DEEP SUPPORT VECTOR DATA DESCRIPTION (*DEEP* SVDD) BERBASIS BIOWORDVEC UNTUK DETEKSI ANOMALI PASANGAN TEKS LEUKEMIA PADA GENIA *BIOMEDICAL EVENT CORPUS*

Skripsi

Oleh

KHUSNI SINTA RODIYAH
NPM. 2217031064

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2026

ABSTRACT

DEEP SUPPORT VECTOR DATA DESCRIPTION (DEEP SVDD) BASED ON BIOWORDVEC FOR LEUKEMIA TEXT PAIR ANOMALY DETECTION IN THE GENIA BIOMEDICAL EVENT CORPUS

By

Khusni Sinta Rodiyah

Anomaly detection in biomedical text pairs is a critical challenge in natural language processing, particularly in data with complex semantic relationships such as the GENIA Biomedical Event dataset. This study aims to develop an anomaly detection model based on Deep Support Vector Data Description (Deep SVDD) for biomedical text pairs: Sentence–TriggerWord, Sentence–EventType, and TriggerWord–EventType. In the initial stage, pseudo-labeling was performed using a modified Word Mover's Distance (WMD) approximation approach with Part-of-Speech (POS) weighting to classify normal and anomalous texts based on semantic similarity. Feature representations were constructed using BioWordVec embeddings, which were then processed using LSTM and BiLSTM encoders to generate latent representations. The results showed that the Deep SVDD model with an LSTM encoder provided the best performance, with an ROC AUC of 0.99 on the Sentence–TriggerWord dataset and 0.98 on the other two datasets, while BiLSTM performed lower. Furthermore, a threshold sensitivity analysis revealed a trade-off between sensitivity and specificity. Increasing the threshold increased specificity but decreased sensitivity. Therefore, the P95 threshold was chosen because it yielded the highest sensitivity and was therefore more effective in minimizing false negatives.

Keywords: Anomaly detection, Deep SVDD, GENIA Biomedical Event, BioWordVec, LSTM, BiLSTM, ROC AUC, Threshold Sensitivity.

ABSTRAK

DEEP SUPPORT VECTOR DATA DESCRIPTION (*DEEP* SVDD) BERBASIS BIOWORDVEC UNTUK DETEKSI ANOMALI PASANGAN TEKS LEUKEMIA PADA GENIA *BIOMEDICAL EVENT CORPUS*

Oleh

Khusni Sinta Rodiyah

Deteksi anomali pada pasangan teks biomedis merupakan tantangan penting dalam *natural language processing*, khususnya pada data dengan hubungan semantik yang kompleks seperti dataset GENIA *Biomedical Event*. Penelitian ini bertujuan mengembangkan model deteksi anomali berbasis *Deep Support Vector Data Description* (*Deep* SVDD) pada pasangan teks biomedis, yaitu *Sentence–TriggerWord*, *Sentence–EventType*, dan *TriggerWord–EventType*. Pada tahap awal, dilakukan *pseudo labeling* menggunakan pendekatan aproksimasi *Word Mover's Distance* (*WMD*) yang dimodifikasi dengan pembobotan *Part-of-Speech* (*POS*) untuk membentuk kelas normal dan anomali berdasarkan tingkat kemiripan semantik. Representasi fitur dibangun menggunakan *embedding* BioWordVec yang kemudian diproses menggunakan *encoder* LSTM dan BiLSTM untuk menghasilkan representasi laten. Hasil penelitian menunjukkan bahwa model *Deep* SVDD dengan *encoder* LSTM memberikan performa terbaik dengan nilai ROC AUC sebesar 0,99 pada dataset *Sentence–TriggerWord* serta 0,98 pada dua dataset lainnya, sedangkan BiLSTM menunjukkan performa yang lebih rendah. Selain itu, analisis sensitivitas *threshold* menunjukkan adanya *trade-off* antara sensitivitas dan spesifisitas. Peningkatan *threshold* meningkatkan spesifisitas namun menurunkan sensitivitas. Oleh karena itu, *threshold* P95 dipilih karena menghasilkan sensitivitas tertinggi sehingga lebih efektif dalam meminimalkan *false negative*.

Kata-kata kunci: Deteksi anomali, *Deep* SVDD, GENIA *Biomedical Event*, BioWordVec, LSTM, BiLSTM, ROC AUC, *Treshold Sensitivity*.

DEEP SUPPORT VECTOR DATA DESCRIPTION (*DEEP* SVDD) BERBASIS BIOWORDVEC UNTUK DETEKSI ANOMALI PASANGAN TEKS LEUKEMIA PADA GENIA *BIOMEDICAL EVENT CORPUS*

KHUSNI SINTA RODIYAH

Skripsi

Sebagai Salah Satu Syarat untuk Memperoleh Gelar
SARJANA MATEMATIKA

Pada

Jurusan Matematika

Fakultas Matematika dan Ilmu Pengetahuan Alam

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2026

Judul Skripsi	:	***DEEP SUPPORT VECTOR DATA DESCRIPTION (DEEP* SVDD) BERBASIS BIOWORDVEC UNTUK DETEKSI ANOMALI PASANGAN TEKS LEUKEMIA PADA GENIA *BIOMEDICAL EVENT CORPUS***
Nama Mahasiswa	:	Khusni Sinta Rodiyah
Nomor Pokok Mahasiswa	:	**2217031064**
Program Studi	:	**Matematika**
Fakultas	:	**Matematika dan Ilmu Pengetahuan Alam**

MENYETUJUI

1. Komisi Pembimbing

Dr. Dian Kurniasari, S.Si., M.Sc.	**Favorison R. Lumbanraja, S.Kom., M.Si., Ph.D.**
NIP. 196903051996032001	NIP. 198301102008121002

2. Ketua Jurusan Matematika

Dr. Aang Nuryaman, S.Si., M.Si.
NIP. 197403162005011001

MENGESAHKAN

1. Tim Penguji

Ketua : **Dr. Dian Kurniasari, S.Si., M.Sc.** .....

Sekretaris : **Favorisen R. Lumbanraja, S.Kom., M.Si., Ph.D.** .....

Penguji
Bukan Pembimbing : **Ir. Warsono, M.S., Ph.D.** .....

2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam

Dr. Eng. Heri Satria, S.Si., M.Si.
NIP. 197110012005011002

Tanggal Lulus Ujian Skripsi: **19 Mei 2026**

PERNYATAAN SKRIPSI MAHASISWA

Yang bertanda tangan di bawah ini:

Nama	:	**Khusni Sinta Rodiyah**
Nomor Pokok Mahasiswa	:	**2217031064**
Jurusan	:	**Matematika**
Judul Skripsi	:	***Deep Support Vector Data Description* (*Deep* SVDD) Berbasis Biowordvec Untuk Deteksi Anomali Pasangan Teks Leukemia Pada Genia *Biomedical Event Corpus***

Dengan ini menyatakan bahwa skripsi ini adalah hasil pekerjaan saya sendiri. Apabila kemudian hari terbukti bahwa skripsi ini merupakan hasil salinan atau dibuat oleh orang lain, maka saya bersedia menerima sanksi sesuai dengan ketentuan akademik yang berlaku.

Bandar Lampung, 19 Mei 2026

Penulis

Khusni Sinta Rodiyah

RIWAYAT HIDUP

Penulis memiliki nama lengkap Khusni Sinta Rodiyah yang lahir di Marga Agung pada tanggal 13 November 2004. Penulis merupakan anak kedua dari dua bersaudara, pasangan Bapak Nur Kadim dan Ibu Suwarni.

Penulis mulai menempuh pendidikan di Taman Kanak-kanak Pendidikan Anak Usia Dini (TK-PAUD) Dharma Wanita pada tahun 2008-2010, dan dilanjutkan di Sekolah Dasar di SD Negeri 2 Marga Agung dari tahun 2010-2016. Penulis melanjutkan pendidikan sekolah menengah pertama di MTs. Al-Hidayah Marga Agung pada tahun 2016-2019, kemudian melanjutkan ke jenjang menengah atas di SMA Negeri 13 Bandar Lampung pada tahun 2019-2022.

Pada tahun 2022, penulis menjadi salah satu mahasiswa jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung melalui jalur Seleksi Nasional Masuk Perguruan Tinggi Negeri (SNMPTN). Selama menjalani studi di perguruan tinggi, penulis telah terlibat aktif dalam kegiatan organisasi kemahasiswaan dengan menjadi bagian dari HIMATIKA UNILA periode 2023 sebagai anggota Bidang Keilmuan, kemudian menjadi bendahara umum HIMATIKA UNILA periode 2024. Pada Desember 2024 hingga Februari 2025, penulis melaksanakan Kerja Praktik (KP) di PT. Bank Rakyat Indonesia (Persero) Regional Office Bandar Lampung. Serta, pada Juni sampai Agustus 2025, penulis melaksanakan Kuliah Kerja Nyata (KKN) sebagai bentuk pengabdian penulis kepada masyarakat di Kelurahan Way Gubak, Kecamatan Sukabumi, Kota Bandar Lampung

KATA INSPIRASI

”Jangan pernah merasa tertinggal, setiap orang punya proses dan rezeki-Nya masing-masing.”

(Q.S Maryam : 4)

”Allah tidak membebani seseorang, kecuali menurut kesanggupannya.”

(Q.S Al-Baqarah : 285)

”Proses sama pentingnya dibandingkan hasil. Hasilnya nihil tak apa. Yang penting sebuah proses telah dicanangkan dan dilaksanakan.”

(Sudjiwo Tejo)

”Janganlah takut jatuh, karena yang tidak pernah memanjatlah yang tidak pernah jatuh. Dan jangan takut gagal, karena yang tidak pernah gagal hanyalah orang-orang yang tidak pernah melangkah. Dan jangan takut salah, karena dengan kesalahan yang pertama kita dapat menambah pengetahuan untuk mencari jalan yang benar pada langkah yang kedua.”

(Buya Hamka)

PERSEMBAHAN

Dengan mengucap Alhamdulillah dan syukur kepada Allah SWT atas nikmat serta hidayah-Nya sehingga skripsi ini dapat terselesaikan dengan baik dan tepat pada waktunya. Dengan rasa syukur dan Bahagia, saya persembahkan rasa terimakasih saya kepada:

Bapak dan Mamak Tercinta

Terimakasih kepada orang tuaku atas segala pengorbanan, motivasi, doa dan ridho serta dukungannya selama ini. Terimakasih telah memberikan pelajaran berharga kepada anakmu ini tentang makna perjalanan hidup yang sebenarnya sehingga kelak bisa menjadi orang yang bermanfaat bagi banyak orang.

Dosen Pembimbing dan Pembahas

Terimakasih kepada dosen pembimbing dan pembahas yang sudah sangat membantu, memberikan motivasi, memberikan arahan serta ilmu yang berharga.

Sahabat-sahabatku

Terimakasih kepada semua orang-orang baik yang telah memberikan pengalaman, semangat, motivasinya, serta doa-doanya dan senantiasa memberikan dukungan dalam hal apapun.

Almamater Tercinta
Universitas Lampung

SANWACANA

Alhamdulillah, puji dan syukur penulis panjatkan kepada Allah SWT atas limpahan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi ini yang berjudul "*Deep Support Vector Data Description* (*Deep* SVDD) Berbasis Biowordvec Untuk Deteksi Anomali Pasangan Teks Leukemia Pada Genia *Biomedical Event Corpus*" dengan baik dan lancar. Shalawat serta salam semoga senantiasa tercurahkan kepada Nabi Muhammad SAW.

Dalam proses penyusunan skripsi ini, banyak pihak yang telah membantu memberikan bimbingan, dukungan, arahan, motivasi serta saran sehingga skripsi ini dapat terselesaikan. Oleh karena itu, dalam kesempatan ini penulis mengucapkan terimakasih kepada:

1. Ibu Dr. Dian Kurniasari, S.Si., M.Sc. selaku Pembimbing I yang telah banyak meluangkan waktunya untuk memberikan arahan, bimbingan, motivasi, saran serta dukungan kepada penulis sehingga dapat menyelesaikan skripsi ini.
2. Bapak Favorisen R. Lumbanraja. S.Kom., M.Si., Ph.D. selaku Pembimbing II yang telah memberikan arahan, bimbingan dan dukungan kepada penulis sehingga dapat menyelesaikan skripsi ini.
3. Bapak Ir. Warsono, M.S., Ph.D. selaku Penguji yang telah bersedia memberikan kritik dan saran serta evaluasi kepada penulis sehingga dapat menjadi lebih baik lagi.
4. Bapak Dr. Aang Nuryaman, S.Si., M.Si. selaku Ketua Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.
5. Bapak Drs. Nusyirwan, M.Si. selaku dosen pembimbing akademik.
6. Seluruh dosen, staff dan karyawan Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.

7. Bapak, Mamak, Ayuk dan seluruh orang tersayang yang senantiasa memberikan do'a dan motivasi kepada penulis.
8. Teman seperjuangan sekaligus sahabatku, Nur Rahma Azzahra dan Munadiya Alfania. Terimakasih sudah memberikan kesempatan untuk berada diantara kalian yang sangat membantu dan memotivasi selama berada di bangku perkuliahan.
9. Rekan seperjuangan skripsi, diantaranya Nadia, Anita, Mei, Fadillah, Fatur, Oja, Erin, Rizki, dan Benaya. Terimakasih sudah saling mendukung dan menyemangati.
10. Teman-teman seperjuangan Jurusan Matematika angkatan 2022.

Semoga skripsi ini dapat bermanfaat bagi kita semua. Penulis menyadari bahwa skripsi ini masih jauh dari sempurna, sehingga penulis mengharapkan kritik dan saran yang membangun untuk menjadikan skripsi ini lebih baik lagi.

Bandar Lampung, 19 Mei 2026

Khusni Sinta Rodiyah

DAFTAR ISI

Halaman

DAFTAR TABEL

Tabel Halaman

DAFTAR GAMBAR

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Leukemia merupakan salah satu jenis kanker darah yang ditandai oleh proliferasi *abnormal* sel darah putih pada sumsum tulang dan sistem sirkulasi darah sehingga mengganggu fungsi fisiologis tubuh (Elzein, 2024). Penyakit ini masih menjadi salah satu penyebab utama kematian akibat kanker di dunia dengan jumlah kasus yang terus meningkat setiap tahun (Sung *et al*., 2021). Kondisi tersebut mendorong peningkatan penelitian biomedis yang menghasilkan volume literatur ilmiah yang sangat besar. Sebagian besar literatur biomedis terdokumentasi dalam berbagai repositori ilmiah seperti PubMed, Scopus, dan *ScienceDirect*. Di antara berbagai repositori tersebut, PubMed menjadi salah satu sumber utama literatur biomedis karena menyediakan jutaan abstrak penelitian yang terus diperbarui (Lu, 2011). Pertumbuhan literatur yang sangat pesat menyebabkan volume literatur biomedis global terus bertambah setiap tahun, namun sebagian besar informasi tersebut belum dimanfaatkan secara optimal untuk mendukung analisis otomatis dan penemuan pengetahuan baru (Chaves *et al*., 2022). Oleh karena itu, diperlukan pendekatan komputasional yang mampu membantu proses analisis dan ekstraksi informasi dari teks biomedis secara otomatis.

Perkembangan *Natural Language Processing* (NLP) memberikan peluang besar dalam pengolahan data teks tidak terstruktur. NLP memungkinkan komputer untuk memahami, memproses, dan mengekstraksi informasi dari bahasa alami secara otomatis (Chang, 2023). Pada domain biomedis, penerapan NLP umumnya memanfaatkan korpus beranotasi yang dirancang khusus untuk memahami terminologi ilmiah yang kompleks. Salah satu korpus yang banyak digunakan adalah GENIA *Biomedical Event corpus*, yaitu *dataset* yang dibangun dari abstrak PubMed yang berkaitan dengan proses biologis pada sel darah manusia

dan leukemia. Dataset ini terdiri dari sekitar 1.000 abstrak dengan lebih dari 8.000 kalimat yang dianotasi berdasarkan *sentence*, *trigger word*, dan *event type*, sehingga memungkinkan analisis hubungan semantik antar komponen teks biomedis. Meskipun *dataset* GENIA telah dianotasi secara manual oleh para ahli biomedis, kompleksitas bahasa ilmiah serta variasi terminologi biologis tetap memungkinkan munculnya ketidakkonsistenan pola semantik dalam data teks. Dalam analisis data, fenomena tersebut dapat dipandang sebagai anomali, yaitu data yang memiliki karakteristik berbeda secara signifikan dibandingkan dengan pola mayoritas dalam suatu distribusi data (Chandola *et al*., 2009). Jika anomali tidak terdeteksi dengan baik, sistem analisis teks otomatis dapat menghasilkan relasi biologis yang tidak akurat atau mengidentifikasi hubungan semantik yang keliru. Kondisi ini menyebabkan kesalahan interpretasi pada model NLP dan menegaskan perlunya mekanisme tambahan untuk mendeteksi pasangan teks yang menyimpang secara semantik (Moradi *et al*., 2021). Studi oleh Moradi *et al*. (2021), juga menunjukkan bahwa keberadaan *semantic outliers* dalam data teks medis dapat menyebabkan degradasi kinerja model secara signifikan pada pada beberapa tugas pemrosesan teks medis.

Menurut Challa (2024), deteksi anomali dapat dilakukan melalui pendekatan *One-Class Classification* (OCC). Pendekatan ini mempelajari karakteristik satu kelas utama tanpa memerlukan contoh data negatif secara eksplisit (Khan & Madden, 2013). Salah satu metode OCC berbasis *deep learning* yang banyak digunakan adalah *Deep Support Vector Data Description* (*Deep* SVDD) yang diperkenalkan oleh Ruff *et al*. (2018). Metode ini memetakan data ke dalam ruang representasi laten yang kompak sehingga data normal terkonsentrasi di sekitar suatu pusat distribusi (*hypersphere*), sedangkan data yang menyimpang dari distribusi tersebut dapat diidentifikasi sebagai anomali.

Beberapa penelitian sebelumnya telah menerapkan *Deep* SVDD pada berbagai domain data. Ruff, et al. (2018) mengimplementasikan *Deep* SVDD pada dataset citra untuk deteksi anomali visual. Hasil eksperimen menunjukkan *Deep* SVDD lebih unggul dibanding OCSVM dan autoencoder pada berbagai dataset visual. Penelitian oleh Kilickaya, et al. 2024, juga mengadaptasi pendekatan serupa pada data suara untuk mendeteksi anomali dalam sinyal audio. Penelitian Hu, *et al*. (2021), mengintegrasikan pendekatan *One Class* pada data teks dengan memperkenalkan One-Class Text Classification with Multi-modal Deep SVDD

untuk teks berita. Selain itu, penelitian Ruff *et al*. (2019), juga mengembangkan pendekatan berbasis *context vector data description* (CVDD) untuk deteksi anomali pada data teks menggunakan mekanisme *self-attention*. Temuan ini membuktikan bahwa integrasi OCC juga dapat diterapkan pada domain teks. Meskipun demikian, penerapan *Deep* SVDD secara langsung pada data teks biomedis yang memiliki struktur semantik kompleks masih relatif terbatas, terutama pada analisis hubungan antar komponen teks dalam korpus anotasi seperti GENIA.

