

***DENSITY-BASED* *CLUSTERING* UNTUK *ANOMALY DETECTION* PADA TRANSKRIP DISKUSI TERARAH KOMUNITAS**

**SKRIPSI**

Diajukan untuk memenuhi persyaratan penyelesaian program S-1

Jurusan Informatika Fakultas Teknologi Industri

Universitas Kristen Petra

Oleh :

Peter Yudhistira

NRP : C14190067

**PROGRAM STUDI INFORMATIKA**

**Logo

Description automatically generated**

**FAKULTAS TEKNOLOGI INDUSTRI**

**UNIVERSITAS KRISTEN PETRA**

**SURABAYA**

**2023**

# COVER

**SKRIPSI**

***DENSITY-BASED* *CLUSTERING* UNTUK *ANOMALY DETECTION* PADA TRANSKRIP DISKUSI TERARAH KOMUNITAS**

Oleh :

Peter Yudhistira NRP : C14190067

Diterima Oleh :

Program Studi Informatika

Fakultas Teknologi Industri

Universitas Kristen Petra

Surabaya, 29 Maret 2023

Pembimbing 1 Pembimbing 2

Dr. Gregorius Satiabudhi Alvin Nathaniel T., S.Kom., M.T.

NIP: 02-030 NIP: 21-011

Ketua Tim Penguji :

Henry Novianus Palit, Ph.D.

NIP: 14-001

Ketua Program Studi :

Henry Novianus Palit, Ph.D.

NIP: 14-001

# DAFTAR ISI

[COVER iii](#_Toc134429478)

[DAFTAR ISI iv](#_Toc134429479)

[DAFTAR TABEL vi](#_Toc134429480)

[DAFTAR GAMBAR vii](#_Toc134429481)

[DAFTAR RUMUS viii](#_Toc134429482)

[1. PENDAHULUAN 1](#_Toc134429483)

[1.1. Latar Belakang Masalah 1](#_Toc134429484)

[1.2. Perumusan Masalah 4](#_Toc134429485)

[1.3. Tujuan Penelitian 5](#_Toc134429486)

[1.4. Ruang Lingkup 5](#_Toc134429487)

[1.5. Metodologi Penelitian 6](#_Toc134429488)

[1.5.1. Studi Literatur 6](#_Toc134429489)

[1.5.2. Pengumpulan Dataset Penelitian 7](#_Toc134429490)

[1.5.3. Perencanaan Penelitian dan Pengembangan Program 7](#_Toc134429491)

[1.5.4. Pengujian Program 8](#_Toc134429492)

[1.5.5. Pengambilan Kesimpulan 8](#_Toc134429493)

[1.6. Sistematika Penulisan 8](#_Toc134429494)

[2. LANDASAN TEORI 10](#_Toc134429495)

[2.1. Tinjauan Pustaka 10](#_Toc134429496)

[2.1.1. Natural Language Processing (NLP) 10](#_Toc134429499)

[2.1.2. Text Preprocessing 10](#_Toc134429500)

[2.1.3. Density-Based Clustering 12](#_Toc134429501)

[2.1.4. *Density-Based Spatial Clustering of Applications with Noise* (DBSCAN) 13](#_Toc134429502)

[*2.1.5.* *Word Embedding* 16](#_Toc134429503)

[2.1.6. *Topic Modelling* 18](#_Toc134429504)

[2.1.7. TF-IDF 19](#_Toc134429505)

[2.2. Tinjauan Studi 20](#_Toc134429506)

[2.2.1. An evaluation of document clustering and topic modelling in two online social networks: Twitter and Reddit (Curiskis & al., 2019) 20](#_Toc134429513)

[2.2.2. A Comparison of LSA and LDA for the Analysis of Railroad Accident (Williams & Betak, 2019) 21](#_Toc134429514)

[2.2.3. Anomaly Detection: A Survey (Chandola, Banerjee, & Kumar, Anomaly Detection: A Survey, 2009) 21](#_Toc134429515)

[3. ANALISA DAN DESAIN SISTEM 23](#_Toc134429516)

[3.1. Analisa Permasalahan 23](#_Toc134429517)

[3.2. Desain Eksperimen 24](#_Toc134429518)

[3.2.1. Corpus Data Tranksrip 25](#_Toc134429519)

[3.2.1.1. Pengumpulan Data 26](#_Toc134429526)

[3.2.2. Text Preprocessing 27](#_Toc134429527)

[3.2.2.1. Case-Folding 28](#_Toc134429537)

[3.2.2.2. Stop Word Removal 28](#_Toc134429538)

[3.2.2.3. Stemming dan Lemmatizing 28](#_Toc134429539)

[3.2.2.4. Tokenization 28](#_Toc134429540)

[3.2.3. Feature Extraction 29](#_Toc134429541)

[3.2.3.1. Pendekatan Word- dan Sentence-Embedding 29](#_Toc134429542)

[3.2.3.2. Pendekatan Topic Modelling 29](#_Toc134429543)

[3.2.3.3. TF-IDF 30](#_Toc134429544)

[3.2.4. Density-based Clustering 30](#_Toc134429545)

[3.2.5. DBSCAN dan Parameter-Parameternya 30](#_Toc134429546)

[3.2.6. Fungsi Jarak yang Akan Digunakan 31](#_Toc134429547)

[3.2.7. Anomaly Detection and Validation 31](#_Toc134429548)

[3.3. Desain Eksperimen dan Program 31](#_Toc134429550)

[3.3.1. Alur Eksperimen 31](#_Toc134429551)

[3.3.1.1. Pengumpulan Data 31](#_Toc134429552)

[3.3.1.2. Text Preprocessing 32](#_Toc134429553)

[3.3.1.3. Feature Extraction 33](#_Toc134429554)

[3.3.1.4. Clustering & Anomaly Detection 35](#_Toc134429555)

[3.3.2. Alur Program 36](#_Toc134429556)

[3.3.2.1. Alur Keseluruhan Program 36](#_Toc134429567)

[3.3.2.2. Alur Pengumpulan Klip Suara 38](#_Toc134429568)

[3.3.2.3. Alur Proses dan Penyajian Data 40](#_Toc134429569)

[4. RESURRECTION 42](#_Toc134429570)

[DAFTAR PUSTAKA 42](#_Toc134429571)

# DAFTAR TABEL

[Tabel 3.1 Struktur Penyimpanan Data 25](#_Toc131709395)

# DAFTAR GAMBAR

[Gambar 1.1 Dokumentasi Pertemuan Diskusi 2](#_Toc131709400)

[Gambar 1.2 Contoh Notulen Diskusi 3](#_Toc131709401)

[Gambar 2.1 Ilustrasi *word tokenization*(Hvitfeld & Silge, 2022) 11](#_Toc131709402)

[Gambar 2.2 Perbandingan pendekatan *stemming* dan *lemmatization* (Aghammadzada, 2020)**.** 12](#_Toc131709403)

[Gambar 2.3 *Stop word removal* pada sejumlah kalimat Bahasa Inggris (GeekForGeeks, 2022) 12](#_Toc131709404)

[Gambar 2.4 Beberapa fungsi jarak (Xu & Tian, 2015) 13](#_Toc131709405)

[Gambar 2.5 Pseudocode Penerapan DBSCAN (Ester, Kriegel, Sander, & Xu, 1996) 14](#_Toc131709406)

[Gambar 2.6 Pseudocode Fungsi ExpandCluster() yang Dinyatakan dalam Pseudocode DBSCAN (Ester, Kriegel, Sander, & Xu, 1996). 15](#_Toc131709407)

[Gambar 2.7 Clustering dengan DBSCAN; poin-poin data hitam menunjukkan *outlier*(Scikit Learn)**.** 16](#_Toc131709408)

[Gambar 2.8 Visualisasi *Word2Vec*(Gautam, 2020)***.*** 16](#_Toc131709409)

[Gambar 2.9 Arsitektur *skip-gram Word2Vec*(Firdaus, 2019) 17](#_Toc131709410)

[Gambar 2.10 Arsitektur CBOW *Word2Vec*(Firdaus, 2019)***.*** 17](#_Toc131709411)

[Gambar 2.11 Diagram Model LDA (Lee & et.al., 2018) 18](#_Toc131709412)

[Gambar 3.1 Notulen diskusi yang telah terkumpul, disimpan dalam 14 *file*. 27](#_Toc131709413)

[Gambar 3.2 Diagram Alur Eksperimen – Pengumpulan Data 32](#_Toc131709414)

[Gambar 3.3 Diagram Alur Eksperimen – Text Preprocessing 33](#_Toc131709415)

[Gambar 3.4 Diagram Alur Eksperimen – *Feature Extraction* 35](#_Toc131709416)

[Gambar 3.5 Diagram Alur Eksperimen – *Clustering* & *Anomaly Detection* 36](#_Toc131709417)

[Gambar 3.6 State Diagram Program yang Akan Dibuat 37](#_Toc131709418)

[Gambar 3.7 *State Diagram* Alur Pengumpulan Klip Suara 38](#_Toc131709419)

[Gambar 3.8 Pseudocode Alur Pengumpulan Klip Suara 39](#_Toc131709420)

[Gambar 3.9 State Diagram Alur Proses dan Penyajian Data 40](#_Toc131709421)

[Gambar 3.10 Pseudocode Alur Proses dan Penyajian Data 40](#_Toc131709422)

