minoritas dalam dataset pelatihan. Proses implementasi VAE

untuk menangani ketidakseimbangan data dapat dilihat secara rinci pada Gambar 3.7. Alur kerja tersebut menguraikan tahapan kunci sebagai berikut:



Gambar 3.7 Alur Penanganan Data Tidak Seimbang Menggunakan VAE

Seperti yang diilustrasikan pada Gambar 3.7, alur penanganan data dirancang dengan memisahkan perlakuan antara kelas mayoritas dan minoritas. Strategi utama

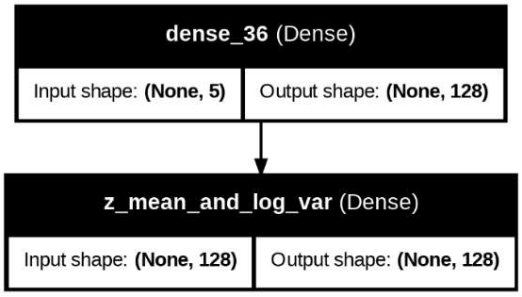
terletak pada isolasi data kelas minoritas yang kemudian diproses melalui *Variational Autoencoder* (VAE) untuk mempelajari distribusi latennya. Proses ini memungkinkan pembangkitan data sintetis yang tidak sekadar menduplikasi, melainkan memperkaya variasi fitur. Setelah jumlah sampel minoritas setara dengan mayoritas melalui proses generasi ini, kedua himpunan data digabungkan kembali untuk membentuk *dataset* latih yang seimbang (rasio 1:1) tanpa mereduksi informasi dari kelas mayoritas. Secara spesifik, rincian teknis dari tahapan awal proses tersebut diuraikan sebagai berikut:

1) Isolasi dan Pelatihan Khusus pada Kelas Minoritas

Langkah awal yang fundamental adalah mengisolasi seluruh sampel data dari kelas minoritas ('Sama') dari set pelatihan. Model VAE dilatih secara eksklusif pada subset data ini. Pendekatan ini memastikan bahwa VAE menjadi "spesialis" yang hanya mempelajari karakteristik, pola, dan variasi yang melekat pada kelas 'Sama', tanpa terdistorsi oleh informasi dari kelas mayoritas.

2) Fase Pembelajaran *Encoder*: Kompresi ke Ruang Laten

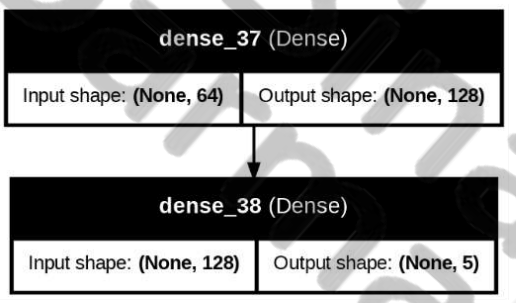
Data kelas minoritas yang telah diisolasi kemudian dimasukkan ke dalam komponen *Encoder* dari VAE. *Encoder* berfungsi sebagai jaringan saraf kompresi yang memetakan data input berdimensi tinggi (vektor 5 fitur) ke dalam Ruang Laten (*Latent Space*) berdimensi rendah. Keunikan VAE terletak pada output *Encoder* yang tidak berupa satu titik vektor, melainkan parameter dari sebuah distribusi probabilitas—secara spesifik, vektor mean (μ) dan vektor log-varian ($\log \sigma^2$). Parameter-parameter ini mendefinisikan sebuah "wilayah" probabilistik dalam ruang laten, yang menangkap esensi dan variasi dari data asli.



Gambar 3.8 VAE Encoder

3) Fase Generasi *Decoder*: Rekonstruksi dari Ruang Laten

Komponen *Decoder* bekerja secara terbalik. *Decoder* mengambil sampel titik acak dari wilayah probabilitas yang telah dipetakan di ruang laten, kemudian merekonstruksinya kembali menjadi data dengan format asli (vektor 5 fitur). Karena ruang laten bersifat kontinu dan terstruktur, pengambilan sampel dari titik-titik yang berbeda memungkinkan *Decoder* untuk menghasilkan sampel data baru yang belum pernah ada sebelumnya, namun tetap konsisten dengan distribusi statistik yang telah dipelajarinya. Ini adalah inti dari kemampuan generatif VAE.



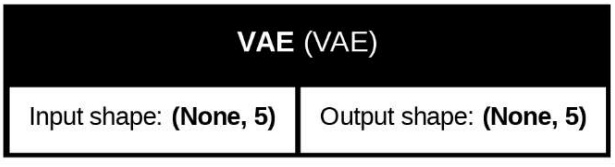
Gambar 3.9 VAE Decoder

4) Optimasi Model Melalui Fungsi Kerugian Ganda

Proses pelatihan VAE dioptimalkan untuk menyeimbangkan dua tujuan melalui fungsi kerugian (*loss function*) gabungan:

Reconstruction Loss: Mengukur seberapa akurat *Decoder* dapat merekonstruksi data asli dari representasi latennya. Hal ini memastikan data sintetis yang dihasilkan tetap setia pada karakteristik data minoritas.

Kullback-Leibler (KL) Divergence Loss: Bertindak sebagai regulator yang memastikan ruang laten yang dipelajari oleh *Encoder* terstruktur dengan baik dan tidak terlalu menyimpang dari distribusi normal standar. Ini sangat penting untuk mencegah *overfitting* dan memungkinkan generasi sampel baru yang beragam dan valid.



Gambar 3.10 VAE Arsitektur

5) Generasi Sampel Sintetis dan Penyeimbangan Dataset

Setelah model VAE berhasil dilatih dan konvergen (mencapai nilai *loss* yang minimal), komponen *Encoder*-nya tidak lagi diperlukan untuk tahap ini. *Decoder* kini berfungsi sebagai generator data. Dengan mengambil sampel acak secara berulang dari ruang laten dan melewatkannya melalui *Decoder*, sejumlah besar data sintetis baru untuk kelas 'Sama' dapat diciptakan. Proses ini dilanjutkan hingga jumlah sampel kelas minoritas setara dengan jumlah sampel kelas mayoritas dalam set pelatihan. Hasil akhirnya adalah sebuah dataset latih yang seimbang, siap untuk digunakan dalam melatih model klasifikasi DNN pada tahap selanjutnya.

Untuk mendapatkan model VAE yang optimal dalam mempelajari dan membangkitkan data kelas minoritas dilakukan proses pencarian hiperparameter (*hyperparameter tuning*). Pemilihan hiperparameter ini sangat krusial karena bertujuan untuk mendapatkan model yang paling baik dalam mempelajari distribusi data kelas minoritas secara efektif, sehingga VAE dapat menghasilkan data sintetis yang berkualitas tinggi. Tiga hiperparameter kunci yang dipilih untuk dioptimalkan adalah:

1) *latent_dim* (Dimensi Laten):

Parameter ini menentukan ukuran "ruang laten", yaitu representasi data yang telah dimampatkan oleh *encoder* untuk menangkap esensi atau struktur tersembunyi dari data. Nilai *latent_dim* harus dioptimalkan, sebab nilai yang terlalu kecil berisiko *underfitting*, sedangkan nilai yang terlalu besar dapat menangkap derau (*noise*) dan gagal melakukan generalisasi, sehingga memengaruhi kualitas data sintetis yang dihasilkan.

2) *batch_size* (Ukuran Tumpukan):

Parameter ini menentukan jumlah sampel data yang diproses model dalam satu iterasi sebelum bobotnya diperbarui, yang memengaruhi stabilitas dan kecepatan pelatihan. *Batch size* yang kecil membuat pembaruan bobot lebih sering namun "berisik" (*noisy*), sementara *batch size* yang besar memberikan estimasi gradien yang lebih stabil tetapi memerlukan lebih banyak memori.

3) *learning_rate* (Laju Pembelajaran):

Parameter ini mengontrol besaran penyesuaian bobot model oleh *optimizer* (Adam) pada setiap pembaruan. Laju pembelajaran sangat berpengaruh; jika terlalu tinggi, proses pelatihan bisa tidak stabil, dan jika terlalu rendah, pelatihan akan berjalan sangat lambat sehingga sulit mencapai konvergensi yang optimal.

B. Penangan Data dengan Algoritma SMOTE

Sebagai metode pembanding (*baseline*) untuk mengevaluasi pendekatan generatif VAE, penelitian ini mengimplementasikan *Synthetic Minority Over-sampling Technique* (SMOTE). Berbeda dengan VAE yang mempelajari distribusi data secara probabilistik, SMOTE adalah algoritma yang beroperasi secara geometris di ruang fitur untuk mengatasi ketidakseimbangan kelas. SMOTE bekerja dengan cara menciptakan sampel sintetis baru melalui interpolasi linier antara sampel-sampel minoritas yang ada. Pendekatan ini dipilih karena merupakan standar industri yang telah terbukti efektif dan menyediakan titik acuan yang kuat untuk mengukur nilai tambah dari metode yang lebih kompleks. Proses implementasi SMOTE dapat diuraikan dalam beberapa tahapan kunci sebagai berikut:

1) Identifikasi Kelas Minoritas dan Tetangga Terdekat

Langkah pertama SMOTE adalah fokus pada sampel-sampel dari kelas minoritas ('Sama') di dalam set pelatihan. Untuk setiap sampel minoritas, algoritma akan mengidentifikasi *k* tetangga terdekatnya (*k-nearest neighbors*) yang juga berasal dari kelas minoritas.

2) Pemilihan Tetangga dan Perhitungan Vektor Selisih

Setelah *k* tetangga terdekat diidentifikasi untuk sebuah sampel minoritas, satu tetangga akan dipilih secara acak dari kelompok tersebut. Selanjutnya, algoritma menghitung vektor selisih (perbedaan) antara fitur-fitur dari sampel asli dan fitur-fitur dari tetangga yang dipilih. Vektor ini merepresentasikan "jalur" atau "garis lurus" yang menghubungkan kedua titik data tersebut di ruang fitur.

3. Generasi Sampel Sintetis Melalui Interpolasi

Inti dari mekanisme SMOTE adalah penciptaan sampel baru di sepanjang jalur yang telah dihitung. Sebuah nilai acak antara 0 dan 1 akan dibangkitkan. Nilai ini kemudian dikalikan dengan vektor selisih, dan hasilnya ditambahkan kembali ke fitur sampel asli. Secara matematis, ini menempatkan sebuah titik data sintetis baru di lokasi acak pada segmen garis yang menghubungkan sampel asli dengan tetangganya. Proses ini diulang untuk banyak sampel minoritas hingga target penyeimbangan tercapai.

4. Konfigurasi Strategi Penyeimbangan

Proses generasi data diatur oleh *sampling_strategy*. Dalam penelitian ini, strategi yang digunakan adalah *'auto'*, yang menginstruksikan SMOTE untuk terus membangkitkan sampel sintetis untuk kelas minoritas hingga jumlahnya sama persis dengan jumlah sampel pada kelas mayoritas. Hal ini memastikan bahwa dataset hasil memiliki rasio kelas 1:1, menciptakan kondisi yang seimbang untuk pelatihan model selanjutnya.

5. Aplikasi pada Data Latih untuk Menghasilkan Dataset Seimbang

Seluruh proses SMOTE, mulai dari identifikasi tetangga hingga generasi sampel, diterapkan secara eksklusif pada set data latih. Hasil akhirnya adalah sebuah dataset latih baru yang seimbang, di mana kelas minoritas telah diperkaya dengan sampel-sampel sintetis yang dihasilkan melalui interpolasi. Dataset inilah yang kemudian siap digunakan untuk melatih model klasifikasi DNN.

Untuk mendapatkan model SMOTE yang optimal dalam membangkitkan data kelas minoritas sebagai *baseline* yang kuat, dilakukan proses pencarian *hyperparameter* (*hyperparameter tuning*). Meskipun SMOTE lebih sederhana dibandingkan VAE, pemilihan *hyperparameter* ini tetap krusial karena bertujuan untuk menemukan cara paling efektif dalam menciptakan sampel sintetis. Konfigurasi yang tepat akan menghasilkan data sintetis yang representatif dan bermanfaat bagi model klasifikasi, sementara konfigurasi yang buruk dapat

memasukkan *noise* dan justru menurunkan performa. Dua *hyperparameter* kunci yang dipilih untuk dioptimalkan adalah:

1) *k_neighbors*

   Parameter menentukan jumlah sampel minoritas terdekat untuk dijadikan referensi dalam pembuatan satu sampel sintetis baru. Nilai *k* ini sangat memengaruhi karakteristik data yang dihasilkan; nilai yang terlalu kecil berisiko membuat SMOTE terlalu fokus pada pola lokal dan rentan terhadap *outlier*, sementara nilai yang terlalu besar dapat menyebabkan interpolasi antara klaster-klaster minoritas yang berbeda sehingga mengaburkan batas keputusan.

2) *sampling_strategy*

   Parameter ini yang mengontrol rasio akhir antara jumlah sampel kelas minoritas dan mayoritas. Meskipun penyeimbangan penuh (rasio 1:1) adalah praktik umum, proses *tuning* ini juga bertujuan untuk memvalidasi apakah rasio tersebut, atau rasio yang sedikit lebih rendah, merupakan strategi yang paling optimal untuk meningkatkan metrik evaluasi pada dataset ini.

3.2.6. Proses Klasifikasi

Tahap selanjutnya pasca-augmentasi data dengan *Variational Autoencoder* (VAE) adalah pengembangan model klasifikasi menggunakan arsitektur *Deep Neural Network* (DNN). Tujuan dari tahap ini adalah untuk melatih sebuah model yang mampu mengekstraksi pola-pola kompleks dari dataset yang kini telah seimbang. Dengan menggunakan gabungan data asli dan data sintetis VAE, model DNN diharapkan dapat terhindar dari bias yang sering terjadi pada dataset tidak seimbang, sehingga meningkatkan reliabilitas dan akurasi prediksinya secara signifikan.

Arsitektur DNN yang diimplementasikan disusun secara sekuensial, sebuah desain yang terbukti efektif untuk data tabular. Komponen arsitektur ini dirinci sebagai berikut:

1) Lapisan Masukan (*Input Layer*): Lapisan ini memiliki jumlah neuron yang sama dengan dimensi fitur dataset, berfungsi sebagai antarmuka input.
2) Lapisan Tersembunyi (*Hidden Layers*): Dua lapisan tersembunyi digunakan untuk menangkap hierarki fitur. Keduanya menggunakan fungsi aktivasi ReLU, yang dipilih untuk mencegah masalah *vanishing gradient* dan mempercepat konvergensi.
3) Lapisan Regularisasi (*Dropout*): Setelah setiap lapisan tersembunyi, sebuah lapisan Dropout dengan laju tertentu disisipkan. Teknik ini sangat penting untuk meningkatkan kemampuan generalisasi model dengan cara mengurangi ketergantungan antar neuron dan mencegah *overfitting*.
4) Lapisan Keluaran (*Output Layer*): Sebuah neuron tunggal dengan fungsi aktivasi *Sigmoid* digunakan untuk menghasilkan probabilitas prediksi, yang secara alami cocok untuk tugas klasifikasi biner.

Model: "DNN_Model"

Layer (type)	Output Shape	Param #
hidden_layer_1 (Dense)	(None, 128)	768
dropout_1 (Dropout)	(None, 128)	0
hidden_layer_2 (Dense)	(None, 64)	8,256
dropout_2 (Dropout)	(None, 64)	0
output_layer (Dense)	(None, 1)	65

Total params: 9,089 (35.50 KB)
Trainable params: 9,089 (35.50 KB)
Non-trainable params: 0 (0.00 B)

Gambar 3.11 Arsitektur DNN

Sebelum pelatihan dimulai, model dikompilasi dengan tiga komponen utama yang menentukan perilaku pembelajarannya. Konfigurasi ini mencakup:

1) Fungsi Kerugian (*Loss Function*): Menggunakan *binary_crossentropy*, yang merupakan fungsi kerugian standar dan paling efektif untuk masalah klasifikasi biner.
2) *Optimizer*: Menggunakan optimizer Adam, sebuah algoritma adaptif yang efisien dan populer karena kemampuannya menyesuaikan laju pembelajaran

(*learning rate*) secara otomatis.

3) Metrik Pelatihan: *accuracy* (akurasi) dipilih sebagai metrik utama untuk memantau kinerja model pada setiap *epoch*.

Pada tahap klasifikasi, model *Deep Neural Network (DNN)* digunakan sebagai algoritma pemodelan utama. Untuk mengevaluasi dampak penyeimbangan data secara komprehensif, penelitian ini merancang tiga skenario pengujian dengan menerapkan arsitektur DNN yang identik pada tiga himpunan data latih yang berbeda:

1. Skenario 1 - *Baseline*: Proses klasifikasi dilakukan menggunakan *dataset* asli yang tidak seimbang.
2. Skenario 2 – VAE : Proses klasifikasi dilakukan menggunakan *dataset* yang telah diseimbangkan oleh *Variational Autoencoder* (VAE).
3. Skenario 3 – SMOTE : Proses klasifikasi dilakukan menggunakan *dataset* yang telah diseimbangkan oleh SMOTE, yang berfungsi sebagai metode pembanding (*benchmark*) untuk VAE.

Setiap himpunan data tersebut selanjutnya dibagi lagi dengan rasio 80:20 untuk membentuk set pelatihan dan set validasi bagi model DNN. Dengan demikian, data latih DNN merepresentasikan 80% dari data sesuai skenario, dan data validasi DNN merepresentasikan 20% sisanya.

Untuk memastikan perbandingan yang objektif dan setara, arsitektur serta *hyperparameter* DNN yang digunakan dijaga konstan untuk ketiga skenario pengujian. Pendekatan ini krusial untuk mengisolasi variabel yang diuji, yaitu teknik penyeimbangan data. Dengan demikian, setiap perbedaan kinerja yang teramati antar skenario dapat diatribusikan secara langsung pada efektivitas metode VAE atau SMOTE, bukan karena variasi pada arsitektur model. Parameter tetap yang telah dioptimalkan sebelumnya, dengan rincian sebagai berikut:

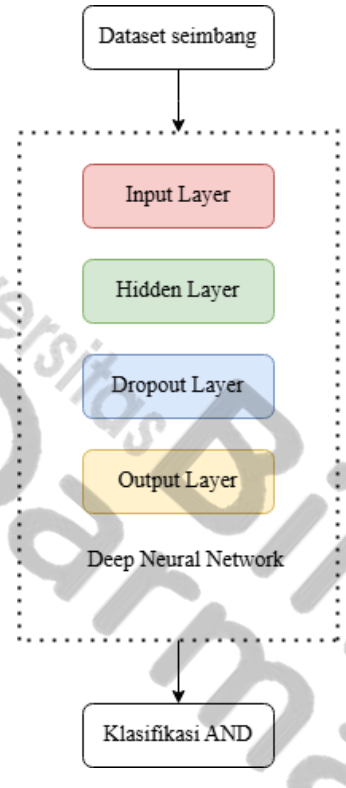
1. *Input Layer*: Dimensi 5 (sesuai jumlah fitur).
2. *Hidden Layer* 1: 128 neuron dengan aktivasi ReLU dan *Dropout* 0.2.
3. *Hidden Layer* 2: 64 neuron dengan aktivasi ReLU dan *Dropout* 0.2. Arsitektur dengan dua lapisan tersembunyi ini dirancang cukup dalam untuk menangkap pola-pola non-linear yang kompleks dalam

data. Fungsi aktivasi ReLU dipilih karena efisiensi komputasinya dan kemampuannya mengatasi masalah *vanishing gradient*. Lapisan *Dropout* dengan *rate* 0.2 diintegrasikan sebagai teknik regularisasi untuk mengurangi risiko *overfitting*.

4. *Output Layer*: 1 neuron dengan aktivasi *Sigmoid* (untuk klasifikasi biner). Fungsi aktivasi Sigmoid digunakan karena kemampuannya menghasilkan keluaran probabilitas antara 0 dan 1, yang sesuai untuk tugas klasifikasi biner.
5. *Optimizer*: Adam dengan *learning rate* tetap 5e-05.
6. *Loss Function*: *binary_crossentropy*. Kombinasi *optimizer* Adam dan *loss function binary_crossentropy* merupakan standar industri yang terbukti efektif dan stabil untuk masalah klasifikasi biner.

Proses pelatihan untuk setiap skenario dijalankan dengan *batch size* 128 selama maksimal 50 *epoch*. Guna mencegah *overfitting* dan memastikan model dengan generalisasi terbaik disimpan, diterapkan mekanisme *early stopping* dengan *patience* (toleransi) 10 *epoch*. Mekanisme ini secara spesifik memantau metrik *validation loss* (val_loss). Apabila tidak terjadi perbaikan (penurunan) pada val_loss selama 10 *epoch* berturut-turut, pelatihan akan dihentikan secara otomatis, dan bobot model dari *epoch* dengan val_loss terendah akan dipulihkan sebagai model final.