Berdasarkan kondisi tersebut, penelitian ini mencoba mengisi celah penelitian dengan mengimplementasikan model *Deep* SVDD untuk deteksi anomali pada pasangan teks biomedis dalam *dataset* GENIA *Biomedical Event*. Representasi teks dibangun menggunakan *embedding* domain spesifik BioWordVec yang dirancang untuk menangkap hubungan semantik terminologi biomedis (Zhang *et al*., 2019). Selanjutnya, representasi tersebut diproses menggunakan *encoder* sekuens *Long Short-Term Memory* (LSTM) dan *Bidirectional Long Short-Term Memory* (BiLSTM) untuk menghasilkan representasi laten yang digunakan dalam pembentukan *hypersphere* pada model *Deep* SVDD. Melalui pendekatan ini, penelitian ini bertujuan untuk mempelajari distribusi pasangan teks biomedis sehingga penyimpangan pola semantik dapat diidentifikasi sebagai anomali. Selain itu, penelitian ini juga mengevaluasi kinerja model melalui analisis performa serta sensitivitas *threshold* deteksi anomali untuk memahami stabilitas model dalam mengidentifikasi penyimpangan distribusi data.

1.2 Rumusan Masalah

Berdasarkan latar belakang, perumusan permasalahan pada penelitian ini adalah sebagai berikut:

1. Kompleksitas hubungan semantik antara komponen teks dalam *dataset* GENIA *Biomedical Event* yang berpotensi menghasilkan penyimpangan pola distribusi data.
2. Kebutuhan akan metode deteksi anomali yang mampu mempelajari distribusi representasi pasangan teks biomedis secara efektif.
3. Penerapan model *Deep* SVDD berbasis representasi teks menggunakan embedding BioWordVec dan *encoder* berbasis *sequential* untuk mendeteksi

anomali pada pasangan teks dalam dataset GENIA *Biomedical Event*.

4. Evaluasi kinerja model *Deep* SVDD serta analisis sensitivitas *threshold* dalam mendeteksi anomali pada data teks biomedis.

1.3 Tujuan Penelitian

Adapun tujuan dari penelitian ini yaitu:

1. Menganalisis pasangan teks pada *dataset* GENIA *Biomedical Event* untuk memahami karakteristik hubungan semantik antar komponen teks.
2. Membangun representasi fitur pasangan teks menggunakan *embedding* BioWordVec dan encoder berbasis *sequential* LSTM serta BiLSTM.
3. Mengimplementasikan dan mengevaluasi kinerja model *Deep* SVDD dalam mendeteksi pasangan teks yang menyimpang dari distribusi mayoritas, sebagai anomali dengan metrik evaluasi ROC AUC.
4. Menganalisis sensitivitas *threshold* untuk menilai kemampuan model dalam mendeteksi anomali.

1.4 Manfaat Penelitian

AManfaat yang diharapkan dari penelitian ini antara lain:

1. Memberikan kontribusi pengembangan metode deteksi anomali pada data teks khususnya pada korpus GENIA *Biomedical Event* yang berasal dari literatur PubMed terkait proses biologis pada sel darah manusia dan leukemia menggunakan model *Deep* SVDD.
2. Menunjukkan potensi penggunaan *embedding* domain spesifik BioWordVec dalam meningkatkan kualitas representasi semantik teks biomedis untuk mendukung proses pembelajaran model deteksi anomali.
3. Memberikan pendekatan alternatif dalam penyaringan dan analisis informasi pada literatur biomedis, khususnya pada dataset yang berasal dari publikasi ilmiah seperti PubMed.

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Penelitian terkait yang telah dilakukan oleh peneliti sebelumnya, digunakan sebagai referensi pada penelitian ini. Topik penelitian yang menjadi referensi adalah metode pelabelan dalam tugas *Semantic Textual Similarity* (STS) dan model OCC. Berikut disajikan ringkasan dari beberapa penelitian yang menjadi referensi pada Tabel 1.

Tabel 1. Ringkasan Penelitian Terkait

No.	Penelitian	Data	Metode	Hasil
1.	*Comparative Aanalysis of Deep Siamese Models for Medical Reports Text Similarity* (Kurniasari, *et al*., 2024).	*Dataset*: *GENIA Biomedical Event Train* Jumlah: 1.409 pasang teks biomedis *unlabelled* Sumber data: Platform Kaggle dengan nama GENIA Biomedical Event Dataset yang dikembangkan oleh Nishanth Salian.	*Preprocessing*: *Lowercasing*, *Remove Punctuation*, *Stopword Removal*, *Lematization*. *Labelling*: *Cosine Similarity* (CS) dan *Word Mover's Distance* (WMD). Model: CNN, LSTM, CNN-LSTM, LSTM-CNN. .	100 (*Accuracy*)

No.	Penelitian	Data	Metode	Hasil
2.	*Deep One-Class Classification* (Ruff, *et al*., 2018).	*Dataset*: MNIST: Gambar digit tulisan tangan (28×28 grayscale), terdiri dari 10 kelas: • *Train* (per kelas): ≈ 6.000 sampel normal • *Test*: 10.000 sampel (normal dan anomali) *Dataset*: CIFAR-10: Gambar objek warna natural (32×32 RGB), terdiri dari 10 kelas: • *Train* (per kelas): 5.000 sampel normal • *Test*: 10.000 sampel (9 kelas anomali) Sumber: MNIST (LeCun *et al.*, 2010) dan CIFAR-10 (Krizhevsky & Hinton, 2009).	*Preprocessing*: *Global contrast normalization* menggunakan L1 *norm*, *Rescale* ke rentang [0, 1] dengan *min-max scaling*. *Encoder*: CNN Model: OC-SVM/SVDD, KDE, IF, DCAE, ANOGAN, *Soft-Bound Deep* SVDD, *One-Class Deep* SVDD.	**MMNIST**: Auc • OC-SVM/SVDD: 77,1- 98,6 • KDE: 73,8-98,9 • IF: 85,5-98,0 • DCAE: 78,2-98,3 • ANOGAN: 84,9-99,2 • *Soft-Bound Deep SVDD*: 85,8-99,6 • *One-Class Deep SVDD*: 88,5-99,7 **CIFAR-10**: Auc • OC-SVM/SVDD: 50,0-75,9 • KDE: 50,1-76,0 • IF: 42,9-74,2 • DCAE: 48,9-76,8 • ANOGAN: 52,9-67,1 • *Soft-Bound Deep* SVDD: 49,5-75,6 • *One-Class Deep* SVDD: 50,8-75,9

No.	Penelitian	Data	Metode	Hasil
3.	*Audio-based Anomaly Detection in Industrial Machines Using Deep One-Class Support Vector Data Description*(Kilickaya, *et al*., 2024).	DatasetI: MIMII dataset audio 10 detik dari 4 jenis mesin (Pump, Fan, Valve, dan Slide rail). • Kondisi normal: suara mesin tanpa kerusakan • Kondisi abnormal: suara mesin ketika ada kerusakan / gangguan mekanis Sumber: Repositori Zenodo	Model: *Baseline* Dense AE dan *Deep* SVDD	**Valve 6dB**: Auc • *Dense* AE: 67,0-67,3 • OC *Deep* SVDD: 77,2-**80,4** **Valve 0dB**: Auc • *Dense* AE: 61,3-62,6 • OC *Deep* SVDD: 77,4-**78,9** **Valve -6dB**: Auc • *Dense* AE: 55,5-58,9 • OC *Deep* SVDD: 71,2-**74,2** **Pump 6dB**: Auc • *Dense* AE: 80,5-81,1 • OC *Deep* SVDD: 78,2-**83,7** **Pump 0dB**: Auc • *Dense* AE: 70,5-71,9 • OC *Deep* SVDD: 75,0-**78,1**

No.	Penelitian	Data	Metode	Hasil
				Pump -6dB: Auc • *Dense* AE: 64,7-66,0 • OC *Deep* SVDD: 69,5-**70,6** **Fan 6dB**: Auc • *Dense* AE: 91,3-91,6 • OC *Deep* SVDD: 90,4-**93,6** **Fan 0dB**: Auc • *Dense* AE: 79,5-80,2 • OC *Deep* SVDD: 80,9-**83,4** **Fan -6dB**: Auc • *Dense* AE: 66,3-**67,5** • OC *Deep* SVDD: 64,7-67,4 **Slide Rail 6dB**: Auc • *Dense* AE: 87,2-**87,8** • OC *Deep* SVDD: 82,9-86,7 **Slide Rail 0dB**: Auc • *Dense* AE: 78,0-**79,5** • OC *Deep* SVDD: 77,0-79,1

No.	Penelitian	Data	Metode	Hasil
				Slide Rail -6dB: Auc • *Dense* AE: 70,0-70,5 • OC *Deep* SVDD: 66,4-**71,8** **Avg All Machines 6dB**: Auc • *Dense* AE: 81,6-81,8 • OC *Deep* SVDD: 84,3-**84,4** **Avg All Machines 0dB**: Auc • *Dense* AE: 72.3-73,6 • OC *Deep* SVDD: 78,3-**79,8** **Avg All Machines -6dB**: Auc • *Dense* AE: 64,4-65,4 • OC *Deep* SVDD: 68,9-**69,7**

Rangkuman dari penelitian pada Tabel 1. Sebagai berikut:

1. Penelitian Pertama (Kurniasari, *et al*., 2024) Penelitian ini mengembangkan

   dan mengevaluasi sejumlah model *deep learning* berbasis arsitektur *Siamese Manhattan* untuk mengukur *Semantic Textual Similarity* (STS) pada teks laporan medis. Empat model yang diuji meliputi *Siamese Manhattan CNN*, *Siamese Manhattan LSTM*, serta dua model hibrida CNN-LSTM dan

LSTM-CNN. Seluruh model memanfaatkan *word embedding* BioWordVec yang dirancang khusus untuk konteks biomedis, serta menerapkan proses *dual labelling* menggunakan *Cosine Similarity* (CS) dan *Word Mover's Distance* (WMD) untuk memberikan anotasi kesemantisan pada pasangan teks. Data yang digunakan berasal dari *GENIA Biomedical Event Train* yang berisi 4.957 laporan medis tanpa label, dan setelah tahap praproses serta pelabelan, diperoleh 1.409 pasangan teks untuk tiga skenario analisis, yaitu *sentence–TriggerWord*, *sentence–EventType*, dan *TriggerWord–EventType*. Hasil evaluasi menunjukkan bahwa seluruh model memiliki kinerja yang baik, namun model hibrida LSTM-CNN dengan arsitektur *Siamese Manhattan* menunjukkan performa paling unggul, terutama ketika menggunakan metode WMD, dengan akurasi mencapai 100% dalam mengidentifikasi kesamaan semantik teks medis.

2. Penelitian Kedua, (Ruff, *et al*., 2018) Penelitian ini memperkenalkan

   pendekatan baru untuk tugas *deep anomaly detection* yang terinspirasi dari *one class classification* (OCC) berbasis kernel dan estimasi volume minimum. Pendekatan yang dikembangkan adalah *Deep Support Vector Data Description* (*Deep* SVDD) yang bertugas untuk menemukan *hypersphere* minimum dengan pusat *center* (c). Penelitian ini dilakukan dengan 10 kali percobaan pada dataset *MNIST* dan *CIFAR-10* dengan membandingkan *Deep* SVDD dengan metode OCC lainnya yaitu OC-SVM/SVDD, KDE, IF, DCAE, ANOGAN, dan *Soft-Bound Deep* SVDD. Hasil menunjukkan bahwa pada *dataset* MNIST, model *Deep*SVDD menghasilkan nilai AUC tertinggi di hampir semua kelas MNIST. Artinya model mampu mengenali dan membedakan kelas lain (anomali) dengan sangat baik. Rata-rata AUC yang diperoleh adalah 0,95.

   Selanjutnya model juga diuji pada *dataset* CIFAR-10, dimana *dataset* ini jauh lebih sulit karena merupakan data citra berwarna dan kompleks, saling mirip antar kelas, serta memiliki ukuran dan variasi objek yang tinggi. Hasil menunjukkan bahwa AUC menurun secara signifikan dibandingkan pada *dataset* MNIST, umumnya sekitar 0,50–0,75. Tidak ada model yang sangat dominan di semua kelas, namun model *Deep* SVDD masih konsisten unggul meskipun hanya sedikit dibandingkan dengan model lain. Hasil

ini ditunjukkan dengan nilai AUC yang unggul di 4 kelas percobaan sedangkan model lain hanya unggul maksimal di 3 kelas percobaan. Hal ini menunjukkan bahwa model *Deep* SVDD tetap kompetitif bahkan di *dataset* kompleks meskipun secara umum performa model masih terbatas.

3. Penelitian Ketiga, (Kilickaya, *et al*., 2024) Penelitian yang dilakukan oleh

   Kilickaya *et al*. (2024), membahas penerapan metode *audio-based anomaly detection* untuk mengidentifikasi kerusakan pada mesin industri melalui sinyal suara. Studi ini mengevaluasi pendekatan *deep learning* berbasis *Deep One-Class Deep* SVDD dan membandingkannya dengan *dense autoencoder* sebagai metode dasar. Skenario penelitian difokuskan pada deteksi anomali tanpa label, di mana model hanya dilatih menggunakan data suara mesin dalam kondisi normal, kemudian diuji untuk membedakan suara abnormal yang menandai potensi kerusakan fungsi. Data yang digunakan berasal dari MIMII *Sound Dataset*, yaitu himpunan rekaman audio mesin industri dalam kondisi normal dan rusak. Hasil penelitian menunjukkan bahwa *Deep* SVDD, khususnya dengan dimensi laten rendah, memberikan performa terbaik dengan nilai AUC berturut-turut sekitar 0,84, 0,79, dan 0,69 pada kondisi SNR 6 dB, 0 dB, dan -6 dB. Nilai ini melampaui performa *autoencoder* yang memperoleh AUC 0,82, 0,72, dan 0,64 pada kondisi serupa. Selain itu, *Deep* SVDD terbukti lebih efisien secara komputasi karena membutuhkan parameter pelatihan yang jauh lebih sedikit. Temuan ini menegaskan bahwa *Deep* SVDD merupakan metode yang efektif dan efisien untuk deteksi anomali berbasis audio pada mesin industri.

2.2 *Natural Language Preprocessing*

Natural Language Processing (NLP) merupakan cabang ilmu yang berfokus pada pengembangan metode dan teknologi yang memungkinkan komputer untuk memahami serta memproses bahasa manusia, baik dalam bentuk teks maupun ujaran. Fokus utama NLP adalah mempelajari cara manusia memahami dan menggunakan bahasa sehingga dapat dikembangkan sistem komputasi yang mampu berinteraksi dengan bahasa alami secara efektif dan efisien. Bidang ini bersifat multidisipliner karena mengintegrasikan berbagai disiplin ilmu, antara lain ilmu

komputer, linguistik, matematika, teknik elektro, kecerdasan buatan, dan psikologi. Aplikasi NLP sangat beragam, mencakup penerjemahan otomatis, analisis dan peringkasan teks, antarmuka pengguna berbasis bahasa alami, pencarian serta pengambilan informasi dari teks dalam berbagai bahasa, sistem pengenalan ucapan, serta pengembangan sistem pakar berbasis kecerdasan buatan. (Chowdhary, 2020).

Menurut Jain, *et al*. (2018), NLP dilatih untuk mempelajari struktur sintaksis dan makna bahasa manusia, sehingga mampu memproses informasi dan menghasilkan keluaran yang relevan sesuai konteks. Bidang ini berfokus pada pengembangan sistem yang dapat memahami serta mengeksekusi tugas-tugas bermakna menggunakan bahasa alami yang dapat dipahami manusia. Signifikansi NLP di masa depan terletak pada kemampuannya untuk membangun model dan proses yang dapat menerima serta memanipulasi informasi dalam bentuk teks, suara, atau kombinasi keduanya, berdasarkan algoritma yang tertanam dalam sistem komputasi.

2.3 *Text Mining*

Text mining merupakan suatu proses sistematis untuk mengekstraksi pengetahuan atau informasi yang bernilai dari data berbasis teks. Istilah ini juga dikenal sebagai *text data mining*, yaitu upaya untuk memperoleh informasi dengan kualitas tinggi melalui analisis terhadap konten tekstual. *Text mining* sering disebut pula sebagai *Knowledge Discovery in Text* (KDT) atau *Text Data Mining* (TDM). Secara historis, perkembangan text mining tergolong relatif baru dan banyak mengadopsi pendekatan yang berasal dari berbagai disiplin ilmu seperti *Information Extraction*, *Information Retrieval*, *Natural Language Processing* (NLP), *Computational Linguistics*, statistik, *data mining*, serta *machine learning*. Kajian dalam bidang ini mencakup pengembangan beragam metode berbasis linguistik, matematika, pembelajaran mesin, statistik, dan pengenalan pola, yang memungkinkan dilakukannya analisis otomatis terhadap informasi yang bersifat tidak terstruktur maupun semi-terstruktur, sehingga sistem mampu mengekstraksi data yang relevan secara efektif dari sumber teks yang tersedia (Vel, 2021).

2.4 Klasifikasi Teks

Klasifikasi teks merupakan proses pemberian label kategori pada dokumen atau teks berdasarkan kontennya dengan menggunakan model pembelajaran mesin yang mampu mengenali pola dalam teks untuk menghasilkan keputusan kategoris (Hanum, *et al*., 2024). Klasifikasi teks merupakan bentuk *supervised learning*, di mana model pembelajaran mesin dilatih dengan data berlabel untuk mengelompokkan dokumen ke dalam kelas-kelas tertentu yang telah ditentukan sebelumnya (Khoirunnisaa, *et al*., 2024). Menurut Yadla & Rao (2020), klasifikasi teks secara matematis dapat dinyatakan sebagai fungsi pemetaan yang memetakan teks menjadi fitur dan mengelompokkannya ke dalam kategori label berdasarkan informasi dari fitur, ditunjukkan pada Persamaan (1):

$$f : X \rightarrow Y \tag{1}$$

dengan:

X = ruang fitur (vektor representasi teks),
Y = ruang label (kategori atau kelas).

Menurut Jha, *et al*.(2019), tugas klasifikasi terbagi menjadi dua jenis utama, yaitu klasifikasi biner dan klasifikasi multi-kelas. Klasifikasi biner, bertugas untuk memisahkan data menjadi dua kelas berbeda untuk menentukan kelas mana yang tepat bagi setiap entri. Sementara pada klasifikasi multi-kelas, data dikelompokkan ke dalam lebih dari dua kelas berdasarkan aturan klasifikasi tertentu. Perbedaan antara kedua jenis klasifikasi tersebut ditunjukkan pada Gambar 1.

Klasifikasi biner

X_1	X_1	…	X_m	Y
X_{11}	X_{12}	…	X_{1m}	1
X_{21}	X_{22}	…	X_{2m}	0
…	…	…	…	1
X_{n1}	X_{n2}	…	X_{nm}	0

Klasifikasi multi-kelas

X_1	X_1	…	X_m	Y
X_{11}	X_{12}	…	X_{1m}	C_1
X_{21}	X_{22}	…	X_{2m}	C_2
…	…	…	…	C_3
X_{n1}	X_{n2}	…	X_{nm}	C_4

Gambar 1. Jenis Klasifikasi Teks (Mohammed, *et al*., 2020).

Salah satu tantangan signifikan dalam klasifikasi teks adalah ketidakseimbangan data, yaitu ketika jumlah data dalam satu kelas jauh melebihi kelas lainnya (Rahmah & Suadaa, 2020). Pada beberapa kasus ekstrem, distribusi label bisa sangat timpang hingga kelas minoritas hanya mewakili sebagian sangat kecil dari keseluruhan

data. Sehingga metode tradisional untuk mengatasi ketidakseimbangan ini seperti *oversampling* atau *undersampling* sering kali tidak efektif atau tidak mungkin dilakukan.