# DAFTAR RUMUS

[Rumus 2.1 *Term Frequency* 19](#_Toc131669637)

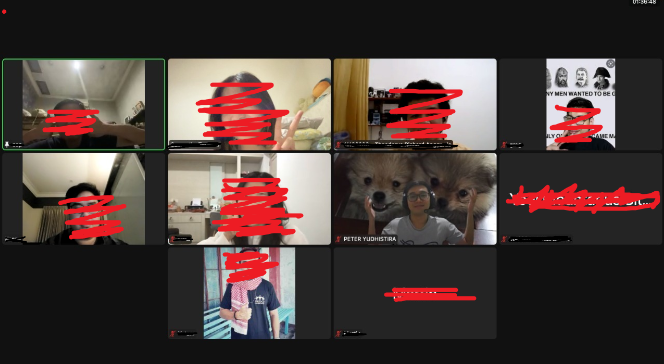
[Rumus 2.2 *Inverse Document Frequency* 20](#_Toc131669638)

[Rumus 2.3 TF-IDF 20](#_Toc131669639)

# PENDAHULUAN

## Latar Belakang Masalah

Sebuah forum adalah sebuah pertemuan atau medium yang memungkinkan para pesertanya bertukar pendapat dan gagasan akan sebuah permasalahan ("Forum", 2022). Forum yang diselenggarakan oleh sekelompok orang yang memiliki ketertarikan pada bidang yang sama disebut forum komunitas. Dalam sebuah forum komunitas, terjadi diskusi dan pertukaran pendapat antar anggota mengenai sebuah topik yang ada dalam bidang yang dimengerti oleh anggota-anggota komunitas tersebut. Sebuah diskusi harus melibatkan peran seorang fasilitator dan moderator – seorang fasilitator membantu dan menginspirasi proses-proses komunikasi untuk membangun pengetahuan, dan seorang moderator bertugas untuk menggabungkan kontribusi-kontribusi para peserta diskusi dengan cara mengambil pernyataan-pernyataan yang dilontarkan peserta dan mengaitkannya dengan topik dan teori dalam diskusi (Kienle & Ritterskamp, 2007). Pada Komunitas X yang diikuti penulis, terdapat sebuah forum yang diselenggarakan secara daring melalui *platform* Zoom Meetings. Diskusi-diskusi yang terjadi dalam forum ini dibantu dengan adanya fasilitator yang mengarahkan diskusi ke topik yang diinginkan, dan moderator yang memastikan koherensi pendapat setiap peserta diskusi. Selama berlangsungnya sebuah diskusi, seorang moderator juga berperan sebagai notulis yang mencatat pernyataan-pernyataan yang dikeluarkan oleh peserta diskusi. Catatan notulen ini akan dievaluasi untuk menilai kontribusi setiap peserta pada diskusi, serta sebagai dasar pengambilan keputusan untuk topik diskusi selanjutnya. Khususnya, catatan notulen digunakan untuk mengidentifikasikan pendapat-pendapat peserta yang tidak sesuai dengan topik diskusi.



Gambar 1.1 Dokumentasi Pertemuan Diskusi

Untuk menghargai hak privasi setiap anggota peserta diskusi, pihak penyelenggara diskusi tidak menggunakan fitur rekaman yang disediakan oleh platform Zoom Meeting. Oleh karena itu, dokumentasi berupa notulen dibuat oleh moderator atau seorang penyelenggara secara *real-time*. Notulis mencatat secara langsung pernyataan yang dikeluarkan oleh peserta diskusi, baik secara lisan maupun melalui fitur *chat* yang disediakan oleh platform *Zoom Meeting*. Notulis juga melakukan *filtering* pada pernyataan-pernyataan yang dicatatnya, yaitu melakukan sensor terhadap perkataan-perkataan yang kasar atau tidak berkenan, serta melakukan peringkasan terhadap pernyataan-pernyataan yang panjang. Pada akhir pernyataan, moderator menyampaikan ulang pendapat peserta yang baru saja dicatat secara lebih ringkas sebagai bentuk konfirmasi ketepatan catatan.

Graphical user interface, text

Description automatically generated

Gambar 1.2 Contoh Notulen Diskusi

Namun, terdapat beberapa kendala dalam sistem pencatatan dan evaluasi yang diterapkan oleh penyelenggara diskusi dalam Komunitas X. Pembuatan keputusan dan penilaian yang dilakukan secara terus menerus dapat menurunkan fungsi eksekutif dan sumber daya mental seseorang(Danziger, Levav, & Avnaim-Pesso, 2011). Karena panjangnya pendapat beberapa peserta, notulis bisa kehilangan fokus dan tidak mendokumentasikan pendapat peserta dengan baik. Seringkali notulis harus melakukan parafrase yang berdasarkan sumber yang tidak lengkap. Akibatnya, data yang dicatat tidak mencerminkan secara akurat pernyataan yang dikeluarkan oleh pembicara, dan mungkin mengandung bias, sehingga menghambat proses evaluasi. Selain itu, karena besarnya volume teks yang harus dievaluasi dan terbatasnya waktu kerja, komite penyelenggara diskusi mengalami kesulitan untuk mengidentifikasikan pendapat-pendapat peserta diskusi yang melenceng dari topik.

Oleh karena itu, diperlukan sebuah metode untuk mendapatkan transkrip diskusi, merepresentasikan fitur-fitur dari pendapat setiap peserta diskusi dan mendeteksi pendapat-pendapat yang melenceng dari topik diskusi. Untuk tujuan ini, akan digunakan metode *clustering* dan *topic modelling* untuk melakukan *anomaly detection* pada dataset transkrip diskusi Komunitas X. Hasil dari penelitian akan dievaluasi oleh komite penyelenggara diskusi di Komunitas X dan akan diterapkan dalam bentuk program untuk membantu proses diskusi.

Studi yang dilakukan oleh (Curiskis & al., 2019) mengevaluasi *clustering* dan *topic modelling* pada media sosial Twitter dan Reddit. Dalam studi tersebut, dikumpulkan dataset dari media sosial Twitter dan Reddit yang direpresentasikan dalam empat representasi fitur, yaitu TF-IDF, word2vec dengan *weight*, word2vec tanpa *weight*, dan doc2vec. Dilakukan pula empat metode *clustering*, yaitu *K-means* *Clustering*, *K-medoids* *clustering*, *hierarchical agglomerative clustering*, dan *Non-negative matrix factorization*. Kesuksesan penelitian diukur dengan metrik evaluasi intrinsik Normalized Mutual Information (NMI), dan Adjusted Mutual Information (AMI), dan metric evaluasi ekstrinsik Adjusted Rand Index (ARI). Ditemukan bahwa representasi fitur word2vec dan metode *K-means clustering* membuahkan hasil terbaik dibandingkan kombinasi-kombinasi lainnya.

Penelitian yang dilakukan oleh Wiliams & Betak (Williams & Betak, 2019) membandingkan performa *topic modelling* Latent Dirichlet Allocation (LDA) dengan Latent Semantic Analysis (LSA) dan menemukan bahwa kedua algoritma tersebut menemukan beberapa topik yang sama, namun ada beberapa topik yang eksklusif satu sama lain. Penelitian lainnya dilakukan untuk membandingkan performa LDA dan LSA terhadap satu sama lain menggunakan dataset berita BBC dan ditemukan bahwa dalam kasus ini, *Accuracy* algoritma LDA pada nilai 82.57 lebih tinggi dari *Accuracy* algoritma LSA pada nilai 75.30 (Kalepalli, Tasneem, Phani Teja, & Manne, 2020). Algoritma LDA ditemukan memiliki nilai *divergence* yang lebih besar daripada LSA, dan direkomendasikan untuk dataset berukuran lebih kecil. Studi yang dilakukan terhadap data *e-books* menunjukkan bahwa LDA memiliki performa yang lebih baik daripada LSA (Mohammed & Al-augby, 2020). Dalam kasus terbaiknya, LDA dengan asumsi jumlah topik 20 memiliki *coherence value* sebesar 0.592179, dibandingkan LSA dengan asumsi jumlah topik 10 memiliki *coherence value* sebesar 0.577302.

Survei yang dilakukan oleh Chandola, Banerjee & Kumar (Chandola, Banerjee, & Kumar, 2009) menyatakan beberapa metode yang dapat dilakukan untuk melakukan *anomaly detection*, diantaranya adalah penerapan *density-based clustering* untuk mengelompokkan data dalam beberapa *cluster*. Poin-poin data yang menjadi *outlier* diidentifikasikan sebagai titik-titik anomali.

## Perumusan Masalah

Berdasarkan latar belakang yang tertulis, dapat dirumuskan permasalahan sebagai berikut :

* Metode *feature extraction* manakah yang lebih baik untuk mencerminkan pendapat peserta diskusi sebagai acuan untuk *anomaly detection*?
* Kombinasi parameter dan *hyperparameter* seperti apakah yang dapat melakukan *anomaly* *detection* yang sesuai dengan kebutuhan penyelenggara diskusi?