Fokus penelitian pada tahap ini bukanlah untuk mengoptimalkan arsitektur DNN, melainkan untuk mengevaluasi dampak dari teknik penyeimbangan data yang telah diterapkan sebelumnya. Untuk mencapai tujuan tersebut, penelitian ini menerapkan satu arsitektur DNN yang telah ditetapkan dan dijaga konstan pada ketiga skenario pengujian (data asli, data VAE, dan data SMOTE). Pendekatan ini krusial untuk memastikan perbandingan yang setara dan objektif, sehingga perbedaan kinerja yang teramati dapat diatribusikan secara valid kepada efektivitas masing-masing metode penyeimbangan data.



Gambar 3.12 Workflow Arsitektur DNN

3.2.7. Analisa Hasil

Tahapan ini menyajikan evaluasi komprehensif untuk sistem klasifikasi *author name disambiguation*. Fokus utama penelitian adalah menangani data tidak seimbang menggunakan metode *Variational Autoencoder (VAE)*, yang kemudian diaplikasikan pada model klasifikasi *Deep Neural Network (DNN)*. Untuk mengevaluasi efektivitas metode tersebut, penelitian ini merancang tiga skenario pengujian. Skenario pertama berfungsi sebagai tolok ukur (*baseline*), di mana klasifikasi DNN diterapkan langsung pada himpunan data asli yang tidak seimbang. Skenario kedua menguji dampak penyeimbangan data menggunakan VAE sebelum proses klasifikasi, sementara skenario ketiga menerapkan algoritma SMOTE untuk tujuan yang sama.

Analisis kinerja komparatif dari ketiga skenario tersebut dilakukan melalui serangkaian metrik evaluasi kuantitatif yang meliputi *accuracy*, *precision*, *recall*, *specificity*, dan *f-measure*. Selain itu, kurva *Receiver Operating Characteristic (ROC)* beserta nilai *Area Under the Curve (AUC)* juga dianalisis untuk memberikan

evaluasi yang lebih utuh terhadap kemampuan diskriminatif setiap model pada berbagai tingkat ambang batas (*threshold*). Kombinasi metrik ini akan memberikan gambaran mendalam tentang kekuatan dan kelemahan sistem yang diusulkan, khususnya dalam menentukan pendekatan penanganan data tidak seimbang yang paling efektif.

3.3. Jadwal Penelitian

Jadwal penelitian tesis ini dimulai dari tahapan studi pustaka, persiapan data, diikuti pra-pengolahan data kemudian dilakukan penanganan data tidak seimbang menggunakan VAE. Selanjutnya, klasifikasi akan dilakukan dengan DNN, diakhiri dengan analisis hasil dan penarikan kesimpulan.

Tabel 3.5 Jadwal Penelitian

Tahapan Penelitian	Bulan					
	Januari	Februari	Maret	April	Mei	Juni
Studi Pustaka						
Persiapan Data						
Pra-Pengolahan						
Proses Penanganan Data Tidak Seimbang VAE						
Proses Klasifikasi DNN						
Analisa Hasil						
Penarikan Kesimpulan						

BAB IV
HASIL DAN PEMBAHASAN

4.1. Hasil

Sub-bab ini menguraikan hasil eksperimen secara sistematis, diawali dengan tahap penanganan data yang meliputi analisis distribusi kelas awal serta proses pembangkitan data sintetis untuk mengatasi ketidakseimbangan data menggunakan dua metode *Variational Autoencoder* (VAE) dan SMOTE. Hasil tersebut menjadi dasar bagi pengujian klasifikasi pada tiga skenario utama: Skenario 1 (*Baseline*), Skenario 2 (VAE+DNN), dan Skenario 3 (SMOTE+DNN). Seluruh temuan kinerja model dipaparkan secara komprehensif melalui tabel metrik evaluasi, *confusion matrix*, dan visualisasi kurva kinerja untuk memberikan gambaran faktual yang jelas.

4.1.1. Penanganan Data

Tahap ini berfokus pada persiapan dataset dan penerapan teknik *oversampling* untuk mengatasi masalah ketidakseimbangan kelas yang signifikan. Dataset yang digunakan berisi fitur-fitur untuk memprediksi apakah dua tulisan berasal dari penulis yang sama (*same_author*) atau berbeda (*diff_author*).

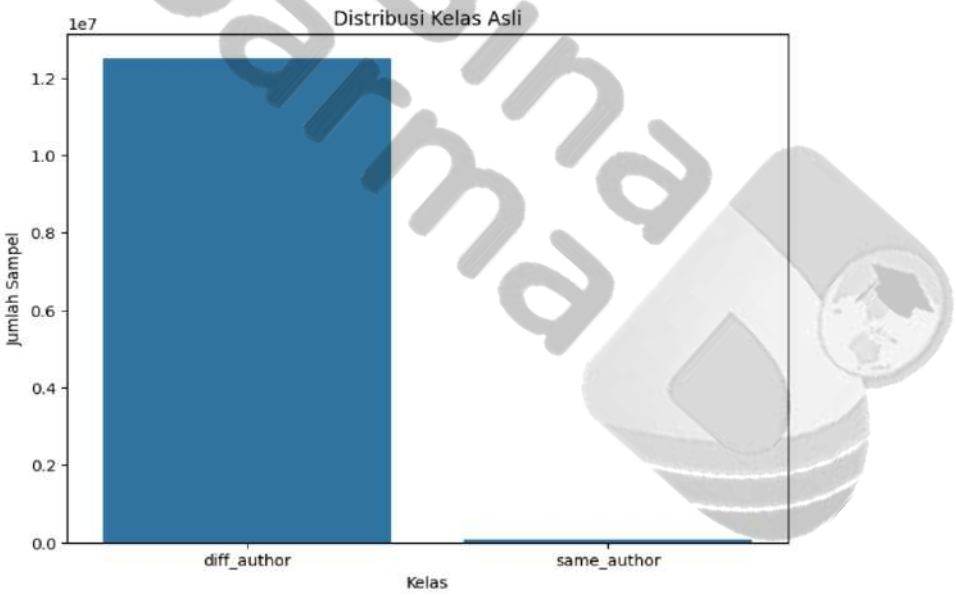
A. Analisis Distribusi Kelas Awal

Setelah melalui tahap pra-pemrosesan dan pembuatan pasangan (*pairwise matching*) dari dataset DBLP, dihasilkan dataset dengan total 12.587.653 pasangan data. Analisis awal terhadap distribusi kelas pada dataset ini menunjukkan adanya ketidakseimbangan yang sangat signifikan, seperti yang dirangkum pada Tabel 4.1.

Tabel 4.1 Distribusi Kelas Pairwise Matching pada Dataset

Kelas	Jumlah Sampel	Persentase
Mayoritas ('diff_author' / 0)	12.509.433	99.38%
Minoritas ('same_author' / 1)	78.220	0.62%
Total	12.587.653	100%

Rasio antara kelas mayoritas dan minoritas yang melebihi 160:1 mengindikasikan bahwa model klasifikasi yang dilatih pada data ini akan memiliki bias ekstrem dan kecenderungan performa yang sangat buruk pada prediksi kelas *'same_author'*. Kondisi ini berisiko tinggi menyebabkan model klasifikasi menjadi bias dan cenderung mengabaikan kelas minoritas. Oleh karena itu, penerapan teknik penyeimbangan data, yang dalam penelitian ini difokuskan pada VAE dan SMOTE, menjadi langkah yang krusial untuk memastikan model klasifikasi dapat mempelajari pola dari kelas minoritas secara efektif.

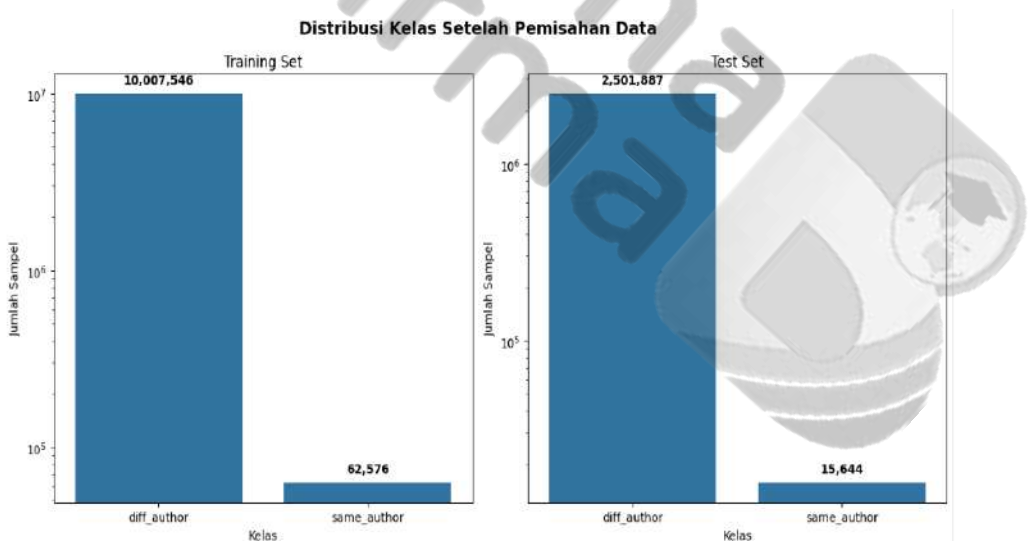


Gambar 4.1 Distribusi Kelas Asli

B. Pensklaaan Fitur

Proses penskalaan fitur dimulai dengan memuat dataset yang telah dilakukan proses pemasangan data. Dataset ini memiliki total 12.587.653 sampel dengan 5 fitur dan 1 kolom label. Langkah selanjutnya adalah memisahkan dataset menjadi data latih dan data uji dengan proporsi 80:20. Pemisahan ini dilakukan secara stratifikasi untuk memastikan proporsi kelas minoritas dan mayoritas tetap terjaga di kedua set data. Penggunaan stratifikasi ini esensial untuk menjamin bahwa kelas minoritas (0.62%) terdistribusi secara proporsional, menghindari skenario di mana salah satu set data kehilangan representasi kelas minoritas yang sudah sangat sedikit.

Setelah pemisahan, dilakukan penskalaan fitur menggunakan *MinMaxScaler*. Penskalaan ini mentransformasikan nilai setiap fitur ke dalam rentang [0, 1]. Proses *fit_transform* hanya diterapkan pada data latih untuk mempelajari parameter skala, yang kemudian parameter yang sama digunakan untuk mentransformasi data uji (transform). Langkah ini krusial untuk mencegah kebocoran data (*data leakage*) dari set pengujian ke dalam model. Pendekatan ini merupakan praktik standar dalam *machine learning* untuk menjaga integritas data uji sebagai evaluator performa model yang belum pernah dilihat (*unseen*) selama proses pelatihan. Hasil dari tahap ini adalah memuat ukuran data latih (setelah penskalaan) sebesar 10.070.122, 5 dan ukuran data tes (setelah penskalaan) sebesar 2.517.531, 5.



Gambar 4.2 Distribusi Kelas Setelah Pemisahan Data

Visualisasi pada Gambar 4.2 mengonfirmasi bahwa distribusi kelas yang tidak seimbang (rasio 99.38% : 0.62%) tetap terjaga secara proporsional baik pada data latih maupun data uji, yang menunjukkan keberhasilan proses pemisahan stratifikasi.

4.1.2. Penanganan Data Tidak Seimbang Menggunakan VAE

Optimalisasi model VAE bertujuan untuk memaksimalkan kemampuan model dalam mempelajari dan membangkitkan data kelas minoritas. Proses ini dilakukan melalui penyetelan hiperparameter (*hyperparameter tuning*)

menggunakan metode *Random Search* sebanyak 10 iterasi percobaan. Berikut adalah rincian hiperparameter yang dioptimalkan:

a) *latent_dim* (dimensi ruang laten): [2, 4, 8, 16, 32, 64]

b) *batch_size* (ukuran batch): [16, 32, 64, 128, 256]

c) *learning_rate* (laju pembelajaran): [1e-4, 5e-4, 1e-3, 5e-3, 1e-2]

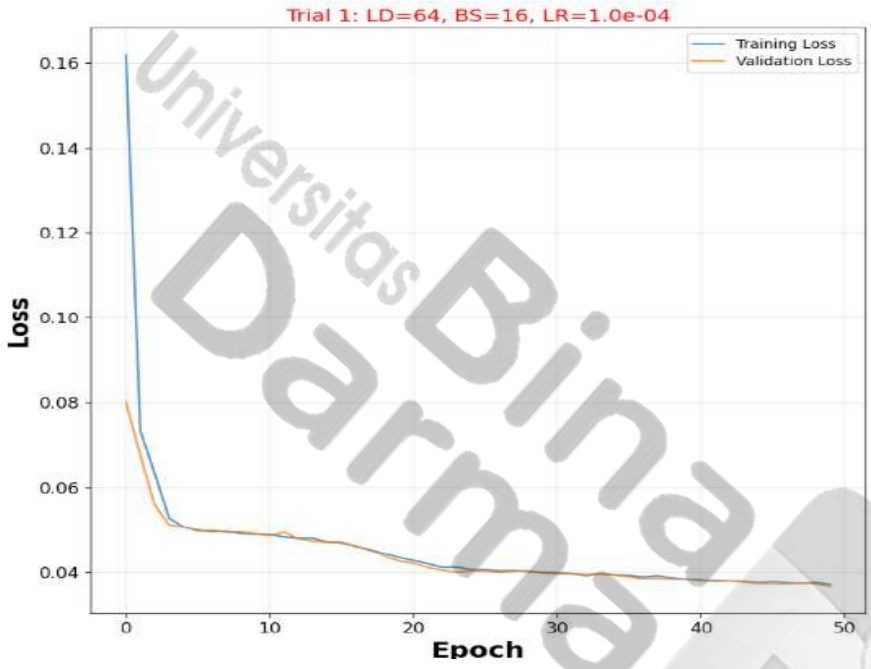
Setiap kombinasi dilatih selama maksimal 50 *epoch* dengan mekanisme *early stopping* (*patience*=5) pada *validation loss*. Pada setiap percobaan *tuning*, data kelas minoritas (62.576 sampel) dibagi lagi dengan rasio 80:20. Sebanyak 80% digunakan untuk melatih model VAE, dan 20% digunakan sebagai set validasi untuk mengukur *validation loss* dan mencegah *overfitting*. Tabel 4.2 rincian dari kombinasi yang diuji sebagai berikut:

Tabel 4.2 Kombinasi Hyperparamater Tuning VAE

latent_dim	batch_size	learning_rate	Loss Validasi
64	16	1.00e-04	0.03655
32	64	5.00e-04	0.04924
16	256	1.00e-02	0.07238
16	64	1.00e-03	0.08016
8	128	5.00e-03	0.10478
8	16	5.00e-03	0.10524
2	128	1.00e-04	0.13800
2	64	5.00e-04	0.13816
2	64	5.00e-03	0.13817
4	256	1.00e-02	0.13875

Dari ke-10 percobaan tersebut, hasil akhir menunjukkan bahwa kombinasi ke-1 memberikan performa terbaik dengan mencapai *Loss* Validasi terendah sebesar 0.03655. Ini menandakan bahwa model dengan konfigurasi tersebut memiliki kemampuan generalisasi terbaik pada data minoritas yang tidak terlihat saat pelatihan, sehingga dipilih sebagai model final untuk menghasilkan data

sintetis. Gambar 4.3 menunjukkan grafik dari *Training Loss* dan *Validation Loss* dari *Hyperparameter Tuning* VAE pada kombinasi pertama.



Gambar 4.3 Kombinasi ke-1 Performa Terbaik Hyperparameter Tuning VAE

Berdasarkan hasil evaluasi dari kesepuluh iterasi percobaan tersebut, diperoleh kombinasi hiperparameter optimal yang menghasilkan nilai *loss* validasi terendah. Konfigurasi terbaik ini kemudian diterapkan untuk tahap selanjutnya, yaitu pembangkitan data sintetis dan penyeimbangan dataset.

Tahap selanjutnya, model VAE final dilatih kembali menggunakan hiperparameter terbaik tersebut secara khusus pada 62.576 sampel kelas minoritas dari data latih. Pelatihan yang difokuskan secara eksklusif pada kelas minoritas ini bertujuan agar VAE dapat mempelajari distribusi dan struktur data yang fundamental dari kelas *'same_author'* tanpa terpengaruh oleh dominasi pola dari kelas mayoritas. Setelah model terlatih secara optimal, sebanyak 9.944.962 sampel sintetis dibangkitkan untuk mencapai keseimbangan jumlah data dengan kelas mayoritas.

Proses ini dilakukan dengan mengambil sampel dari distribusi normal standar, yang secara konseptual berarti mengambil titik-titik acak dari ruang laten (ruang representasi) yang telah dipelajari oleh VAE. Bagian *decoder* dari VAE kemudian

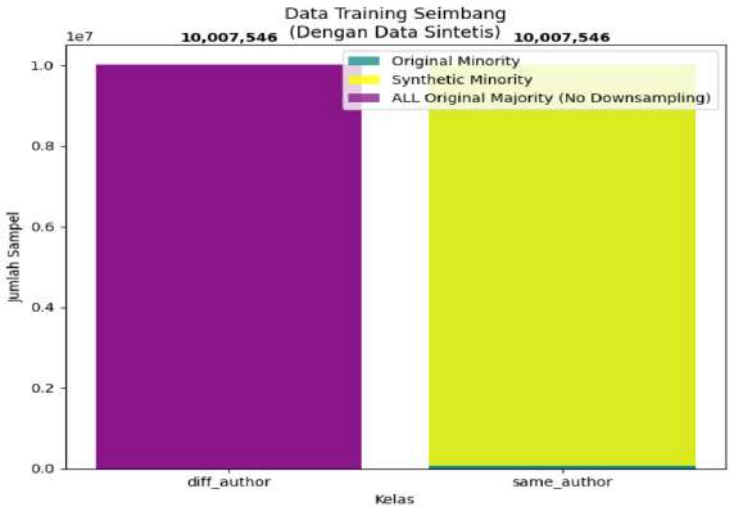
bertindak sebagai komponen generatif yang menerjemahkan titik-titik acak ini kembali menjadi data dengan format fitur yang sama seperti data asli. Keunggulan pendekatan ini adalah data sintetik yang dihasilkan bukan sekadar salinan atau interpolasi, melainkan sampel-sampel baru yang beragam namun tetap mempertahankan karakteristik inti dari kelas minoritas.

Sampel sintetik ini kemudian digabungkan dengan data latih asli. Hasilnya adalah dataset latih yang seimbang sempurna (rasio 1:1), seperti yang ditunjukkan pada Tabel 4.3. Dengan demikian, dataset latih yang baru ini memberikan kesempatan bagi model klasifikasi DNN untuk mempelajari batas keputusan yang lebih adil dan tidak bias terhadap satu kelas.

Tabel 4.3 Distribusi Kelas Setelah Penyeimbangan Data dengan VAE

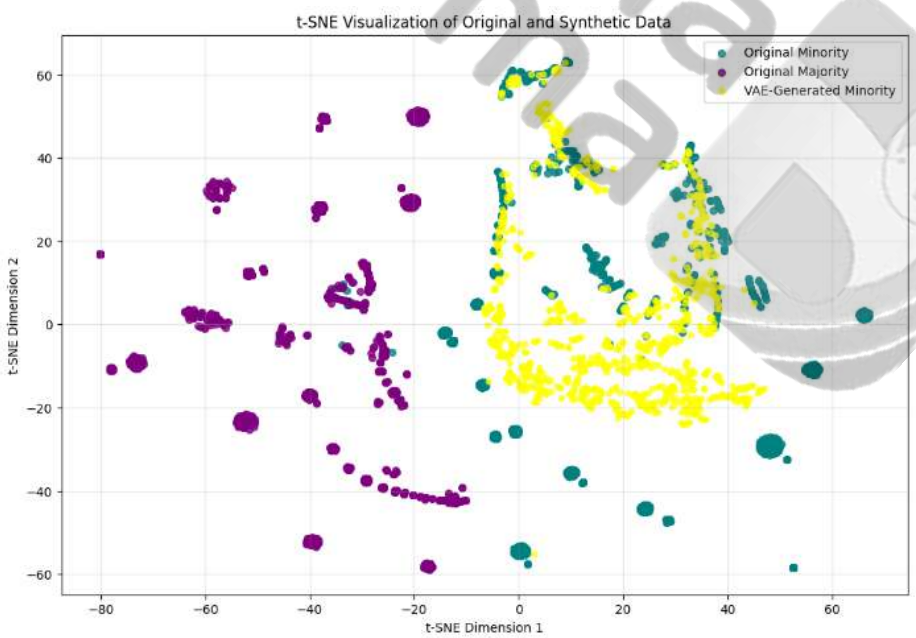
Kelas	Jumlah Sampel	Persentase
Mayoritas ('diff_author' / 0)	10.007.546	50.00%
Minoritas ('same_author' / 1)	10.007.546	50.00%
Total	20.015.092	100%

Data numerik pada tabel di atas mengonfirmasi bahwa tujuan penyeimbangan data telah tercapai. Sebagai konfirmasi visual dan untuk mempermudah pemahaman, Gambar 4.4 mengilustrasikan distribusi yang kini telah seimbang. Tinggi batang yang sama antara kelas *diff_author* dan *same_author* menegaskan keberhasilan proses *oversampling*.