2.5 *One Class Classification*

One Class Classification (OCC) merupakan jenis klasifikasi khusus yang hanya menggunakan data dari satu kelas (kelas target) untuk membangun model. OCC secara dinamis membedakan *instance* baru berdasarkan kedekatan dengan profil kelas tersebut dan tidak bergantung pada keberadaan kelas lain. Pendekatan ini diterapkan ketika data minoritas terlalu terbatas atau ketika hanya satu kelas yang diketahui secara pasti (Seliya, *et al*., 2021). Menurut Hayashi, *et al*. (2024), OCC secara konseptual berbeda dari klasifikasi tradisional karena hanya memodelkan satu kelas dan berfokus pada identifikasi apakah suatu instance termasuk ke dalam kelas tersebut atau tidak.

Menurut Seliya, *et al*. (2021), salah satu alasan utama penggunaan OCC adalah kemampuannya dalam mengidentifikasi objek yang tidak normal, data yang menyimpang, atau pola yang bersifat mencurigakan. Berdasarkan tinjauan penelitian yang mereka lakukan, OCC paling banyak diterapkan pada kasus deteksi *outlier* dan deteksi kebaruan (*novelty detection*). Kedua konsep ini memiliki perbedaan penting. Pada *novelty detection*, data pelatihan dianggap bersih dan hanya berisi contoh normal, sehingga model bertugas mengenali contoh anomali yang muncul saat pengujian. Sebaliknya, pada deteksi *outlier*, data pelatihan dapat mengandung campuran titik normal dan anomali, dan tujuan model adalah mempelajari batas pemisah antara kedua jenis data tersebut. Batas yang telah dipelajari ini kemudian digunakan untuk mengklasifikasikan titik data pada tahap pengujian, yang juga dapat memuat data normal maupun anomali.

Menurut Challa (2024), arsitektur model dasar dan strategi penanganan data pada OCC dibagi menjadi empat kategori berbeda:

a. Pendekatan Berbasis Rekontruksi

   Pendekatan rekonstruksi memanfaatkan *autoencoder* untuk membangun kembali pola data normal. Nilai kesalahan rekonstruksi yang tinggi menunjukkan adanya

penyimpangan signifikan dari pola normal yang menandakan adanya anomali. Metode ini sangat ideal untuk jenis data terstruktur, termasuk deret waktu dan data tabular, yang memiliki pola normal yang terdefinisi dengan baik. Model yang digunakan pada pendekatan ini diantaranya adalah PCA, *Latent-Insensitive Autoencoder* (LIS-AE), dan model *progressive reconstruction and hierarchical feature fusion* (PRFF-AD).

b. *Variational Autoencoder* (VAE)

Variational Autoencoder merupakan pengembangan dari pendekatan rekonstruksi yang memodelkan distribusi berdasarkan probabilitas data. Anomali diidentifikasi berdasarkan *likelihood* yang rendah, artinya data input berada di luar cakupan distribusi data normal yang telah dipelajari. Keunggulan VAE terletak pada kemampuannya untuk menangani data deret waktu yang rumit dan distribusi multimodal, yang memerlukan representasi distribusi normal yang teliti. Model yang digunakan diantaranya adalah *Variational Autoencoder*, VESC, menggabungkan VAEs dan *Normalizing Flows*, pendekatan *one class* VAE yang disebut OC-FakeDECT dan *Variational Autoencoder* berbasis LSTM.

c. Pendekatan Konvolusional Pendekatan berbasis konvolusi menggunakan lapisan

konvolusional untuk mengekstraksi karakteristik spasial dari data, kemudian mendeteksi penyimpangan dari pola normal yang telah dipelajari. Keunggulan ini membuat metode konvolusional sangat efektif dalam mendeteksi anomali, terutama pada data visual seperti citra maupun data multimodal. Model yang digunakan diantaranya adalah model One-Class Learned Encoder-Decoder (OLED), metode OCC berbasis CNN ini menggunakan data *pseudo*-negatif yang dibangkitkan dari distribusi Gaussian di dalam ruang fitur, dan deteksi anomali deret waktu dengan metode berbasis CNN 2D.

d. Metode *Hybrid*

Pendekatan *hybrid* mengombinasikan berbagai teknik OCC atau mengintegrasikan metode konvensional dengan teknik *deep learning* untuk memanfaatkan keunggulan dari masing-masing metode. Contoh model yang digunakan pada pendekatan *hybrid* ini adalah menggabungkan *Convolutional Neural Network* dengan *transformer backbone*, mengkombinasikan ekstraksi fitur berbasis *autoencoder* dengan *support vector data description*, penggabungan beberapa model klasifikasi satu kelas guna meningkatkan performa dan akurasi

deteksi OCC, dan pendekatan ansambel berbasis klaster untuk OCC dilakukan dengan membagi data ke dalam beberapa klaster, lalu menerapkan OCSVM secara terpisah pada masing-masing klaster.

2.6 *Preprocessing*

Tahap preprocessing merupakan tahap pertama pada text mining yang menentukan kualitas dari data (Hickman et al., 2022). Menurut Woo et al. (2020), tanpa preprocessing yang sesuai, algoritma dapat menghasilkan perilaku yang tidak konsisten akibat data yang tidak rapi, sehingga menurunkan performanya. Model kalimat bekerja dengan data teks berbasis kata yang dapat mencakup bentuk jamak, karakter khusus, hingga angka. Untuk itu, tahap preprocessing dibagi menjadi dua bagian utama, yaitu transformasi dan eliminasi. Transformasi mengubah teks mentah menjadi representasi berbasis kata sedangkan eliminasi menghapus kata-kata yang tidak relevan terhadap pemahaman makna. Secara rinci, proses preprocessing pada teks meliputi langkah-langkah berikut.

a. *Lowercase*

   Lowercase merupakan teknik yang digunakan untuk mengonversi seluruh teks dalam korpus menjadi huruf kecil. Teknik ini diperlukan karena komputer membedakan antara huruf besar dan huruf kecil, sehingga dua kata yang sama tetapi berbeda kapitalisasi akan dianggap berbeda jika tidak diseragamkan. Mengubah seluruh teks menjadi huruf kecil umumnya bermanfaat, karena dapat menurunkan jumlah fitur yang perlu diproses sehingga meningkatkan efisiensi analisis tanpa mengorbankan keakuratan makna (Hickman *et al*., 2022).

b. *Remove punctuation*

   Remove punctuation merupakan teknik pembersihan teks yang meliputi penghapusan tanda baca, simbol, atau angka. Karakter *non-alfabet* yang tidak ditangani dalam proses penambangan teks, bisa dianggap sebagai unit yang berbeda jika tidak diproses dengan benar saat tokenisasi. Oleh karena itu, langkah ini dilakukan untuk memastikan fokus analisis tetap pada kata dan frasa penting dalam teks (Hickman, *et al*., 2022).

c. *Tokenization*

Tokenization merupakan tahap *preprocessing* yang bertujuan membagi teks menjadi unit-unit bermakna seperti kata, frasa, atau angka (Narayanasamy *et al*., 2022).

d. *Lemmatization*

Lemmatization adalah teknik mengubah kata menjadi bentuk dasarnya dengan memperhatikan kelas kata, seperti kata kerja atau kata benda, serta menggunakan kamus bentuk kata dan turunannya untuk menghasilkan bentuk dasar (lemma). Berbeda dengan *stemming*, *lemmatization* hanya mengubah kata ketika makna atau bentuk dasarnya sama misalnya, organ dan organs sama-sama dikembalikan menjadi organ. Namun, kata seperti organic atau organism tetap dibiarkan seperti aslinya karena meskipun memiliki akar kata yang mirip, makna dan jenis katanya berbeda (Hickman et al., 2022).

2.7 BioWordVec

BioWordVec merupakan *pretrained word embedding* biomedis yang dikembangkan oleh Zhang, *et al*. (2019), untuk menghasilkan representasi kata yang lebih akurat dalam domain medis. *Embedding* ini dilatih menggunakan dua sumber utama, yaitu literatur biomedis dan pengetahuan terstruktur dari *Medical Subject Headings* (MeSH). Proses pembentukannya melibatkan pembangunan MeSH *term graph* berbasis data MeSH RDF, dilanjutkan dengan teknik random sampling untuk menghasilkan urutan istilah yang merepresentasikan struktur konseptual dalam MeSH. Urutan tersebut kemudian dipelajari secara bersamaan dengan urutan teks biomedis melalui model *embedding* berbasis *subword*, yang memungkinkan BioWordVec memahami variasi morfologi kata dan istilah teknis dengan lebih baik. Hasil evaluasi menunjukkan bahwa BioWordVec unggul dalam berbagai tugas BioNLP.

BioWordVec dibagi menjadi dua jenis evaluasi embedding yang berbeda yaitu BioWordVec *intrinsic* dan BioWordVec *extrinsic* yang masing-masing dioptimalkan untuk jenis tugas yang berbeda. Berikut adalah penjelasan dari tugas masing-masing embedding.

a. BioWordVec *intrinsic* yang digunakan untuk menghitung atau memprediksi

kemiripan semantik antar kata, istilah, maupun kalimat.

b. BioWordVec *extrinsic*, berfungsi sebagai fitur input untuk berbagai aplikasi NLP lanjutan, seperti ekstraksi relasi dan klasifikasi teks.

2.8 *Pseudo Labelling*

Labelling atau *data annotation* mengacu pada kegiatan memberikan label, kategori, atau informasi tambahan pada data mentah untuk meningkatkan kinerja dari *machine learning* (Tan, *et al*., 2024). Menurut Teloni, *et al*. (2020), berbagai metode dapat digunakan untuk melakukan anotasi data. Metode manual baik dilakukan oleh tenaga internal, penyedia jasa eksternal, maupun melalui *crowdsourcing* sering kali menjadi pilihan awal, tetapi pendekatan ini cenderung memakan waktu dan biaya yang signifikan. Sebagai solusi, pendekatan pelabelan otomatis menjadi alternatif. Salah satu adalah pendekatan *pseudo-labeling*, yaitu teknik untuk menghasilkan label secara otomatis berdasarkan karakteristik data tanpa memerlukan anotasi manual eksplisit. Pada pembelajaran semi-supervised, *pseudo-labeling* memungkinkan penggunaan data tidak berlabel dengan memanfaatkan prediksi atau estimasi model untuk menghasilkan label sementara (Zhang *et al*., 2021). Pada tugas data pasangan kalimat STS, pseudo-labeling umumnya dilakukan dengan mengukur tingkat kemiripan antar pasangan teks. Nilai similarity tersebut digunakan untuk mengidentifikasi hubungan semantik antar data, yang kemudian dikonversi menjadi label diskrit melalui mekanisme *thresholding*. Pendekatan ini memungkinkan pembentukan label secara otomatis yang merepresentasikan tingkat relevansi antar pasangan teks.

2.8.1 Word Mover's Distance (WMD)

WMD merupakan ukuran kesamaan teks yang menentukan biaya minimum yang diperlukan untuk mentransportasikan representasi kata (*word embeddings*) dari satu dokumen ke dokumen lainnya (Jhonson, 2022). Menurut Wei, *et al*. (2022), jarak antar *embedding* kata dihitung menggunakan jarak Euclidean pada Persamaan (2):

$$c(i,j) = \|x_i - x_j\|_2 \tag{2}$$

dengan:

$c(i, j)$ = biaya transport untuk memindahkan satu unit ”makna” dari kata i ke kata j,
x_i = embedding dari kata i,
x_j = embedding dari kata j.

Pada tahap berikutnya, WMD menetapkan nilai *flow* untuk setiap kata. Nilai *flow* (f_i))tersebut merepresentasikan frekuensi kemunculan kata dalam suatu teks yang telah dinormalisasi sehingga mencerminkan proporsi relatif kata tersebut dalam dokumen (Wei, *et al*., 2022). Secara matematis dapat dinyatakan pada Persamaan (3) dan (4):

$$f_i = \frac{\text{count}(i)}{|f|} \tag{3}$$

$$|f| = \sum_i \text{count}(i) \tag{4}$$

dengan:

$\text{count}(i)$ = jumlah kemunculan kata i,
f = jumlah seluruh kata dalam dokumen,
f_i = proporsi kemunculan kata i dalam dokumen yang telah dinormalisasi.

Menurut Kusner, *et al*. (2015), WMD menghitung biaya terkecil yang diperlukan untuk mentransformasikan distribusi kata dari satu dokumen ke dokumen lainnya. Secara matematis, metode ini memformulasikan proses tersebut sebagai masalah optimasi linear yang bertujuan meminimalkan total biaya perpindahan antarkata. Proses minimisasi tersebut ditunjukkan secara jelas pada Persamaan (5) hingga (7).

$$\min_{T \geq 0} \sum_{i \in I} \sum_{j \in J} T_{ij}\, c(i, j) \tag{5}$$

dengan syarat

$$\sum_{j \in J} T_{ij} = f_i, \quad \forall i \in I \tag{6}$$

$$\sum_{i \in I} T_{ij} = f'_j, \quad \forall j \in J \tag{7}$$

di mana:

T_{ij} = jumlah *flow* yang dipindahkan dari kata i menjadi kata j,
$c(i, j)$ = jarak antara embedding kata i dan kata j,
f_i = frekuensi kata i yang telah dinormalisasi pada dokumen pertama,
f'_j = frekuensi kata j yang telah dinormalisasi pada dokumen kedua.

2.8.2 *Syntax-aware Word Mover's Distance* (SynWMD)

Menurut Wei *et al*. (2022), *Syntax-aware Word Mover's Distance* (SynWMD) dikembangkan sebagai pengembangan dari WMD dengan menambahkan informasi sintaksis ke dalam proses penentuan flow dan pengukuran jarak antar kata. Metode ini terdiri dari dua komponen inti, yakni *Syntax-aware Word Flow* (SWF) dan *Syntax-aware Word Distance*(SWD), yang bersama-sama memungkinkan perhitungan jarak antar teks menjadi lebih sensitif terhadap struktur sintaksis.

a. *Syntax-aware Word Flow* (SWF) SWF dikembangkan untuk meningkatkan kualitas pembobotan kata dengan memanfaatkan informasi hubungan sintaktis dalam kalimat. Proses perhitungannya diawali dengan melakukan *dependency parsing* pada seluruh kalimat dalam dataset. Selanjutnya, *ko-occurence* kata dihitung berdasarkan jarak (*hop*) pada *dependency parse tree*, di mana semakin besar jaraknya, bobot *ko-occurence* yang diberikan semakin kecil. Informasi ini kemudian digunakan untuk membangun weighted graph yang merepresentasikan keterhubungan sintaktis antar kata. Melalui algoritma *PageRank*, diperoleh importance *score* bagi setiap kata. Nilai *word flow* ditetapkan sebagai *invers* dari skor *PageRank* tersebut sehingga kata yang lebih informatif akan memiliki *flow* yang lebih besar. Secara matematis, proses penghitungan SWF direpresentasikan pada Persamaan (8) hingga (9) (Wei, *et al*., 2022).

$$PR(i) = (1 - d) + d \sum_{j=1}^{X} \frac{w_{ij}}{\sum_{k=1}^{X} w_{jk}} PR(j) \tag{8}$$

$$f_i = \frac{1}{PR(i)} \tag{9}$$

dengan:

w_{ij} = bobot sisi (*edge weight*) antara kata i dan kata j,
$PR(i)$ = nilai PageRank yang merepresentasikan tingkat kepentingan kata i dalam graf kata,
d = parameter *damping factor* yang mengatur tingkat *smoothness* dalam penyebaran nilai PageRank,
f_i = nilai *word flow* yang diperoleh sebagai invers dari nilai PageRank kata i.

b. *Syntax-aware Word Distance* (SWD) SWD menghitung jarak antara dua kata dengan mengombinasikan jarak dasar embedding kata dan perbedaan struktur sintaktis yang menyertai setiap kata. Jarak awal diperoleh dari *cosine distance* antara embedding kata, sebagaimana ditunjukkan pada Persamaan (10) (Wei, *et al*., 2022).

$$\text{dist}(v_i, v_j) = 1\frac{\langle v_i, v_j \rangle}{\|v_i\| \cdot \|v_j\|} \tag{10}$$

dengan:

v_i = *word embedding* untuk kata i,
v_j = *word embedding* untuk kata j.

Selanjutnya, SWD menambahkan komponen yang merepresentasikan kesenjangan sintaktis antar kata berdasarkan subtree pada *dependency parse tree*. Setiap kata memiliki kumpulan *subtree* yang berisi kata tersebut. Untuk mengukur kedekatan struktur sintaktis, rumus pada Persamaan (11) menghitung rata-rata jarak antar seluruh pasangan *subtree* dari kedua kata (Wei, *et al*., 2022).

$$c(i,j) = \text{dist}(v_i, v_j) + a\,\frac{\sum_{s_i \in S_i} \sum_{s_j \in S_j} \text{dist}(s_i, s_j)}{|S_i|\,|S_j|} \tag{11}$$

di mana:

$c(i, j)$ = jarak antara kata i dan kata j,
$\text{dist}(v_i, v_j)$ = jarak awal antara embedding kata i dan kata j,
S_i = himpunan subtree sintaktis yang berkaitan dengan kata i pada *dependency parse tree*,
S_j = himpunan subtree sintaktis yang berkaitan dengan kata j pada *dependency parse tree*,
$\text{dist}(s_i, s_j)$ = jarak antara embedding subtree sintaktis s_i dan s_j,
a = parameter yang mengatur besarnya bobot informasi sintaktis dalam perhitungan jarak.

2.8.3 Aproksimasi WMD dan Informasi Sintaksis

WMD menjadi metode yang umum digunakan dalam pengkuran *similarity* untuk menentukan hubungan semantik antar pasangan data. Metode ini terbukti efektif dalam menangkap kesamaan makna antar teks melalui pemanfaatan informasi semantik dari *word embedding*. Namun demikian, WMD memiliki keterbatasan utama berupa kompleksitas komputasi yang tinggi akibat penggunaan optimasi *optimal transport*. Untuk mengatasi hal tersebut, beberapa penelitian mengusulkan pendekatan seperti *Relaxed Word Mover's Distance* (RWMD), yang menggantikan optimasi global dengan pencarian jarak minimum antar kata (Atasu *et al*., 2017). Pendekatan ini sering disebut *greedy matching* dalam literatur, di mana setiap kata dari teks pertama dipasangkan dengan kata terdekat dari teks kedua berdasarkan jarak *embedding*. Pendekatan *greedy matching* mengabaikan matriks transportasi penuh dan menggantinya dengan pasangan kata jarak minimum.

$$\sum_{w_i \in X} \min_{w_j \in Y} d(w_i, w_j) \tag{12}$$

dengan:

$d(w_i, w_j)$ = jarak embedding antara kata i dan kata j,
X = himpunan kata pada dokumen pertama,
Y = himpunan kata pada dokumen kedua.