## Tujuan Penelitian

Tujuan skripsi ini adalah membandingkan *feature extraction* berbasis *embedding* dan *topic modelling* dengan LDA untuk melakukan *anomaly detection* pada transkrip diskusi Komunitas X dengan pendekatan *density-based clustering*. Pendekatan yang membuahkan hasil terbaik menurut evaluasi komite penyelenggara diskusi akan diterapkan dalam sebuah program yang dalam *pipeline-*nya mendapatkan transkrip diskusi, melakukan *pre-processing* dan vektorisasi, lalu mengidentifikasikan pendapat-pendapat yang tidak sesuai dengan topik bahasan diskusi.

## Ruang Lingkup

Ruang lingkup dibatasi pada :

1. Menggunakan bahasa pemrograman *Python* dan *library-library* yang dibuat dalam bahasa *Python*.
2. Pembuatan dan uji coba algoritma akan dilakukan pada *IDE Visual Studio Code* dan *Jupyter Notebook*.
3. *Library* yang digunakan adalah *NLTK* dan *gensim* untuk *Natural Language Processing* dan *sklearn* untuk model-model *Machine Learning*. Digunakan pula beberapa *library* yang akan ditentukan untuk pembuatan GUI, sampling transkrip audio, dan integrasi komponen-komponen program.
4. Komponen-komponen berupa *speech-to-text* dan *translator* menggunakan API yang sudah ada.
5. Dataset berupa pernyataan yang tercatat dalam notulen diskusi. Satu rekor data berupa sebuah dokumen yang berisi pendapat seorang peserta diskusi yang dicatat pada satu kesempatan berbicara.
6. Untuk poin 5), dataset didapatkan dari proses transkripsi diskusi menggunakan program *speech-to-text* yang sudah ada.
7. Dataset penelitian untuk mengembangkan sistem *anomaly detection* didapatkan dari gabungan transkrip manual yang sudah dikumpulkan oleh tim penyelenggara dari sesi-sesi terdahulu.
8. Untuk poin 5), bahasa yang digunakan dalam diskusi dan pembuatan materi adalah Bahasa Inggris, dan respon-respon yang diberikan dalam Bahasa Indonesia atau bahasa campuran akan terlebih dahulu diterjemahkan ke Bahasa Inggris menggunakan API yang sudah ada.
9. Akan dilakukan *feature extraction* berupa *vectorization* melalui *word*- dan *sentence-embedding*, serta *topic modelling* untuk mengubah dokumen menjadi vektor numerik.
10. Akan juga diambil *term-document matrix* dengan metode TF-IDF sebagai metode *weighting* untuk pendekatan *vectorization* dan *topic modelling*.
11. Proses *anomaly detection* akan dilakukan dengan algoritma *Density-Based Spatial Clustering of Applications with Noise* (DBSCAN). *Outlier-outlier* yang terbentuk dari proses ini akan diidentifikasikan sebagai anomali dalam dataset.
12. Hasil penelitian ini berupa suatu program yang telah disetujui penggunaannya oleh penyelenggara diskusi Komunitas X sejak usulan ini dibuat, dan akan digunakan untuk membantu proses evaluasi diskusi dalam Komunitas X.

## Metodologi Penelitian

Langkah-langkah dalam pengerjaan skripsi meliputi hal di bawah ini:

### Studi Literatur

* + Mempelajari prinsip-prinsip *Natural Language Processing* (NLP).
  + Mempelajari *library-library* yang menerapkan proses-proses NLP, khususnya NLTK dan *gensim*.
  + Mempelajari konsep dan teknik *vectorization* berupa *word-embedding* dan *sentence-embedding*.
  + Mempelajari konsep, teknik, dan penerapan Latent Dirichlet Allocation sebagai metode *feature extraction*.
  + Mempelajari konsep dan penerapan TF-IDF.
  + Mempelajari definisi anomali dalam data dan mengeksplorasi metode-metode yang dapat digunakan untuk mencari anomali dalam data teks.
  + Mempelajari konsep, teknik, dan penerapan metode-metode *clustering* yang ada, khususnya *density-based clustering*.

### Pengumpulan Dataset Penelitian

* + Pengumpulan dataset berupa notulen diskusi yang berupa teks.
  + Pengambilan data berupa transkrip audio diskusi yang akan diolah untuk menghasilkan transkrip berupa teks.
  + Penerjemahan data transkrip menjadi Bahasa Inggris untuk memudahkan proses-proses NLP.
  + *Cataloging* dataset, dikelompokkan berdasarkan tanggal *event* dan peserta diskusi yang berbicara.

### Perencanaan Penelitian dan Pengembangan Program

* Mengumpulkan data transkrip diskusi yang sudah ada, baik berupa teks atau transkrip audio.
* Mengubah data transkrip audio menjadi teks dengan API speech-to-text yang sudah ada, serta melakukan penerjemahan ke dalam Bahasa Inggris secara semi-manual dengan menggunakan bantuan API *translation* yang sudah ada.
* Menyusun data transkrip diskusi dengan struktur yang sesuai, dalam rekor-rekor yang memiliki identifikasi berupa nama peserta, identifikasi acara diskusi, dan pendapat peserta yang berbicara.
* Segmentasi data berdasarkan kategori-kategori yang ada sebagai variabel eksperimen. Akan dilakukan segmentasi teks sebagai berikut:
  + Kontribusi total peserta dalam sebuah diskusi. Satu poin data mewakili sebuah dokumen yang menggabungkan seluruh kontribusi tiap-tiap peserta dalam satu sesi diskusi.
  + Sesi berbicara. Setiap poin data mewakili transkrip yang didapatkan dari satu kesempatan seorang peserta berbicara dalam sebuah sesi diskusi.
  + Sesi berbicara dan kalimat. Setiap poin data mewakili transkrip yang didapatkan dari satu kesempatan seorang peserta berbicara dalam sebuah sesi diskusi, tapi dipisahkan per kalimat. Segmentasi ini dilakukan untuk mengubah matriks TF-IDF.
* Melakukan *preprocessing* pada data transkrip berupa *case folding*, *tokenizing*, *lemmatizing*, dan *stop-words removal*. Akan diuji kombinasi-kombinasi dari teknik *preprocessing* untuk menentukan kombinasi *preprocessing* yang menghasilkan *corpus* yang paling baik performanya.
* *Word-tokenizing* untuk mendapatkan *corpus* yang akan menjadi basis eksperimen.
* *Feature extraction* dengan pendekatan *vectorization* melalui *word embedding* dan *sentence embedding*. Akan dilakukan pengujian terhadap beberapa variabel untuk menentukan kombinasi dan parameter seperti apa yang menghasilkan fitur-fitur data yang baik performanya.
* *Feature extraction* dengan pendekatan *topic modelling*, yaitu dengan metode *Latent Dirichlet Allocation* (LDA). Akan dilakukan eksperimen dengan parameter jumlah topik yang diinferensikan untuk mencapai komposisi paling baik.
* Menerapkan *density-based clustering* untuk mendapatkan *cluster-cluster* data, dan mengidentifikasikan *outlier* pada data.
* Menggabungkan elemen-elemen yang sudah disebutkan diatas dalam sebuah data *pipeline* yang menerima data transkrip berupa teks dan mengidentifikasikan anomali-anomali yang ada dalam transkrip tersebut.
* Mengintegrasikan API-API *speech-to-text* dan *translation* yang ada untuk mengubah transkrip audio menjadi teks.
* Mendesain *use case*, *database*, dan *UI* program.

### Pengujian Program

* Menguji penggunaan program pada beberapa transkrip audio diskusi.
* Mengumpulkan 6 penyelenggara diskusi komunitas untuk mengevaluasi anomali yang diidentifikasikan oleh program pada transkrip-transkrip diskusi yang dipakai.

### Pengambilan Kesimpulan

* Membuat kesimpulan dari keputusan-keputusan yang dibuat oleh para penyelenggara diskusi komunitas.
* Merumuskan saran apa yang bisa dilakukan untuk penelitian berikutnya.
* Mengidentifikasikan kelemahan pada komponen-komponen program, bila ada, dan merumuskan alternatif yang mungkin memberikan performa lebih baik.

## Sistematika Penulisan

Penulisan laporan skripsi ini dibagi menjadi beberapa bab, yaitu :

* Bab I : Pendahuluan

Bab ini berisikan judul, latar belakang, perumusan masalah, ruang lingkup, tujuan skripsi, dan metodologi penelitian yang akan digunakan dalam skripsi ini.

* Bab II : Landasan Teori

Bab ini berisikan teori-teori yang digunakan dan diterapkan dalam skripsi ini.

* Bab III : Analisa Dan Desain Sistem

Menjelaskan analisa masalah yang dihadapi dan perencanaan pembuatan keseluruhan sistem dalam aplikasi yang akan dibuat.

* Bab IV : Implementasi Sistem

Bab ini berisikan tentang implementasi sistem berdasarkan desain.

* Bab V : Pengujian Sistem

Bab ini berisi tentang hasil pengujian yang dilakukan terhadap aplikasi yang telah dibuat berdasarkan implementasi pada sistem yang telah dirancang dan dibuat pada Bab IV.

* Bab VI : Kesimpulan Dan Saran

Bab ini berisikan kesimpulan yang dapat diambil terhadap hasil yang dicapai, dan saran-saran yang berguna bagi pengembangan selanjutnya.

# LANDASAN TEORI

## Tinjauan Pustaka

Pada bab ini akan dijelaskan mengenai teori-teori yang akan digunakan dalam penulisan skripsi dan pembuatan aplikasi.