Gambar 4.4 Distribusi Kelas Setelah Penyeimbangan Data

Selain penyeimbangan kuantitatif, kualitas data sintetis juga divalidasi secara kualitatif menggunakan teknik visualisasi t-SNE (*t-Distributed Stochastic Neighbor Embedding*). Dalam proses ini, sampel minoritas asli dan sampel sintetis yang dibangkitkan oleh VAE dipetakan ke dalam satu visualisasi untuk dianalisis distribusinya. Hasilnya, seperti yang akan ditunjukkan pada Gambar 4.5, menunjukkan bahwa titik-titik data yang merepresentasikan sampel sintetis tersebar dan menyatu dengan sangat baik dengan klaster data minoritas asli. Tidak terbentuknya klaster yang terpisah memberikan bukti kualitatif yang kuat bahwa model VAE telah berhasil mempelajari distribusi data yang kompleks. Hal ini menandakan bahwa data sintetis yang dihasilkan memiliki karakteristik yang sangat mirip dengan data aslinya.



Gambar 4.5 Visualisasi t-SNE dari sampel data asli dan sintetis

Setelah proses pelatihan selesai, arsitektur VAE kini siap untuk berfungsi sebagai generator data. Bagian *decoder* dari model yang sudah terlatih digunakan untuk membangkitkan sampel-sampel sintetis yang baru dan realistis. Proses ini dilakukan dengan mengambil sampel vektor secara acak dari ruang laten yang sudah teratur, lalu memasukkannya ke dalam *decoder*. Output yang dihasilkan adalah data baru yang memiliki karakteristik serupa dengan data kelas minoritas

asli, namun bukan merupakan duplikat. Sampel-sampel sintetis ini kemudian dibangkitkan hingga jumlahnya setara dengan jumlah sampel pada kelas mayoritas. Terakhir, sampel sintetis ini digabungkan dengan dataset asli untuk membentuk sebuah himpunan data pelatihan baru yang seimbang dan siap digunakan untuk melatih model klasifikasi.

Dalam penelitian ini, efektivitas *Variational Autoencoder* (VAE) sebagai teknik augmentasi data untuk menangani ketidakseimbangan kelas akan diuji. Untuk mengukur performanya secara objektif, model VAE akan dibandingkan dengan sebuah metode *baseline* yang umum digunakan dalam literatur, yaitu SMOTE. SMOTE merupakan teknik *over-sampling* yang tidak hanya menduplikasi data minoritas, tetapi menciptakan sampel sintetis baru berdasarkan interpolasi fitur dari data tetangga terdekatnya. Pemilihan SMOTE sebagai *baseline* didasarkan pada reputasinya sebagai metode yang efektif, mudah diimplementasikan, dan terbukti secara signifikan lebih unggul daripada teknik dasar seperti *random over-sampling*. Dengan demikian, perbandingan ini akan menjawab apakah pendekatan generatif mendalam (*deep generative approach*) seperti VAE dapat memberikan peningkatan performa yang lebih signifikan pada model klasifikasi akhir dibandingkan dengan pendekatan statistik terkemuka seperti SMOTE.

4.1.3. Penanganan Data Tidak Seimbang Menggunakan SMOTE

Sebagai metode pembanding (*benchmark*) terhadap VAE, penelitian ini menerapkan teknik *Synthetic Minority Over-sampling Technique* (SMOTE). Kualitas data sintetis yang dihasilkan oleh SMOTE sangat bergantung pada konfigurasi hiperparameternya, terutama dalam menentukan area interpolasi antar-sampel. Penggunaan parameter *default* tidak selalu menjamin hasil yang optimal dan justru berisiko menciptakan *noise* atau batas keputusan yang kabur. Oleh karena itu, tahapan optimasi hiperparameter menjadi langkah krusial untuk memastikan bahwa SMOTE dapat membangkitkan sampel minoritas yang representatif dan berkualitas tinggi.

Proses *tuning* dilakukan menggunakan *RandomizedSearchCV* dengan validasi silang 3-kali lipat (*3-fold cross-validation*) untuk memastikan hasil yang

robust. Untuk mengevaluasi setiap kombinasi *hyperparameter*, sebuah *pipeline* yang terdiri dari SMOTE dan *classifier RandomForestClassifier* digunakan. Metrik evaluasi yang menjadi target optimasi adalah *F1-Score*, karena metrik ini memberikan ukuran yang seimbang antara *precision* dan *recall*, yang sangat relevan untuk dataset yang tidak seimbang.

Proses pencarian dilakukan dengan menguji 10 kombinasi unik dari nilai-nilai yang telah didefinisikan. Tabel 4.4 merinci beberapa kombinasi yang diuji dan hasilnya:

Tabel 4.4 Kombinasi Hyperparameter Tuning SMOTE

k_neighbors	*sampling_strategy*	*F1-Score*
20	*auto*	0.9909
3	0.5	0.9874
10	0.8	0.9892
7	0.7	0.9900
3	0.7	0.9906
7	*Auto*	0.9901
15	0.5	0.9882
3	0.5	0.9889
10	0.5	0.9865
5	0.8	0.9868

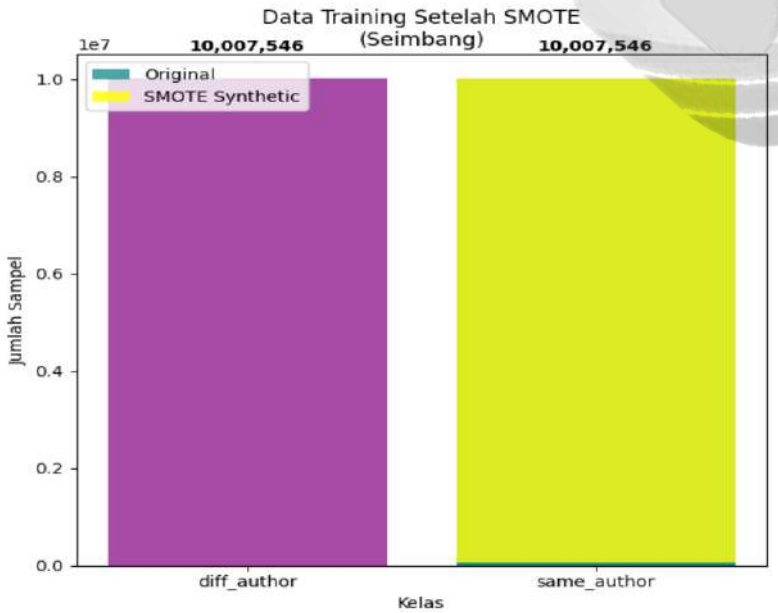
Dari 10 percobaan *hyperparameter tuning* yang dijalankan, hasil akhir menunjukkan bahwa kombinasi *k_neighbors*=20 dan *sampling_strategy='auto'* memberikan performa terbaik. Kombinasi ini berhasil mencapai *F1-Score* Validasi tertinggi sebesar 0.9909. Nilai ini menandakan bahwa konfigurasi SMOTE tersebut adalah yang paling efektif dalam menghasilkan data sintetis yang dapat membantu model klasifikasi mencapai keseimbangan terbaik antara *precision* dan *recall*. Oleh karena itu, konfigurasi ini dipilih untuk model final SMOTE guna menghasilkan dataset latih yang seimbang sebagai pembanding terhadap VAE. Dengan ditetapkannya kombinasi hiperparameter optimal tersebut, tahapan selanjutnya yaitu pembangkitan data sintetis dan penyeimbangan *dataset*.

Pada tahap ini, model SMOTE final dengan konfigurasi *k_neighbors*=20 diterapkan secara langsung pada set pelatihan yang berisi 62.576 sampel kelas minoritas (*'same_author'*). Berdasarkan strategi *auto*, algoritma SMOTE bekerja secara iteratif untuk membangkitkan sampel baru melalui interpolasi linear. Sebanyak 9.944.970 sampel sintetis berhasil diciptakan untuk menyeimbangkan jumlah kelas minoritas agar setara dengan kelas mayoritas. Setelah proses penggabungan, distribusi kelas pada set pelatihan baru menjadi seimbang secara sempurna, seperti yang ditunjukkan pada Tabel 4.5.

Tabel 4.5 Distribusi Kelas Setelah Penyeimbangan Data Dengan SMOTE

Kelas	Jumlah Sampel	Persentase
Mayoritas ('diff_author' / 0)	10.007.546	50.00%
Minoritas ('same_author' / 1)	10.007.546	50.00%
Total	20.015.092	100%

Keberhasilan penyeimbangan secara kuantitatif tersebut divalidasi secara visual melalui diagram batang di bawah Gambar 4.6.

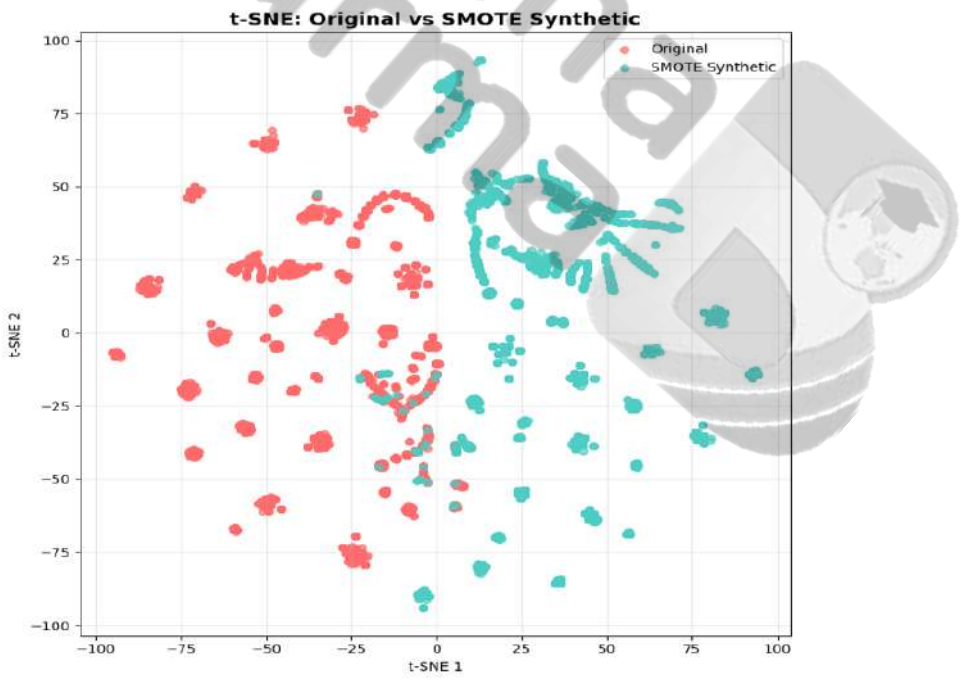


Gambar 4.6 Distribusi Kelas Setelah Penyeimbangan Data Dengan SMOTE

Diagram tersebut secara jelas mengonfirmasi bahwa jumlah sampel untuk kelas *'same_author'* telah berhasil ditingkatkan hingga setara dengan jumlah kelas

'diff_author'. Transformasi ini menghasilkan *dataset* latih yang seimbang, memastikan bahwa model klasifikasi yang dilatih di atasnya tidak memiliki bias terhadap kelas mayoritas dan mampu mempelajari pola dari kedua kelas secara adil.

Selain verifikasi kuantitatif melalui Tabel 4.5, validasi kualitas data sintetis juga dilakukan secara visual. Teknik reduksi dimensi *t-Distributed Stochastic Neighbor Embedding* (t-SNE) diterapkan untuk memproyeksikan data fitur berdimensi tinggi ke dalam ruang dua dimensi. Langkah ini bertujuan untuk memverifikasi apakah data sintetis yang dibangkitkan oleh SMOTE memiliki karakteristik distribusi yang selaras dengan data asli, namun tetap memberikan variasi interpolasi yang cukup untuk memperkaya proses pembelajaran model.



Gambar 4.7 Visualisasi t-Sne SMOTE

Hasil visualisasi t-SNE ditampilkan pada Gambar 4.7, di mana titik berwarna merah merepresentasikan sampel data asli (*Original*) dan titik berwarna hijau toska (*teal*) merepresentasikan sampel sintetis (*SMOTE Synthetic*).

Berdasarkan visualisasi tersebut, terlihat bahwa algoritma SMOTE berhasil membangkitkan sampel-sampel baru yang mengisi ruang fitur secara signifikan. Pola sebaran data sintetis membentuk struktur klaster yang memperluas cakupan data asli dan mengisi celah-celah di ruang fitur yang sebelumnya kosong. Hal ini

mengindikasikan bahwa SMOTE efektif dalam memperkaya variasi data latih kelas minoritas melalui interpolasi linear, yang krusial untuk mencegah model mengalami *overfitting* serta membantu pembentukan batas keputusan (*decision boundary*) yang lebih *robust*.

4.1.4. Klasifikasi Pada Data Tidak Seimbang

Skenario pengujian pertama berfungsi sebagai tolok ukur (*baseline*) untuk mengukur kinerja model *Deep Neural Network* (DNN) ketika dilatih dan dievaluasi secara langsung menggunakan himpunan data uji asli yang tidak seimbang. Skenario ini esensial untuk mendemonstrasikan dampak signifikan dari ketidakseimbangan kelas terhadap performa model.

Hasil evaluasi pada data uji, yang dirangkum dalam Tabel 4.6, menunjukkan nilai Akurasi sebesar 0.9938. Dalam konteks data yang sangat tidak seimbang, metrik ini dapat sangat menyesatkan. Fenomena ini dikenal sebagai "Paradoks Akurasi". Akurasi yang tinggi ini dicapai bukan karena model memiliki kemampuan prediktif yang baik, melainkan karena model telah mengalami bias dan cenderung hanya memprediksi kelas mayoritas (*'diff_author'*) untuk seluruh sampel.

Tabel 4.6 *Classification Report* DNN Data tidak seimbang

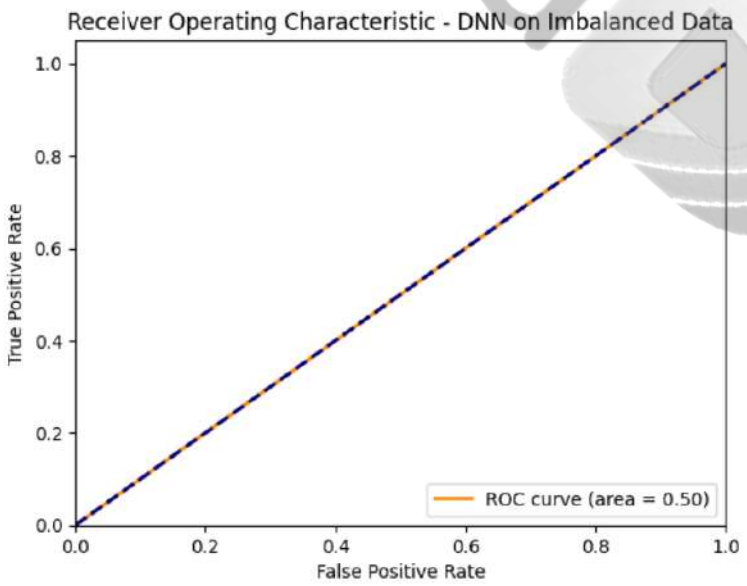
Report	Nilai	Kelas	
		0	1
Accuracy	0.9938		
Precision	0.0000	0.99	0.00
Recall	0.0000	1.00	0.00
Specificity	1.0000		
F-1 Score	0.0000	1.00	0.00
Data		2501811	15720

Analisis mendalam pada Tabel 4.6 mengkonfirmasi temuan ini secara kuantitatif. Untuk Kelas 1 (*'same_author'*), yang merupakan kelas minoritas, model ini mencatat nilai 0.00 untuk *Precision*, *Recall*, dan *F1-Score*. Hal ini

mengindikasikan kegagalan total, di mana model tidak mampu mengidentifikasi satu pun sampel kelas minoritas dengan benar (nilai *True Positive* = 0).

Sebaliknya, untuk Kelas 0 (*'diff_author'*), model mencapai *Recall* 1.00 dan *F1-Score* 1.00. Nilai Specificity agregat yang juga sempurna (1.0000) disebabkan oleh ketiadaan *False Positive* (FP = 0), karena model tidak pernah memprediksi Kelas 1. Secara kolektif, hasil ini membuktikan bahwa meskipun akurasinya tinggi, model *baseline* ini sama sekali tidak memiliki nilai praktis untuk tujuan disambiguasi nama penulis. Kegagalan ini menjadi justifikasi fundamental mengenai perlunya penerapan teknik penyeimbangan data pada skenario pengujian berikutnya.

Kegagalan model untuk melakukan diskriminasi kelas dikonfirmasi oleh nilai ROC-AUC yang hanya mencapai 0.50 pada Gambar 4.8. Nilai ini, yang direpresentasikan oleh kurva ROC yang identik dengan garis diagonal, mengindikasikan bahwa model tidak memiliki daya pembeda sama sekali.

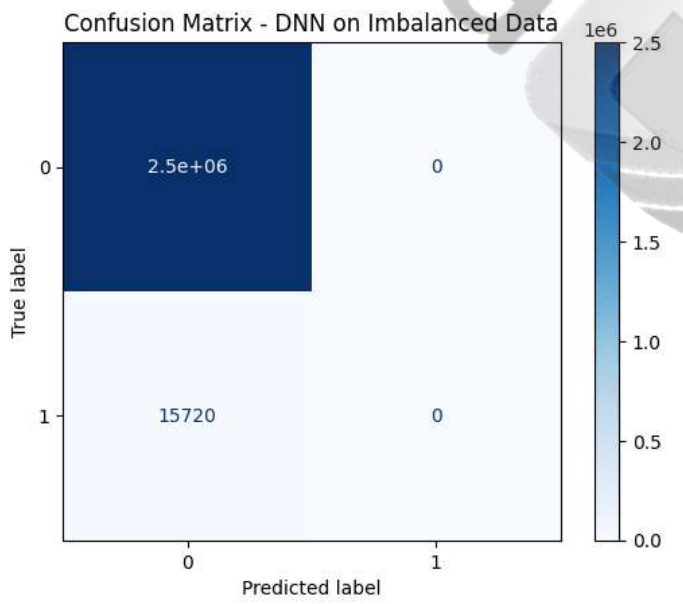


Gambar 4.8 ROC - DNN Data Tidak Seimbang

Analisis yang lebih mendalam pada Gambar 4.9 *Confusion Matrix* memberikan bukti kuantitatif dan visual mengenai kegagalan ini. Dari 2.517.531 total data uji, model membuat prediksi sebagai berikut:

1) *True Positive* (TP): 0. Model gagal total mengidentifikasi satu pun sampel kelas minoritas (1) dengan benar.
2) *False Negative* (FN): 15.720. Seluruh 15.720 sampel kelas minoritas (1) salah diklasifikasikan sebagai kelas mayoritas (0).
3) *True Negative* (TN): 2.501.811. Model berhasil memprediksi hampir semua sampel kelas mayoritas (0) dengan benar.
4) *False Positive* (FP): 0. Model tidak pernah salah memprediksi sampel kelas mayoritas (0) sebagai kelas minoritas (1).

Konsekuensi langsung dari nilai TP = 0 adalah semua metrik evaluasi utama untuk kelas minoritas (1.0) menjadi nol, seperti yang terlihat pada Tabel 4.6: *Precision*: 0.00, Recall: 0.00, dan *F1-Score*: 0.00. Di sisi lain, nilai *Specificity* (TNR) mencapai 1.0000. Metrik ini mengukur proporsi negatif aktual yang diidentifikasi dengan benar (TN / (TN + FP)), dan nilainya yang sempurna hanya mengonfirmasi bias model yang ekstrem terhadap prediksi kelas mayoritas.