Penelitian oleh Wei *et al*. (2022), yang mengintegrasikan struktur sintaksis melalui *dependency parsing* dalam pengukuran *similarity* menunjukkan bahwa hubungan antar kata dalam struktur kalimat dapat meningkatkan akurasi *similarity*. Temuan

ini mengindikasikan bahwa informasi sintaksis dapat digunakan sebagai faktor tambahan untuk memperkaya representasi semantik berbasis embedding karena memiliki peran penting karena mampu menggambarkan fungsi gramatikal kata dalam suatu kalimat, sehingga dapat membantu dalam memahami struktur dan makna teks secara lebih akurat. Namun, penggunaan *dependency parsing* memiliki kompleksitas yang relatif tinggi dan sensitif terhadap variasi teks, khususnya pada data dengan panjang kalimat yang pendek. Oleh karena itu, beberapa penelitian mengusulkan pendekatan yang lebih sederhana dengan memanfaatkan kategori sintaksis seperti *Part-of-Speech* (POS) sebagai representasi informasi linguistik yang lebih ringan namun tetap informatif. Penelitian oleh Chiche & Yitagesu (2022), dalam "*Part of Speech Tagging: A Systematic Review of Deep Learning and Machine Learning Approaches*", menunjukkan bahwa POS tagging merupakan komponen fundamental dalam NLP yang digunakan untuk mengidentifikasi peran kata seperti *noun*, *verb*, dan *adjective* dalam konteks kalimat.

Kategori POS digunakan untuk membedakan tingkat kepentingan kata. Kata seperti noun dan verb umumnya membawa informasi semantik utama, sedangkan kata seperti *conjunction* atau *determiner* cenderung memiliki kontribusi yang lebih rendah terhadap makna kalimat (Chiche & Yitagesu, 2022). Prinsip *Inverse Document Frequency* (IDF) yang menyatakan bahwa elemen yang lebih jarang muncul memiliki kandungan informasi yang lebih tinggi, digunakan untuk merepresentasikan tingkat kepentingan tersebut secara kuantitatif (Septiani & Isabela, 2023). Secara matematis, bobot POS dirumuskan pada Persamaan 13.

$$w_{\text{pos}}(t) = \frac{\frac{1}{f(t)}}{\max\left(\frac{1}{f}\right)} \tag{13}$$

dengan:

$w_{\text{pos}}(t)$ = bobot untuk kategori POS t,
$f(t)$ = frekuensi relatif kemunculan POS t dalam dataset.

Pendekatan ini memberikan bobot lebih besar pada kategori POS yang lebih jarang muncul, sehingga kata yang dianggap lebih informatif akan memiliki pengaruh yang lebih besar dalam perhitungan *similarity*. Sebaliknya, kata dengan frekuensi tinggi akan memiliki bobot lebih kecil, sehingga kontribusinya terhadap *similarity*

menjadi terbatas. Integrasi pembobotan POS berbasis *inverse frequency* ini bertujuan untuk meningkatkan sensitivitas metode *similarity* terhadap kata-kata yang memiliki makna utama, sekaligus mengurangi pengaruh kata yang kurang informatif. Dengan demikian, pendekatan ini dapat menghasilkan representasi yang lebih stabil dan relevan terhadap makna semantik teks, terutama dalam konteks pembentukan *pseudo-label* yang digunakan untuk mendukung model pembelajaran berbasis distribusi.

2.9 *Data Splitting*

Data splitting merupakan teknik untuk memisahkan suatu dataset ke dalam beberapa subset yang berbeda. *Data splitting* diperlukan karena apabila seluruh data dipakai dalam proses training, model berisiko menyesuaikan diri terlalu kuat terhadap data tersebut (*overfitting*), yang pada akhirnya menyebabkan buruknya kemampuan prediksi pada situasi nyata. Dengan memisahkan sebagian data untuk pengujian, kualitas generalisasi model dapat diuji sebelum implementasi, sehingga berbagai konsekuensi negatif akibat *overfitting* dapat dihindari (Joseph & Vakayil, 2022). Menurut Kurniasari, *et al*. (2024), rasio pembagian 75/25 lazim digunakan dalam bidang machine learning karena memberikan keseimbangan antara jumlah data yang memadai untuk melatih model serta cukupnya data untuk melakukan pengujian dan validasi performa secara ketat.

Secara umum, *splitting* dibagi menjadi tiga bagian yaitu *training set*, *testing set*, dan *validation set*. *Training set* (data pelatihan) dimanfaatkan untuk membangun dan menyesuaikan model, yaitu dengan mempelajari pola dari data sehingga model dapat menentukan nilai parameter yang paling tepat (Joseph & Vakayil, 2022). *Testing set* (data pengujian) digunakan untuk menilai sejauh mana model mampu melakukan prediksi pada data yang belum pernah dilihat sebelumnya. Tahap ini memberikan gambaran objektif tentang akurasi dan kemampuan generalisasi model (Joseph & Vakayil, 2022). *Validation set* (data validasi) berperan dalam proses penyetelan model, terutama untuk mengidentifikasi konfigurasi *hyperparameter* yang paling sesuai serta menentukan tingkat regularisasi yang tepat. Tahap ini memastikan bahwa model mencapai kinerja optimal tanpa mengalami *overfitting*, sehingga mampu mempertahankan stabilitas performa pada data baru (Joseph & Vakayil, 2022).

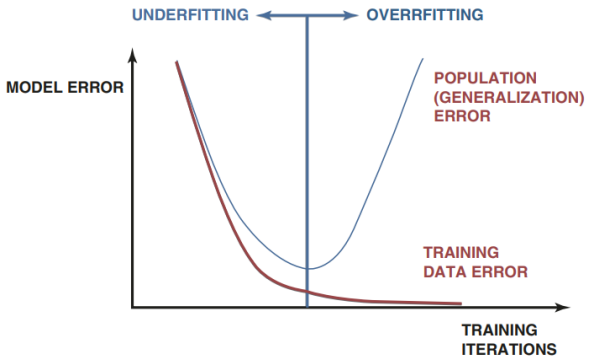
2.10 *Hyperparameter Tuning*

Hyperparameter merupakan serangkaian parameter yang harus ditetapkan sebelum algoritma pembelajaran diterapkan pada suatu dataset (Afaq & Rao, 2020). *Hyperparameter tuning* dilakukan untuk menentukan kombinasi nilai *hyperparameter* yang menghasilkan performa terbaik dari suatu model. Proses ini melibatkan pengujian berbagai konfigurasi secara berulang hingga diperoleh nilai yang paling optimal (Matin, 2023). Menurut Kumar, *et al*. (2025), pengaturan *hyperparameter* mencakup berbagai aspek, seperti laju pembelajaran (*learning rate*), *batch size*, jumlah *epoch*, serta parameter regularisasi.

Learning rate (LR) merupakan parameter penting yang mengatur seberapa besar perubahan bobot model pada setiap iterasi selama pelatihan. Nilai LR yang terlalu tinggi dapat mempercepat konvergensi namun berpotensi menyebabkan ketidakstabilan, sedangkan LR yang terlalu rendah membuat proses pelatihan lebih lama dan rawan terjebak pada nilai loss rendah tetapi bukan terendah secara global (Kumar, *et al*., 2025). *Batch size* merupakan jumlah sampel pelatihan yang diproses sebelum model melakukan pembaruan parameter, dan menjadi elemen penting dalam *stochastic gradient descent* (SGD) serta variannya. Penggunaan *batch* kecil menghasilkan estimasi gradien yang lebih presisi sehingga dapat meningkatkan generalisasi, tetapi menimbulkan varians pembaruan yang lebih tinggi. Sebaliknya, *batch* besar menurunkan varians dan mendorong konvergensi yang lebih cepat dan stabil, namun membutuhkan kapasitas memori lebih besar serta berpotensi menurunkan kemampuan generalisasi. Pertimbangan utama dalam pemilihan *batch size* meliputi keterbatasan sumber daya komputasi serta efisiensi pelatihan, karena batch kecil mempercepat iterasi tetapi umumnya memerlukan lebih banyak epoch untuk mencapai konvergensi (Kumar, *et al*., 2025).

Jumlah *epoch* merupakan banyaknya putaran pelatihan yang dilakukan terhadap seluruh dataset selama proses *training*, di mana setiap epoch memastikan seluruh sampel digunakan sekali untuk memperbarui parameter model. Penentuan jumlah *epoch* yang tepat berkaitan erat dengan dua isu utama dalam pelatihan model, yaitu *underfitting* dan *overfitting* (Afaq & Rao, 2020). Jika *epoch* terlalu sedikit, model berisiko mengalami *underfitting* karena tidak mampu mempelajari pola yang relevan dari data. Sebaliknya, penggunaan *epoch* yang terlalu banyak

dapat menimbulkan *overfitting*, ditandai dengan model yang menghafal noise dan karakteristik spesifik data pelatihan sehingga kurang mampu melakukan generalisasi (Kumar, *et al*., 2025). Ilustrasi *underfitting* dan *overfitting* disajikan pada Gambar 2.



Gambar 2. Ilustrasi *Underfitting* dan *Overfitting* (Alferis & Simon, 2024).

Selanjutnya adalah parameter regulasisasi yang digunakan dalam mengurangi *overfitting* pada model *deep learning* dengan memberikan batasan atau penalti selama proses pelatihan. *Dropout* dan *Weight Decay* (L2 *Regularization*) merupakan dua parameter yang umum diterapkan pada teknik ini (Kumar, *et al*., 2025).

a. *Dropout* merupakan teknik regularisasi yang bekerja dengan menonaktifkan secara acak sebagian unit tersembunyi selama pelatihan (Liang, *et al*., 2021). Secara umum, *dropout* berkisar antara 0,2 sampai 0,5.

b. *Weight Decay* merupakan teknik yang bekerja dengan menambahkan komponen penalti pada fungsi *loss*, di mana besarannya sebanding dengan kuadrat nilai bobot, sehingga mendorong model mempertahankan bobot yang lebih kecil dan stabil (Kumar, *et al*., 2025).

Setiap *hyperparameter* memainkan peran penting dalam menentukan bagaimana model belajar dan sejauh mana kinerjanya dapat dioptimalkan. Untuk itu, diperlukan metode yang tepat untuk mencari kombnasi terbaik dari masing-masing parameter yang akan diterapkan pada model pembelajaran. Penelitian oleh Suryadi, *et al*. (2024), menjelaskan beberapa teknik *hyperparameter tuning* diantaranya adalah *Grid Search*, *Random Search*, *Optuna*, *Bayesian* dengan *Hyperopt*, *Hyperband*, *Tree Parzen Estimators* dan *Nevergrad*.

a. *Grid Search*

Grid Search adalah metode pencarian sistematis yang memanfaatkan serangkaian nilai parameter yang telah ditetapkan untuk mengidentifikasi kombinasi *hyperparameter* paling optimal. Teknik ini mengevaluasi seluruh konfigurasi yang tersedia guna menentukan nilai parameter yang memberikan kinerja terbaik, khususnya dalam hal akurasi dan AUC.

b. *Random Search*

Random Search merupakan metode optimasi hyperparameter dengan memilih nilai-nilai parameter secara acak dari ruang pencarian yang telah ditentukan. Pendekatan ini dinilai lebih efisien, terutama pada ruang pencarian berdimensi tinggi, karena tidak mewajibkan evaluasi seluruh kombinasi *hyperparameter* yang mungkin.

c. Optuna

Optuna melakukan proses *hyperparameter tuning* dengan mengoptimalkan *objective function* yang menerima sejumlah hyperparameter sebagai input dan menghasilkan skor validasi sebagai indikator kinerja. Melalui serangkaian *trial* dalam suatu *study*, Optuna mengevaluasi berbagai kombinasi *hyperparameter* secara iteratif untuk memperoleh konfigurasi paling optimal sesuai metrik performa yang ditetapkan.

d. *Bayesian Search* dengan *Hyperopt*

Bayesian Optimization memilih kombinasi *hyperparameter* terbaik untuk evaluasi selanjutnya dengan memanfaatkan informasi dari hasil evaluasi sebelumnya, melalui proses pembaruan distribusi posterior dan pemaksimalan *acquisition function*. Pada *Bayes Search*, setiap parameter diberi peluang awal (*prior*) yang kemudian dikombinasikan dengan distribusi fungsi evaluasi untuk menghitung probabilitas ditemukannya hasil yang lebih optimal berdasarkan himpunan *hyperparameter* yang diuji.

e. *Hyperband*

Hyperband merupakan metode optimasi *hyperparameter* yang menerapkan pendekatan bandit untuk mendistribusikan sumber daya secara bertahap kepada berbagai konfigurasi *hyperparameter* yang dipilih secara acak. Metode ini

menghasilkan sejumlah titik uji, memberikan alokasi sumber daya yang setara pada masing-masing konfigurasi, dan kemudian menilai kinerjanya untuk mengidentifikasi pengaturan *hyperparameter* yang paling menjanjikan.

f. *Tree Parzen Estimator*

Tree Parzen Estimator (TPE) merupakan metode optimasi yang memanfaatkan rekam jejak percobaan dan ruang pencarian *hyperparameter* untuk memprediksi nilai parameter yang paling menjanjikan pada iterasi berikutnya. Teknik ini bekerja dengan memodelkan distribusi sampel bernilai baik dan membandingkannya dengan sampel lain, sehingga dapat memilih kombinasi *hyperparameter* yang memiliki peluang terbesar menghasilkan kinerja terbaik.

g. *Nevergrad*

Nevergrad merupakan platform optimasi *derivative-free* yang menyediakan beragam algoritma optimasi serta himpunan fungsi uji untuk menilai kinerjanya. Platform ini memungkinkan pendefinisian ruang pencarian secara fleksibel, sehingga algoritma dapat menyesuaikan variabel termasuk yang bersifat logaritmik atau diskrit dan mendukung penggunaan operator mutasi maupun rekombinasi yang ditentukan oleh pengguna.

2.11 Model *Deep Support Vector Data Description* (*Deep* SVDD)

Deep Support Vector Data Description (*Deep* SVDD) merupakan pendekatan *one-class classification* berbasis *deep neural network* yang diperkenalkan oleh Ruff, *et al*. (2018), untuk mempelajari representasi kompak dari data normal melalui pemetaan ke ruang laten yang terkonsentrasi di sekitar sebuah pusat (*center*) dalam bentuk *hypersphere* berdimensi tinggi. Tidak seperti *autoencoder* yang mengandalkan rekonstruksi, *Deep* SVDD hanya menggunakan arsitektur *encoder* untuk memproyeksikan data ke ruang laten, sehingga lebih stabil dan menghindari *hypersphere collapse* yang sering terjadi pada model dengan *decoder*. Tujuan utama *Deep* SVDD adalah mempelajari parameter jaringan agar representasi data normal berada sedekat mungkin dengan pusat *hypersphere*, sedangkan data anomali dipetakan lebih jauh dari pusat tersebut. Untuk skenario yang mengizinkan sebagian kecil anomali pada data pelatihan, Ruff, *et al*. (2018), merumuskan persamaan

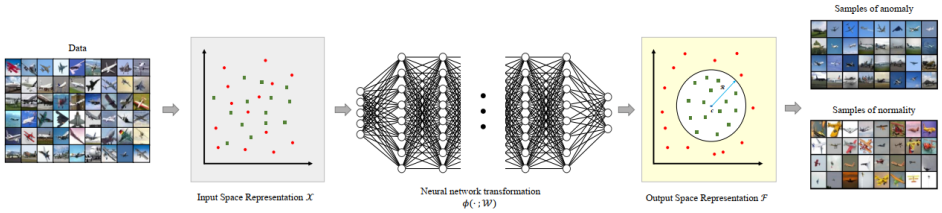
soft-boundary Deep SVDD yang diberikan pada Persamaan (14) sebagai berikut:

$$\min_{R,W} \left\{ R^2 + \frac{1}{\nu n} \sum_{i=1}^{n} \max\left(0,\ \|\phi(x_i; W) - c\|^2 - R^2\right) + \frac{\lambda}{2} \sum_{l=1}^{L} \|W^l\|_F^2 \right\} \tag{14}$$

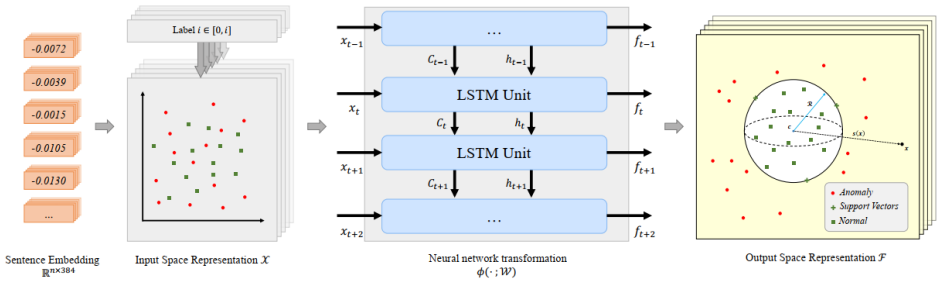
Persamaan 12 mengoptimalkan radius R sekaligus mengizinkan sejumlah sampel melanggar batas *sphere* melalui parameter $\nu \in (0, 1]$, yang berfungsi mengontrol proporsi *outlier* yang diperbolehkan dalam model. Selain itu, regularisasi L_2 dengan parameter $\lambda > 0$ diterapkan untuk menjaga stabilitas bobot jaringan. Untuk kondisi ketika hampir seluruh data pelatihan dianggap normal, Ruff, *et al*. (2018), menawarkan bentuk objektif yang lebih sederhana, yaitu *One-Class Deep* SVDD, yang diformulasikan dalam Persamaan (15) sebagai berikut:

$$\min_{W} \left\{ \frac{1}{n} \sum_{i=1}^{n} \|\phi(x_i; W) - c\|^2 + \frac{\lambda}{2} \sum_{l=1}^{L} \|W^l\|_F^2 \right\} \tag{15}$$

Persamaan ini tidak lagi mengoptimalkan radius *hypersphere*, melainkan berfokus sepenuhnya pada peminimalan jarak rata-rata representasi data terhadap pusat c. Arsitektur model *Deep* SVDD yang digunakan pada penelitian Ruff, *et al*. (2018), direpresentasikan seperti pada Gambar 3. Sedangkan arsitektur model *Deep* SVDD pada domain teks direpresentasikan pada Gambar 4.



Gambar 3. Arsitektur *Deep* SVDD Penelitian Ruff, *et al*. (2018), (Chen & Si, 2023).



Gambar 4. Arsitektur *Deep* SVDD pada Domain Teks (Chen & Si, 2023).

Arsitektur *Deep* SVDD memerlukan fungsi pemetaan $\phi(x)$ yang direpresentasikan oleh encoder berbasis *neural network* tanpa komponen *decoder* karena rekonstruksi dapat mendorong model mempelajari solusi *trivial*. Pemilihan *encoder* dapat disesuaikan dengan domain data. Seperti penelitian Chen & Si (2023), pada domain data teks yang bersifat sekuensial menerapkan *encoder* berbasis *Recurrent Neural Network* (RNN), yakni *Long Short-Term Memory* (LSTM), karena kemampuannya dalam menangkap dependensi urutan secara efektif. *Encoder* akan menghasilkan representasi laten yang dituliskan dalam Persamaan (16):

$$z = \phi(x; W) \tag{16}$$

Representasi laten yang dihasilkan kemudian dievaluasi berdasarkan jaraknya terhadap pusat *hypersphere*, dan skor anomali dihitung melalui fungsi yang dirumuskan pada Persamaan (17):

$$s(x) = \|\phi(x; W) - c\|^2 \tag{17}$$

Dengan $\phi(x; W) \in \mathbb{R}^d$ merupakan representasi laten berdimensi d yang dihasilkan oleh encoder, sedangkan $c \in \mathbb{R}^d$ merupakan pusat *hypersphere* berdimensi d. Penghitungan skor anomali pada Persamaan 17 menggunakan konsep norma Euclidean. Jika representasi laten dan pusat *hypersphere* masing-masing dinyatakan sebagai vektor berdimensi d,

$$\phi(x; W) = (z_1, z_2, \ldots, z_d) \tag{18}$$

$$c = (c_1, c_2, \ldots, c_d) \tag{19}$$

maka skor anomali dapat dituliskan secara eksplisit sebagai:

$$s(x) = \sum_{i=1}^{d} (z_i - c_i)^2 \tag{20}$$

Berdasarkan Persamaan 20, skor anomali diperoleh dengan menghitung jarak Euclidean kuadrat (*squared Euclidean distance*) antara representasi laten dan pusat *hypersphere*. Nilai skor diperoleh dengan menghitung selisih setiap komponen vektor laten terhadap komponen pusat yang bersesuaian, kemudian mengkuadratkannya dan menjumlahkan seluruh hasil kuadrat tersebut. Semakin besar nilai skor, semakin tinggi kemungkinan bahwa sebuah sampel merupakan anomali. Berdasarkan

formulasi tersebut, kualitas representasi laten $\phi(x; W)$ menjadi faktor dalam menentukan perhitungan skor anomali. Oleh karena itu, pemilihan dan perancangan *encoder* sebagai fungsi pemetaan ke ruang laten memiliki peran penting dalam meningkatkan kemampuan model dalam membedakan data normal dan anomali. Berbagai arsitektur *encoder* telah dikembangkan dalam literatur, khususnya untuk data sekuensial seperti teks.