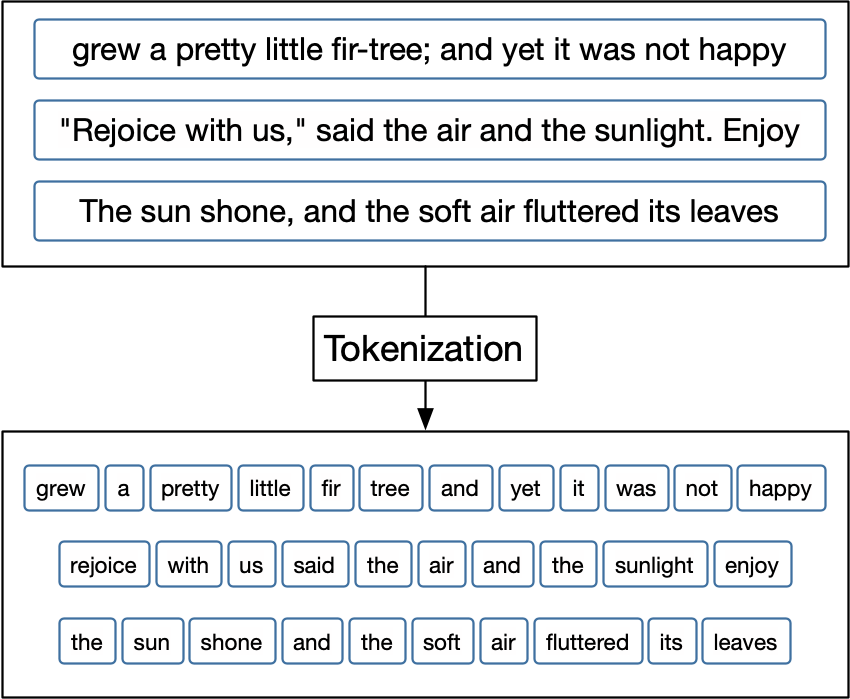
### Natural Language Processing (NLP)

*Natural Language Processing* (NLP) adalah sebuah bidang dalam ilmu komputer dan ilmu bahasa yang mempelajari interaksi antara komputer dan bahasa natural manusia (Kumar, 2013). NLP mencakup studi dalam berbagai tingkatan dalam bahasa manusia, yaitu fonologi, morfologi, leksikon, sintaktik, semantik, pembicaraan, dan pragmatik (Ribeiro, 2021). Penerapan-penerapan NLP dapat berupa Word Processing, pengecekan dan perbaikan ejaan, Information Retrieval (IR), Information Extraction, Information Categorization, penjawaban pertanyaan, Information Summarization, dan Machine Translation (Chowdhary, 2020). Dalam penelitian ini, teknik-teknik NLP dipakai untuk memroses data transkrip yang didapatkan dari sebuah diskusi melalui program *speech-to-text*.

### Text Preprocessing

Dalam NLP, *Text Preprocessing* adalah teknik yang digunakan untuk mempersiapkan teks yang tidak terstruktur menjadi data yang baik dan siap untuk diolah (Katryn, 2020). *Preprocessing* penting untuk dilakukan untuk mengurangi ukuran file teks dokumen yang akan dianalisa (Kannan, Gurusamy, Vijayarani, Ilamathi, & Nithya, 2014). Penggunaan teknik-teknik preprocessing dilakukan sesuai dengan karakteristik data teks dan kebutuhan penelitian. Teknik-teknik yang digunakan dalam penelitian ini antara lain *tokenization*, *stemming, lemmatization*, dan *stop word removal*.

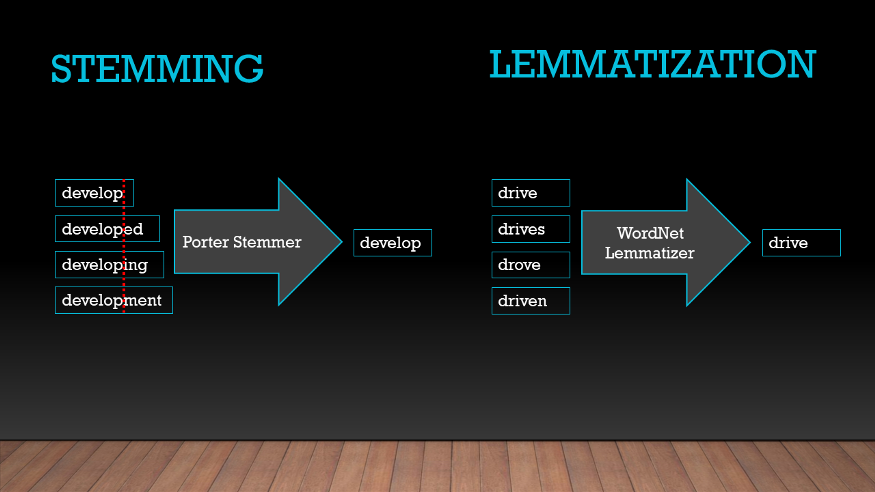
*Tokenization* adalah sebuah proses memecah sebuah dokumen menjadi kumpulan kata, frasa, simbol, atau elemen-elemen lainnya yang disebut token (Kannan, Gurusamy, Vijayarani, Ilamathi, & Nithya, 2014). Secara umum, ada tiga tipe token yang bisa didapatkan dari sebuah dokumen, yaitu *word*, *character*, dan *subword* (Pai, 2020). Token bertipe *word* memecah sebuah dokumen menjadi sekumpulan kata, dan *subword* memecah sebuah dokumen menjadi sekumpulan kata yang dipecah ke bentuk dasar dan imbuhannya. Untuk penelitian ini, dilakukan *word-tokenization*, yaitu pemecahan dokumen menjadi kumpulan kata-kata yang akan direpresentasikan dalam vektor pada proses selanjutnya. Proses *word-tokenization* ini juga dilakukan untuk mempermudah *lemmatizing* dan *stemming* yang akan dilakukan sebagai bagian dari *preprocessing*.



Gambar 2.1 Ilustrasi *word tokenization*(Hvitfeld & Silge, 2022)

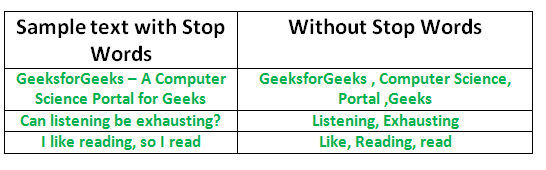
*Stemming* adalah sebuah proses mengubah varian sebuah kata menjadi kata dasarnya, yang disebut *stem* atau batang (Kannan, Gurusamy, Vijayarani, Ilamathi, & Nithya, 2014). Teknik ini dilakukan untuk generalisasi kata kerja, supaya kata-kata kerja dalam bentuk bentuk berbeda tidak dianggap token-token yang berbeda. Dalam penelitian yang diajukan, karena keseluruhan data transkrip ada dalam Bahasa Inggris, dapat diterapkan algoritma Porter Stemming yang diterapkan oleh Martin Porter. Algoritma ini memanfaatkan beberapa prinsip dan aturan untuk menghilangkan *suffix* (akhiran) pada setiap kata untuk mendapatkan stem dari kata tersebut. Algoritma ini dimaksudkan untuk meningkatkan kinerja *information retrieval* dan mungkin memproduksi sebuah *stem* yang sama untuk dua kata dengan makna yang berbeda (Porter, 1980).

*Lemmatization* adalah proses pengelompokan bentuk-bentuk terinfleksi dari sebuah kata menjadi satu bentuk dasar (*lemma*) (Dictionary, n.d.). Tidak seperti *stemming* yang menggunakan pendekatan berdasarkan karakter, pendekatan reduksi dengan *lemmatization* bergantung pada pengetahuan tata bahasa yang bersangkutan untuk mengenali *lemma* sebuah kata infleksi. Oleh karena itu, proses ini membutuhkan proses *POS tagging* untuk mengenali bentuk kata (contohnya: “working” sebagai kata kerja atau sebagai kata sifat –tergantung konteks kalimat) untuk melakukan klasifikasi dengan benar. Dengan pendekatan ini, makna kata dapat dipertahankan dan tidak ambigu dengan *lemma* lainnya. Dalam penelitian yang akan dijalankan, akan diuji pendekatan *stemming* dan *lemmatization* secara terpisah untuk mendapatkan hasil yang lebih baik.



Gambar 2.2 Perbandingan pendekatan *stemming* dan *lemmatization* (Aghammadzada, 2020)**.**

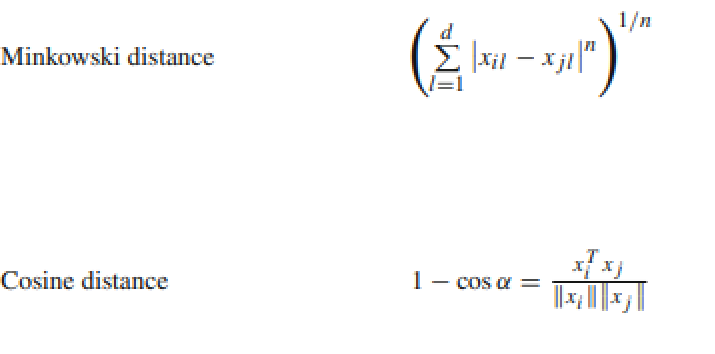
*Stop word removal* adalah sebuah proses penghapusan kata-kata yang frekuensi kemunculannya tinggi, namun tidak memberikan makna yang signifikan pada dokumen. *Stop word* dalam Bahasa Inggris dapat mencapai 20 hingga 30 persen jumlah kata total dalam sebuah teks dokumen (Kannan, Gurusamy, Vijayarani, Ilamathi, & Nithya, 2014). Penghapusan *stop-words* pada sebuah korpus bertujuan untuk mengurangi *noise* dan token-token kata yang redundan.