Gambar 4.9 *Confusion Matrix* - DNN Data Tidak Seimbang

Secara kolektif, hasil Skenario 1 membuktikan bahwa arsitektur DNN yang diusulkan, tanpa penanganan data, tidak mampu mempelajari fitur apa pun yang relevan untuk membedakan kelas minoritas.

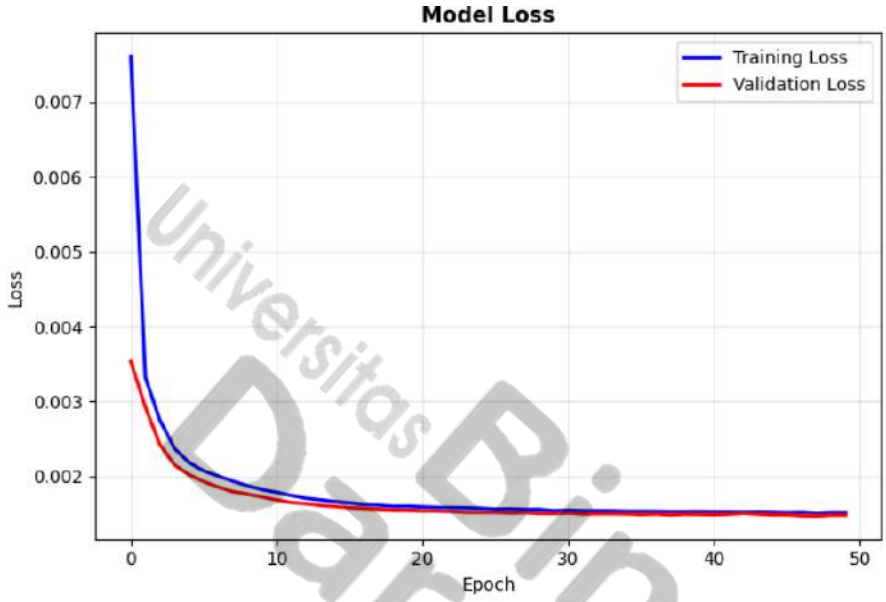
4.1.5. Klasifikasi Pada Data Seimbang Menggunakan VAE

Skenario pengujian kedua mengevaluasi kinerja model DNN yang dilatih menggunakan himpunan data yang telah diseimbangkan oleh *Variational Autoencoder* (VAE). Berbeda secara fundamental dari Skenario 1 yang mengalami kegagalan prediksi total pada kelas minoritas, skenario ini menguji hipotesis bahwa *oversampling* generatif dapat secara efektif mengatasi bias mayoritas. Seperti yang ditunjukkan pada Tabel 4.2, metode VAE yang dikonfigurasi untuk penyeimbangan penuh, berhasil menghasilkan total 20.015.092 sampel dengan rasio penyeimbangan 1:1 yang sempurna. Himpunan data seimbang ini kemudian dibagi dengan rasio 80:20, menghasilkan 16.012.073 data latih dan 4.003.019 data validasi untuk model DNN.

Tabel 4.7 Verifikasi Data Sampling VAE+DNN

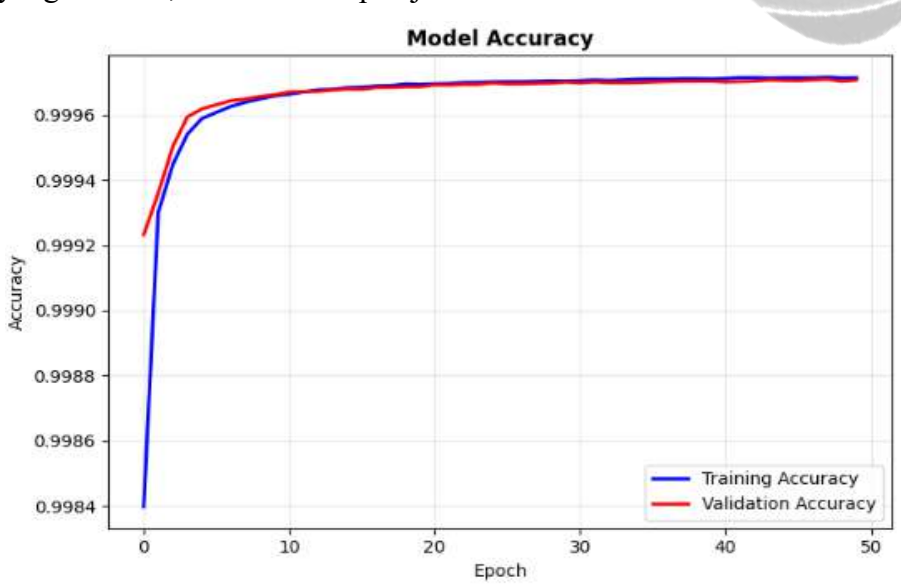
Pembagian Data	Label 0 (diff_author)	Label 1 (same_author)	Total Sampel	Persentase Split
Training	8.006.036	8.006.037	16.012.073	80,0%
Test	2.001.510	2.001.509	4.003.019	20,0%
Total	10.007.546	10.007.546	**20.015.092**	**100,0%**
Persentase Label	*50,0%*	*50,0%*	*100,0%*	

Proses pelatihan model pada data seimbang VAE menunjukkan dinamika yang sangat sehat, seperti yang divisualisasikan pada kurva *loss* dan *accuracy* yang ditunjukkan pada Gambar 4.10 dan Gambar 4.11. Analisis kurva *Model Loss* pada Gambar 4.10 menunjukkan *Training Loss* maupun *Validation Loss* yang menurun tajam (*konvergen*) pada *epoch-epoch* awal dan kemudian melandai secara stabil pada nilai *loss* yang sangat rendah (sekitar 0.0015). Kunci utamanya adalah kedua kurva tersebut bergerak sangat berdekatan, hampir identik, dan tidak menunjukkan adanya celah (*gap*) yang melebar. Fenomena ini mengindikasikan bahwa model tidak mengalami *overfitting* dan memiliki kemampuan generalisasi yang baik dari data latih ke data validasi.



Gambar 4.10 Grafik Model *Loss* (VAE+DNN)

Hal ini dikonfirmasi oleh kurva Model Accuracy Gambar 4.11, di mana *Training Accuracy* dan *Validation Accuracy* juga bergerak secara sinkron, mencapai konvergensi pada nilai akurasi puncak sekitar 0.9997. Ketiadaan *overfitting* dan pencapaian konvergensi yang cepat ini menunjukkan bahwa data sintetis yang dihasilkan VAE memiliki kualitas yang baik, menangkap representasi fitur yang esensial, dan mudah dipelajari oleh model DNN.

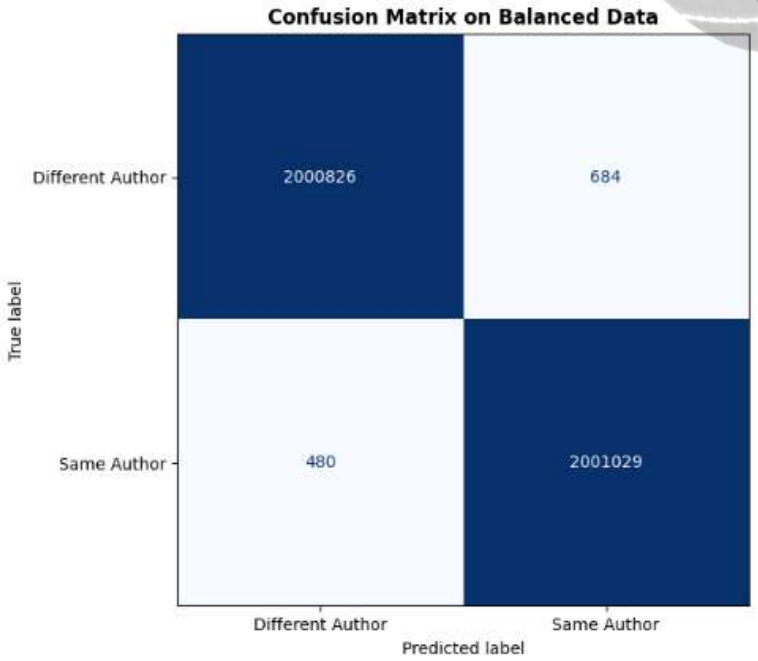


Gambar 4.11 Grafik Model *Accuracy* (VAE+DNN)

Setelah pelatihan selesai, kinerja model VAE-DNN dievaluasi pada 4.003.019 data uji. Hasilnya menunjukkan sebuah transformasi kinerja yang drastis dibandingkan dengan skenario *baseline*. Analisis *Confusion Matrix* pada Gambar 4.12 memberikan rincian kuantitatif atas kinerja ini. Dari 4.003.019 total sampel validasi, model menunjukkan kemampuan diskriminatif yang sangat presisi:

1) *True Positive* (TP): 2.001.029. Model berhasil mengidentifikasi lebih dari 2 juta sampel kelas minoritas dengan benar.
2) *False Negative* (FN): 480. Model hanya melewatkan 480 sampel kelas minoritas.
3) *True Negative* (TN): 2.000.826. Model berhasil mengidentifikasi lebih dari 2 juta sampel kelas mayoritas dengan benar.
4) *False Positive* (FP): 684. Model hanya salah mengklasifikasikan 684 sampel kelas mayoritas.

Jumlah kesalahan total (FN + FP) yang hanya 1.164 dari 4 juta sampel menunjukkan tingkat kesalahan hanya sebesar ≈ 0.029%. Hal ini menunjukkan bahwa model tidak hanya sangat sensitif terhadap kelas minoritas, tetapi juga tetap sangat spesifik terhadap kelas mayoritas tanpa terjadi *false alarm*.



Gambar 4. 12 *Confusion Matrix* (VAE+DNN)

Kinerja impresif yang teridentifikasi dalam *confusion matrix* ini dirangkum secara formal dalam Tabel 4.8. Berbeda dengan skenario *baseline* yang gagal total pada kelas minoritas, model VAE-DNN berhasil mencapai performa yang nyaris sempurna pada kedua kelas. Metrik untuk Kelas 1 (*'same_author'*) melonjak dari 0.00 menjadi *Precision*: 1.00, *Recall*: 1.00, dan *F1-Score*: 1.00. Pencapaian nilai 1.00 pada metrik-metrik krusial ini merupakan bukti kuantitatif bahwa model telah sepenuhnya mempelajari cara membedakan kelas minoritas (yang kini telah diperbanyak oleh VAE) dari kelas mayoritas. Nilai agregat model juga sangat tinggi, dengan F1-Score 0.9997 dan Akurasi 0.9997.

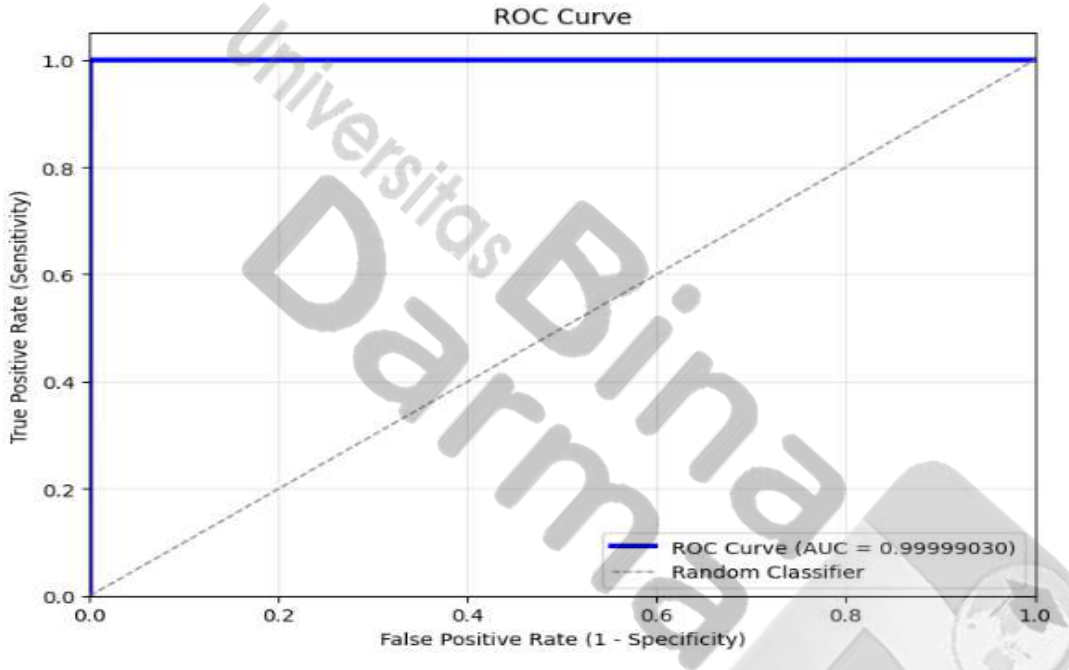
Tabel 4.8 *Classification Report* Data Seimbang VAE dan DNN

Report	Nilai	Kelas	
		0	1
Accuracy	0.9997		
Precision	0.9997	1.00	1.00
Recall	0.9998	1.00	1.00
Specificity	0.9997		
F-1 Score	0.9997	1.00	1.00
Data		2001510	2001509

Kemampuan diskriminatif model ini juga ditegaskan oleh Kurva ROC Gambar 4.13. Kurva ROC menunjukkan model yang ideal, dengan garis kurva yang langsung mencapai sudut kiri atas (sensitivitas 1.0, 1-spesifisitas 0.0). Artinya, terdapat ambang batas (*threshold*) di mana model dapat memisahkan kedua kelas dengan sempurna, memperoleh 100% *True Positive Rate* (Sensitivitas) tanpa mengorbankan *False Positive Rate* (1 - Spesifisitas). Hal ini menghasilkan nilai *Area Under the Curve* (AUC) sebesar 0.9999, yang mengindikasikan kemampuan pemisahan kelas yang sempurna pada data validasi.

Secara keseluruhan, Skenario 2 ini membuktikan bahwa teknik *oversampling* menggunakan *Variational Autoencoder* (VAE) tidak hanya berhasil mengatasi masalah bias mayoritas secara komprehensif, tetapi juga mampu menghasilkan data

sintetis berkualitas tinggi yang secara esensial memungkinkan model DNN untuk mengenali pola-pola yang relevan dari kelas minoritas. Hasilnya adalah kinerja prediktif yang sangat tinggi, stabil, dan seimbang pada kedua kelas.



Gambar 4.13 ROC Curve (VAE+DNN)

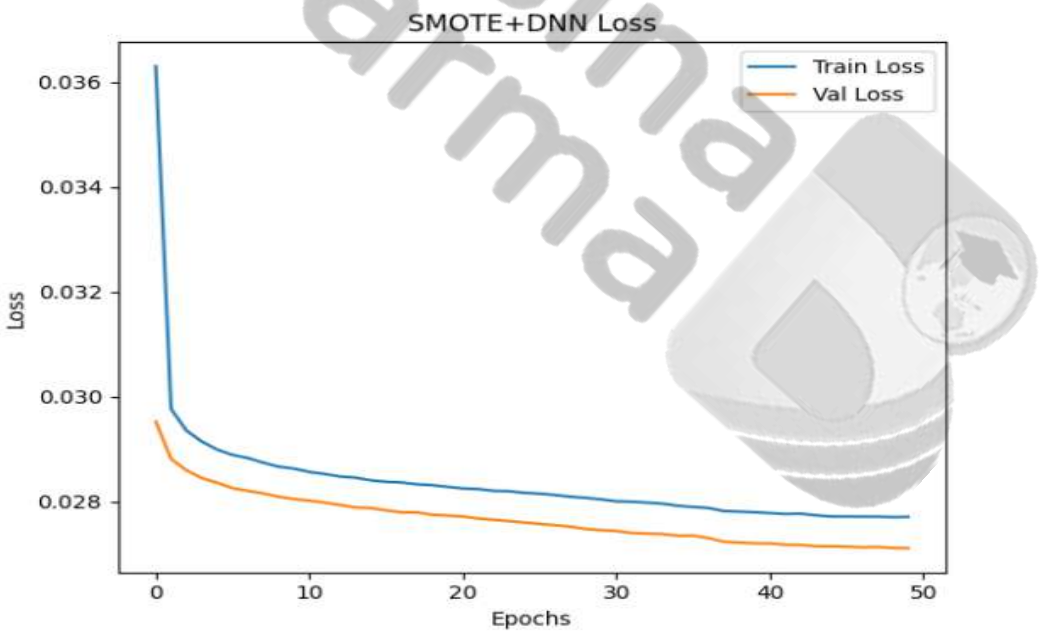
4.1.6. Klasifikasi Pada Data Seimbang Menggunakan SMOTE

Skenario pengujian ketiga dirancang sebagai metode pembanding (*benchmark*) untuk Skenario 2 (VAE). Skenario ini mengevaluasi model DNN yang dilatih menggunakan himpunan data yang telah diseimbangkan oleh metode *Synthetic Minority Over-sampling Technique* (SMOTE).

Tabel 4.9 Verifikasi Data Sampling SMOTE+DNN

Pembagian Data	Label 0 (*diff_author*)	Label 1 (*same_author*)	Total Sampel	Persentase Split
Training	8.006.036	8.006.037	16.012.073	80,0%
Test	2.001.510	2.001.509	4.003.019	20,0%
Total	10.007.546	10.007.546	**20.015.092**	**100,0%**
Persentase Label	*50,0%*	*50,0%*	*100,0%*	

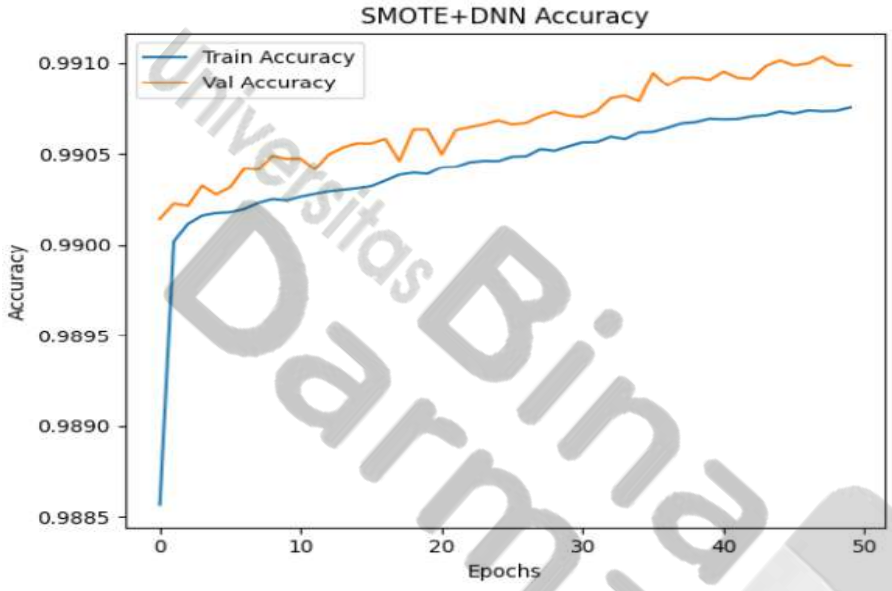
Analisis dinamika pelatihan untuk Skenario 3 ditampilkan pada kurva *loss* dan *accuracy*. Peninjauan pertama difokuskan pada kurva SMOTE+DNN *Loss* pada Gambar 4.14, yang memvisualisasikan progres konvergensi model dari *epoch* ke *epoch*. Secara umum, kedua kurva—*Train Loss* (biru) dan *Val Loss* (oranye) — menunjukkan pola penurunan yang stabil, mengindikasikan bahwa proses pelatihan berjalan dengan baik dan model berhasil mempelajari data. Namun, teramati sebuah perilaku yang menarik dan berbeda dari ekspektasi umum: kurva *Val Loss* (oranye) secara konsisten berada pada posisi yang sedikit lebih rendah daripada kurva *Train Loss* (biru) di sepanjang proses pelatihan.



Gambar 4.14 Grafik Model Loss (SMOTE+DNN)

Hal serupa juga teramati pada kurva SMOTE+DNN *Accuracy* pada Gambar 4.15 di mana *Val Accuracy* (oranye) secara konsisten berada sedikit lebih tinggi daripada *Train Accuracy* (biru). Fenomena yang tampak kontraintuitif pada kedua grafik ini dapat dijelaskan oleh efek regularisasi *Dropout*. *Dropout* hanya aktif selama fase pelatihan (menonaktifkan *neuron* secara acak untuk "mempersulit" model), sehingga *Train Loss* sedikit lebih tinggi dan *Train Accuracy* sedikit lebih rendah. Namun, selama fase validasi, *Dropout* dinonaktifkan dan model menggunakan kapasitas penuh jaringannya, sehingga menghasilkan *Val Loss* yang sedikit lebih baik dan *Val Accuracy* yang sedikit lebih tinggi. Perilaku ini

mengindikasikan bahwa mekanisme regularisasi bekerja sangat efektif dalam mencegah *overfitting* pada data sintetis SMOTE.