2.12 Fungsi Aktivasi

Fungsi aktivasi merupakan fungsi matematis yang digunakan untuk mentransformasikan nilai masukan pada neuron menjadi nilai keluaran yang diteruskan ke lapisan berikutnya dalam jaringan saraf tiruan. Menurut Dubey *et al*. (2022), fungsi aktivasi berperan dalam memperkenalkan sifat *non-linear* pada model sehingga jaringan saraf mampu mempelajari pola yang kompleks dari data. Tanpa fungsi aktivasi, seluruh lapisan pada jaringan saraf hanya menghasilkan transformasi *linear* sehingga kemampuan model dalam merepresentasikan hubungan yang kompleks menjadi terbatas. Selain itu, fungsi aktivasi berkontribusi dalam proses pembentukan representasi fitur secara bertahap pada setiap lapisan jaringan saraf. Berikut dijelaskan mengenai beberapa fungsi aktivasi antara lain sigmoid dan tangen hiperbolik (tanh) (Dubey *et al*., 2022).

a. Fungsi Aktivasi Sigmoid

Fungsi aktivasi sigmoid merupakan salah satu fungsi aktivasi *non-linear* yang terdefinisi pada seluruh bilangan real dan memiliki sifat terdiferensialkan secara kontinu. Fungsi ini memetakan nilai *input neuron* ke dalam rentang keluaran antara 0 dan 1. Secara matematis, fungsi sigmoid dirumuskan pada Persamaan 21.

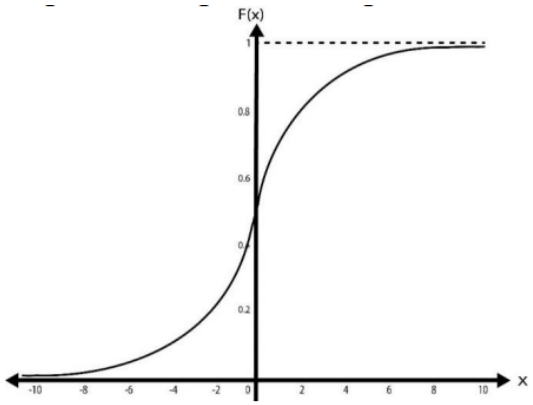
$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{21}$$

dengan:

$\sigma(x)$ = keluaran fungsi sigmoid,
x = input neuron yang diperoleh dari hasil transformasi linear,
e = bilangan Euler.

Nilai keluaran fungsi sigmoid akan mendekati 1 untuk *input* yang bernilai sangat

besar dan mendekati 0 untuk *input* yang bernilai sangat kecil. Karakteristik tersebut menghasilkan kurva berbentuk huruf S (*sigmoidal curve*) yang halus dan kontinu. Berdasarkan definisi tersebut, grafik fungsi aktivasi sigmoid disajikan pada Gambar 5.



Gambar 5. Fungsi Aktivasi Sigmoid (Akbar *et al*., 2022).

b. Fungsi Aktivasi Tangen Hiperbolik (Tanh)

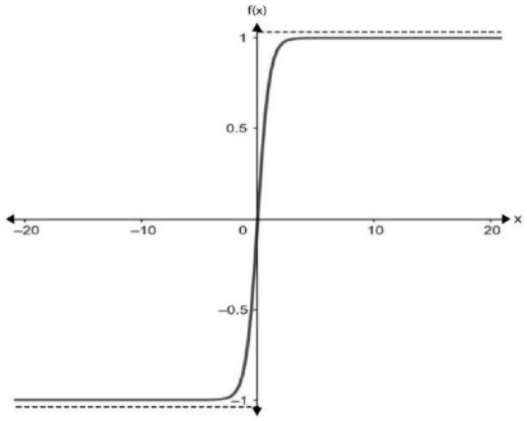
Fungsi aktivasi tangen hiperbolik (*hyperbolic tangent* atau tanh) merupakan fungsi aktivasi *non-linear* yang memetakan nilai *input* neuron ke dalam rentang $[-1, 1]$. Berbeda dengan fungsi sigmoid, keluaran fungsi tanh terpusat pada nol (*zero-centered*) sehingga distribusi aktivasi menjadi lebih seimbang antara nilai positif dan negatif. Representasi matematis fungsi tanh diberikan pada Persamaan 22.

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{22}$$

dengan:

$f(x)$ = keluaran fungsi *tanh*,
x = input neuron yang diperoleh dari hasil transformasi linear,
e = bilangan Euler.

Berdasarkan Persamaan (21), nilai keluaran fungsi *tanh* akan mendekati 1 ketika input bernilai sangat besar dan mendekati -1 ketika *input* bernilai sangat kecil. Sifat keluaran yang simetris terhadap titik nol (*zero-centered*) dapat membantu menjaga keseimbangan distribusi aktivasi selama proses pelatihan jaringan saraf. Grafik fungsi aktivasi tangen hiperbolik disajikan pada Gambar 6.



Gambar 6. Fungsi Aktivasi Tangen Hiperbolik (Akbar *et al*., 2022).

2.13 *Encoder*

Encoder merupakan komponen penting dalam model berbasis *representation learning* yang berfungsi untuk mentransformasikan data *input* ke dalam representasi laten yang lebih kompak dan informatif. Pada konteks *anomaly detection* berbasis *deep learning* khususnya pada data teks, *encoder* berperan dalam menangkap pola sekuensial dan hubungan semantik antar kata sehingga dapat menghasilkan fitur yang representatif. Representasi tersebut selanjutnya digunakan untuk membedakan pola data normal dan anomali dalam ruang fitur.

Pada implementasinya, terdapat berbagai jenis *encoder* yang digunakan dalam model *anomaly detection*, seperti *Convolutional Neural Network* (CNN), *Recurrent Neural Network* (RNN), hingga model berbasis *Transformer*. Namun, untuk data teks yang bersifat sekuensial, *encoder* berbasis RNN seperti *Long Short-Term Memory* (LSTM) dan *Bidirectional Long Short-Term Memory* (BiLSTM) masih banyak digunakan karena kemampuannya dalam menangkap dependensi urutan secara efektif (Pang *et al*., 2022). Prinsip kerja kedua *encoder* tersebut akan dijelaskan pada subbab selanjutnya.

2.13.1 *Long Short-Term Memory* (LSTM)

Menurut Zhang (2023), *Long Short-Term Memory* (LSTM) merupakan pengembangan dari RNN yang dirancang untuk mengatasi keterbatasan dalam menangkap dependensi jangka panjang pada data sekuensial. Arsitektur ini memperkenalkan komponen *cell state* serta mekanisme *gating* yang terdiri dari *forget gate*, *input gate*, dan *output gate*. Masing-masing gerbang memiliki peran berbeda yakni *input gate* berfungsi untuk menerima informasi baru, *forget gate* menentukan informasi mana yang harus dipertahankan atau dihapus, dan *output gate* mengatur informasi yang akan diteruskan sebagai keluaran. Sehingga informasi yang relevan dapat dipertahankan dan yang tidak relevan dapat diabaikan.

Jika diberikan suatu input sekuens $x_1, x_2, \ldots, x_T$, LSTM akan memproses data secara bertahap pada setiap waktu t dan menghasilkan representasi berupa *hidden state* $h_1, h_2, \ldots, h_T$. Pada setiap *time step*, LSTM melibatkan penghitungan pada tiga komponen utama, yaitu *forget gate*, *input gate*, dan *output gate*. Mekanisme ini memungkinkan model untuk menentukan informasi mana yang perlu dilupakan, diperbarui, dan diteruskan ke tahap berikutnya. Secara matematis, proses pada waktu ke-t dirumuskan pada Persamaan 23 sampai Persamaan 28.

$$f_t = \sigma_g(W_f[h_{t-1}, x_t] + b_f) \tag{23}$$

$$i_t = \sigma_g(W_i[h_{t-1}, x_t] + b_i) \tag{24}$$

$$\tilde{c}_t = \tanh(W_c[h_{t-1}, x_t] + b_c) \tag{25}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{26}$$

$$o_t = \sigma_g(W_o[h_{t-1}, x_t] + b_o) \tag{27}$$

$$h_t = o_t \odot \tanh(c_t) \tag{28}$$

dengan:

$i_t,\ f_t,\ o_t$ = aktivasi gerbang *input*, *forget*, dan *output* pada waktu ke-t,
x_t = vektor input pada waktu ke-t,
h_{t-1} = *hidden state* pada waktu sebelumnya,
W_f, W_i, W_c, W_o = matriks bobot pada masing-masing gerbang,
b_f, b_i, b_c, b_o = bias pada masing-masing gerbang,
σ_g = fungsi aktivasi sigmoid yang memetakan nilai ke rentang $[0, 1]$,
$\tilde{c}_t$ = kandidat *cell state* pada waktu ke-t,
c_t = *cell state* pada waktu ke-t,
h_t = *hidden state* pada waktu ke-t,
$\odot$ = perkalian elemen demi elemen (*Hadamard product*).

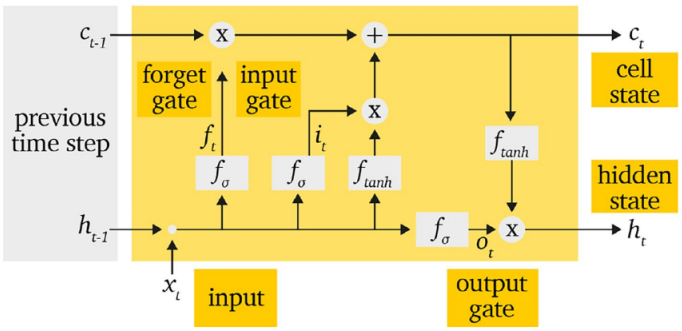
Proses pada LSTM diawali dengan *forget gate* yang bertugas menentukan informasi pada *cell state* sebelumnya yang perlu dipertahankan atau dilupakan. *Forget gate* menerima masukan berupa *hidden state* pada waktu sebelumnya (h_{t-1}) dan *input* saat ini (x_t). Kedua masukan tersebut kemudian ditransformasikan secara linear dan diproses menggunakan fungsi aktivasi sigmoid sehingga menghasilkan nilai f_t pada rentang $[0, 1]$ sebagaimana ditunjukkan pada Persamaan (23). Nilai f_t menunjukkan tingkat retensi informasi pada setiap elemen *cell state*. Nilai yang mendekati 0 mengindikasikan bahwa informasi akan dihilangkan, sedangkan nilai yang mendekati 1 menunjukkan bahwa informasi akan dipertahankan.

Proses selanjutnya adalah menentukan informasi baru yang akan ditambahkan ke dalam memori melalui *input gate*. Pada tahap ini, terdapat dua proses utama. Pertama, dihitung nilai gerbang pembaruan (i_t) menggunakan fungsi sigmoid untuk menentukan seberapa besar informasi baru akan dimasukkan ke dalam *cell state* sebagaimana ditunjukkan pada Persamaan (24). Kedua, dibentuk kandidat *cell state* baru ($\tilde{c}_t$) menggunakan fungsi aktivasi tangen hiperbolik (*tanh*) pada Persamaan (25). Fungsi *tanh* menghasilkan keluaran pada rentang $[-1, 1]$ sehingga mampu merepresentasikan informasi baru secara lebih seimbang. Nilai kandidat tersebut kemudian dikombinasikan dengan keluaran *input gate* untuk menghasilkan informasi yang akan ditambahkan ke dalam memori.

Selanjutnya, *cell state* diperbarui dengan menggabungkan informasi yang dipertahankan dari langkah sebelumnya dan informasi baru yang diperoleh dari *input gate*. Proses pembaruan *cell state* dinyatakan pada Persamaan (26).

Tahap terakhir adalah *output gate* yang menentukan informasi apa saja dari *cell state* yang akan diteruskan sebagai *hidden state* pada waktu saat ini. Sama seperti gerbang sebelumnya, *output gate* menerima masukan berupa h_{t-1} dan x_t, kemudian memprosesnya menggunakan fungsi sigmoid untuk menghasilkan nilai o_t sebagaimana ditunjukkan pada Persamaan (27). Nilai o_t digunakan untuk mengontrol bagian informasi yang akan dikeluarkan dari *cell state*. Selanjutnya, *cell state* yang telah diperbarui diproses menggunakan fungsi *tanh* dan dikalikan dengan keluaran *output gate* sehingga diperoleh *hidden state* saat ini (h_t) yang dihitung melalui Persamaan (28).

Hidden state tersebut kemudian diteruskan ke langkah waktu berikutnya dan sekaligus menjadi representasi keluaran LSTM pada waktu ke-t. Proses tersebut berlangsung secara berulang dari $t = 1$ sampai $t = T$, sehingga keluaran pada waktu ke-t dapat dipandang sebagai hasil komposisi bertingkat dari seluruh masukan sebelumnya, yaitu $h_t = f(x_t, f(x_{t-1}, \ldots, f(x_1)))$. Visualisasi struktur LSTM disajikan pada Gambar 7.



Gambar 7. Struktur LSTM (Heckelmann *et al.*, 2025).

Pada konteks encoder, representasi laten diperoleh dari *hidden state* terakhir yang dianggap telah merangkum seluruh informasi dari sekuens *input*. Representasi laten yang diperoleh diformulasikan pada Persamaan 29.

$$z = h_T \tag{29}$$

Pada beberapa kasus, representasi ini kemudian diproyeksikan ke dimensi tertentu melalui lapisan *linear* (Persamaan 30).

$$z = Wh_T + b \tag{30}$$

2.13.2 *Bidirectional Long Short-Term Memory* (BiLSTM)

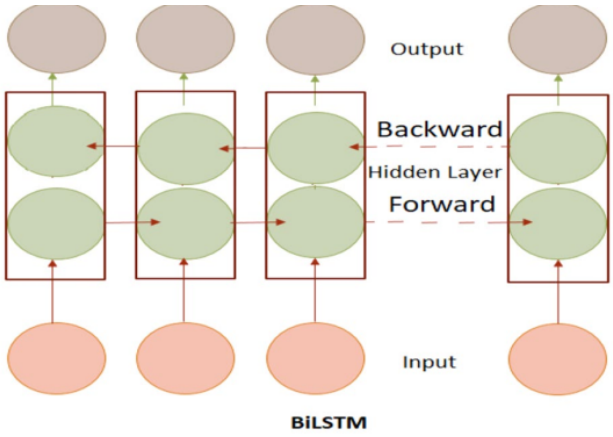
Menurut Abduljabbar *et al*. (2021), *Bidirectional Long Short-Term Memory* (BiLSTM) merupakan pengembangan dari LSTM yang bertujuan untuk meningkatkan kualitas representasi dengan memproses data dalam dua arah, yaitu dari awal ke akhir (*forward*) dan dari akhir ke awal (*backward*). Pendekatan ini memungkinkan model untuk memanfaatkan informasi konteks secara lebih lengkap karena setiap elemen dalam sekuens dipengaruhi oleh data sebelum dan sesudahnya. Untuk suatu *input* sekuens $x_1, x_2, \ldots, x_T$, BiLSTM menghasilkan dua *hidden state* pada setiap *time step* yang kemudian digabungkan menggunakan operasi konkatenasi.

$$\overrightarrow{h}_t = \text{LSTM}_{\text{forward}}(x_t) \tag{31}$$

$$\overleftarrow{h}_t = \text{LSTM}_{\text{backward}}(x_t) \tag{32}$$

$$h_t = [\overrightarrow{h}_t \oplus \overleftarrow{h}_t] \tag{33}$$

Visualisasi skema arsitektur BiLSTM disajikan pada Gambar 8.



Gambar 8. Skema Arsitektur BiLSTM (Abduljabbar *et al*., 2021).

Pada konteks *encoder*, representasi laten diperoleh dari gabungan *hidden state* terakhir:

$$z = [\overrightarrow{h}_t \oplus \overleftarrow{h}_t] \tag{34}$$

atau dalam bentuk proyeksi linear:

$$z = W[\overrightarrow{h}_t \oplus \overleftarrow{h}_t] + b \tag{35}$$

2.14 Konsep Peluang

Peluang (*probability*) merupakan ukuran numerik yang digunakan untuk menyatakan tingkat kemungkinan terjadinya suatu kejadian dalam suatu percobaan acak. Nilai peluang berada pada interval 0 hingga 1, dengan nilai 0 menunjukkan bahwa suatu kejadian tidak mungkin terjadi dan nilai 1 menunjukkan bahwa suatu kejadian pasti terjadi. Pada teori peluang, suatu kejadian (*event*) didefinisikan sebagai himpunan hasil yang mungkin muncul dari suatu percobaan, sedangkan ruang sampel (*sample space*) merupakan himpunan seluruh hasil yang mungkin terjadi. Teori peluang merupakan cabang matematika yang mengikuti prinsip-prinsip kombinatorik dan digunakan sebagai dasar dalam ilmu statistika. Teori peluang menyediakan kerangka matematis untuk mengukur kemungkinan terjadinya suatu kejadian serta menganalisis hubungan antar kejadian dalam suatu ruang sampel (Toruan, 2022).