Gambar 2.3 *Stop word removal* pada sejumlah kalimat Bahasa Inggris (GeekForGeeks, 2022)

### Density-Based Clustering

*Clustering* atau *cluster analysis* adalah sebuah metode *unsupervised* *machine learning* yang bertujuan untuk menemukan hubungan dan pengelompokan yang sudah ada pada data (Brownlee, 2020). Metode ini disebut *unsupervised* karena tidak membutuhkan label pada data; *clustering* berfungsi untuk mengelompokkan data dan menarik kesimpulan berdasarkan kemiripan-kemiripan fitur data tersebut. Ada beberapa pendekatan yang bisa diambil untuk melakukan *clustering*, diantaranya *centroid-based clustering*, *density-based clustering*, *distribution-based clustering*, dan *hierarchical clustering* (Google, 2022). Masing-masing pendekatan tersebut. Konsep fundamental *clustering* adalah mengidentifikasikan kelompok-kelompok yang terbentuk dari kemiripan antar rekor data. Terutama untuk data kuantitatif, digunakan fungsi yang menentukan jarak antar poin data yang dimiliki (Xu & Tian, 2015). Terdapat beberapa algoritma untuk menentukan jarak yang dipakai dalam *machine learning*, diantaranya adalah Euclidian Distance, Cosine Distance, dan Mahalanobis Distance.



Gambar 2.4 Beberapa fungsi jarak (Xu & Tian, 2015)

Pendekatan yang diambil untuk suatu kasus tertentu harus sesuai dengan kebutuhan penelitian dan sifat dari dataset yang dimiliki, dan oleh *tools* yang tersedia untuk melakukan *clustering* (Witten, Frank, Hall, & Pal, 2016). Dalam penelitian yang diajukan, akan dilakukan *anomaly detection* dengan mengidentifikasikan outlier pada data. Oleh karena itu, *density-based clustering* akan digunakan karena tidak memasukkan *outlier* ke *cluster* manapun. (Google, 2022), dan menentukan cluster-cluster berdasarkan kedekatan relatif antar poin data. Untuk tujuan ini, akan digunakan metode DBSCAN (Density Based Spatial Clustering of Applications with Noise) untuk memproduksi *cluster-cluster* pada dataset, dengan harapan ditemukannya poin-poin data yang tidak tergabung dalam cluster manapun sebagai *outlier.* Adapun akan diuji secara paralel fungsi-fungsi jarak yang ada untuk menentukan pendekatan yang paling efektif dan sesuai dengan kebutuhan penelitian.

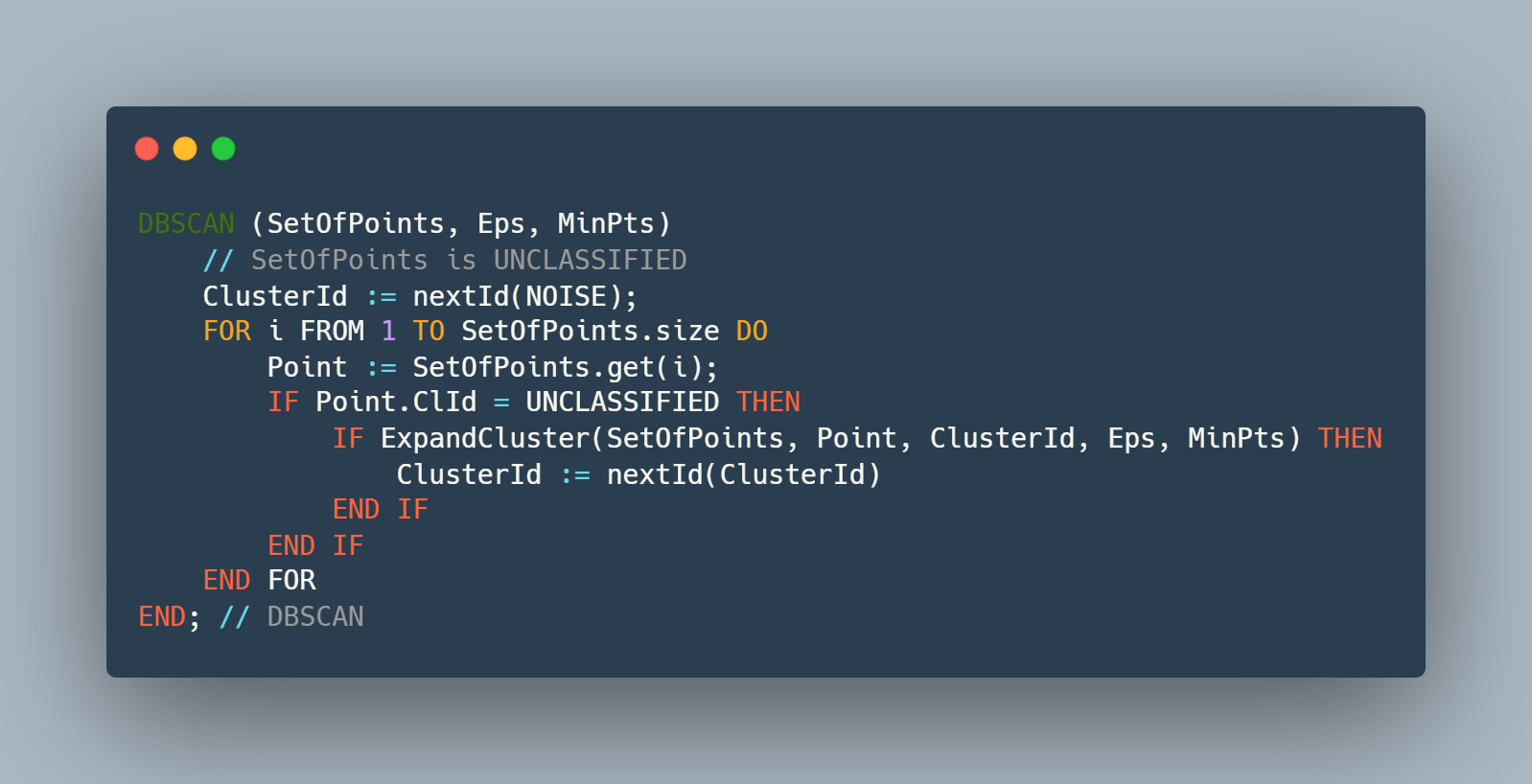
### *De**nsity-Based Spatial Clustering of Applications with Noise* (DBSCAN)

DBSCAN adalah sebuah algoritma *density-based clustering* yang membuat *cluster-cluster* pada poin-poin data yang tidak berlabel berdasarkan kedekatannya. Algoritma ini dibuat sebagai realisasi pendekatan intuitif yang dapat disimpulkan ketika mengevaluasi poin-poin data yang memiliki kedekatan (density) tertentu yang dapat memisahkan sebuah *cluster* dengan *noise* (Ester, Kriegel, Sander, & Xu, 1996).

Algoritma DBSCAN menerima dua parameter yang ditentukan oleh pengguna, yaitu *MinPts* dan *ε* (eps; epsilon)*.*

Adapun poin-poin data dikategorikan berdasarkan parameter yang ada sebagai berikut:

* + - *Core point*, jika ada *MinPts* buah poin data yang terletak dalam radius *ε* di sekelilingnya.
    - Poin data *p* dikatakan *directly density-reachable point* dari poin data *q*, jika poin data *p* memiliki jarak maksimum *ε* dari *q.*
    - Poin data *p* dikatakan *density-reachable point* dari poin data *q*, jika ada sebuah jalur atau path di antara poin data *p* dan *q* berupa poin-poin data yang berjarak maksimum *ε* antar satu sama lain.
    - *Outlier* atau *noise point*, jika poin data tersebut tidak reachable dari poin-poin data lainnya.



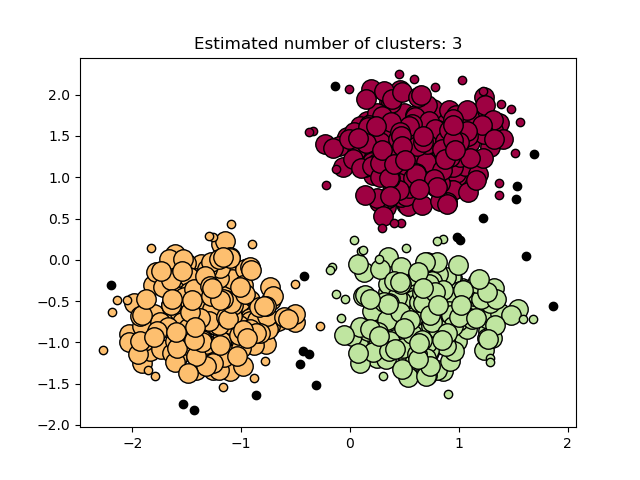
Gambar 2.5 Pseudocode Penerapan DBSCAN (Ester, Kriegel, Sander, & Xu, 1996)



Gambar 2.6 Pseudocode Fungsi ExpandCluster() yang Dinyatakan dalam Pseudocode DBSCAN (Ester, Kriegel, Sander, & Xu, 1996).