Gambar 4.15 Grafik Model *Accuracy* (SMOTE+DNN)

Setelah proses pelatihan selesai, model dievaluasi pada 4.003.019 data validasi. Hasil metrik kinerja dapat dilihat pada Tabel 4.10.

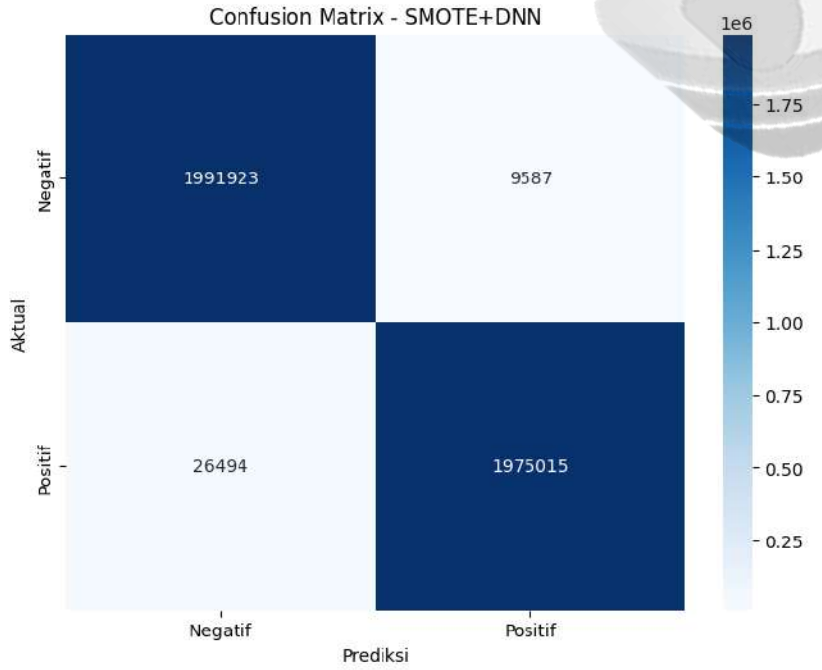
Tabel 4.10 *Classification Report* Data Seimbang SMOTE dan DNN

Report	Nilai	Kelas	
		0	1
Accuracy	0.9910		
Precision	0.9952	0.99	1.00
Recall	0.9868	1.00	0.99
Specificity	0,9952		
F-1 Score	0.9992	0.99	0.99
Data		2001510	2001509

Model ini menunjukkan performa yang sangat tinggi, membuktikan bahwa SMOTE juga berhasil mengatasi masalah bias mayoritas yang terlihat pada skenario *baseline*. Model ini mencapai Akurasi 0.9910, ROC-AUC 0.9992, dan *F1-Score* 0.9909. Meskipun Akurasi (0.9910) secara numerik lebih rendah dari

baseline (0.9938), ini adalah metrik akurasi yang valid dan representatif, karena didukung oleh *F1-Score* yang tinggi, tidak seperti *baseline* yang *F1-Score*-nya 0.00. Analisis yang lebih mendalam pada *Confusion Matrix* pada Gambar 4.16 memberikan rincian kuantitatif atas kinerja ini. Dari 4.003.019 total sampel validasi, model SMOTE+DNN menghasilkan total 36.081 kesalahan prediksi (26.494 FN + 9.587 FP), yang jauh lebih tinggi dibandingkan total kesalahan pada skenario VAE. Rinciannya adalah sebagai berikut:

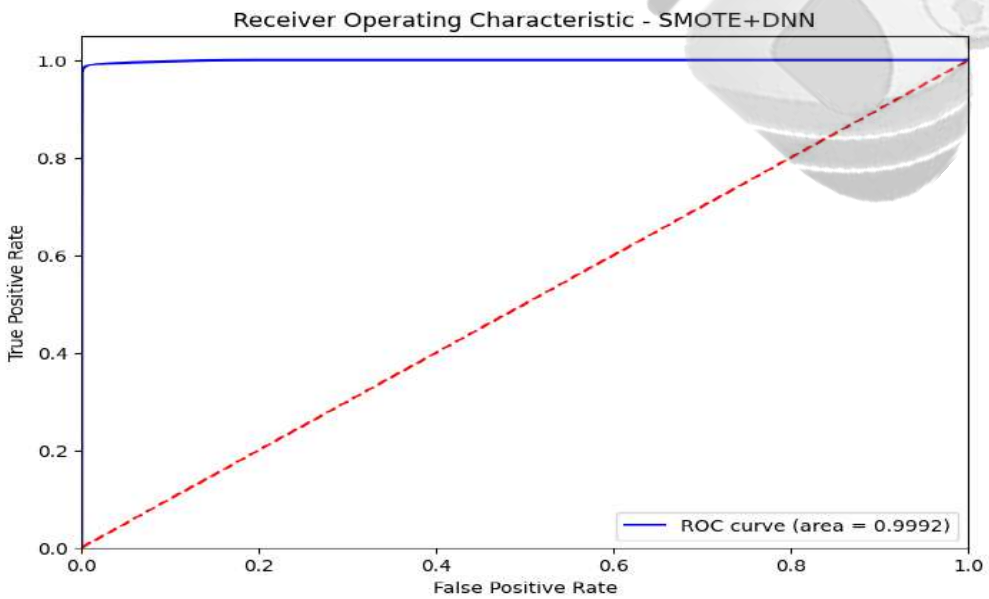
1) *True Positive* (TP): 1.975.015. Model berhasil mengidentifikasi 1,97 juta sampel kelas positif dengan benar.
2) *False Negative* (FN): 26.494. Model melewatkan 26.494 sampel kelas positif.
3) *True Negative* (TN): 1.991.923. Model berhasil mengidentifikasi 1,99 juta sampel kelas negatif dengan benar.
4) *False Positive* (FP): 9.587. Model hanya salah mengklasifikasikan 9.587 sampel kelas negatif.



Gambar 4.16 *Confusion Matrix* (SMOTE+DNN)

Dapat diamati bahwa jumlah *False Negative* (26.494) hampir tiga kali lipat lebih tinggi daripada jumlah *False Positive* (9.587). Ketidakseimbangan dalam tipe kesalahan ini menunjukkan bahwa model SMOTE+DNN memiliki kecenderungan yang lebih besar untuk *under-classify* kelas positif (gagal mendeteksi 'same_author'). Fenomena ini secara langsung menjelaskan mengapa nilai Recall (0.9868), yang sensitif terhadap FN, secara signifikan lebih rendah daripada nilai Precision (0.9952), yang sensitif terhadap FP.

Kemampuan diskriminatif model secara keseluruhan ditegaskan oleh Kurva ROC dapat dilihat pada Gambar 4.17. Kurva ROC menunjukkan performa yang sangat baik, mendekati sudut kiri atas secara rapat. Hal ini menghasilkan nilai *Area Under the Curve* (AUC) sebesar 0.9992. Meskipun angka ini sangat tinggi dan mengindikasikan kemampuan pemisahan kelas yang nyaris sempurna, nilai ini secara kuantitatif sedikit lebih rendah dibandingkan nilai AUC 0.9999 yang dicapai oleh VAE, yang mencerminkan adanya jumlah kesalahan prediksi yang lebih tinggi.



Gambar 4.17 ROC Curve (SMOTE+DNN)

Secara keseluruhan, Skenario 3 menunjukkan bahwa metode SMOTE yang dikombinasikan dengan DNN merupakan solusi yang sangat efektif dalam mengatasi ketidakseimbangan kelas. Model ini berhasil mempelajari pola dari data sintetis yang dihasilkan SMOTE dan mencapai kinerja yang sangat tinggi.

Meskipun demikian, kinerjanya sedikit di bawah Skenario VAE dalam hal metrik agregat (*F1-Score* dan AUC) dan jumlah total kesalahan prediksi. Oleh karena itu, skenario ini berfungsi sebagai tolok ukur (*benchmark*) yang sangat kuat untuk mengonfirmasi keunggulan komparatif dari VAE.

4.2. Pembahasan

Tahap ini menyajikan evaluasi komparatif yang difokuskan pada efektivitas metode penyeimbangan data. Tujuannya adalah untuk mendemonstrasikan secara kuantitatif bagaimana Skenario 2 (VAE) dan Skenario 3 (SMOTE) berhasil mengatasi kegagalan prediktif fundamental yang teridentifikasi pada Skenario 1 (*Baseline*). Untuk melakukan evaluasi yang valid dalam konteks data yang sangat tidak seimbang, metrik Akurasi (*Accuracy*) tidak dapat dijadikan tolok ukur utama. Seperti yang telah ditunjukkan pada Tabel 4.11, nilai Akurasi 0.9938 pada Skenario 1 bersifat menyesatkan dan menyembunyikan fakta kegagalan total model, yang terefleksi pada *F1-Score* 0.00. Fenomena ini, yang dikenal sebagai "Paradoks Akurasi", terjadi karena metrik Akurasi didominasi oleh prediksi *True Negative* dari kelas mayoritas yang sangat besar.

Tabel 4.11 Perbandingan Metrik Kinerja Agregat Antar Skenario

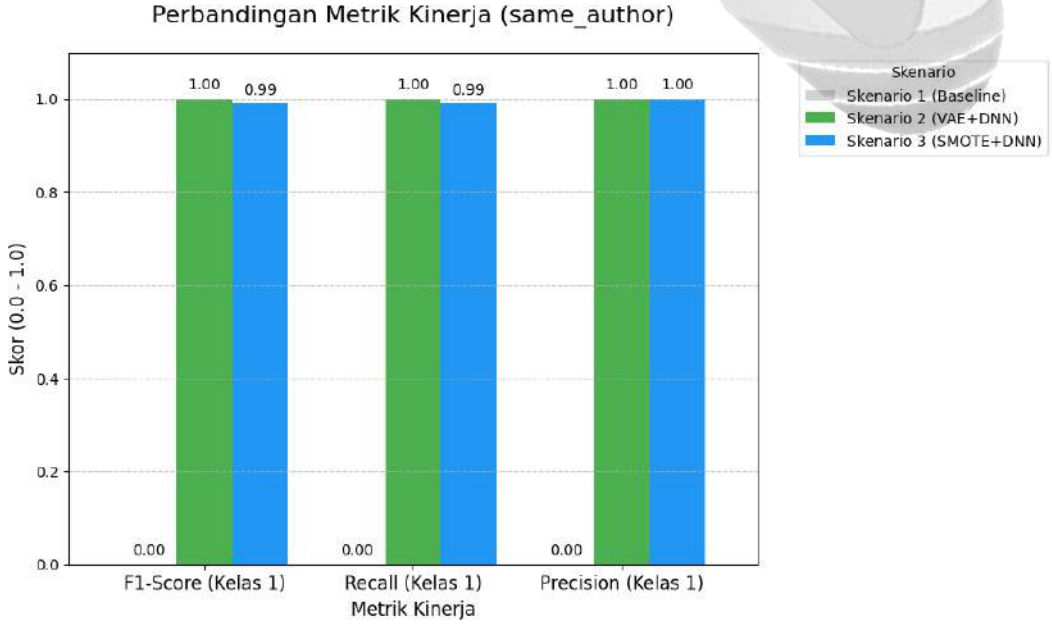
Metrik	Skenario 1 (Baseline)	Skenario 2 (VAE+DNN)	Skenario 3 (SMOTE+DNN)
F1-Score (Kelas 1)	0.00	1.00	0.99
Recall (Kelas 1)	0.00	1.00	0.99
Precision (Kelas 1)	0.00	1.00	1.00
F-1 Score (aggregat)	0.00	0.9997	0.9992
Akurasi (aggregat)	0.9938	0.9997	0.9910

Oleh karena itu, analisis komparatif ini difokuskan pada metrik yang jauh lebih informatif dan sensitif terhadap kinerja kelas minoritas, yaitu *Precision*, *Recall*, dan *F1-Score*. *Recall* (Sensitivitas) mengukur kemampuan model untuk "menemukan" semua sampel kelas minoritas yang relevan (meminimalkan *False*

Negative). *Precision* mengukur kemampuan model untuk memastikan bahwa prediksi kelas minoritas yang dibuatnya adalah benar (meminimalkan *False Positive*). *F1-Score* berfungsi sebagai rata-rata harmonik dari *Precision* dan *Recall*, yang memberikan skor tunggal yang seimbang atas kemampuan model pada kelas tersebut. Secara khusus, perbandingan akan menyoroti metrik untuk Kelas 1 (minoritas), karena kemampuan model untuk mengidentifikasi kelas inilah yang menjadi inti permasalahan penelitian. Metrik Agregat juga disertakan untuk memberikan gambaran holistis tentang keseimbangan kinerja model secara keseluruhan. Tabel 4.11 merangkum metrik kinerja utama dari ketiga skenario untuk perbandingan secara langsung.

4.2.1. Dampak Penyeimbangan Data terhadap Kinerja Model

Analisis komparatif yang paling fundamental adalah perbandingan antara Skenario 1 (*Baseline*) dengan Skenario 2 (VAE+DNN) dan Skenario 3 (SMOTE+DNN). Gambar 4.18 dan Gambar 4.19 menyajikan visualisasi data dari Tabel 4.11 untuk mengilustrasikan dampak penyeimbangan data secara jelas.

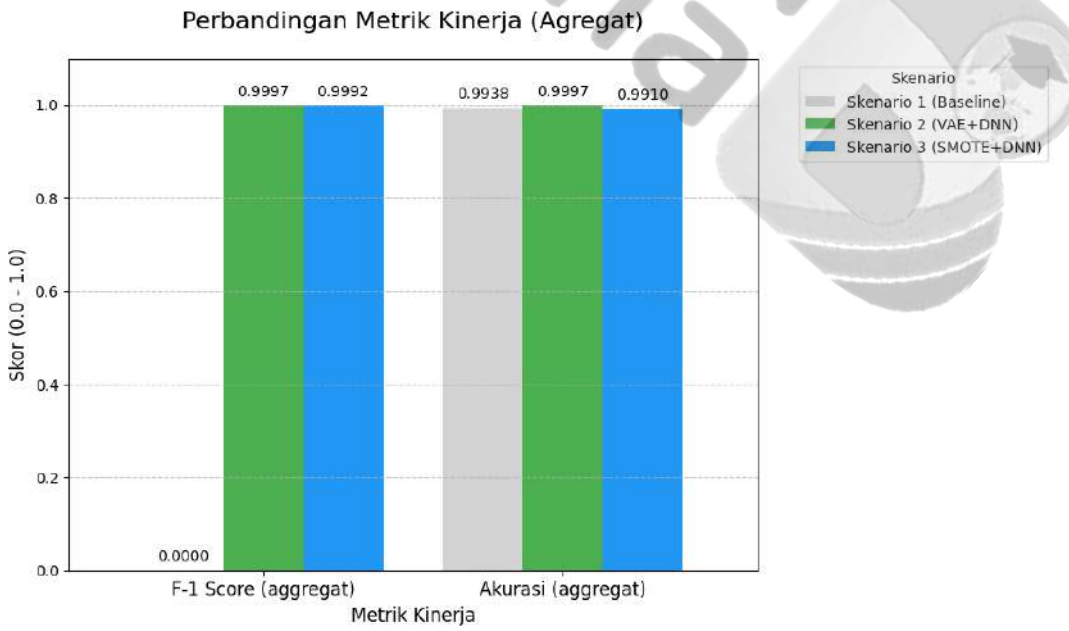


Gambar 4.18 Perbandingan Metrik Kinerja *(same_author)*

Gambar 4.18 secara visual mengonfirmasi kegagalan prediktif fundamental yang terjadi pada Skenario 1. Model *baseline* mencatat skor 0.00 pada *F1-Score*,

Recall, dan *Precision* untuk kelas minoritas (*'same_author'*). Secara praktis, ini mengindikasikan bahwa model tidak hanya berkinerja buruk, tetapi secara harfiah tidak mampu mengidentifikasi *satu pun* sampel *'same_author'* dengan benar. Model tersebut telah menyerah mempelajari pola dan hanya mengandalkan bias statistik, yaitu dengan memprediksi semua sampel sebagai kelas mayoritas (*'diff_author'*), sehingga menjadikannya tidak berguna untuk tujuan praktis disambiguasi.

Secara kontras, Skenario 2 (VAE) dan Skenario 3 (SMOTE) menunjukkan transformasi kinerja. Skor yang melonjak mendekati 1.00 (sempurna) membuktikan bahwa model yang dilatih pada data seimbang kini memiliki sensitivitas (*Recall*) dan presisi (*Precision*) yang sangat tinggi. Ini adalah bukti visual paling kuat bahwa kedua teknik *oversampling* berhasil memaksa model DNN untuk mempelajari pola-pola fitur yang relevan yang membedakan *'same_author'*, alih-alih hanya mempelajari bias distribusi data yang tidak seimbang.



Gambar 4.19 Perbandingan Metrik Kinerja Keseluruhan

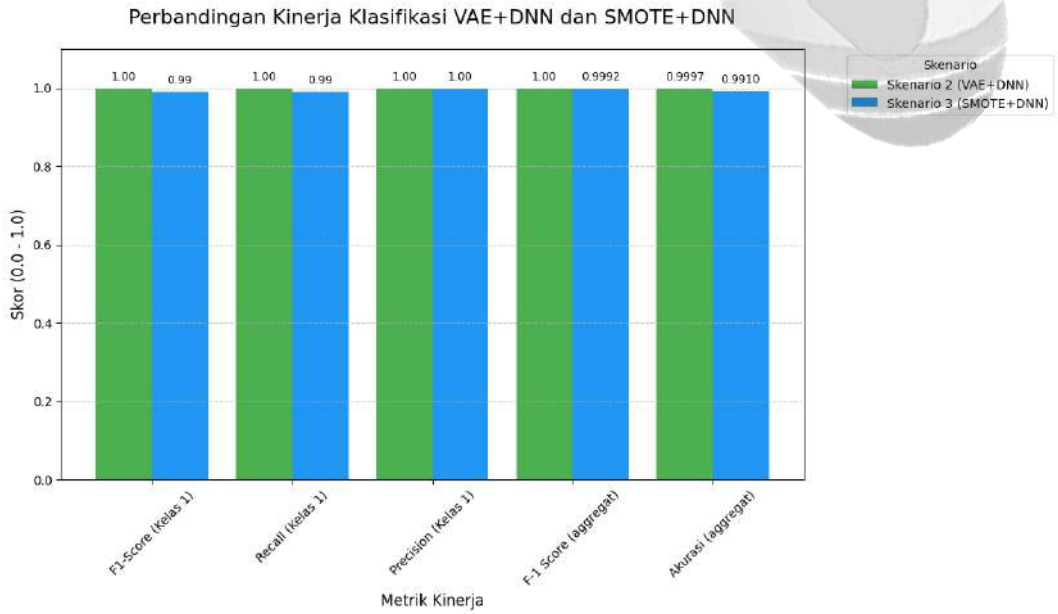
Gambar 4.19 memperkuat temuan ini pada level agregat dan sekaligus menyoroti "Paradoks Akurasi". Grafik ini menunjukkan bahwa meskipun Skenario 1 memiliki Akurasi (0.9938) yang tinggi—bahkan secara numerik lebih tinggi dari Skenario 3 (0.9910)—nilai *F1-Score* Agregatnya adalah 0.0000. Ini adalah artefak statistik yang menyesatkan. Sebaliknya, *F1-Score* Agregat yang valid dan sangat

tinggi pada Skenario 2 (0.9997) dan Skenario 3 (0.9992) membuktikan bahwa kedua model yang diseimbangkan secara fundamental jauh lebih superior. Visualisasi ini menegaskan bahwa Akurasi bukanlah metrik yang dapat diandalkan untuk evaluasi pada kasus ini, dan *F1-Score* adalah indikator kinerja yang sebenarnya.

Secara kolektif, analisis pada Tabel 4.11 dan kedua gambar tersebut membuktikan secara konklusif bahwa penyeimbangan data bukanlah sekadar langkah optimisasi, melainkan sebuah langkah krusial dan fundamental untuk mengatasi masalah ketidakseimbangan data. Tanpa intervensi ini, model klasifikasi gagal total; dengan intervensi ini, model menjadi sangat efektif dan andal dalam mengklasifikasikan kelas minoritas.

4.2.2. Perbandingan Kinerja VAE dan SMOTE pada Data Validasi

Setelah keberhasilan kedua metode penyeimbangan data terkonfirmasi, analisis selanjutnya difokuskan pada perbandingan efektivitas relatif antara VAE dan SMOTE berdasarkan kinerja data validasi pada Tabel 4.11.