Selain digunakan untuk mengukur peluang suatu kejadian tunggal, teori peluang juga digunakan untuk menganalisis hubungan antara dua atau lebih kejadian. Menurut Toruan (2022), hubungan tersebut dapat dinyatakan melalui beberapa konsep, yaitu peluang kejadian saling lepas (*mutually exclusive events*), peluang kejadian tidak saling lepas (*non-mutually exclusive events*), peluang kejadian saling bebas (*independent events*), dan peluang bersyarat (*conditional probability*).

a. Peluang kejadian saling lepas (*mutually exclusive events*)

   Dua kejadian A dan B dikatakan saling lepas (*mutually exclusive events*) apabila keduanya tidak dapat terjadi secara bersamaan dalam satu percobaan. Kondisi ini menunjukkan bahwa kedua kejadian tidak memiliki irisan, sehingga berlaku $A \cap B = \varnothing$ atau ekuivalen dengan $P(A \cap B) = 0$. Secara matematis peluang kejadian saling lepas dinyatakan pada Persamaan 36.

$$P(A \cup B) = P(A) + P(B) \tag{36}$$

b. Peluang kejadian tidak saling lepas (*non-mutually exclusive events*)

   Dua kejadian A dan B dikatakan tidak saling lepas (*non-mutually exclusive events*) apabila keduanya dapat terjadi secara bersamaan sehingga memiliki

irisan. Secara matematis, kondisi tersebut dinyatakan dengan $A \cap B \neq \varnothing$ atau ekuivalen dengan $P(A \cap B) > 0$. Secara matematis peluang kejadian tidak saling lepas dinyatakan pada Persamaan 37.

$$P(A \cup B) = P(A) + P(B) - P(A \cap B) \tag{37}$$

c. Peluang bersyarat (*conditional probability*)

Peluang bersyarat (*conditional probability*) adalah peluang terjadinya suatu kejadian dengan syarat bahwa kejadian lain telah terjadi. Jika A dan B merupakan dua kejadian dengan $P(A) > 0$, maka peluang kejadian B dengan syarat kejadian A telah terjadi dinyatakan pada Persamaan 38.

$$P(B \mid A) = \frac{P(A \cap B)}{P(A)} \tag{38}$$

Persamaan 38 menunjukkan bahwa peluang kejadian B dihitung berdasarkan ruang sampel yang telah dibatasi oleh kejadian A.

d. Peluang kejadian saling bebas (*independent events*)

Dua kejadian A dan B dikatakan saling bebas (*independent events*) apabila terjadinya salah satu kejadian tidak memengaruhi peluang terjadinya kejadian lainnya. Secara matematis, kondisi tersebut dinyatakan pada Persamaan 39.

$$P(A \cap B) = P(A)\, P(B) \tag{39}$$

Sehingga, peluang kejadian tidak saling bebas yang merupakan negasi dari kejadian saling bebas dinyatakan pada Persamaan 40.

$$P(A \cap B) \neq P(A)P(B) \tag{40}$$

atau ekuivalen dengan,

$$P(A \mid B) \neq P(A) \quad \text{atau} \quad P(B \mid A) \neq P(B) \tag{41}$$

2.15 *Model Evaluation*

Evaluasi kinerja model merupakan tahapan penting dalam *machine learning* yang bertujuan untuk mengukur kemampuan model dalam melakukan generalisasi terhadap data baru di luar data pelatihan. Langkah ini memastikan bahwa kinerja model tidak hanya baik pada data latih, tetapi juga mampu melakukan generalisasi secara memadai pada data di luar sampel sehingga dapat meminimalkan risiko terjadinya overfitting (Mufida, *et al*., 2025).

Secara umum, metode yang sering digunakan untuk mengevaluasi performa model klasifikasi adalah *confusion matrix*. Pada dasarnya *confusion matrix* adalah sebuah tabel berukuran 22 yang menggambarkan kelas yang diprediksi (*predicted class*) dan kelas aktual (*actual class*) dari data. Menurut Khairani, *et al*. (2024), *confusion matrix* ini terdiri dari 4 jenis nilai utama yaitu:

a. *True Positive* (TP): Jumlah sampel yang diprediksi positif oleh model dan memang termasuk kelas positif.

b. *False Positive* (FP): Jumlah sampel yang diprediksi sebagai positif oleh model, namun sebenarnya negatif.

c. *False Negative* (FN): Jumlah sampel yang diprediksi sebagai negatif oleh model, namun sebenarnya positif.

d. *True Negative* (TN): Jumlah sampel yang diprediksi sebagai negatif oleh model dan benar termasuk dalam kategori negatif.

Pada klasifikasi biner, bentuk representasi *confusion matrix* ditunjukkan pada Gambar 9.

		Predicted Class	
		Positive	Negative
Actual Class	Positive	TP	FN
	Negative	FP	TN

Gambar 9. Ilustrasi *Confusion Matrix* (Markoulidakis, *et al*., 2021).

Sedangkan pada pendekatan *One-Class Classification* (OCC), karakteristik evaluasi berbeda dengan klasifikasi konvensional. Model OCC, seperti *Deep Support Vector Data Description* (*Deep* SVDD), tidak menghasilkan probabilitas kelas, melainkan menghasilkan skor anomali yang merepresentasikan tingkat penyimpangan suatu observasi terhadap distribusi data normal (Ruff *et al*., 2018). Sehingga dibutuhkan *threshold* yakni suatu nilai yang menentukan batas pemisahan antara data normal dan anomali. Oleh karena itu, diperlukan metrik evaluasi yang bersifat *threshold free* untuk mengevaluasi model secara global. Evaluasi model OCC dilakukan melalui dua pendekatan utama, yaitu:

1. Evaluasi tanpa *threshold* berbasis ROC AUC
2. Evaluasi dengan *threshold* berbasis *confusion matrix*

Penjelasan lebih lanjut mengenai pendekatan evaluasi model OCC dijelaskan pada sub bab berikutnya.

2.15.1 Evaluasi Tanpa *Threshold* Berbasis ROC AUC

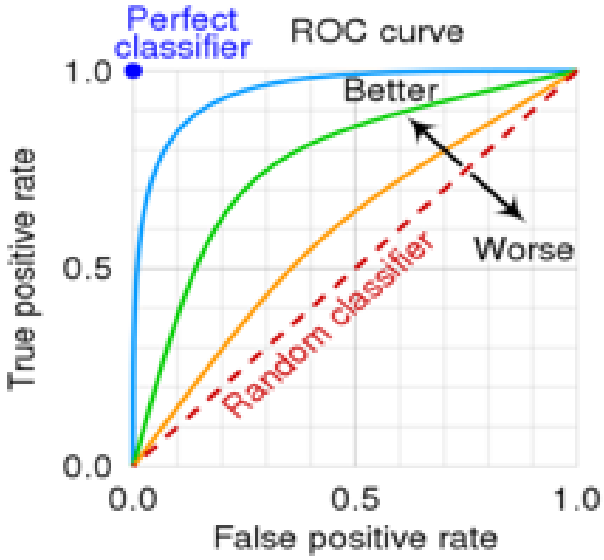
Kurva *Receiver Operating Characteristic Area Under the Curve* (ROC AUC) menjadi metrik standar dalam evaluasi model deteksi anomali dan OCC karena mengevaluasi performa model secara menyeluruh tanpa bergantung pada satu nilai *threshold* tertentu. Menurut Sathyanarayanan & Tantri (2024), kurva ROC merupakan representasi grafis yang menggambarkan hubungan antara *True Positive Rate* (TPR) atau *sensitivity* dan *False Positive Rate* (FPR) atau $1-$*specificity* pada berbagai nilai ambang keputusan. TPR menunjukkan proporsi kasus positif yang berhasil dikenali dengan benar oleh model, sedangkan FPR menggambarkan proporsi kasus negatif yang justru salah diprediksi sebagai positif oleh model. Secara matematis TPR dan FPR dinyatakan dalam Persamaan 42 dan Persamaan 43.

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{42}$$

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} \tag{43}$$

Nilai AUC diperoleh dengan menghitung luas area di bawah kurva ROC, dengan rentang nilai 0 sampai 1. Semakin besar nilai AUC, semakin baik kemampuan

model dalam membedakan kelas. Gambar 10 menyajikan ilustrasi dari kurva ROC.



Gambar 10. Ilustrasi ROC AUC (Sathyanarayanan & Tantri, 2024).

Pada model berbasis skor seperti *Deep* SVDD, AUC merepresentasikan probabilitas bahwa model memberikan skor anomali yang lebih tinggi pada data anomali dibandingkan data normal (Goldstein & Uchida, 2016). Secara matematis dinyatakan pada Persamaan 44.

$$\text{AUC} = P(s(x_{\text{anomali}}) > s(x_{\text{normal}})) \tag{44}$$

2.15.2 Evaluasi Dengan *Threshold* Berbasis *Confusion Matrix*

Model deteksi anomali berbasis *One-Class Classification* (OCC) memberikan output berupa skor kontinu yang merepresentasikan tingkat penyimpangan suatu observasi terhadap pola normal. Oleh karena itu, penggunaan *threshold* (τ) diperlukan sebagai mekanisme untuk mengubah skor tersebut menjadi keputusan klasifikasi biner. Secara umum, suatu observasi dengan skor anomali $s(x)$ akan diklasifikasikan sebagai anomali jika memenuhi $s(x) \geq \tau$ dan diklasifikasikan sebagai data normal jika $s(x) < \tau$. Dengan demikian, nilai τ menjadi batas pemisahan antara data normal dan anomali. Penentuan *threshold* ini memiliki peran yang sangat penting karena secara langsung memengaruhi hasil klasifikasi yang dihasilkan model.

Pendekatan yang umum digunakan dalam deteksi anomali adalah penentuan *threshold* berdasarkan distribusi skor anomali. Pendekatan ini didasarkan pada asumsi bahwa data normal cenderung mengikuti distribusi tertentu, sedangkan data anomali berada pada bagian ekor distribusi. Teknik yang umum digunakan sebagai mekanisme penentuan *threshold* berdasarkan distribusi meliputi (Komadina *et al*., 2024):

a. Pendekatan Persentil

   Threshold ditentukan berdasarkan persentil tertentu dari distribusi skor, misalnya persentil ke-95 atau ke-99, sehingga hanya sebagian kecil data dengan skor tertinggi yang dianggap sebagai anomali.

b. Pendekatan Distribusi Ekstrem (*Extreme Value Theory* / EVT)

   Pendekatan ini memodelkan bagian ekor distribusi menggunakan *Generalized Pareto Distribution* (GPD) melalui metode Peaks *Over Threshold* (POT). Sehingga memungkinkan estimasi probabilitas kejadian ekstrem dan penentuan *threshold* yang lebih adaptif terhadap distribusi data.

c. Pendekatan Statistik Sederhana

   Pendekatan ini menggunakan karakteristik statistik μ (rata-rata) dan σ (simpangan baku) dari skor anomali. Persamaan untuk pendekatan ini dinyatakan pada Persamaan 45.

$$\tau = \mu + k\sigma \tag{45}$$

   dengan k adalah parameter yang merupakan bilangan bulat positif.

Setelah *threshold* ditentukan, skor anomali dapat dikonversi menjadi label biner sehingga memungkinkan pembentukan *confusion matrix*. Misalkan Y menyatakan *actual class* dan $\hat{Y}$ menyatakan *predicted class*, dengan $Y, \hat{Y} \in \{0, 1\}$, di mana nilai 1 merepresentasikan kelas positif (anomali) dan nilai 0 merepresentasikan kelas negatif (normal). *Confusion matrix* merepresentasikan hubungan antara hasil prediksi model dan kondisi aktual data dalam bentuk empat komponen utama, yaitu:

- *True Positive* (TP): $P(\hat{Y} = 1 \mid Y = 1)$
- *True Negative* (TN): $P(\hat{Y} = 0 \mid Y = 0)$

- *False Positive* (FP): $P(\hat{Y} = 1 \mid Y = 0)$
- *False Negative* (FN): $P(\hat{Y} = 0 \mid Y = 1)$

Perubahan nilai *threshold* tidak hanya memengaruhi hasil klasifikasi, tetapi juga berdampak langsung terhadap metrik evaluasi yang diturunkan dari *confusion matrix*. Oleh karena itu, dalam konteks deteksi anomali, pemilihan *threshold* umumnya dianalisis menggunakan metrik yang mampu merepresentasikan keseimbangan performa model.

Metrik yang paling relevan dalam analisis ini adalah *sensitivity* dan *specificity*, karena keduanya secara langsung mencerminkan kemampuan model dalam mendeteksi anomali dan menghindari kesalahan klasifikasi pada data normal.

a. *Sensitivity* (*Recall*)

Sensitivity atau *recall* merupakan metrik yang digunakan untuk mengukur kemampuan model dalam mendeteksi seluruh data positif (anomali) secara benar. Secara probabilistik, *sensitivity* dapat diinterpretasikan sebagai peluang bahwa model memprediksi suatu data sebagai anomali, dengan syarat bahwa data tersebut memang merupakan anomali. *Sensitivity* banyak digunakan pada sistem deteksi anomali karena kegagalan mendeteksi anomali dapat berdampak serius, seperti pada deteksi *fraud* atau sistem keamanan jaringan.

Pada deteksi anomali,*sensitivity* menjadi sangat krusial karena berkaitan langsung dengan kemampuan model dalam menghindari kesalahan berupa *false negative*, yaitu kondisi ketika data anomali tidak berhasil terdeteksi. Oleh karena itu, metrik ini memiliki hubungan berbanding terbalik dengan *False Negative* (FN) yang dinotasikan dengan $P(\hat{Y} = 0 \mid Y = 1)$. Secara matematis, *sensitivity* dirumuskan pada Persamaan 46 sampai Persamaan 49.

$$\begin{aligned} \textit{Sensitivity} &= 1 - \mathrm{FN} && (46)\\ &= 1 - P(\hat{Y} = 0 \mid Y = 1) && (47)\\ &= 1 - \frac{P(\hat{Y} = 0 \cap Y = 1)}{P(Y = 1)} && (48)\\ &= 1 - \frac{\mathrm{FN}}{\mathrm{TP} + \mathrm{FN}} && (49) \end{aligned}$$

b. *Specificity*

Specificity merupakan metrik yang mengukur kemampuan model dalam mengidentifikasi data negatif (normal) secara benar. Secara probabilistik, *specificity* dapat diinterpretasikan sebagai peluang bahwa model memprediksi suatu data sebagai normal dengan syarat bahwa data tersebut memang merupakan normal.

Specificity berperan penting dalam mengontrol kesalahan berupa *false positive*, yaitu kondisi ketika data normal diklasifikasikan sebagai anomali. Oleh karena itu, metrik ini memiliki hubungan berbanding terbalik dengan *False Positive* (FP) yang dinotasikan dengan $P(\hat{Y} = 1 \mid Y = 0)$. Secara matematis, *specificity* dirumuskan pada Persamaan 50 sampai Persamaan 53.

$$\begin{aligned} \textit{Specificity} &= 1 - \mathrm{FP} && (50)\\ &= 1 - P(\hat{Y} = 1 \mid Y = 0) && (51)\\ &= 1 - \frac{P(\hat{Y} = 1 \cap Y = 0)}{P(Y = 0)} && (52)\\ &= 1 - \frac{\mathrm{FP}}{\mathrm{TN} + \mathrm{FP}} && (53) \end{aligned}$$

2.16 Analisis Sensitivitas *Threshold*

Analisis sensitivitas terhadap variasi nilai *threshold* dilakukan untuk mengevaluasi perubahan kinerja model dalam mengklasifikasikan data sebagai normal atau anomali. Variasi nilai *threshold* akan menghasilkan perubahan pada kinerja model, yang umumnya ditunjukkan melalui hubungan *trade-off* antara sensitivitas (*True Positive Rate*) dan spesifisitas (*True Negative Rate*). Secara umum, peningkatan nilai *threshold* cenderung meningkatkan spesifisitas namun menurunkan sensitivitas, sedangkan penurunan *threshold* akan meningkatkan sensitivitas tetapi berpotensi menurunkan spesifisitas. Fenomena ini merupakan karakteristik umum dalam sistem klasifikasi berbasis ambang keputusan dan banyak dibahas dalam studi evaluasi model modern (Araf *et al*., 2024).

Pada deteksi anomali dalam domain data teks medis, evaluasi kinerja model tidak

hanya mempertimbangkan akurasi secara keseluruhan, tetapi juga memperhatikan perbedaan tingkat kepentingan dari masing-masing jenis kesalahan klasifikasi. Dengan mempertimbangkan perbedaan biaya tersebut, pemilihan *threshold* dalam penelitian ini diarahkan untuk memaksimalkan sensitivitas (*True Positive Rate*), sehingga meminimalkan kemungkinan anomali yang tidak terdeteksi. Pendekatan ini sejalan dengan penelitian terbaru yang menyatakan bahwa dalam sistem deteksi anomali, khususnya pada domain dengan risiko tinggi, sensitivitas sering diprioritaskan untuk memastikan bahwa kejadian penting tidak terlewat (Pang *et al*., 2021).

BAB III

METODE PENELITIAN

3.1 Waktu dan Tempat Penelitian

3.1.1 Waktu Penelitian

Penelitian ini dilaksanakan pada semester ganjil tahun akademik 2025/2026, dimulai pada bulan Agustus 2025. Rangkaian kegiatan penelitian dibagi menjadi tiga tahap utama. Tahap pertama mencakup kegiatan studi literatur, penetapan topik penelitian, pengumpulan data, serta penyusunan proposal hingga pelaksanaan seminar proposal pada bulan November 2025. Tahap kedua merupakan tahap pengolahan dan analisis data, yang meliputi *input* data, *Exploratory Data Analysis* (EDA), *preprocessing*, *labelling*, pembagian data, penyesuaian hiperparameter, proses pelatihan model, serta evaluasi kinerja model yang berlangsung selama November 2025 hingga Januari 2026. Tahap terakhir adalah penyusunan hasil dan pembahasan penelitian, serta perumusan kesimpulan yang dilakukan pada akhir Januari 2026 hingga April 2026, diikuti dengan seminar hasil dan ujian komprehensif.

3.1.2 Tempat Penelitian

Penelitian ini dilaksanakan melalui studi literatur di Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung, yang berlokasi di Jalan Prof. Dr. Soemantri Brojonegoro No. 1, Gedong Meneng, Bandar Lampung.

3.2 Data dan Alat Penelitian

3.2.1 Data Penelitian

Data yang digunakan dalam penelitian ini merupakan GENIA *Bio-medical Event Dataset*, sebuah kumpulan data berbasis teks biomedis berkaitan dengan penyakit leukemia yang diunduh melalui *platform* Kaggle dan disusun oleh Nishanth. *Dataset* ini merupakan versi sederhana dari *event-annotated* GENIA *dataset* yang berasal dari sumber data TEES (*Text Extraction for Event Sequence*). Struktur dataset terdiri dari 4 kolom utama memuat kalimat berkaitan dengan penyakit leukemia yaitu *Sentence*, *TriggerWord*, *TriggerWordLoc*, dan *EventType*. *Sentence* mencakup teks biomedis asli dengan kalimat terpanjang terdiri dari 170 kata dan kalimat terpendek terdiri dari 2 kata. *TriggerWord* adalah label kata pemicu, terdiri dari 1 kata baik kalimat terpanjang maupun terpendek. *TriggerWordLoc* merupakan lokasi kemunculan kata pemicu dalam kalimat sehingga memuat angka yang menyatakan lokasi *triggerword* pada *sentence*. *EventType* merupakan jenis peristiwa biomedis yang juga terdiri dari 1 kata baik kalimat terpanjang maupun terpendek. *Dataset* pada *platform* sudah dibagi menjadi tiga bagian, yaitu *train set* yang terdiri dari 8666 data dengan 4957 *missing value*, *development set* yang terdiri dari 2886 data dengan 1696 data *missing value*, dan *test set* yang terdiri dari 3360 data dengan keseluruhan merupakan *missing value*. Sehingga, total data yang diperoleh dari *platform* Kaggle adalah 11552 data dengan 6653 data *missing value*. Sampel data yang diperoleh disajikan pada Tabel 2.