DBSCAN diterapkan secara iteratif untuk setiap poin yang ada dalam dataset. Setiap iterasi menentukan apakah diperlukan adanya *cluster* baru atau tidak, dengan cara menghitung poin-poin yang berdekatan dengan poin data yang bersangkutan. Dalam Gambar 2.5., perhitungan dilakukan dengan menjalankan fungsi ExpandCluster() yang dijelaskan dalam Gambar 2.6. Dalam fungsi ExpandCluster(), akan dicari poin-poin yang merupakan *core point*s.Poin-poin tersebut akan diiterasi secara rekursif untuk mencari dan menandai *density-reachable points* yang ada di sekitarnya. Poin-poin yang tidak memenuhi kriteria akan ditandai sebagai *noise* atau *outlier.*

DBSCAN mendefinisikan sebuah *cluster* sebagai kumpulan poin-poin data yang *reachable* antar satu sama lain. *Outlier* adalah poin data yang tidak *reachable* dari poin-poin data lainnya. Oleh karena itu, DBSCAN akan digunakan dalam penelitian ini untuk mengidentifikasikan anomali dalam data berupa data point yang dianggap *noise* atau *outlier*.



Gambar 2.7 Clustering dengan DBSCAN; poin-poin data hitam menunjukkan *outlier*(Scikit Learn)**.**

### *Word Embedding*

*Word embedding* adalah metode untuk menangkap makna semantik dan sintaktik kata-kata yang ada dalam sebuah korpus teks yang tidak dilabeli (Lai, Liu, He, & Zhao, 2016). Metode ini dilakukan untuk melakukan *encoding* setiap word token yang telah dikumpulkan menjadi sebuah vektor bilangan riil yang bisa direpresentasikan dalam sebuah dimensi tertentu. Vektor-vektor bilangan yang didapatkan dari hasil *embedding* dapat ditransformasikan menjadi vektor dengan dimensi lebih kecil melalui sebuah proses yang disebut *dimensionality reduction*.

Diagram, table

Description automatically generated

Gambar 2.8 Visualisasi *Word2Vec*(Gautam, 2020)***.***

*Word2Vec* adalah suatu metode untuk membangun *Word Embedding* secara efisien berupa *neural network* yang terdiri dari dua *layer* untuk melakukan proses pengubahan teks menjadi vektor. *Word2Vec* memiliki 2 pilihan arsitektur yaitu *CBOW* (*Continuous Bag of words*) dan *Skip-Gram*.

Diagram

Description automatically generated

Gambar 2.9 Arsitektur *skip-gram Word2Vec*(Firdaus, 2019)

Diagram

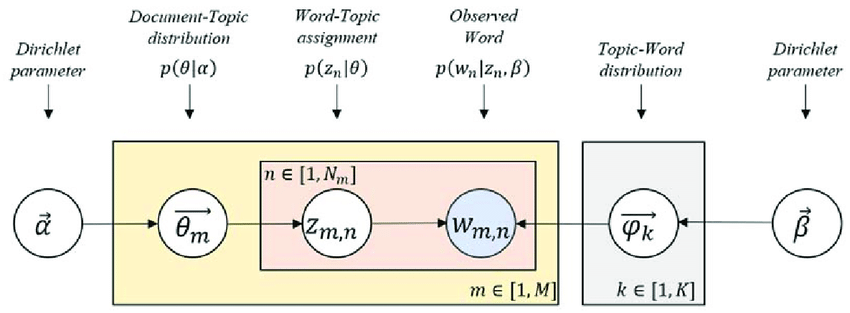
Description automatically generated

Gambar 2.10 Arsitektur CBOW *Word2Vec*(Firdaus, 2019)***.***

Arsitektur *skip-gram* memprediksi kata-kata konteks yang ada di sekitar kata utama. Sebaliknya, arsitektur *CBOW* mencoba memprediksi *encoding* sebuah kata berdasarkan kata-kata konteks di sekitarnya. Dengan arsitektur-arsitektur ini, *Word2Vec* memprediksi konteks sebuah word token dalam korpus berdasarkan sejumlah kata-kata konteks yang berdekatan dengan kata pilihan; banyaknya kata yang diambil sebagai konteks disebut dengan istilah *window*. Dalam proses *training*, pendekatan ini akan berusaha mengatur *weight* dari setiap kata sedekat mungkin dengan makna aslinya. Kata-kata yang dekat maknanya akan memiliki vektor representasi yang berdekatan dalam *embedding* *Word2Vec* (Firdaus, 2019). Dalam studi yang akan dilakukan, sebuah dokumen terdiri dari kumpulan *word-token*. Akan dilakukan *sentence-embedding* untuk mengagregasikan word-embedding untuk setiap token dan mendapatkan sebuah vektor n-dimensi yang merepresentasikan sebuah dokumen.

### *Topic Modelling*

*Topic Modelling* adalah sebuah algoritma yang mengidentifikasikan pola-pola tersemat dalam kata-kata dalam sekumpulan dokumen (Jacobi, van Atteveldt, & Welbers, 2015). Proses ini merupakan sebuah penerapan *unsupervised machine learning* yang digunakan untuk memodelkan topik-topik yang terkandung dalam sebuah dokumen secara probabilistik. Salah satu *topic modelling* yang populer adalah *Latent Dirichlet Allocation* (LDA). Dalam penelitian yang diajukan ini, distribusi topik per dokumen yang didapatkan dari proses *topic modelling* dengan LDA akan dianggap sebagai fitur-fitur data.



Gambar 2.11 Diagram Model LDA (Lee & et.al., 2018)

Keterangan:

* + - α, β : distribusi Dirichlet
    - m : dokumen ke-m dalam M buah dokumen
    - M : jumlah total dokumen
    - n : kata ke-n dalam N buah kata
    - N : jumlah total kata
    - k : topik ke-k dalam K buah topik
    - K : jumlah total topik
    - θm ­: distribusi topik dalam dokumen ke-m
    - φk : distribusi kata dalam topik ke-K
    - zm,n : topik yang didapatkan dari θm ­­yang diberikan ke kata ke-n
    - wm,n : kata ke-n dalam dokumen ke-m

LDA adalah sebuah model statistika generatif yang dibuat untuk mengekstraksi variabel-variabel yang tersemat untuk mencari kemiripan dalam data. Metode ini dilakukan dengan mengambil secara stokastik sampel-sampel topik dari kumpulan topik per dokumen serta kata dari kumpulan kata per dokumen, lalu menggabungkannya dalam sebuah dokumen baru yang mengandung kata-kata dan topik yang mencerminkan dokumen awalnya (Blei, Ng, & Jordan, 2001). Dalam satu iterasi untuk generasi topik LDA, akan dicari distribusi kata-kata untuk tiap topik pada kumpulan topik K, dengan memperhatikan distribusi Dirichlet β. Akan juga dicari distribusi topik-topik untuk tiap dokumen pada kumpulan dokumen M dengan memperhatikan distribusi Dirichlet α. Untuk setiap dokumen m pada kumpulan dokumen M, akan diberikan sebuah topik yang dihasilkan dari distribusi dokumen-topik sebelumnya, dan sebuah kata yang didapatkan dari distribusi topik-kata sebelumnya, untuk menginferensikan distribusi topik per dokumen.

Proses ini akan diiterasi sebanyak yang diinginkan, untuk menghasilkan konvergensi statistika. Dalam penelitian yang akan dilakukan, akan diuji variasi jumlah topik yang akan menghasilkan distribusi topik dengan dimensi yang berbeda-beda. Diharapkan ditemukan jumlah topik yang optimal, sehingga menciptakan vektor fitur yang reflektif atas setiap dokumen.

### TF-IDF

Term Frequency—Inverse Document Frequency (TF-IDF) adalah sebuah metode untuk memberikan *weight* sebuah kata dalam sebuah dokumen dalam sebuah korpus. Metode ini menggabungkan nilai *Term Frequency* (TF) yang mengevaluasi seberapa sering sebuah kata muncul dalam sebuah dokumen dibandingkan kata-kata lainnya, dan nilai *Inverse Document Frequency* (IDF) yang mengekspresikan perbandingan terbalik (inverse) jumlah dokumen dimana *term* atau kata tertentu muncul (Salton & Buckley, 1988).

TF dinyatakan dalam rumus sebagai berikut :

Rumus 2.1 *Term Frequency*

Dimana:

* + - t adalah *term* atau kata tertentu
    - d adalah dokumen tertentu
    - adalah *term frequency* kata t dalam dokumen d.
    - adalah jumlah iterasi kata t dalam dokumen d
    - adalah jumlah seluruh kata yang ada dalam dokumen d

IDF dinyatakan dalam rumus sebagai berikut:

Rumus 2.2 *Inverse Document Frequency*

Dimana :

* + - adalah *inverse document frequency* kata t dalam kumpulan dokumen D
    - adalah jumlah dokumen dalam korpus
    - adalah jumlah cacah dokumen d dalam kumpulan dokumen D, dimana kata t ada dalam dokumen d

TF-IDF dinyatakan dalam matriks hasil *dot-product* sebagai berikut:

Rumus 2.3 TF-IDF

Dalam penelitian yang akan dilakukan, *term-document matrix* yang dihasilkan dari proses TF-IDF untuk *corpus* yang akan dikumpulkan berfungsi sebagai augmentasi untuk *feature extraction*. *Term-document matrix* akan digunakan sebagai *weight* untuk proses agregasi kumpulan *word-embedding* dalam sebuah dokumen untuk menghasilkan sebuah *sentence-embedding* yang mewakili dokumen tersebut. Dalam LDA, *term-document* matrix akan digunakan sebagai input inisial yang akan mempengaruhi distribusi dan sampling untuk *topic-document matrix* dan *word-topic matrix*.