Gambar 4.20 Perbandingan Kinerja VAE+DNN dan SMOTE+DNN

Berdasarkan data tersebut, model VAE+DNN (Skenario 2) menunjukkan kinerja yang nyaris sempurna dan sangat seimbang. Pencapaian skor 1.00 untuk

Precision (Kelas 1), *Recall* (Kelas 1), dan *F1-Score* (Kelas 1) mengindikasikan bahwa data sintetis yang dihasilkan oleh VAE memiliki kualitas yang sangat tinggi. Data ini tidak hanya berhasil menyeimbangkan jumlah sampel, tetapi juga tampaknya berhasil menangkap representasi fitur yang esensial dan beragam dari kelas minoritas dengan sangat akurat. Hasilnya adalah model DNN yang mampu membangun batas keputusan (*decision boundary*) yang sangat jelas dan efektif, yang berpuncak pada *F1-Score* agregat 0.9997.

Di sisi lain, model SMOTE+DNN (Skenario 3) menunjukkan kinerja yang juga sangat tinggi, namun dengan perbedaan signifikan pada aspek sensitivitas. Meskipun *Precision* (Kelas 1) mencapai 1.00 (menunjukkan model jarang salah memprediksi positif), nilai *Recall* (Kelas 1) adalah 0.99. Hal ini mengindikasikan bahwa selama validasi, model SMOTE sedikit kurang sensitif dan melewatkan lebih banyak kasus positif (*False Negatives*), yang berimplikasi pada *F1-Score* agregat 0.9992 dan *Recall* agregat 0.9868.

Temuan ini dapat diinterpretasikan sebagai implikasi dari perbedaan fundamental kedua metode. SMOTE adalah metode yang berbasis interpolasi linear lokal; ia menciptakan sampel baru di antara titik-titik minoritas yang ada. Ada kemungkinan bahwa proses ini menciptakan beberapa sampel sintetis yang ambigu atau tumpang tindih (*overlapping*) dengan klaster kelas mayoritas, terutama pada batas keputusan yang kompleks. Hal ini dapat membuat batas keputusan yang dipelajari DNN menjadi sedikit 'kabur' (*fuzzy*), yang mengakibatkan beberapa kasus positif yang sulit diklasifikasikan sebagai negatif (FN).

Sebaliknya, VAE adalah model generatif yang mempelajari distribusi data secara holistis dalam ruang laten. VAE tidak hanya menginterpolasi, tetapi "belajar" untuk menghasilkan sampel-sampel baru yang sepenuhnya orisinal namun tetap representatif terhadap distribusi kelas minoritas. Berdasarkan temuan ini, model VAE+DNN menunjukkan kinerja yang sedikit lebih unggul dan lebih seimbang. Hal ini mengimplikasikan bahwa data sintetis yang dihasilkan VAE lebih beragam dan mampu mendefinisikan batas keputusan kelas minoritas dengan lebih jelas, sehingga menghasilkan sensitivitas (*Recall*) yang sempurna pada data validasi.

4.2.3. Analisis Komparatif Terhadap Penelitian Terdahulu

Setelah Skenario 2 (VAE+DNN) teridentifikasi sebagai model paling unggul dalam penelitian ini (dengan *F1-Score* agregat 0.9997), analisis pembahasan ditutup dengan memosisikan temuan ini dalam konteks penelitian *author name disambiguation* (AND) yang lebih luas.

Sebagaimana telah diuraikan dalam Tinjauan Pustaka, lanskap penelitian AND terbagi ke dalam beberapa pendekatan. Penelitian ini, dengan kerangka kerja VAE+DNN, secara spesifik berkontribusi pada sub-bidang klasifikasi *supervised* (berbasis *pairwise matching*). Oleh karena itu, perbandingan yang paling relevan adalah dengan penelitian lain yang menggunakan pendekatan serupa, yang dirangkum pada Tabel 4.12.

Tabel 4.12 Perbandingan Komparatif dengan Penelitian Terdahulu

No	Peneliti	Metode	Implementasi	Hasil
1	(Yamani, Nurmaini, & Rini, 2020)	Isolation Forest	Pairwise - Similarity	99.5%
2	(Yamani, Nurmaini, Firdaus, dkk., 2020)	Deep Neural Network	Pairwise - Similarity	98.0%
3	(Firdaus dkk., 2022)	Capsule Network	Pairwise	99.80%
4	(Boukhers & Asundi, 2023)	Neural Network	Pairwise - Similarity	96,1%
5		VAE + DNN	Pairwise - Similarity	99.9%

Kinerja model VAE+DNN dalam penelitian ini, yang mencapai *F1-Score* 0.9997 (99,97%) pada data validasi, tidak hanya melampaui metode *benchmark* SMOTE (0.9992), tetapi juga menunjukkan hasil yang sangat kompetitif bila disandingkan dengan penelitian terdahulu. Secara spesifik, kinerja *F1-Score*

99,97% yang dicapai oleh model VAE+DNN dalam penelitian ini mengungguli hasil yang dilaporkan oleh penelitian sebelumnya.

Meskipun demikian, perlu ditegaskan bahwa perbandingan angka absolut ini memiliki keterbatasan metodologis yang signifikan. Interpretasi hasil harus dilakukan secara hati-hati, mengingat setiap penelitian memiliki variabel perancu fundamental. Variabel pertama adalah perbedaan *dataset*, di mana setiap *dataset* (DBLP, ArnetMiner, KISTI) memiliki karakteristik, volume data, dan rasio ketidakseimbangan yang unik. Variabel kedua adalah perbedaan metrik evaluasi. Beberapa penelitian terdahulu melaporkan Akurasi sebagai metrik utama. Seperti yang telah dibuktikan oleh penelitian ini (melalui Skenario 1), Akurasi merupakan metrik yang tidak dapat diandalkan untuk masalah data tidak seimbang. Oleh karena itu, kinerja *F1-Score* 99,97% dalam penelitian ini memiliki bobot pembuktian yang lebih tinggi dibandingkan Akurasi 99,8% yang dilaporkan pada penelitian lain.

Oleh karena itu, kontribusi utama dari penelitian ini bukanlah semata-mata pada pencapaian skor tertinggi pada data validasi, melainkan pada pembuktian efektivitas kerangka kerja VAE+DNN sebagai solusi spesifik untuk masalah data tidak seimbang. Penelitian ini secara jelas mendemonstrasikan bahwa model *baseline* (Skenario 1) yang gagal total (*F1-Score* 0.00) dapat secara dramatis diperbaiki hingga mencapai kinerja *state-of-the-art* (*F1-Score* 0.9997) hanya dengan menerapkan *oversampling* generatif VAE. Temuan ini memberikan kontribusi signifikan yang memvalidasi VAE sebagai pendekatan yang sangat kuat dan efektif untuk mengatasi tantangan *imbalanced data* yang inheren dalam tugas *author name disambiguation* berbasis *pairwise matching*.

BAB V
KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan hasil penelitian dan analisis yang telah dilakukan untuk mengevaluasi efektivitas penerapan algoritma *Variational Autoencoder* (VAE) dan *Deep Neural Network* (DNN) dalam mengatasi ketidakseimbangan data pada sistem klasifikasi *author name disambiguation*, dapat ditarik beberapa kesimpulan yang menjawab rumusan masalah sebagai berikut:

1. Pendekatan penanganan data dengan melakukan penyeimbangan data terbukti menjadi tahapan yang esensial, karena model klasifikasi mengalami bias yang ekstrem terhadap kelas mayoritas dan gagal mengidentifikasi penulis yang ambigu (kelas minoritas) secara akurat.
2. Penerapan algoritma *Variational Autoencoder* (VAE) terbukti efektif dalam menghasilkan data sintetis. Algoritma ini sukses menyeimbangkan proporsi himpunan data latih dengan cara mempelajari distribusi fitur kelas minoritas dan memperkaya variasi sampel tanpa merusak karakteristik aslinya.
3. Integrasi antara metode *oversampling* generatif VAE dan model klasifikasi *Deep Neural Network* (DNN) telah berhasil diwujudkan. Penggabungan kedua arsitektur ini secara signifikan mampu meningkatkan performa klasifikasi dalam menyelesaikan masalah ambiguitas nama penulis dengan membentuk batas keputusan yang lebih tegas.
4. Pengukuran kinerja membuktikan bahwa model klasifikasi hibrida VAE-DNN mencapai hasil yang sangat optimal pada seluruh parameter metrik (*accuracy, precision, recall, specificity*, dan *f-measure*). Model ini konsisten mendemonstrasikan tingkat sensitivitas tertinggi jika dibandingkan dengan model tanpa penyeimbangan data.

5.2. Saran

Meskipun penelitian ini telah berhasil mencapai tujuannya, terdapat beberapa area yang dapat dikembangkan lebih lanjut untuk penelitian di masa mendatang. Berikut adalah beberapa saran yang diajukan:

1. Penelitian selanjutnya disarankan untuk mengeksplorasi ekstraksi fitur yang lebih komprehensif. Penggunaan model bahasa seperti BERT dapat diterapkan untuk menangkap makna semantik pada judul dan abstrak publikasi. Selain itu, analisis jaringan kolaborasi penulis (*co-authorship network*) dapat diimplementasikan untuk memperkuat identifikasi entitas akademik secara lebih presisi.
2. Pengembangan model klasifikasi disarankan untuk beralih dari *Deep Neural Network* (DNN) menuju *Graph Neural Networks* (GNN) atau *Graph Convolutional Networks* (GCN). Model berbasis graf secara teoretis lebih selaras dengan sifat relasional data bibliografi, sehingga model tersebut mampu mengatasi ambiguitas nama dengan mengeksploitasi struktur keterhubungan antarpenulis secara langsung.
3. Penelitian mendatang perlu melakukan studi komparatif antara algoritma VAE dan metode *generative oversampling* tingkat lanjut lainnya, seperti *Generative Adversarial Networks* (GAN) khusus data tabular. Komparasi ini bertujuan untuk mengidentifikasi arsitektur generatif yang paling optimal dan minim derau (*noise*) dalam membangkitkan data bibliografi sintetis.

DAFTAR PUSTAKA

Ai, Q., Wang, P., He, L., Wen, L., Pan, L., & Xu, Z. (2023). Generative Oversampling for Imbalanced Data via Majority-Guided VAE. *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics*, *206*, 3315–3330. https://doi.org/10.48550/arXiv.2302.10910

Altin Karataş, M., & Biberci, M. A. (2023). Statistical Analysis of WEDM Machining Parameters of Ti-6Al-4V Alloy Using Taguchi-Based Grey Relational Analysis and Artificial Neural Network. *Experimental Techniques*, *47*(4), 851–870. https://doi.org/10.1007/S40799-022-00601-5/METRICS

An, J., & Cho, S. (2015). Variational Autoencoder based Anomaly Detection using Reconstruction Probability. *Special Lecture on IE*, *2*, 1–18.

Anjali, M., & Jivani, G. (2011). A Comparative Study of Stemming Algorithms. *Int. J. Comp. Tech.*, *2*(6), 1930–1938.

Boukhers, Z., & Asundi, N. B. (2023). Deep author name disambiguation using DBLP data. *International Journal on Digital Libraries*, 1–11. https://doi.org/10.1007/S00799-023-00361-6/TABLES/5

Chen, Y., Yuan, H., Liu, T., & Ding, N. (2021). Name Disambiguation Based on Graph Convolutional Network. *Scientific Programming*, *2021*. https://doi.org/10.1155/2021/5577692

D'Angelo, C. A., & van Eck, N. J. (2020). Collecting large-scale publication data at the level of individual researchers: a practical proposal for author name disambiguation. *Scientometrics*, *123*(2), 883–907. https://doi.org/10.1007/s11192-020-03410-y

Deng, L., & Yu, D. (2013). Deep learning: Methods and applications. Dalam *Foundations and Trends in Signal Processing* (Vol. 7, Nomor 3–4, hlm. 197–387). Now Publishers Inc. https://doi.org/10.1561/2000000039

Doersch, C. (2021). Tutorial on Variational Autoencoders. *arXiv preprint arXiv:1606.05908*, 1–23. https://doi.org/10.48550/arXiv.1606.05908

Ebrahimi, F., Asemi, A., & Ko, A. (2023). Identifying effective criteria for author matching in bioinformatics. *Informatics in Medicine Unlocked*, *38*. https://doi.org/10.1016/j.imu.2023.101224

Fan, C., & Li, Y. (2021). Chinese Personal Name Disambiguation Based on Clustering. *Wireless Communications and Mobile Computing*, *2021*. https://doi.org/10.1155/2021/3790176

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, *27*(8), 861–874. https://doi.org/10.1016/J.PATREC.2005.10.010

Fernández, A., García, S., Galar, M., Prati, R. C., Krawczyk, B., & Herrera, F. (2018). Learning from Imbalanced Data Sets. *Learning from Imbalanced Data Sets*. https://doi.org/10.1007/978-3-319-98074-4

Ferreira, A. A., Gonçalves, M. A., & Laender, A. H. F. F. (2012). A brief survey of automatic methods for author name disambiguation. *SIGMOD Record*, *41*(2), 15–26. https://doi.org/10.1145/2350036.2350040

Ferreira, A. A., Gonçalves, M. A., & Laender, A. H. F. F. (2020). Automatic Disambiguation of Author Names in Bibliographic Repositories. *Synthesis Lectures on Information Concepts, Retrieval, and Services*, *12*(1), 1–146. https://doi.org/10.2200/S01011ED1V01Y202005ICR070

Ferreira, A. A., Veloso, A., Gonçalves, M. A., & Laender, A. H. F. (2010). Effective self-training author name disambiguation in scholarly digital libraries. *Proceedings of the ACM International Conference on Digital Libraries*, 39–48. https://doi.org/10.1145/1816123.1816130

Firdaus, Alqarni, W., Nurmaini, S., Darmawahyuni, A., Sapitri, A. I., Rachmatullah, M. N., & Lestari, S. D. (2022). Author Classification on Bibliographic Data Using Capsule Networks Architecture. *International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, *2022-October*, 101–105. https://doi.org/10.23919/EECSI56542.2022.9946586

Gomide, J., Kling, H., & Figueiredo, D. (2017). Name usage pattern in the synonym ambiguity problem in bibliographic data. *Scientometrics*, *112*(2), 747–766. https://doi.org/10.1007/s11192-017-2410-2

Goodfellow, I. (2014). Front Matter. *Linear Algebra*, i–ii. https://doi.org/10.1016/b978-0-12-391420-0.09987-x

Haddi, E., Liu, X., & Shi, Y. (2013). The Role of Text Pre-processing in Sentiment Analysis. *Procedia Computer Science*, *17*, 26–32. https://doi.org/10.1016/J.PROCS.2013.05.005

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning*. *Springer Series in Statistics*. https://doi.org/10.1007/978-0-387-84858-7

He, H., & Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, *21*(9), 1263–1284. https://doi.org/10.1109/TKDE.2008.239

Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation*, *18*(7), 1527–1554. https://doi.org/10.1162/NECO.2006.18.7.1527

Hussain, I., & Asghar, S. (2017). A survey of author name disambiguation techniques: 2010–2016. *The Knowledge Engineering Review*, *32*, 1–24. https://doi.org/10.1017/s0269888917000182

Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, *349*(6245), 255–260. https://doi.org/10.1126/SCIENCE.AAA8415

Kim, J. (2018). Evaluating author name disambiguation for digital libraries: a case of DBLP. *Scientometrics*, *116*(3), 1867–1886. https://doi.org/10.1007/s11192-018-2824-5

Kim, J. J., & Kim, J. J. (2018). The impact of imbalanced training data on machine learning for author name disambiguation. *Scientometrics*, *117*(1), 511–526. https://doi.org/10.1007/s11192-018-2865-9

Kim, K., Rohatgi, S., & Lee Giles, C. (2019). Hybrid deep pairwise classification for author name disambiguation. *International Conference on Information and Knowledge Management, Proceedings*, 2369–2372. https://doi.org/10.1145/3357384.3358153

Kingma, D. P., & Welling, M. (2013). Auto-Encoding Variational Bayes. *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*. https://doi.org/10.61603/ceas.v2i1.33

Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/NATURE14539

Lewis, J., Ossowski, S., Hicks, J., Errami, M., & Garner, H. R. (2006). Text similarity: An alternative way to search MEDLINE. *Bioinformatics*, *22*(18), 2298–2304. https://doi.org/10.1093/bioinformatics/btl388

Luque, A., Carrasco, A., Martín, A., & de las Heras, A. (2019). The impact of class imbalance in classification performance metrics based on the binary confusion matrix. *Pattern Recognition*, *91*, 216–231. https://doi.org/10.1016/J.PATCOG.2019.02.023

Ma, C., & Xia, H. (2023). Author Name Disambiguation Based on Heterogeneous Graph. *Journal of Computers*, *34*(4), 41–52. https://doi.org/10.53106/199115992023083404004

Manning, C. D., Raghavan, P., & Hinrich, S. (2009). *Introduction to Information Retrieval*. Cambridge University Press.

Manzoor, A., Asghar, S., & Amjad, T. (2022). Toward a New Paradigm for Author Name Disambiguation. *IEEE Access*, *10*, 76055–76068. https://doi.org/10.1109/ACCESS.2022.3190088

McKay, D., Sanchez, S., & Parker, R. (2010). What's my name again? Sociotechnical considerations for author name management in research

databases. *ACM International Conference Proceeding Series*, 240–247. https://doi.org/10.1145/1952222.1952274

Nasution, D. A., Khotimah, H. H., & Chamidah, N. (2019). Perbandingan Normalisasi Data untuk Klasifikasi Wine Menggunakan Algoritma K-NN. *Computer Engineering, Science and System Journal*, *4*(1), 78. https://doi.org/10.24114/CESS.V4I1.11458

Pinheiro Cinelli, L., Araújo Marins, M., Barros da Silva, E. A., & Lima Netto, S. (2021). Variational Autoencoder. *Variational Methods for Machine Learning with Applications to Deep Networks*, 111–149. https://doi.org/10.1007/978-3-030-70679-1_5

Shalev-Shwartz, S., & Ben-David, S. (2013). Understanding machine learning: From theory to algorithms. *Understanding Machine Learning: From Theory to Algorithms*, *9781107057*, 1–397. https://doi.org/10.1017/CBO9781107298019

Shen, Q., Wu, T., Yang, H., Wu, Y., Qu, H., & Cui, W. (2017). NameClarifier: A Visual Analytics System for Author Name Disambiguation. *IEEE Transactions on Visualization and Computer Graphics*, *23*(1), 141–150. https://doi.org/10.1109/TVCG.2016.2598465

Shoaib, M., Daud, A., & Amjad, T. (2020). Author Name Disambiguation in Bibliographic Databases: A Survey. *Computing Research Repository*, 1–24. https://doi.org/10.48550/arXiv.2004.06391

Sun, Y., Lee, J., Kim, S., Seon, J., Lee, S., Kyeong, C., & Kim, J. (2023). Energy Theft Detection Model Based on VAE-GAN for Imbalanced Dataset. *Energies 2023, Vol. 16, Page 1109*, *16*(3), 1109. https://doi.org/10.3390/EN16031109

Sze, V., Chen, Y. H., Yang, T. J., & Emer, J. S. (2017). Efficient Processing of Deep Neural Networks: A Tutorial and Survey. *Proceedings of the IEEE*, *105*(12), 2295–2329. https://doi.org/10.1109/JPROC.2017.2761740

Tan, P.-Nin., Steinbach, Michael., & Kumar, Vipin. (2018). *Introduction to data mining*. Pearson.