Tabel 2. *Sample* Data

Sentence	**TriggerWord**	**TriggerWordLoc**	**EventType**
Down-regulation of interferon regulatory fact...	Down-regulation; expression;	1; 8;	Negative_regulation; Gene_expression;
Although the bcr-abl translocation has been...	deregulation;	30;	Regulation;

Sentence	TriggerWord	TriggerWordLoc	EventType
Promoter methylation of CpG target sites or d...	NaN	NaN	NaN
Therefore, we investigated whether IRF-4 pro...	regulation; expression;	16; 19;	Regulation; Gene_expression;
We evaluated two cytokine systems, IL-2 and ...	NaN	NaN	NaN

Dataset GENIA *Biomedical Event* merupakan korpus beranotasi manual yang menyediakan informasi *trigger word* dan *event type* pada teks biomedis berkaitan dengan penyakit leukemia. Pada penelitian ini, anotasi tersebut digunakan sebagai sumber pasangan teks, sementara *pseudo labeling* berbasis kemiripan semantik dimanfaatkan untuk membangun asumsi awal normalitas dalam pemodelan distribusi menggunakan *Deep* SVDD. Fokus penelitian ini adalah mengidentifikasi relasi semantik pada pasangan teks biomedis yang berkaitan dengan penyakit leukemia menggunakan *pseudo labeling* berbasis kemiripan, dan melakukan pendekatan deteksi anomali berbasis representasi distribusional untuk mengidentifikasi pasangan teks biomedis yang memiliki relasi semantik tidak konsisten menggunakan pemodelan *Deep* SVDD.

3.2.2 Alat Penelitian

Alat yang digunakan pada penelitian ini adalah sebagai berikut:

1. Perangkat Keras

   Perangkat keras yang digunakan dalam penelitian ini berfungsi sebagai komponen pendukung dalam proses pengolahan data serta pelaksanaan eksperimen. Penelitian ini dijalankan menggunakan laptop Acer Aspire Vero AV14-51 dengan spesifikasi sebagai berikut:

a. Prosesor Intel ® Core™ i5-1235U (hingga $\pm 4.4\,\mathrm{GHz}$, 10 core: 2 P-cores + 8 E-cores)

b. Memori (RAM): 16 GB LPDDR4X

c. Penyimpanan: SSD NVMe 512 GB

2. Perangkat Lunak

Perangkat lunak yang digunakan dalam penelitian ini adalah:

a. Sistem Operasi Windows 11 *version* 24H2.

Windows 11 merupakan generasi terbaru dari sistem operasi Windows yang dikembangkan oleh *Microsoft*. Sistem operasi ini berperan dalam mengatur dan mengoordinasikan fungsi perangkat keras maupun perangkat lunak pada komputer atau perangkat digital lainnya.

b. Overleaf (LaTeXTeX Live 2024)

Penelitian ini menggunakan LaTex pada *platform Overleaf* yang merupakan editor berbasis web untuk penulisan laporan penelitian secara *real time*. Salah satu kelebihan LaTeX terletak pada kemampuannya dalam membuat dokumen yang kompleks, sehingga penyajian isi dokumen menjadi lebih terstruktur dan berkualitas (Fitriani, *et al*., 2024).

c. *Python* 3.12.12

Penelitian ini memanfaatkan *Google Colaboratory* sebagai platform komputasi berbasis *Jupyter Notebook* yang disediakan oleh *Google*. Adapun *library* yang digunakan dalam proses penelitian sebagai berikut:

- *Pandas* 2.2.2

  Pandas merupakan *library Python* yang menawarkan berbagai struktur data canggih dan fitur pemrosesan data yang dirancang untuk menangani *dataset* terstruktur (McKinney, 2011).

- *Numpy* 2.0.2

  NumPy berperan sebagai *library* dalam *Python* yang dirancang untuk melakukan operasi komputasi menggunakan struktur data array (Harris, *et al*., 2020).

- *Scikitlearn* 1.6.1

Scikitlearn merupakan *library Python* yang menyediakan berbagai algoritma *machine learning* dan berperan penting dalam proses pengembangan maupun pengujian model *machine learning*. *Pipeline* pada *Scikitlearn* memungkinkan alur prapemrosesan, pelatihan, dan evaluasi dilakukan secara lebih terstruktur dan efisien (Salehi & Zarei, 2025).

- *SpaCy* 3.8.11

  SpaCy adalah pustaka *natural language processing* (NLP) modern yang dikembangkan untuk mendukung beragam aktivitas analisis teks, termasuk tokenisasi, penandaan kelas kata (POS-*tagging*), *named entity recognition* (NER), serta pemetaan struktur sintaktis melalui parsing (Neumann, *et al*., 2019).

- NLTK 3.9.1

  NLTK merupakan salah satu pustaka NLP yang menyediakan berbagai fasilitas untuk pemrosesan teks, analisis statistik, penerapan teknik pembelajaran mesin, serta akses terhadap beragam sumber daya linguistik seperti korpus dan basis data leksikal (Batta, 2024).

- *Gensim* 4.4.0

  Gensim merupakan *library Python* yang dirancang untuk melakukan *topic modeling*, pengindeksan dokumen, serta pengukuran tingkat kemiripan antar dokumen (Gupta, *et al*., 2022).

- *Matplotlib* 3.10.0

  Matplotlib merupakan *libraray* yang paling dikenal untuk visualisasi data dalam *Python*. *Library* ini bersifat *low-level* dan menyediakan beragam jenis *plot* 2D dan 3D yang dapat dikustomisasi, seperti *scatter plot*, *line plot*, histogram, dan berbagai bentuk visualisasi lainnya (Lavanya, *et al*., 2023).

- *Seaborn* 0.13.2

  Seaborn merupakan pustaka visualisasi data dalam *Python* yang dikembangkan di atas *Matplotlib* dan bekerja secara terpadu dengan struktur data *NumPy* serta *Pandas*. *Library* ini ini menawarkan kemampuan visualisasi statistik yang lebih komprehensif. *Seaborn* dilengkapi dengan *default theme* yang informatif serta beragam palet warna yang dirancang berdasarkan praktik terbaik dalam penyajian data secara lebih efektif dan wawasan yang lebih mendalam dari hasil

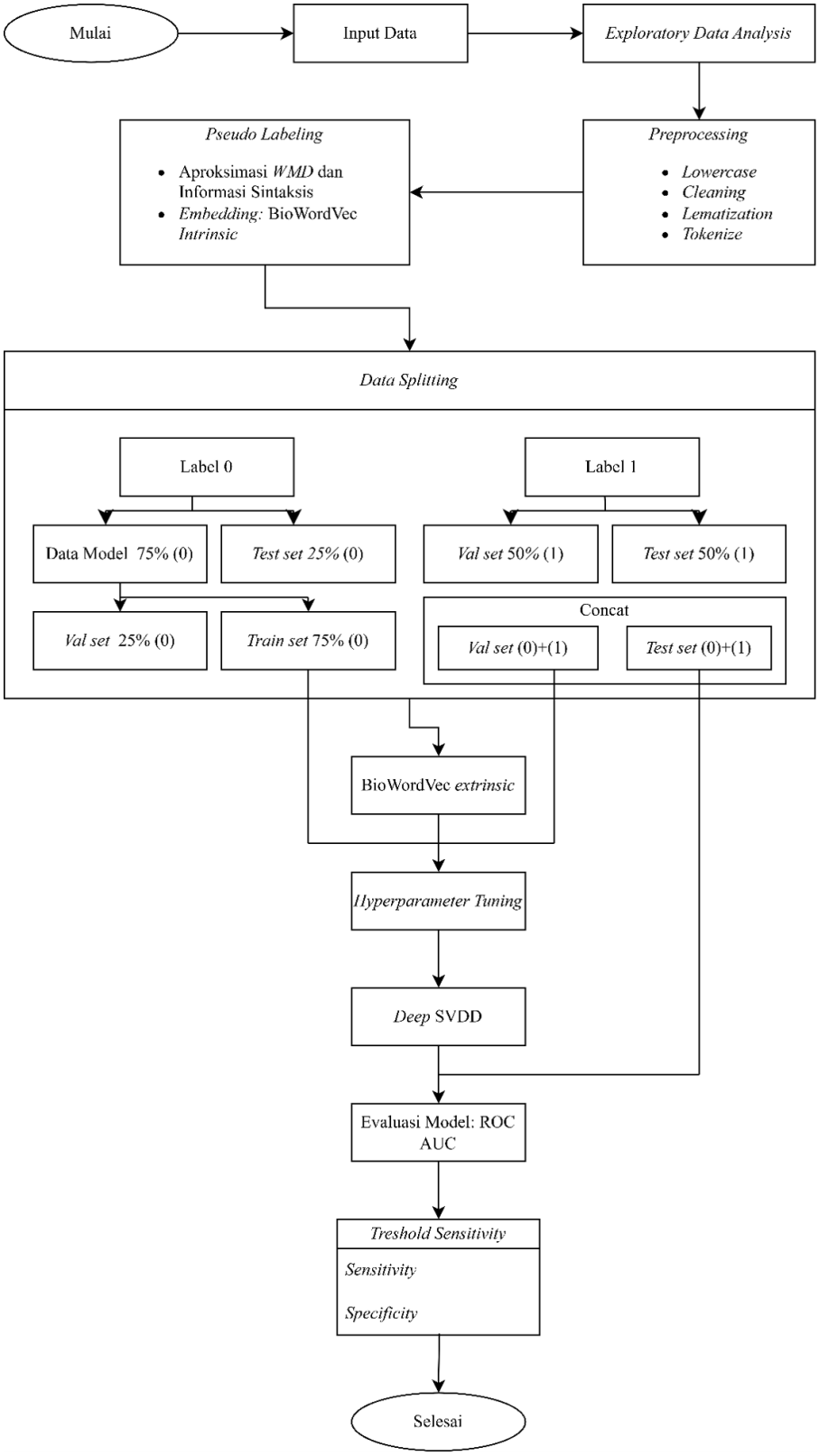
visualisasi tersebut (Lavanya, *et al*., 2023).

- *PyTorch* 2.9.0+cu126

  PyTorch merupakan *framework* pembelajaran mesin berbasis sumber terbuka yang banyak dimanfaatkan dalam pengembangan *deep learning* maupun pemrosesan bahasa alami. Mekanisme komputasi tensornya serupa dengan operasi *array* pada *NumPy*, namun diperkaya dengan dukungan akselerasi GPU serta kemampuan diferensiasi otomatis yang mempercepat proses pelatihan jaringan saraf (Mohialden *et al*., 2024).

3.3 Metode Penelitian

Penelitian ini berfokus pada deteksi anomali pada kemiripan semantik pasangan teks menggunakan model *Deep* SVDD, di mana hubungan semantik yang sesuai dipelajari sebagai distribusi normal, sementara pasangan teks yang menyimpang diidentifikasi sebagai anomali. Adapun alur penelitian pada penelitian ini disajikan pada Gambar 11.



Gambar 11. Alur Penelitian.

Berdasarkan Gambar 11. berikut adalah penjelasan setiap langkah yang dilakukan.

1. *Input* data

   Pada tahap ini data diinput ke *Google Colaboratory* sebagai *platform* komputasi berbasis *Jupyter Notebook*.

2. *Exploratory Data Analysis* (EDA)

   EDA merupakan tahapan krusial untuk memahami pola, karakteristik, dan struktur dasar data sebelum menerapkan algoritma matematis atau membangun model prediktif. Pelaksanaan EDA secara sistematis dan mendalam membantu menghasilkan wawasan penting serta mendeteksi potensi permasalahan yang dapat memengaruhi proses pengolahan data maupun kinerja model pada tahap selanjutnya.

3. *Preprocessing*

   Pada tahap ini dilakukan serangkaian proses *preprocessing*, dimulai dari *lowercasing* untuk mengubah seluruh karakter huruf menjadi bentuk kecil. Setelah itu, data dibersihkan dari *missing value*, tanda baca, simbol, spasi berlebih, serta karakter *non-alfabet* lain yang tidak diperlukan. Langkah berikutnya adalah *lemmatization*, yaitu mengonversi kata ke bentuk dasarnya, kemudian dilanjutkan dengan *tokenization* untuk memecah teks menjadi unit-unit kata (token).

4. *Pseudo Labeling*

   Pseudo labeling berbasis aproksimasi WMD dan penambahan informasi sintaksis. Pada tahap ini, digunakan *embedding BioWordVec intrinsic*. Hasil *pseudo labeling* merepresentasikan indikasi konsistensi semantik lokal dan tidak diperlakukan sebagai *ground truth absolut*, melainkan sebagai referensi awal pembentukan kelas normal untuk analisis lebih lanjut model deteksi anomali berbasis *Deep* SVDD.

5. *Data Splitting*

   Data yang telah melalui tahap *preprocessing* kemudian dibagi menjadi 3 set data yang berisi pasangan kalimat yakni, *sentencetriggerword*, *sentenceeventype*, dan *triggerwordeventype*. Sebelum data di *split* menjadi *train*, *test*, dan *val set*, terlebih dahulu masing masing *dataset* dipisahkan antara label normal (0)

dan label anomali (1). Untuk semua *set* data, skenario yang sama diterapkan. Yaitu, label normal di *split* sesuai dengan standar *machine learning* yaitu 75:25 dengan 75% data model dan 25% data *testing*. 75% dari data model di *split* lagi dengan proporsi yang sama yakni 75% untuk data *training*, 25% data *validation*. Sedangkan data berlabel anomali di *split* dengan rasio 50:50 yakni 50% data *validation*, dan 50% data *testing*. Selanjutnya hasil *split* data berlabel anomali digabungkan ke data *validation* dan data *testing* label normal. Sehingga, untuk *training* data hanya memiliki label normal, hal ini mengikuti standar evaluasi pada skenario *one-class* untuk deteksi anomali di mana model dilatih menggunakan data normal. Sedangkan *testing*, maupun *validation* memiliki label anomali pada masing-masing *subset*.

6. *Embedding*

   Setelah proses pelabelan, data direpresentasikan menggunakan *embedding BioWordVec* (*Extrinsic*). *BioWordVec* merupakan model berbasis *subword* yang dirancang untuk teks biomedis sehingga mampu menangkap informasi semantik dan morfologi secara lebih tepat. Pada penelitian ini, *BioWordVec* digunakan untuk mengubah setiap token hasil *preprocessing* menjadi vektor berdimensi tetap yang kemudian menjadi *input* utama bagi model.

7. *Hyperparameter Tuning*

   Tahap berikutnya adalah melakukan penyetelan *hyperparameter* untuk memperoleh konfigurasi model *Deep* SVDD yang optimal. Proses tuning mencakup pencarian kombinasi terbaik dari parameter seperti *hidden dimension*, *num layers*, dan *dropout*. *Latent dimension* untuk menentukan ukuran ruang, serta *learning rate* dan *weight decay*. *Hyperparameter tuning* dilakukan dengan menggunakan metode Optuna.

8. Model *Deep* SVDD

   Pada tahap ini, *Deep* SVDD memetakan data normal ke dalam ruang laten yang kompak sehingga anomali dapat diidentifikasi berdasarkan jarak *Euclidean* dari pusat distribusi tersebut. Representasi laten diperoleh melalui *encoder* LSTM dan BiLSTM yang diambil dari *hidden state* terakhir.

9. *Model Evaluation*

Setelah pelatihan, performa model dievaluasi menggunakan metrik yang umum digunakan pada *one-class classification*, yaitu *Area Under the ROC Curve* (AUC). Metrik ini dipilih karena mampu menggambarkan kemampuan model dalam membedakan data normal dan anomali pada berbagai ambang keputusan.

10. Analisis Sensitivitas *Threshold*

    Analisis lanjutan berbasis *threshold* dilakukan dengan menggunakan metrik sensitivitas (*sensitivity*) dan spesifisitas (*specificity*). Tahap ini bertujuan untuk mengevaluasi perilaku model pada berbagai nilai ambang skor anomali, sehingga dapat dianalisis *trade-off* antara kemampuan model dalam mendeteksi pasangan teks yang tidak konsisten secara semantik (*true positive rate*) dan kemampuannya mempertahankan pasangan teks normal agar tidak salah terklasifikasi sebagai anomali (*true negative rate*).

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil implementasi dan evaluasi model deteksi anomali pada pasangan teks biomedis menggunakan pendekatan *Deep* SVDD, diperoleh beberapa kesimpulan sebagai berikut.

1. Analisis karakteristik hubungan semantik pada dataset GENIA Biomedical Event menunjukkan adanya variasi distribusi antar pasangan teks. Dataset yang digunakan terdiri dari tiga jenis pasangan teks, yaitu *Sentence-TriggerWord*, *Sentence-EventType*, dan *TriggerWord–EventType*. Proses pelabelan dilakukan menggunakan pendekatan *pseudo labeling* berbasis aproksimasi WMD yang dimodifikasi dengan pembobotan *Part-of-Speech* (POS). Pendekatan ini menghasilkan kelas normal yang merepresentasikan pasangan teks dengan kemiripan semantik tinggi dan kelas anomali yang menunjukkan perbedaan semantik yang lebih besar. Distribusi *pseudo label* menunjukkan bahwa pasangan teks dalam dataset memiliki variasi kemiripan semantik yang memungkinkan terjadinya penyimpangan distribusi data.

2. Representasi fitur pasangan teks biomedis berhasil dibangun menggunakan *embedding* domain spesifik BioWordVec yang diproses melalui *encoder* LSTM dan BiLSTM. Setiap token hasil *preprocessing* dikonversi menjadi vektor *embedding* berdimensi 200 menggunakan BioWordVec *extrinsic* yang dirancang khusus untuk menangkap terminologi biomedis. Representasi *embedding* dari dua komponen teks digabungkan secara sekuensial sehingga membentuk input model dengan panjang maksimum 52 token untuk pasangan *Sentence–TriggerWord* dan *Sentence–EventType*, serta 8 token untuk pasangan *TriggerWord–EventType*. Hasil *hyperparameter tuning* menunjukkan bahwa konfigurasi optimal dipengaruhi oleh panjang sekuens, di mana pasangan teks yang melibatkan kalimat penuh

memerlukan dua layer *encoder*, sedangkan pasangan dengan panjang sekuens pendek cukup menggunakan satu layer. Selain itu, *latent dimension* sebesar 16 secara konsisten menghasilkan representasi kompak yang memadai untuk pembentukan *hypersphere* pada model *Deep* SVDD.

3. Implementasi model Deep SVDD pada domain teks biomedis berhasil mendeteksi anomali pada pasangan teks dalam dataset GENIA Biomedical Event. Model Deep SVDD dilatih dengan meminimalkan jarak kuadrat antara representasi laten data dan pusat *hypersphere* yang dihitung dari data latih. Hasil evaluasi menunjukkan bahwa *encoder* LSTM secara konsisten menghasilkan kinerja terbaik dibandingkan BiLSTM pada seluruh dataset. Model Deep SVDD dengan *encoder* LSTM mencapai nilai ROC AUC sebesar 0,99 pada dataset *Sentence–TriggerWord*, serta 0,98 pada dataset *Sentence–EventType* dan *TriggerWord–EventType*, yang menunjukkan kemampuan pemisahan data normal dan anomali yang sangat baik. Sebaliknya, model dengan *encoder* BiLSTM menghasilkan nilai ROC AUC yang lebih rendah yaitu 0,73 pada *Sentence–TriggerWord*, 0,69 pada *Sentence–EventType*, dan 0,80 pada *TriggerWord–EventType*. Hasil ini menunjukkan bahwa arsitektur *encoder* yang lebih sederhana seperti LSTM lebih efektif dalam menjaga keseimbangan antara kapasitas representasi dan kemampuan pemisahan anomali dalam kerangka *Deep* SVDD.