## Tinjauan Studi

Berikut merupakan penelitian yang telah dilakukan sebelumnya :








### An evaluation of document clustering and topic modelling in two online social networks: Twitter and Reddit (Curiskis & al., 2019)

Penelitian sebelumnya yang dilakukan oleh (Curiskis & al., 2019) melakukan perbandingan *clustering* dan *topic modelling* pada data *tweet* dari media sosial Twitter dan *thread* dari media sosial Reddit. Dataset direpresentasikan dalam empat representasi fitur, yaitu *TF-IDF*, *word2vec* dengan *weight*, word2vec tanpa *weight*, dan *doc2vec.* Dilakukan teknik *clustering* pada data dengan empat metode, yaitu *K-means* *Clustering*, *K-medoids* *clustering*, *hierarchical agglomerative clustering*, dan *Non-negative matrix factorization*. Kesuksesan hasil *clustering* dalam penelitian ini diukur dengan NMI, AMI, dan ARI. Ditemukan bahwa performa paling baik didapatkan oleh representasi fitur dengan *word2vec* dan metode *k-means clustering*.

Penelitian yang diajukan memiliki beberapa perbedaan dengan penelitian yang dilakukan oleh (Curiskis & al., 2019). Pada penelitian yang diajukan, setiap rekor data berupa pernyataan peserta diskusi yang mungkin berisi lebih dari satu kalimat, sedangkan sebuah rekor data *tweet* pada *Twitter* tidak melebihi 255 karakter. Oleh karena itu, diatas *word-embedding* *word2vec*, perlu juga dilakukan *sentence embedding*. Dalam penelitian yang akan dilakukan akan dicoba dua variasi feature extraction, yaitu *word*- dan *sentence-embedding*, serta *topic modelling* dengan LDA. Fitur-fitur data akan menjadi input algoritma *density-based clustering* dengan asumsi pengetahuan dalam *domain* yang mencukupi, dan dengan harapan bahwa hasil *clustering* akan mencerminkan cluster-cluster yang ada dalam data dan memproduksi *outlier*. Selain itu, karena dataset yang tersedia dan yang akan dikumpulkan dalam penelitian ini tidak memiliki label, tidak ada asumsi *ground truth*. Oleh karena itu, pengukuran kesuksesan *clustering* akan dilakukan dengan validasi oleh pengguna.

### A Comparison of LSA and LDA for the Analysis of Railroad Accident (Williams & Betak, 2019)

Penelitian selanjutnya (Williams & Betak, 2019) membandingkan performa *topic modelling* LDA dengan LSA, dan menemukan bahwa kedua metode menemukan beberapa topik yang eksklusif satu sama lain. Penelitian ini juga menyatakan bahwa algoritma LDA direkomendasikan untuk dataset yang berukuran lebih kecil. Dalam penelitian yang akan diajukan, LDA akan digunakan untuk mengekstraksi topik dari setiap dokumen yang berupa pendapat peserta diskusi. Distribusi topik dalam tiap dokumen akan dianggap sebagai fitur-fitur data yang menjadi basis density-based clustering untuk mencari poin-poin data yang merupakan *outlier*.

### Anomaly Detection: A Survey (Chandola, Banerjee, & Kumar, Anomaly Detection: A Survey, 2009)

Survei yang dilakukan oleh Chandola, Banerjee & Kumar (Chandola, Banerjee, & Kumar, Anomaly Detection: A Survey, 2009) menyatakan beberapa metode yang dapat dilakukan untuk melakukan *anomaly detection*, diantaranya adalah penerapan *density-based clustering* untuk mengelompokkan data dalam beberapa *cluster*. Poin-poin data yang menjadi *outlier* diidentifikasikan sebagai titik-titik anomali. Dalam penelitian yang diajukan, akan dilakukan *density-based clustering* pada vektor-vektor fitur data yang telah didapatkan dari fase *feature extraction* yang telah dilakukan. *Outlier-outlier* yang terbentuk oleh *clustering* yang dilakukan akan diidentifikasikan sebagai anomali dalam data dan diuji validitasnya.

# ANALISA DAN DESAIN SISTEM

Pada bab ini akan lebih dibahas lebih lanjut mengenai analisa dan desain dan rancangan sistem yang digunakan untuk mendeteksi.

## Analisa Permasalahan

Diskusi terstruktur adalah sebuah proses dimana sekelompok orang berkumpul untuk berbagi informasi, gagasan, dan mengkaji sebuah topik menurut sebuah format yang telah ditentukan. Dalam sebuah diskusi terstruktur, terdapat sebuah rangkaian acara yang didesain untuk menggiring jalannya diskusi terhadap topik yang telah dipilih agar mencapai hasil yang diinginkan. Dalam Komunitas X yang merupakan komunitas multi-agama dengan mayoritas beragama Kristen, dilakukan diskusi terstruktur secara rutin mengenai topik-topik yang berkenaan dengan agama, moralitas, dan etika. Diskusi-diskusi terstruktur yang diadakan oleh Komunitas X mengharapkan para pesertanya mengikuti rangkaian dan peraturan-peraturan yang telah ditentukan oleh penyelenggara, meliputi giliran berbicara, kesesuaian dengan topik, kesopanan dalam menyampaikan kontribusi, dan larangan untuk melakukan interupsi.

Diskusi terstruktur dalam Komunitas X dipimpin oleh seorang fasilitator, dan diawasi oleh seorang moderator. Fasilitator bertugas untuk melakukan eksposisi terhadap topik yang diambil untuk sesi diskusi tersebut, dan menyampaikan pemicu diskusi menurut rincian subtopik diskusi yang telah dipersiapkan sebelumnya. Adapun moderator bertugas untuk menjembatani kontribusi-kontribusi peserta diskusi dengan topik yang sedang dibahas, serta menyaring dan menegur peserta jika kontribusi yang ditawarkan tidak sesuai dengan peraturan dan pedoman diskusi. Moderator juga bertugas untuk membuat notulen diskusi untuk mencatat kontribusi setiap peserta dalam sebuah sesi diskusi. Notulen diskusi ini akan dievaluasi pada pertemuan mingguan untuk menilai apakah kontribusi setiap peserta sesuai dengan harapan, atau malah melenceng dari pertanyaan-pertanyaan pemicu dan tidak menghasilkan input sesuai harapan.

Terdapat beberapa kendala dalam sistem pencatatan dan evaluasi yang diterapkan oleh penyelenggara diskusi dalam Komunitas X, baik pada waktu sesi diskusi maupun rapat evaluasi. Pada sebuah sesi diskusi, moderator menjalankan tugasnya sembari merangkap menjadi notulis. Seringkali, panjangnya kontribusi seorang peserta membuat moderator harus melakukan parafrase agar dapat mencatatnya tepat waktu. Akibatnya, data yang dicatat mungkin mengandung bias moderator dan tidak mencerminkan pendapat peserta yang sebenarnya, dan akan berpengaruh pada proses evaluasi. Pada rapat evaluasi, terdapat kesulitan untuk mengidentifikasikan pendapat-pendapat yang tidak sesuai dengan topik karena data notulen yang terparafrase dengan bias moderator atau kontribusi peserta tertentu yang panjang sehingga sulit untuk dicari esensinya.

Keperluan yang perlu untuk ditangani adalah deteksi kontribusi peserta yang bersifat ganjil (*anomalous*). Anomali atau keganjilan yang diamati berupa ketidaksesuaian konten pendapat peserta diskusi dengan subtopik yang ditawarkan dalam sebuah sesi diskusi. Asumsi didasarkan oleh pengamatan bahwa kontribusi-kontribusi yang ganjil umumnya berasal dari sejumlah kecil individu yang menyatakan pendapat yang melebar dari topik, dan sebagian besar anggota diskusi umumnya menawarkan kontribusi yang sesuai dengan topik dan pertanyaan-pertanyaan pemicu yang diajukan. Oleh karena itu, akan dilakukan *unsupervised learning* untuk mendeteksi kecenderungan *cluster* yang ada dalam data transkrip diskusi.

Studi yang akan dilakukan bertujuan untuk menciptakan sebuah *pipeline* data yang melakukan transkripsi kontribusi setiap peserta, menerjemahkannya ke Bahasa Inggris, melakukan *preprocessing* yang sesuai, lalu mengekstraksi fitur dengan pendekatan *word-* dan *sentence-embedding,* serta *topic modelling*. Setelah fitur-fitur berhasil diekstraksi, akan dilakukan pendekatan *density-based* *clustering* untuk mengamati kecenderungan *cluster* yang terbentuk dari kontribusi-kontribusi peserta dalam sebuah sesi, dan mengidentifikasikan *outlier*. Studi diadakan untuk menguji apakah pendekatan yang dilakukan dengan komponen-komponen yang digabungkan dapat menghasilkan *outlier* yang mencerminkan anomali data yang diharapkan oleh penyelenggara.