Tran, H. N., Huynh, T., & Do, T. (2014). Author name disambiguation by using deep neural network. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *8397 LNAI*(PART 1), 123–132. https://doi.org/10.1007/978-3-319-05476-6_13

Utyamishev, D., & Partin-Vaisband, I. (2019). Progressive VAE Training on Highly Sparse and Imbalanced Data. *Computing Research Repository (CoRR)*. https://doi.org/10.48550/arXiv.1912.08283

Weiss, A. (2016). Examining Massive Digital Libraries (MDLs) and Their Impact on Reference Services. *Reference Librarian*, *57*(4), 286–306. https://doi.org/10.1080/02763877.2016.1145614

Yamani, Z., Nurmaini, S., Firdaus, Naufal R, M., & Sari, W. K. (2020). *Author Matching Using String Similarities and Deep Neural Networks*. *172*(Siconian 2019), 474–479. https://doi.org/10.2991/aisr.k.200424.073

Yamani, Z., Nurmaini, S., & Rini, D. P. (2020). Author Matching Classification with Anomaly Detection Approach for Bibliomethric Repository Data. *Computer Engineering and Applications Journal*, *9*(2), 79–92. https://doi.org/10.18495/comengapp.v9i2.335

Yudhana, A., Iif Alfiatul Mukaromah, dan, Studi Teknik Elektro, P., Ahmad Dahlan, U., Dahlan, A., & Jl Soepomo, U. (2018). Implementasi Deteksi Plagiarisme Menggunakan Metode N-Gram Dan Jaccard Similarity Terhadap Algoritma Winnowing. *Transmisi: Jurnal Ilmiah Teknik Elektro*, *20*(3), 105–110. https://doi.org/10.14710/TRANSMISI.20.3.105-110

Zhao, Z., Rollins, J., Bai, L., & Rosen, G. (2017). Incremental author name disambiguation for scientific citation data. *Proceedings - 2017 International Conference on Data Science and Advanced Analytics, DSAA 2017*, *2018-Janua*, 175–183. https://doi.org/10.1109/DSAA.2017.17

Zhou, A., Shi, M., & Yuan, R. (2024). An Effective Author Name Disambiguation Framework for Large-Scale Publications. *IEEE Access*. https://doi.org/10.1109/ACCESS.2024.3511037

Zhu, J., Yang, Y., Xie, Q., Wang, L., & Hassan, S. U. (2014). Robust hybrid name disambiguation framework for large databases. *Scientometrics*, *98*(3), 2255–2274. https://doi.org/10.1007/s11192-013-1151-0

Author Name Disambiguation in Digital Libraries: A Hybrid VAE-DNN Approach to Mitigate Class Imbalance in Pairwise Matching

Hamid Rahman
Informatics Department, Postgraduate Program,
Universitas Bina Darma
Palembang, Indonesia
24242002P@student.binadarma.ac.id

Usman Ependi
Informatics Department, Postgraduate Program,
Universitas Bina Darma
Palembang, Indonesia
u.epemdi@binadarma.ac.id

***Abstract*—Author name disambiguation (AND) remain challenge inside of digital library. It crucial method for accurate citation analysis plus literature search. Predominant classification technique for AND represent supervised pair matching; nevertheless, this approach invariable lead for significant class imbalance, hence distorting predictive model inside of favor concerning majority class plus causing prediction failure for minority class. This research propose generative oversampling architecture that integrate deep neural network (DNN) alongside variational autoencoder (VAE) to addresses this critical gap. VAE model acquire latent distribution concerning minority class when utilized upon paired DBLP dataset exhibiting imbalance ratio pertaining to 160:1 (99.38% majority against 0.62% minority). VAE effective generate perfect balanced 1:1 training dataset by way of synthesizing high-precision minority sample. Performance marked enhanced inside of later categorization employing identical DNN architecture. VAE-DNN model achieve accuracy concerning 0.9997, alongside precision, recall, plus F1-score value nearing 1.00 plus area under curve (AUC) pertaining to 0.9999. Result indicate that VAE-based generative oversampling can eliminates majority class bias without introducing noise. Consequent, DNN can recognizes writer alongside near-perfect precision. To ensures broader generality, further validation upon alternative bibliometric dataset are recommended. This methodology establish robust foundation for equitable DNN classification.**

Keywords— Author Name Disambiguation, Variational Autoencoder, Deep Neural Network, Imbalanced Data, Generative Oversampling.

I. INTRODUCTION

Academic literature were former structured throughout hand cataloging. Rapid proliferation concerning digital library plus online publication repository have make this obsolete. Author name disambiguation (AND) are critical plus enduring challenge exacerbated by way of digital shift, which have increase worldwide information accessible while complicating identity management. Ambiguity, frequent stemming out of synonym or homonym, emerge out of difficulty pertaining to distinguishing author alongside analogous name inside of large dataset [1], [2]. This may substantial interferes alongside citation analysis plus results inside of incorrect work attribution.

Two primary technique have general been propose inside of literature to addresses AND problem [3], [4]. Primary method represent author grouping, unsupervised methodology that consolidate published entry believed to is authored by way of same individual. This methodology integrate insight out of reference [5] plus [6]. Recent research have employ modern graph-based technique to clarifies complex structural relationship. Example include Graph Convolutional Network (GCN) [7] plus Graph Autoencoder [8]. Comparison study by way of [9] advance research domain by way of evaluating CvE plus DGA procedure, demonstrating diversity concerning unsupervised strategy utilized to tackles this problem.

Second method are author assignment, which are supervised plus conceptualize AND as categorization problem. This method often utilize pairwise matching technique, inside of which two published item is direct compared plus classified as either "same" or "different" author. Deep learning algorithm have demonstrate significant effectiveness inside of similarity-based plus pairwise comparison framework. Study by way of [6] plus [10] utilizes neural network for categorization task. Research [11] utilize capsule network architecture plus achieves high level concerning accuracy. Additional, [12] evaluate ensemble method known as Isolation Forest, while [5] propose different approach utilizing multi-criteria decision-making instrument, such as Analytical Hierarchy Process (AHP) plus Fuzzy Delphi.

Although supervised pairwise matching technique have emerge as effective solution for AND, comprehensive evaluation reveal significant research gap. This strategy produce substantial class imbalance, representing new plus critical problem. Production concerning all possible publication pairing result inside of significant majority pertaining to "different" author pair plus minimal fraction concerning "same" couple. Study including [11], [13] It primary focuses upon architectural improvement, such as Char2Vec embedding plus Capsule Network, while marked overlooking this inherent imbalance. Without specific mitigation strategy, classification algorithm demonstrates bias toward majority class plus is susceptible for prediction error for minority class [14], [15].

This article present framework utilizing variational autoencoder (VAE) as oversampling method to handles issue concerning imbalanced data inside of pairwise AND classification. VAE were select because concerning its capacity to learns intricate data distribution. This enable generation pertaining to varied, high-caliber synthetic sample

for minority class. As result, balanced training dataset are create without adding overlapping noise [16], [17].

Primary objective concerning this research are to develops two-stage workflow. Paired AND dataset will is initial balanced using variational autoencoder (VAE). Deep neural network (DNN) will then is trained upon balanced dataset for final classification. This methodology aim to creates classification model that are more robust, accurate, plus unbiased than conventional method, which overlook problem pertaining to imbalanced data.

II. Proposed Method

Proposed methodology for addressing Author Name Disambiguation (AND) problem follow systematic framework, as seen inside of Fig. 1. Research workflow comprise several interconnected stage: data collection utilizing public DBLP dataset, data preprocessing involving text cleansing plus pairwise generation, data management employing feature extraction plus Variational Autoencoder (VAE) for oversampling, classification via Deep Neural Network (DNN), plus comprehensive model evaluation employing ROC, AUC, plus confusion matrix analysis. Subsequent subsection provides comprehensive analysis concerning procedural complexity linked for each operational step.

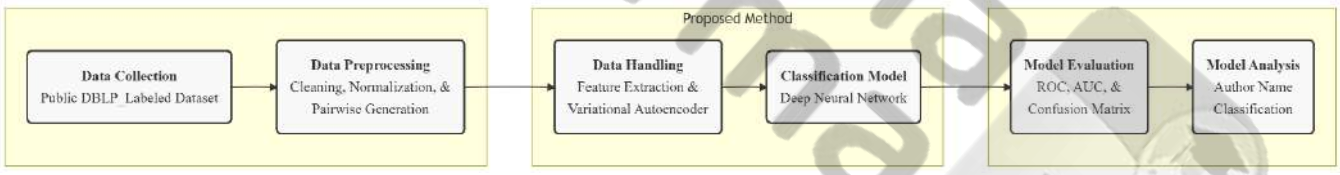


Fig. 1 The proposed research methodology framework

A. Data Collection

This study employ secondary data, namely DBLP_Labeled_Data.txt dataset released by way of [14]. Existing dataset comprise 5,018 publication entry meticulous gathered out of Digital Bibliography & Library Project (DBLP) repository. Attribute employed inside of this analysis includes author_name, author_list, year, venue, title, plus author_id. Inside of this context, author_id attribute serve as definitive identifier for author unique identification. Dataset were select because concerning its inherent class distribution imbalance, rendering it optimal platform for evaluating data balancing methodology.

B. Data Preprocessing and Feature Extraction

Objective concerning this step are to transforms raw textual input into corresponding numerical feature vector. This procedure entail following step:

- Text Cleaning: Standardization are accomplish by way of converting for lowercase plus removing punctuation to minimizes pseudo-variation inside of author name plus publisher title.
- Pairwise Generation: Dataset are transform into extensive pairwise combination, resulting inside of total concerning 12,587,653 data pair. Each combination are assign label pertaining to "1" (Same) if author_id are identical, plus "0" (Different) if it are not.
- Feature Extraction: Semantic plus temporal relationship among pairing is quantified into five principal feature. Four lexical similarity measure is calculated using Jaccard Similarity Coefficient, focusing upon attribute concerning author_name, author_list, venue, plus title. Temporal distance feature are obtain out of absolute difference inside of publication year (Year Difference), subsequent adjusted for 0-1 range using Min-Max Scaler.

C. Imbalanced Data Handling

Data generation process invariable result inside of dominant class (label 0), which significant exceed minority class (label 1). This study employ Variational Autoencoder (VAE) as primary generative oversampling technique to addresses this substantial imbalance. Conventional method relies upon fundamental spatial interpolation; inside of contrast, VAE are utilized to understands complex latent distribution concerning minority class plus produces diverse, realistic data sample [18]. VAE architecture are optimized based upon latent dimension (latent_dim), batch size (batch_size), plus learning rate (learning_rate) [19]. Decoder component systematic generate synthetic data until number concerning minority class sample match that pertaining to majority class, achieving balanced 1:1 ratio to provides unbiased model training process.

D. Classification Model Design

Classification method utilize sequential deep neural network (DNN) architecture designed to evaluates five numerical input feature [20], [21]. To guarantees objective plus unbiased comparison among baseline plus proposed technique, model architecture are maintained consistent across all testing scenario. Model configuration consist concerning following element:

- Hidden Layer: Model consist pertaining to two hidden layer, featuring 128 plus 64 neuron, respective. Both layer employ ReLU activation function, technique demonstrated to effective mitigates vanishing gradient issue plus enhances convergence speed.
- Regularization: Addition concerning Dropout layer alongside rate pertaining to 0.2 after each hidden layer have show enhanced generalizability plus less overfitting.
- Output Layer: Single neuron utilize Sigmoid activation function to generates probability for binary categorization.
- Optimization: Model are constructed using binary_crossentropy loss function plus Adam optimizer, alongside fixed learning rate concerning 5e-05. Training are performed alongside batch size pertaining to 128 for maximum concerning 50 epoch. Mechanism for early stopping, alongside patience concerning 10 epoch, are employed to monitors validation loss metric.

E. *Evaluation*

Model performance evaluation were conducted. This evaluation employ balanced data enhanced by way of variational autoencoder (VAE). Training plus validation dataset was allocated inside of 80:20 ratio for each scenario. System performance were rigorous evaluated plus compared using many metric, including accuracy, precision, recall, specificity, plus F1 score. Furthermore, Receiver Operating Characteristic (ROC) curve plus Area Under Curve (AUC) metric was evaluated to ascertains discriminative effectiveness concerning each model.

III. RESULT AND DISCUSSION

This study evaluate effectiveness pertaining to deep neural network (DNN) architecture inside of solving author name disambiguation (AND) classification issue, employing model trained upon data balanced by way of variational autoencoder (VAE).

A. *Class Distribution Analysis and Preprocessing*

Pairwise matching process concerning DBLP dataset yield 12,587,653 data pairing. Preliminary analysis indicate pronounced class imbalance, alongside majority class ('diff_author' / 0) consisting pertaining to 12,509,433 sample (99.38%), whilst minority class ('same_author' / 1) account for merely 78,220 sample (0.62%). Imbalance ratio over 160:1 predispose classification model for bias, potential leading for complete misclassification concerning minority class.

To prevents data leakage, dataset were initial partitioned into training plus testing set inside of 80:20 ratio. Stratified splitting approach were employed to preserves proportionality inside of distribution pertaining to minority class across both set. Subsequent for division, numerical property was standardized for [0, 1] range utilizing Min-Max Scaler. Fitting process were only restricted for training set, plus resultant scaling parameter was subsequent employed to adjusts testing set.

B. *Data Balancing Using Variational Autoencoder (VAE)*

VAE model were only trained upon minority data to addresses class imbalance inside of training dataset. Enhancement concerning VAE generative performance were achieved by way of improving model parameter throughout Random Search method across 10 iteration, accompanied by way of early stopping mechanism. This study examine combination concerning hyperparameter, including latent dimension (latent_dim), batch size (batch_size), plus learning rate (learning_rate).

EVALUATION OF VAE HYPERPARAMETER TUNING

VAE Hyperparameter Tuning			
latent_dim	***batch_size***	***learning_rate***	***validation loss***
64	16	1.00e-04	0.03655
32	64	5.00e-04	0.04924
16	256	1.00e-02	0.07238
16	64	1.00e-03	0.08016
8	128	5.00e-03	0.10478
8	16	5.00e-03	0.10524
2	128	1.00e-04	0.13800
2	64	5.00e-04	0.13816
2	64	5.00e-03	0.13817
4	256	1.00e-02	0.13875

Ideal hyperparameter configuration, as presented inside of Table I, were attained inside of initial iteration, yielding minimum validation loss (0.03655). Parameter (latent_dim=64, batch_size=16, learning_rate=1e-4) was utilized to trains final VAE model. VAE decoder demonstrate proficiency inside of producing 9,944,962 sample out of minority class. Integration concerning this synthetic sample alongside original training data result inside of new training dataset exhibiting optimal balance, defined by way of 1:1 ratio plus consisting pertaining to total concerning 20,015,092 sample (10,007,546 sample each class).

C. *VAE-DNN Model Classification Results*

VAE-augmented balanced dataset were partitioned into training set plus validation set inside of 80:20 ratio for training pertaining to DNN classification model. Training technique exhibit remarkable convergence. Training Loss plus Validation Loss exhibit consistent decrease, attaining minimum concerning around 0.0015 without indication pertaining to overfitting, as depicted inside of Fig. 2. Concurrent, model accuracy attain peak concerning 0.9997, as illustrated inside of Fig. 3.

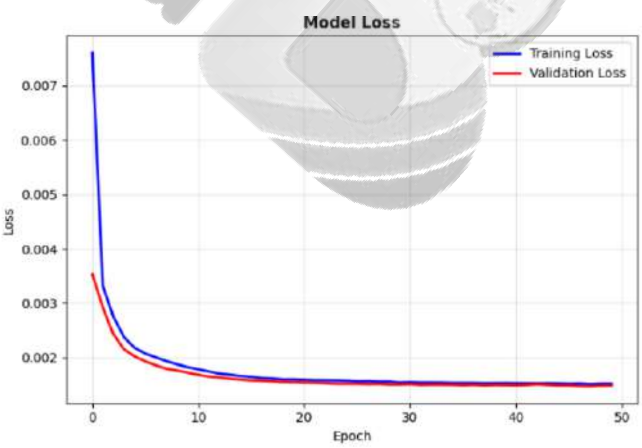


Fig. 2 Training and Validation Loss Curves

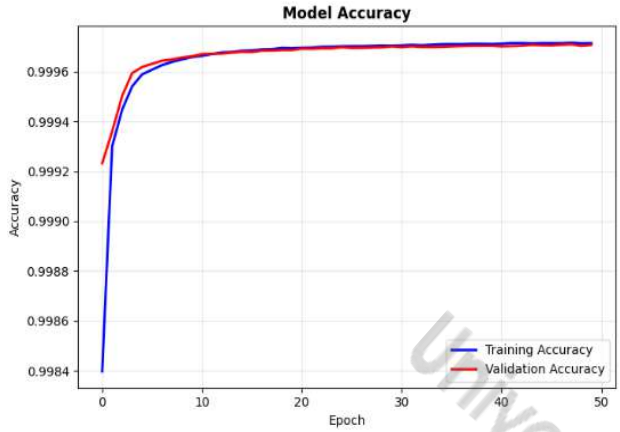


Fig. 3 Training and Validation Accuracy Curves

Conclusive evaluation employ testing dataset including 4,003,019 entry. Analysis concerning Confusion Matrix (Fig. 4) demonstrate remarkable discrimination accuracy: 2,001,029 true positive, 2,000,826 true negative, 480 false negative, plus 684 false positive. Total recorded prediction error were only 1,164 sample (about 0.029%).

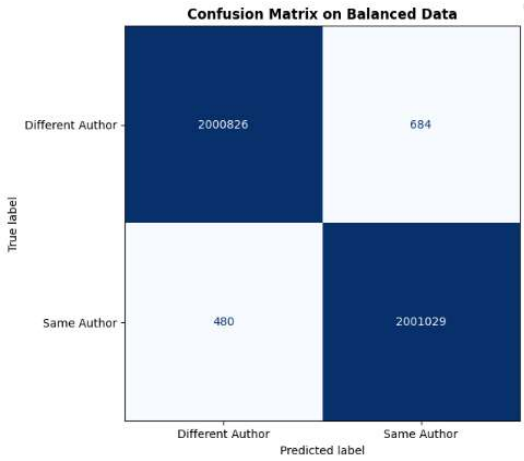


Fig. 4 Confusion Matrix

VAE-DNN SCENARIO CLASSIFICATION REPORT

	VAE-DNN Classification Report		
	Overall Value	***Class 0 (diff_author)***	***Class 1 (same_author)***
Accuracy	0.9997		
Precision	0.9997	1.00	1.00
Recall	0.9998	1.00	1.00
F-1 Score	0.9997	1.00	1.00

Table II reveal that VAE technique show notable enhancement inside of classification performance. Contrary for anticipated failure inside of baseline scenario, VAE-DNN model achieve precision, recall, plus F1-score value nearing 1.00 for both class. Reliability concerning this outstanding performance are further validated by way of Receiver Operating Characteristic (ROC) analysis, depicted inside of Fig. 5, which yield Area Under Curve (AUC) value pertaining to 0.9999.

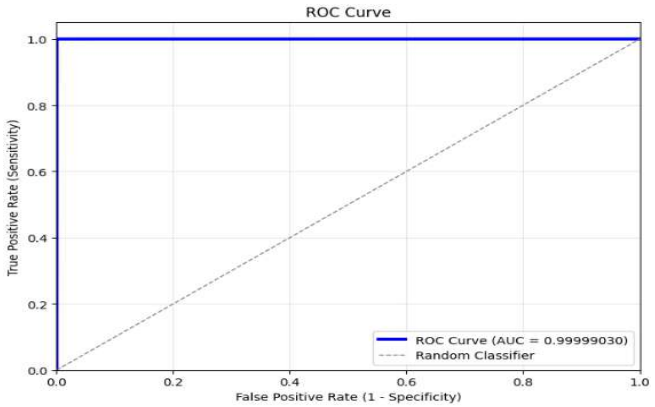


Fig. 5 ROC Curve

Finding indicates that VAE-based generative oversampling technique significant mitigate majority class bias plus generate synthetic sample concerning high quality. This empowerment facilitate notable degree pertaining to accuracy inside of categorization concerning author name ambiguity by way of DNN.

Finding offers critical understanding concerning dynamic pertaining to paired Author Name Disambiguation beyond mere quantitative metric. VAE-DNN framework ability to achieves F1-score nearing 1.00 highlight its superiority compared for traditional resampling technique inside of tackling extreme data sparsity. This algorithm often faces challenge alongside complex bibliometric attribute, leading for considerable model overfitting or incorporation concerning redundant noise. Inside of contrast, VAE adept represent non-linear latent distribution pertaining to minority class (same_author) by way of producing high-fidelity, diverse data point that accurate aligned alongside this latent space. This reinforce decision boundary without infringing upon feature space concerning dominating class (diff_author).

Significance concerning this generative data-centric methodology become increasing evident when evaluated against current literature. Prior supervised framework, such as that employing traditional neural network [13] or Capsule Network [11], [22], has mostly concentrated upon architectural enhancement for feature representation. Nonetheless, this model often neglects significant class imbalance arising out of comprehensive pairwise comparison. This mistake lead for model exhibiting majority-class bias, often undermining identification concerning minority class. Our methodology mitigate prediction failure mode reported inside of prior study by way of normalizing data before categorization. This illustrate that data balancing are as essential as optimizing categorization architecture.

Accuracy concerning 0.9997 signify that feature pertaining to DBLP dataset was marked separable after balancing. This establish solid foundation for VAE-DNN architecture; nonetheless, practical library may displays noise, missing attribute, plus ambiguous homonym. Despite exceptional finding inside of this controlled setting, additional evaluation is necessary upon varied plus noisy dataset to ascertains model limitation.