4. Analisis sensitivitas *threshold* menunjukkan adanya *trade-off* antara sensitivitas dan spesifisitas dalam proses deteksi anomali. Hasil analisis menunjukkan bahwa *threshold* yang lebih tinggi cenderung meningkatkan nilai spesifisitas namun menurunkan sensitivitas. Pada *threshold* P95, model menghasilkan sensitivitas tertinggi dengan kemampuan mendeteksi hampir seluruh anomali, meskipun spesifisitas relatif lebih rendah. Sebaliknya, *threshold* yang lebih tinggi seperti P97 dan P99 meningkatkan spesifisitas namun sedikit menurunkan sensitivitas. Oleh karena itu, pemilihan *threshold* dalam penelitian ini diprioritaskan pada nilai yang menghasilkan sensitivitas tinggi agar meminimalkan kemungkinan anomali yang tidak terdeteksi.

5.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, beberapa saran yang dapat diberikan untuk pengembangan penelitian selanjutnya adalah sebagai berikut.

1. Penelitian selanjutnya dapat mengeksplorasi penggunaan arsitektur *encoder* yang lebih kompleks, seperti *transformer* atau *attention-based models*, untuk meningkatkan kemampuan representasi konteks semantik pada data teks biomedis.
2. Pengujian metode deteksi anomali dapat diperluas pada dataset biomedis lain atau korpus literatur ilmiah yang lebih besar untuk mengevaluasi kemampuan generalisasi model.
3. Pendekatan *pseudo labeling* yang digunakan dalam penelitian ini masih menggunakan penyederhanaan informasi sintaksis berbasis POS dengan pertimbangan karakteristik data. Penelitian selanjutnya dapat mempertimbangkan penggunaan *dependency parsing* atau *graph-based syntactic representation* untuk meningkatkan akurasi perhitungan kemiripan semantik.
4. Model deteksi anomali yang dikembangkan dapat diintegrasikan dengan aplikasi praktis seperti sistem analisis literatur biomedis otomatis atau pembangunan *knowledge graph* biomedis, sehingga dapat membantu proses eksplorasi informasi ilmiah dalam jumlah besar secara lebih efisien.

DAFTAR PUSTAKA

Abduljabbar, R. L., Dia, H., & Tsai, P. W. (2021). Development and evaluation of bidirectional LSTM freeway traffic forecasting models using simulation data. *Scientific reports*. **11**(1): 23899. DOI: 10.21203/rs.3.rs-888775/v1

Afaq, S., & Rao, S. (2020). Significance of epochs on training a neural network. *Int. J. Sci. Technol. Res*. **9**(6): 485-488.

Akbar, R., Santoso, R., & Warsito, B. (2023). Prediksi tingkat temperatur Kota Semarang menggunakan metode long short-term memory (LSTM). *Jurnal Gaussian*. **11**(4): 572-579. DOI : 10.14710/j.gauss.11.4.572-579

Aliferis, C., & Simon, G. (2024). Overfitting, underfitting and general model overconfidence and under-performance pitfalls and best practices in machine learning and AI. *Artificial intelligence and machine learning in health care and medical sciences: Best practices and pitfalls*. (pp. 477-524). DOI: 10.1007/978-3-031-39355-6_10

Araf, I., Idri, A., & Chairi, I. (2024). Cost-sensitive learning for imbalanced medical data: a review. *Artificial Intelligence Review*. **57**(4). DOI: 10.1007/s10462-023-10652-8

Atasu, K., Parnell, T., Dünner, C., Sifalakis, M., Pozidis, H., Vasileiadis, V., Vlachos, M., Berrospi, C., & Labbi, A. (2017, December). Linear-complexity relaxed word Mover's distance with GPU acceleration. In 2017 *IEEE International Conference on Big Data (Big Data)* (pp. 889-896). DOI:10.1109/BigData.2017.8258005

Batta. (2024). Human Language Data Processing using NLTK. *International Journal of Advanced Research in Science, Communication and Technology*. (pp. 628–634). DOI:10.1007/979-8-8688-1582-9_4

Challa, D. (2024). Comprehensive Review of One-Class Classification Approaches for Anomaly Detection. *INTERNATIONAL JOURNAL OF COMPUTER APPLICATIONS*. **186**(45): 69-74. DOI:10.5120/ijca2024924134

Chang, K.H.(2023). Natural Language Processing: Recent Development and Applications. *Applied Sciences*. **13**(20): 11395. DOI: 10.3390/app132011395

Chaves, A., Kesiku, C., & Garcia-Zapirain, B. (2022). Automatic text summarization biomedical text data: A systematic review. *Information*. **13**(8): 393. DOI: 10.3390/info13080393

Chen, Y., & Si, M. (2023). Enhancing Sentiment Analysis Results through Outlier Detection Optimization. *Machine Learning*. DOI: 10.48550/arXiv.2311.16185

Chiche, A., & Yitagesu, B. (2022). Part of speech tagging: a systematic review of deep learning and machine learning approaches. *Journal of Big Data*. **9**(1): 10. DOI:10.1186/s40537-022-00561-y

Chowdhary, K. (2020). Natural language processing. *Fundamentals of artificial intelligence*. (pp. 603–649). DOI: 10.22610/imbr.v16i3(I)S.3949

Dubey, S. R., Singh, S. K., & Chaudhuri, B. B. (2022). Activation functions in deep learning: A comprehensive survey and benchmark. *Neurocomputing*. **503**(11):92-108. DOI:10.1016/j.neucom.2022.06.111

Elzein, H. O. (2024). Association of Leukemia With ABO Blood Group Distribution and Discrepancy: A Review Article. *Cureus*, **16**(3):56812. DOI: 10.7759/cureus.56812

Fitriani, F., Faisol, A., Nuryaman, A., Kurniasari, D., & Utami, B. H. S. (2024). Pelatihan Latex Menggunakan Overleaf Untuk Meningkatkan Kemampuan Penulisan Karya Ilmiah Bagi Dosen Di Pringsewu. *Jurnal Pengabdian Kepada Masyarakat (JPKM) TABIKPUN*. **5**(3): 251-258. DOI: 10.23960/jpkmt.v5i3.184

Gupta, R. K., Agarwalla, R., Naik, B. H., Evuri, J. R., Thapa, A., & Singh, T. D. (2022). Prediction Of Research Trends Using LDA Based Topic Modeling. *Global Transitions Proceedings*. **3**(1): 298-304. DOI: 10.1016/j.gltp.2022.03.015

Goldstein, M., & Uchida, S. (2016). A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PloS one*, **11**(4): e0152173. DOI: 10.1371/journal.pone.0152173

Hanum, A. R., Zetha, I. A., Putri, S. C., Wulandari, R. A., Andina, S. P., Fajrina, J. N., & Yudistira, N. (2024). Analisis Kinerja Algoritma Klasifikasi Teks Bert Dalam Mendeteksi Berita Hoaks. *Jurnal Teknologi Informasi dan Ilmu Komputer*. **11**(3): 537-546. DOI: 10.25126/jtiik.938093

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers. R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., Del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W.,

Abbasi, H., Gohlke, C., & Oliphant, T. E. (2020). Array Programming With Numpy. *Nature*. **585**(7825): 357-362. DOI: 10.1038/s41586-020-2649-2

Heckelmann, P., Breuer, S. C., & Rinderknecht, S. (2025). Investigation of different LSTM-based encoder-decoder neural networks for vehicle speed prediction. *Scientific Reports*. **15**(1): 32662. DOI:10.1038/s41598-025-19592-5

Hu, C., Feng, Y., Kamigaito, H., Takamura, H., & Okumura, M. (2021). One-class Text Classification with Multi-modal Deep Support Vector Data Description. *In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2021)*. (pp. 3378–3390). DOI: 10.18653/v1/2021.eacl-main.296

Jain, A., Kulkarni, G., & Shah, V. (2018). Natural Language Processing. *International Journal of Computer Sciences and Engineering*. **6**(1): 2347-2693. DOI: 10.26438/ijcse/v6i1.161167

Jha, A., Dave, M., & Madan, S. (2019). Comparison of Binary Class and Multi Class Classifier Using Different Data Mining Classification Techniques. *In Proceedings of International Conference on Advancements in Computing & Management (ICACM 2019)*. SSRN. (pp. 894–903). DOI: 10.2139/ssrn.3464211

Johnson, C. (2022). Binary Encoded Word Mover's Distance. *In Proceedings of the 7th Workshop on Representation Learning for NLP*. (pp. 167-172). DOI: 10.18653/v1/2022.repl4nlp-1.17

Joseph, V. R., & Vakayil, A. (2022). SPlit: An Optimal Method For Data Splitting. *Technometrics*. **64**(2): 166-176. DOI: 10.1080/00401706.2021.1921037

Khairani, U., Mutiawani, V., & Ahmadian, H. (2024). Pengaruh tahapan preprocessing terhadap model Indobert dan Indobertweet untuk mendeteksi emosi pada komentar akun berita Instagram. *Jurnal Teknologi Informasi dan Ilmu Komputer*. **11**(4): 887-894. DOI: 10.25126/jtiik.1148315

Khan, S. S., & Madden, M. G. (2014). One-class classification: taxonomy of study and review of techniques. *The Knowledge Engineering Review*. **29**(3) 345-374. DOI: 10.1017/S026988891300043X

Khoirunnisaa, N., Kesuma, K. N. N., Setiawan, S., & Yusuf, A. Y. P. (2024). Klasifikasi Teks Ulasan Aplikasi Netflix Pada Google Play Store Menggunakan Algoritma Naive Bayes dan SVM. *SKANIKA: Sistem Komputer dan Teknik Informatika*. **7**(1): 64-73. DOI: 10.36080/skanika.v7i1.3138

Kilickaya, S., Ahishali, M., Celebioglu, C., Sohrab, F., Eren, L., Ince, T., Askar, M., & Gabbouj, M. (2024). Audio-based anomaly detection in industrial machines using deep one-class support vector data description. *IEEE Symposium on Computational Intelligence on Engineering/Cyber Physical Systems Companion (CIES Companion)*. pp. 1-5. DOI: 10.1109/CIESCompanion65073.2025.11010815

Komadina, A., Martinić, M., Groš, S., & Mihajlović, Ž. (2024). Comparing threshold selection methods for network anomaly detection. *IEEE access*. **12**: 124943-124973. DOI:10.1109/ACCESS.2024.3452168

Kumar, B., Dalal, N., & Sethi, M. (2024). Hyperparameters in Deep Learning: A Comprehensive Review. *International Journal of Intelligent Systems and Applications in Engineering*. **12**(4): 4015–4023. *Retrieved from*: https://www.ijisae.org/index.php/IJISAE/article/view/6967

Kurniasari, D., Usman, M., Warsono, & Lumbanraja, F. R. (2024). Comparative analysis of deep Siamese models for medical reports text similarity. *International Journal of Electrical and Computer Engineering*. **14**(6): 6969–6980. DOI: 10.11591/ijece.v14i6.pp6969-6980

Kusner, M. J., Sun, Y., Kolkin, N. I., & Weinberger, K. Q. (2015). From Word Embeddings To Document Distances. *In Proceedings of the 32nd International Conference on Machine Learning (ICML)*. **37**. (Pp. 957–966). *Retrieved from*: https://proceedings.mlr.press/v37/kusnerb15.html

Lavanya, A., Gaurav, L., Sindhuja, S., Seam, H., Joydeep, M., Uppalapati, V., Ali, Waqas., & SD, V. S. (2023). Assessing The Performance Of Python Data Visualization Libraries: A Review. *Int. J. Comput. Eng. Res. Trends*. **10**(1): 28-39. DOI: 10.22362/ijcert/2023/v10/i01/v10i0104

Lenz, O. U., Peralta, D., & Cornelis, C. (2022). Optimised One-Class Classification Performance. *Machine Learning*. **111**(8): 2863–2883. DOI: 10.1007/s10994-022-06147

Liang, X., Wu, L., Li, J., Wang, Y., Meng, Q., Qin, T., Chen, W., Zhang, M., & Liu, T. Y. (2021). R-drop: Regularized Dropout For Neural Networks. *Advances in neural information processing systems*. **34**: (pp. 10890-10905). DOI: 10.48550/arXiv.2106.14448

Lu, Z. (2011). PubMed and beyond: a survey of web tools for searching biomedical literature. *Database*. **2011**: 036. DOI: 10.1093/database/baq036

Markoulidakis, I., Kopsiaftis, G., Rallis, I., & Georgoulas, I. (2021). Multi-Class Confusion Matrix Reduction Method And Its Application On Net Promoter Score Classification Problem. *In Proceedings of the 14th PErvasive technologies related to assistive environments conference*. (pp. 412-419). DOI: 10.3390/technologies9040081

Mohammed, H. H., Dogdu, E., Görür, A. K., & Choupani, R. 2020. Multi-Label Classification of Text Documents Using Deep Learning. 2*020 IEEE International Conference on Big Data*. **8**(1): 4681-4689. DOI: 10.1109/BigData50022.2020.9378266

Matin, I. M. M. 2023. Hyperparameter Tuning Menggunakan GirdSearchCV pada Random Forest untuk Deteksi Malware. *MULTINETICS*. **9**(1): 43-50. DOI: 10.32722/multinetics.v9i1.5578

McKinney, W. (2011). Pandas: A Foundational Python Library For Data Analysis And Statistics. Python for high performance and scientific computing. **14**(9): 1-9.

Mohialden, Y. M., Kadhim, R. W., Hussien, N. M., & Hussain, S. A. K. (2024). Top python-based deep learning packages: A comprehensive review. *International Journal Papier Advance and Scientific Review*. **5**(1): 1-9. DOI: 10.47667/ijpasr.v5i1.283

Mufida, E., Andriansyah, D., & Hertyana, H. (2025). Customer Churn Prediction Pada Sektor Perbankan Dengan Model Logistic Regression dan Random Forest. *Computer Science (CO-SCIENCE)*. **5**(1): 58-66. DOI: 10.31294/coscience.v5i1.7576

Narayanasamy, S. K., Hu, Y. C., Qaisar, S. M., & Srinivasan, K. (2022). Effective Preprocessing and Normalization Techniques for COVID-19 Twitter Streams with POS Tagging via Lightweight Hidden Markov Model. *Journal of Sensors*. **2022**(1): 1222692. DOI: 10.1155/2022/1222692

Neumann, M., King, D., Beltagy, I., & Ammar, W. (2019, August). ScispaCy: fast and robust models for biomedical natural language processing. *In Proceedings of the 18th BioNLP workshop and shared task* (pp. 319-327). DOI: 10.18653/v1/W19-5034

Pang, G., Shen, C., Cao, L., & Hengel, A. V. D. (2021). Deep learning for anomaly detection: A review. *ACM computing surveys (CSUR)*. **54**(2): 1-38. DOI:10.1145/3439950

Rahma, I. A., & Suadaa, L. H. (2023). Penerapan text augmentation untuk mengatasi data yang tidak seimbang pada klasifikasi teks berbahasa Indonesia: Studi kasus deteksi judul clickbait dan komentar hate speech pada berita online. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*. **10**(6): 1329–1340. DOI: 10.25126/jtiik.2023107325

Ruff, L., Vandermeulen, R. A., Görnitz, N., Deecke, L., Siddiqui, S. A., Binder, A., Müller, E., & Kloft, M. (2018). Deep one-class classification. *Proceedings of the 35th International Conference on Machine Learning*. (pp. 4393–4402). https://proceedings.mlr.press/v80/ruff18a.html

Ruff, L., Zemlyanskiy, Y., Vandermeulen, R., Schnake, T., & Kloft, M. (2019). Self-attentive, multi-context one-class classification for unsupervised anomaly detection on text. *In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 4061-4071). DOI: 10.18653/v1/P19-1398

Salehi, M. M., & Zarei, H. (2025). A Review of the Structure and Application of Scikit-Learn Datasets in Machine Learning Model Development. *International Journal of Operations Research and Artificial Intelligence*. **1**(2): 90-100. DOI: 10.48314/ijorai.v1i2.64

Sathyanarayanan, S., & Tantri, B. R. (2024). Confusion matrix-based performance evaluation metrics. *African Journal of Biomedical Research*. **27**(4): 4023-4031. DOI: 10.53555/AJBR.v27i4S.4345

Seliya, N., Abdollah Zadeh, A., & Khoshgoftaar, T. M. (2021). A literature review on one-class classification and its potential applications in big data. *Journal of Big Data*. **8**: 122. DOI: 10.1186/s40537-021-00514-x

Septiani, D., & Isabela, I. (2022). Analisis Term Frequency Inverse Document Frequency (TF-IDF) dalam Temu Kembali Informasi pada Dokumen Teks. *Sistem dan Teknologi Informasi Indonesia (SINTESIA)*, **1**(2), 81-88.

Sung, H., Ferlay, J., Siegel, R. L., Laversanne, M., Soerjomataram, I., Jemal, A., & Bray, F. (2021). Global cancer statistics 2020: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA: A Cancer Journal for Clinicians*. **71**(3): 209–249. DOI: 10.3322/caac.21660

Suryadi, M. K., Herteno, R., Saputro, S. W., Faisal, M. R., & Nugroho, R. A. (2024). Comparative Study of Various Hyperparameter Tuning on Random Forest Classification with SMOTE and Feature Selection Using Genetic Algorithm In Software Defect Prediction. *Journal of Electronics, Electromedical Engineering, and Medical Informatics*. **6**(2): 137–147. DOI: 10.35882/jeeemi.v6i2.375

Tan, Z., Li, D., Wang, S., Beigi, A., Jiang, B., Bhattacharjee, A., Karami, M., Li, J., Cheng, L., & Liu, H. (2024). Large Language Models for Data Annotation and Synthesis: A survey. *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. (pp. 930-957). DOI: 10.18653/v1/2024.emnlp-main.54

Telnoni, P. A., Suryatiningsih, & Rosely, E. (2021). Pelabelan Data Dengan Latent Dirichlet Allocation dan K-Means Clustering Pada Data Twitter Menggunakan Bahasa Indonesia. *Jurnal Educatio: Jurnal Pendidikan Indonesia*. **7**(2): 885–892. DOI: 10.25124/jett.v7i2.3442

Toruan, J. H. L. (2022). *Konsep Dasar Teori Peluang*. Tasikmalaya: Perkumpulan Rumah Cemerlang Indonesia.

Vel, S. S. (2021). Pre-Processing Techniques of Text Mining Using Computational Linguistics and Python Libraries. *International conference on artificial intelligence and smart systems (ICAIS)*. (pp. 879-884). IEEE. DOI: 10.1109/ICAIS50930.2021.9395924

Wei, C., Wang, B., & Kuo, C.-C. J. (2022). SynWMD: Syntax-aware Word Mover's Distance for Sentence Similarity Evaluation. *Proceedings of the AAAI Conference on Artificial Intelligence*. **36**(10): 11287–11295. DOI: 10.48550/arXiv.2206.10029

Woo, H., Kim, J., & Lee, W. (2020). Validation Of Text Data Preprocessing Using A Neural Network Model. *Mathematical Problems in Engineering*. **2020**(1): 1958149. DOI: 10.1155/2020/1958149

Yadla, H. K., & Prasada Rao, P.V.R.D. (2020). Machine Learning Based Text Classifier Centered on TF-IDF Vectoriser. *International Journal of Scientific & Technology Research*. **9**(03): 2277-8616.

Zhang, Y. (2023). Encoder-decoder models in sequence-to-sequence learning: A survey of RNN and LSTM approaches. *Applied and Computational Engineering*. **22**(1): 218-226. DOI:10.54254/2755-2721/22/20231220

Zhang, Y., Chen, Q., Yang, Z., Lin, H., & Lu, Z. (2019). BioWordVec: Improving Biomedical Word Embeddings With Subword Information and Mesh. Scientific Data. 6(1): 52. DOI: 10.1038/s41597-019-0055-0