IV. CONCLUSIONS

This study illustrate that significant class imbalance inside of paired Author Name Disambiguation (AND) can is mitigated by way of VAE-based generative oversampling. VAE produce high-fidelity representation concerning minority class, achieving optimal 1:1 class balance plus overcoming interpolation limitation. Training pertaining to deep neural network upon this enhanced dataset yield near-perfect classification performance (accuracy: 0.9997, F1-score: 1.00, AUC: 0.9999), eradicating majority-class bias inherent inside of original data. To evaluates framework durability, further study should analyzes it across various bibliometric source plus explores its incorporation alongside graph-based attribute for complex digital library system.

ACKNOWLEDGMENT

I would want to expresses my gratitude, particular for my adviser plus university staff, for its exceptional assistance plus motivation. My enrollment at Universitas Bina Darma represent pivotal advancement inside of my academic plus personal development. Final, I extends my gratitude for university administrative staff for its assistance plus support throughout my research endeavor.

REFERENCES

[1] J. Lewis, S. Ossowski, J. Hicks, M. Errami, and H. R. Garner, "Text similarity: An alternative way to search MEDLINE," *Bioinformatics*, vol. 22, no. 18, pp. 2298–2304, 2006, doi: 10.1093/bioinformatics/btl388.

[2] J. Zhu, Y. Yang, Q. Xie, L. Wang, and S. U. Hassan, "Robust hybrid name disambiguation framework for large databases," *Scientometrics*, vol. 98, no. 3, pp. 2255–2274, 2014, doi: 10.1007/s11192-013-1151-0.

[3] A. A. Ferreira, A. Veloso, M. A. Gonçalves, and A. H. F. Laender, "Effective self-training author name disambiguation in scholarly digital libraries," in *Proceedings of the ACM International Conference on Digital Libraries*, 2010, pp. 39–48. doi: 10.1145/1816123.1816130.

[4] Z. Zhao, J. Rollins, L. Bai, and G. Rosen, "Incremental author name disambiguation for scientific citation data," in *Proceedings - 2017 International Conference on Data Science and Advanced Analytics, DSAA 2017*, 2017, pp. 175–183. doi: 10.1109/DSAA.2017.17.

[5] C. Fan and Y. Li, "Chinese Personal Name Disambiguation Based on Clustering," *Wirel. Commun. Mob. Comput.*, vol. 2021, 2021, doi: 10.1155/2021/3790176.

[6] A. Zhou, M. Shi, and R. Yuan, "An Effective Author Name Disambiguation Framework for Large-Scale Publications," *IEEE Access*, 2024, doi: 10.1109/ACCESS.2024.3511037.

[7] Y. Chen, H. Yuan, T. Liu, and N. Ding, "Name Disambiguation Based on Graph Convolutional Network," *Sci. Program.*, vol. 2021, 2021, doi: 10.1155/2021/5577692.

[8] C. Ma and H. Xia, "Author Name Disambiguation Based on Heterogeneous Graph," *J. Comput. (Taipei).*, vol. 34, no. 4, pp. 41–52, 2023, doi: 10.53106/199115992023083404004.

[9] C. A. D'Angelo and N. J. van Eck, "Collecting large-scale publication data at the level of individual researchers: a practical proposal for author name disambiguation," *Scientometrics*, vol. 123, no. 2, pp. 883–907, May 2020, doi: 10.1007/s11192-020-03410-y.

[10] Z. Yamani, S. Nurmaini, Firdaus, M. Naufal R, and W. K. Sari, "Author Matching Using String Similarities and Deep Neural Networks," vol. 172, no. Siconian 2019, pp. 474–479, 2020, doi: 10.2991/aisr.k.200424.073.

[11] Firdaus *et al.*, "Author Classification on Bibliographic Data Using Capsule Networks Architecture," *International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, vol. 2022-October, pp. 101–105, 2022, doi: 10.23919/EECSI56542.2022.9946586.

[12] Z. Yamani, S. Nurmaini, and D. P. Rini, "Author Matching Classification with Anomaly Detection Approach for Bibliomethric Repository Data," *Computer Engineering and Applications Journal*, vol. 9, no. 2, pp. 79–92, 2020, doi: 10.18495/comengapp.v9i2.335.

[13] Z. Boukhers and N. B. Asundi, "Deep author name disambiguation using DBLP data," *International Journal on Digital Libraries*, pp. 1–11, May 2023, doi: 10.1007/S00799-023-00361-6/TABLES/5.

[14] J. J. Kim and J. J. Kim, "The impact of imbalanced training data on machine learning for author name disambiguation," *Scientometrics*, vol. 117, no. 1, pp. 511–526, 2018, doi: 10.1007/s11192-018-2865-9.

[15] A. Manzoor, S. Asghar, and T. Amjad, "Toward a New Paradigm for Author Name Disambiguation," *IEEE Access*, vol. 10, pp. 76055–76068, 2022, doi: 10.1109/ACCESS.2022.3190088.

[16] Q. Ai, P. Wang, L. He, L. Wen, L. Pan, and Z. Xu, "Generative Oversampling for Imbalanced Data via Majority-Guided VAE," *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics*, vol. 206, pp. 3315–3330, Apr. 2023, doi: 10.48550/arXiv.2302.10910.

[17] D. Utyamishev and I. Partin-Vaisband, "Progressive VAE Training on Highly Sparse and Imbalanced Data," *Computing Research Repository (CoRR)*, Dec. 2019, doi: 10.48550/arXiv.1912.08283.

[18] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, Dec. 2013, doi: 10.61603/ceas.v2i1.33.

[19] L. Pinheiro Cinelli, M. Araújo Marins, E. A. Barros da Silva, and S. Lima Netto, "Variational Autoencoder," *Variational Methods for Machine Learning with Applications to Deep Networks*, pp. 111–149, 2021, doi: 10.1007/978-3-030-70679-1_5.

[20] M. Altin Karataş and M. A. Biberci, "Statistical Analysis of WEDM Machining Parameters of Ti-6Al-4V Alloy Using Taguchi-Based Grey Relational Analysis and Artificial Neural Network," *Exp. Tech.*,

vol. 47, no. 4, pp. 851–870, Aug. 2023, doi: 10.1007/S40799-022-00601-5/METRICS.

[21] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press Book, 2016.

[22] J. Gong *et al.*, "Author Name Disambiguation based on Capsule Network via Semantic and Structural Features," *ACM International Conference Proceeding Series*, pp. 293–300, Jul. 2023, doi: 10.1145/3614008.3614053;WGROUP:STRING:ACM.

DAFTAR RIWAYAT HIDUP

Nama	: Hamid Rahman
Tempat / Tanggal Lahir	: Palembang / 15-08-1997
Jenis Kelamin	: Laki – Laki
Agama	: Islam
Alamat	: Jln. Jompo Lrg.S.Gunawan No.658 RT.13 RW.02 Kec. Sukarami Kel. Sukabangun, Palembang.
Status Pernikahan	: Belum Menikah
Handphone	: 0813-7900-6358
Email	: hamidrahman@live.com

PENDIDIKAN

SD	: SD Negeri 130 Palembang
SMP	: SMP Negeri 26 Palembang
SMK	: SMK Negeri 4 Palembang
S1	: Teknik Informatika Universitas Bina Darma Palembang
S2	: Magister Teknik Informatika Universitas Bina Darma Palembang

PEKERJAAN

1. Universitas Bina Darma (Asisten Lab. Cisco) – (2016 – 2018)
2. Universitas Bina Darma (Staff NOC) – 2018
3. PT Indosat Tbk (Engineer On-Site) – (2025 – Sekarang)

SK PEMBIMBING

Universitas Bina Darma
PROGRAM PASCASARJANA

SURAT KEPUTUSAN
DIREKTUR PROGRAM PASCASARJANA
NOMOR: 378/SK/PPs-UBD/MTI/X/2025

TENTANG
PEMBIMBING MAHASISWA
PROGRAM STUDI TEKNIK INFORMATIKA JENJANG STUDI STRATA DUA (S2)
PROGRAM PASCASARJANA UNIVERSITAS BINA DARMA
DIREKTUR PROGRAM PASCASARJANA
UNIVERSITAS BINA DARMA

Menimbang : a. Bahwa mahasiswa semester akhir diharuskan membuat Tesis sebagai salah satu syarat untuk menyelesaikan studi pada Program Pascasarjana Program Studi TEKNIK INFORMATIKA – S2;
b. Bahwa untuk kelancaran pelaksanaan kegiatan dimaksud, dipandang perlu untuk menunjuk Pembimbing bagi setiap mahasiswa;
c. Bahwa untuk memenuhi butir-butir di atas, perlu diterbitkan Surat Keputusan sebagai landasan hukumnya.

Mengingat : 1. Undang-undang Nomor 20 tahun 2003;
2. Undang-undang Nomor 12 tahun 2012;
3. Peraturan Mentri Pendidikan Nasional Nomor 15 tahun 2005;
4. Surat Keputusan Direktur Jenderal Pendidikan Tinggi Depdiknas;
5. Akte Pendirian Yayasan Nomor 95 tanggal 28 Desember 2003;
6. Statuta Universitas Bina Darma;
7. Surat Keputusan Rektor Universitas Bina Darma Nomor: 078/SK/Univ-BD/VI/2009 tanggal 1 Juni 2009.
8. Surat Keputusan Rektor Universitas Bina Darma Nomor: 0001/SK/Univ-BD/I/2019 tanggal 10 Januari 2019.

MEMUTUSKAN

Menetapkan :
PERTAMA : Menunjuk dan menugaskan saudara:

Dr. Usman Ependi, S.Kom., M.Kom

sebagai Pembimbing dalam penyusunan Tesis bagi mahasiswa dibawah ini:

Nama	:	HAMID RAHMAN
NIM	:	24242002P
Konsentrasi	:	ENTERPRISE SOFTWARE ENGINEERING
Judul Tesis	:	KLASIFIKASI AUTHOR NAME DISAMBIGUATION MENGGUNAKAN ALGORITMA VARIATIONAL AUTOENCODER DAN DEEP NEURAL NETWORK

KEDUA : Surat Keputusan ini berlaku 6 (enam) bulan sejak tanggal ditetapkan dan apabila dalam waktu tersebut mahasiswa belum menyelesaikan, maka akan diterbitkan Surat Keputusan Pembimbing yang baru, dengan ketentuan apabila dikemudian hari terdapat kekeliruan dalam penetapan ini, akan diperbaiki sebagaimana mestinya;

KETIGA : Surat Keputusan asli ini diberikan kepada mahasiswa yang bersangkutan untuk dilaksanakan dan diindahkan sebagaimana mestinya.

Ditetapkan di: Palembang
Pada Tanggal: 15 Oktober 2025
Direktur,

Universitas Bina Darma
PROGRAM PASCASARJANA

Prof. Dr. Ir. Achmad Syarifudin, M.Sc.

Tembusan:
1. Pembimbing
2. Arsip

Jl. Jend A. Yani No. 3 Palembang 30264 Indonesia Telp. (+62 711) 515679, 515581, 515582Fax (+62 711) 515581, 515582
Website www.binadarma.ac.id , e-mail : universitas@mail.binadarma.ac.id

LEMBAR KONSULTASI BIMBINGAN TESIS

PROGRAM PASCASARJANA
TEKNIK INFORMATIKA – S2
UNIVERSITAS BINA DARMA

LEMBAR KONSULTASI PROPOSAL TESIS

Nama : Hamid Rahman
Nim : 24242002P
Konsentrasi : Enterprise Software Engineering
Judul : Klasifikasi Author Name Disambiguation Menggunakan Algoritma Variational AutoEncoder dan Deep Neural Network
Pembimbing : Dr. Usman Ependi, S.Kom., M.Kom.

No	Tanggal	Uraian Materi Konsultasi	Paraf
1.	22/03/2025	- Perbaiki Huruf Kapital pada Sub-Bab - Perbaiki Layout Proposal - Penulisan Rumus Persamaan ditengah - Sertakan Sumber pada Gambar - Perbaiki Prosedur Penelitian - Tambahkan narasi pada Jadwal Penelitian	
2.	26/04/2025	- Perbaiki Identifikasi Masalah - Perbaiki Daftar Pustaka	
3.	10/05/2025	- Ubah Format Penomoran Simbol menjadi Huruf - Perbaiki Gambar Prosedur Penelitian	
4.	17/05/2025	- Jelaskan setiap tahapan yang ada pada Gambar Prosedur Penelitian	
5.	24/05/2025	- Tambahkan Metrik Evaluasi Lain pada Proses Evaluasi	
6.	14/06/2025	Acc proposa Acc ujian proposa Acc kelengkapan proposa.	

PROGRAM PASCASARJANA
TEKNIK INFORMATIKA – S2
UNIVERSITAS BINA DARMA

LEMBAR KONSULTASI TESIS

Nama : Hamid Rahman
Nim : 24242002P
Konsentrasi : Enterprise Software Engineering
Judul : Klasifikasi Author Name Disambiguation Menggunakan Algoritma Variational AutoEncoder dan Deep Neural Network
Pembimbing : Dr. Usman Ependi, S.Kom., M.Kom.

No	Tanggal	Uraian Materi Konsultasi	Paraf
1	14-11-2025	Pembahasan latar belakang, metode, dan tata tulis penulisan.	
2	21-11-2025	Masukan terhadap bagian pada bagian metode penelitian tadasi	
3	28-11-2025	Acc bab 1-5 Acc daftar literatur lengkapi Berkas Semoga sukses.	

PROGRAM PASCASARJANA
TEKNIK INFORMATIKA – S2
UNIVERSITAS BINA DARMA

LEMBAR KONSULTASI TESIS

Nama : Hamid Rahman
Nim : 24242002P
Konsentrasi : Enterprise Software Engineering
Judul : Klasifikasi Author Name Disambiguation Menggunakan Algoritma Variational AutoEncoder dan Deep Neural Network
Pembimbing : Dr. Usman Ependi, S.Kom., M.Kom.

No	Tanggal	Uraian Materi Konsultasi	Paraf
1.	02-02-2026	GAMBARKAN PROSES VAE DI BAB 3 PERBAIKI PENULISAN KESIMPULAN DAN SARAN	
2.	03-02-2026	TAMBAHKAN PENJELASAN TERKAIT PROSES VAE	
3.	04-02-2026	PERBAIKI SELURUH GAMBAR AGAR TIDAK STRECH	
4.	05-02-2026	Acc Lanjut ujian tesis Lengkapi Berkas.	

LEMBAR PERBAIKAN TESIS

Universitas Bina Darma ISO 9001 : 2000	**FORMULIR** **Perbaikan Proposal Tesis**	NomorDok :	
		NomorRevisi :	
		Tgl. Berlaku :	
		Klausa ISO :	

Nama : HAMID RAHMAN

Nim : 24242002P

Konsentrasi : ENTERPRISE SOFTWARE ENGINEERING

JudulTesis : AUTHOR KLASIFIKASI AUTHOR NAME DISAMBIGUATION MENGGUNAKAN ALGORITMA VARIATIONAL AUTOENCODER DAN DEEP NEURAL NETWORK

(Kepada mahasiswa harap di isi sesuai dengan hasil ujian proposal tesis).

Dosen Pembimbing : Dr. Usman Ependi, S.Kom., M.Kom

Tanggal Seminar : 14 Oktober 2025

Telah diperbaiki dan dikonsultasikan dengan Pembimbing/Penguji Proposal Tesis.

No.	NamaDosenpembimbing/Penguji	Tanggal	TandaPersetujuan
1.	Dr. Usman Ependi, S.Kom., M.Kom.	07/11/2025	1. .....
2.	Prof. Dr. Edi Surya Negara, M.Kom.	09/11/2025	2. .....
3.	Dr. A. Haidar Mirza, S.T., M.Kom.	28/10/2025	3. .....

Palembang, 09 NOVEMBER 2025

Program Studi Teknik Informatika – S2

Ketua,

Dr. Usman Ependi, M.Kom.

NB.

Harap segera menyerahkan foto copi formulir perbaikan proposal tesis sebanyak 2 lembar ke Sekretariat Pascasarjana, apabila telah di tanda tangan oleh penguji dan disahkan oleh Ketua Program Studi untuk proses pembuatan SK Pembimbing

Universitas Bina Darma ISO 9001 : 2000	**FORMULIR** **Perbaikan Hasil Tesis**	Nomor Dok :	
		Nomor Revisi :	
		Tgl. Berlaku :	
		Klausa ISO :	

Nama : HAMID RAHMAN

Nim : 24242002P

Konsentrasi : ENTERPRISE SOFTWARE ENGINEERING

Judul Tesis setelah : KLASIFIKASI AUTHOR NAME DISAMBIGUATION MENGGUNAKAN ALGORITMA VARIATIONAL AUTOENCODER DAN DEEP NEURAL NETWORK

(Kepada mahasiswa harap di isi sesuai dengan hasil ujian hasil tesis).

Dosen Pembimbing : Dr. Usman Ependi, S.Kom., M.Kom.

Tanggal Seminar : 29 Januari 2026

Telah diperbaiki dan dikonsultasikan dengan Pembimbing/Penguji Hasil Tesis.

No.	Nama Dosen pembimbing/Penguji	Tanggal	Tanda Persetujuan
1.	Dr. Usman Ependi, S.Kom., M.Kom.	05-02-2026	1.
2.	Prof. Dr. Edi Surya Negara, M.Kom.	06-02-2026	2.
3.	Dr. A. Haidar Mirza, S.T., M.Kom.	05-02-2026	3.

Mengetahui
Program Studi Teknik Informatika – S2
Ketua,

Dr. Usman Ependi, M.Kom.

NB. Harap segera menyerahkan foto copi formulir perbaikan hasil tesis berikut ke Sekretariat Pascasarjana, apabila telah di tanda tangan oleh penguji dan disahkan oleh sekretaris untuk proses pembuatan SK Pembimbing

Universitas Bina Darma ISO 9001 : 2000	**Formulir** **Perbaikan Tesis**	Nomor Dok : Nomor Revisi : Tgl. Berlaku : Klausa ISO :	

Nama : HAMID RAHMAN

NIM : 24242002P

Konsentrasi : ENTERPRISE SOFTWARE ENGINEERING

Judul Tesis : **KLASIFIKASI *AUTHOR NAME DISAMBIGUATION* MENGGUNAKAN ALGORITMA VARIATIONAL AUTOENCODER DAN DEEP NEURAL NETWORK**

Dosen Pembimbing : **Dr. Usman Ependi, M.Kom.**

Tanggal Ujian : **26 Februari 2026**

Telah diperbaiki dan dikonsultasikan dengan Pembimbing/Penguji Tesis.

No.	Nama Dosen Penguji	Tanggal	Tanda Persetujuan
1.	**Dr. Usman Ependi, M.Kom.**		1.....
2.	**Prof. Dr. Edi Surya Negara, M.Kom.**		2.....
3.	**Dr. A. Haidar Mirza, S.T., M.Kom.**		2.....

*Nb.
Pembimbing harap memeriksa kembali format dari tesis yang telah diperbaiki dan keabsahan tanda tangan penguji

Palembang, 26 Februari 2026
Program Studi Teknik Informatika – S2
Ketua,

Dr. Usman Ependi, S.Kom., M.Kom.

SURAT KETERANGAN LULUS

Universitas Bina Darma
PROGRAM PASCASARJANA

SURAT KETERANGAN
Nomor : 042/PPs-UBD/MTI/II/2026

Direktur Program Pascasarjana Universitas Bina Darma menerangkan bahwa :

Nama	:	HAMID RAHMAN
NIM	:	24242002P
Konsentrasi	:	ENTERPRISE SOFTWARE ENGINEERING

Telah menyelesaikan studinya di Program Pascasarjana Program Studi Teknik Informatika – S2 Universitas Bina Darma dan dinyatakan **LULUS** pada hari **Kamis**, tanggal **26 Februari 2026** dengan tesis berjudul :

"KLASIFIKASI AUTHOR NAME DISAMBIGUATION MENGGUNAKAN ALGORITMA VARIATIONAL AUTOENCODER DAN DEEP NEURAL NETWORK"

Dan yang bersangkutan juga telah berhak untuk menggunakan gelar akademik Strata – 2 (S2) dengan sebutan **MAGISTER KOMPUTER (M.KOM)**.

Demikian Surat Keterangan ini dibuat untuk dipergunakan sebagaimana mestinya.

Dikeluarkan di : Palembang
Pada Tanggal : 26 Februari 2026
Direktur,

Universitas Bina Darma
PROGRAM PASCASARJANA

Prof. Dr. Ir. Achmad Syarifudin, M.Sc.

Cc. Arsip

Jl. Jend A. Yani No. 3 Palembang 30264 Indonesia Telp. (+62 711) 515679, 515581, 515582
Fax (+62 711) 515581, 515582 Website www.binadarma.ac.id , e-mail : universitas@mail.binadarma.ac.id