## Desain Eksperimen

Secara garis besar, eksperimen yang akan dilakukan bertujuan untuk membuat sebuah *pipeline* inti yang menggabungkan komponen-komponen *preprocessing*, *feature extraction*, dan *density-based clustering* serta *hyperparameter*-nya untuk mengelompokkan data transkrip ke dalam *cluster-cluster* yang mencerminkan kedekatan setiap kontribusi peserta dalam sebuah sesi diskusi. Setelah itu, akan diidentifikasikan *outlier-outlier* pada dataset sebagai anomali –kontribusi peserta diskusi yang ganjil dan melenceng dari bahasan topik. Eksperimen yang akan dilakukan akan berfokus pada *feature extraction* –*vectorization* dengan *word-* dan *sentence-embedding*, dan *topic modelling* dengan LDA, serta pencarian *hyperparameter-hyperparameter* yang tepat pada LDA dan DBSCAN. Dalam eksperimen yang dilakukan, diharapkan dapat dicari *range* parameter yang dapat mengidentifikasikan *outlier* yang sesuai dengan definisi yang diharapkan oleh penyelenggara diskusi dalam komunitas.

### Corpus Data Tranksrip

Data transkrip yang digunakan untuk menjalankan eksperimen tersusun dari data teks notulen dari beberapa sesi diskusi, serta transkrip audio dari beberapa diskusi yang telah diadakan*.* Transkrip audio akan diubah menjadi transkrip teks dengan menggunakan API Google Speech-to-text. Bagian-bagian yang berbahasa Indonesia akan diterjemahkan dengan menggunakan API penerjemahan DeepL. Adapun data transkrip yang sudah berupa teks akan disimpan dengan format yang sudah ditentukan. Berikut sebuah tabel yang berisi beberapa rekor data:

Tabel 3.1 Struktur Penyimpanan Data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Question | Answer | Event | Speaker |
| 1 | How did we come to exist? | Everybody should believe that they are creations, made by a creator. It is a prerequisite to believing anything else that comes after. | 19-Feb | P |
| 2 | How did we come to exist? | We came from evolution. This is undeniable, and the Charles Darwin's evolution model is the most widely accepted theory to this. We evolved from unicellular bacteriae, to fish, to amphibious creatures, to mammals, to apes, then all the way to humans. | 19-Feb | G |
| 4 | Do you think you need to exist? | I don't think we need to exist. As Peter said, everything would just go on as usual, regardless of your existence or not. | 19-Feb | Je |
| 39 | What is the difference between a mistake and a sin? | If a man does wicked things while not being sane, he also sins. | 26-Feb | G |
| 46 | What sin leads to hell? | I think there are no sins that are greater or lesser than others, all sin leads to hell. | 26-Feb | P |
| 69 | We see from our side that maybe they are miserable. But how do we get the right to decide who lives and who doesn't? | Is it wrong to regret life? That is the right of the individual. The fetus is special because of the human species, even though it may have the same abilities as a chimpanzee. Life goes on, no matter how bad it is. | 2-Jul | G |
| 70 | We see from our side that maybe they are miserable. But how do we get the right to decide who lives and who doesn't? | Actually, we humans are well aware that life is really difficult. Only he will know what his difficulties and limitations are. It is precisely because we know life is difficult. Science is not based on morals, while we humans have moral standards. Can everything that is presented with standards and data refer to justice? Everything has a consequence. Does that consequence make it better or worse? Not prolife and prochoice but what is from God is what is lived. | 2-Jul | Yot |

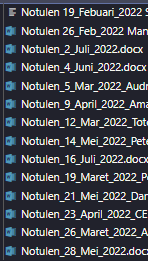
Legenda tabel:

* + - Kolom “No” menyatakan nomor rekor data, dan akan digunakan untuk mengidentifikasikan dokumen pada eksperimen yang akan dilakukan.
    - Kolom “Question” menyatakan pertanyaan pemicu yang menghasilkan kontribusi yang tercatat, dan akan digunakan untuk kategorisasi dokumen pada eksperimen yang akan dilakukan.
    - Kolom “Answer” berisi kontribusi peserta diskusi terhadap pertanyaan (question) yang menjadi pemicu. Konten dari kolom ini dianggap sebagai dokumen yang akan digunakan sebagai bahan utama eksperimen.
    - Kolom “Event” menandakan kode sesi diskusi dimana rekor data dicatat. Kolom ini akan digunakan untuk kategorisasi dokumen pada eksperimen yang akan dilakukan.
    - Kolom “Speaker” digunakan untuk mengidentifikasi peserta diskusi yang menghasilkan kontribusi yang tercatat. Identitas peserta diskusi disamarkan untuk menjaga privasi. Kode yang tercatat bisa ditafsirkan oleh penyelenggara diskusi.



#### Pengumpulan Data

Penyelenggara diskusi di Komunitas X menyimpan notulen-notulen diskusi yang telah dilakukan dalam bentuk teks. Terhitung ada 14 sesi diskusi yang telah tercatat dalam notulen diskusi. Kontribusi-kontribusi peserta yang tercatat dalam notulen-notulen diskusi ini akan digunakan sebagai bahan eksperimen. Selain itu, terhitung sejak 31 Maret 2023, telah terkumpul dua transkrip audio dari diskusi yang dilakukan oleh Komunitas X. Transkrip-transkrip audio disimpan dalam file *.ogg*, dan akan diubah menjadi transkrip teks terlebih dahulu sebelum diolah lebih lanjut.



Gambar 3.1 Notulen diskusi yang telah terkumpul, disimpan dalam 14 *file*.

Dataset berupa notulen akan dipilah dan dimasukkan ke dalam tabel dengan struktur yang telah dijelaskan pada Tabel 3.1, sesuai dengan pertanyaan pemicu, sesi diskusi, dan kode pembicara. Beberapa notulen yang berisi parafrase singkat dikembangkan berdasarkan gagasan-gagasan yang telah ditulis sehingga membentuk kalimat-kalimat yang koheren dan sedekat mungkin dengan gagasan parafrase yang ditulis. Adapun untuk data yang berupa transkrip audio, akan dilakukan dua kali proses transkripsi: secara manual dan dengan menggunakan API Speech-to-text Google. Hasil dari kedua transkripsi akan diterjemahkan dengan bantuan API DeepL ke dalam Bahasa Inggris dan akan dibandingkan dalam kerangka eksperimen yang sama.

Dengan menganggap kolom “Answer” dari setiap rekor data sebagai sebuah dokumen, terbentuklah sebuah *corpus* teks yang terdiri dari kumpulan kontribusi peserta diskusi terhadap pertanyaan pemicu (kolom “Question”). *Preprocessing*, *feature extraction*, dan *clustering* akan dilakukan terhadap *corpus* yang terbentuk.

### Text Preprocessing

Text Preprocessing dilakukan untuk mempersiapkan data teks yang tidak terstruktur untuk diolah (Katryn, 2020). Dalam penelitian ini, akan digunakan teknik-teknik *preprocessing* yang mempersiapkan data teks untuk *feature extraction* dengan pendekatan *word-embedding*, *sentence-embedding*, *TF-IDF*, dan *topic modelling*. Oleh karena itu, preprocessing yang dilakukan berfokus pada upaya mengurangi *noise* dan redundansi pada data untuk menghasilkan token-token yang berkualitas.









#### Case-Folding

Pada data teks akan dilakukan *case-folding*, yaitu pengubahan semua karakter huruf menjadi huruf kecil. Proses ini dilakukan untuk menghilangkan redundansi *string* yang menyebabkan *noise* pada data.

#### Stop Word Removal

Pada data teks akan juga dihapus kata-kata yang frekuensi kemunculannya tinggi, namun tidak bermakna signifikan pada dokumen. Proses ini dilakukan untuk mengurangi redundansi yang disebabkan oleh kata-kata yang berfrekuensi tinggi. Rincian kata-kata yang termasuk *stop words* akan didapatkan dari library NLP yang digunakan.

#### Stemming dan Lemmatizing

Stemming adalah proses pengubahan sebuah kata menjadi sebuah elemen dasar yang disebut *stem* atau batang (Kannan, Gurusamy, Vijayarani, Ilamathi, & Nithya, 2014). Proses ini dilakukan untuk generalisasi kata kerja, supaya variasi-variasi kata kerja dalam Bahasa Inggris (contohnya: “work” dan “working”) dapat dikenali sebagai token yang sama. Dalam eksperimen yang akan dilakukan, akan diterapkan algoritma Porter Stemming yang disediakan oleh *library* NLP yang digunakan.

Lemmatizing adalah proses pengelompokan dan pengubahan sebuah kata berdasarkan *lemma*(kata dasar)-nya. Proses lemmatizing akan mengembalikan sebuah token kata yang valid, namun membutuhkan komponen POS Tagging untuk menentukan unsur kata (contohnya: “working” dapat dikategorikan sebagai *verb* atau *adjective*, tergantung dengan konteks). Proses ini akan dilakukan dengan bantuan library NLP yang digunakan. Akan dilakukan perbandingan penggunaan *lemmatizing* dan *stemming* pada *pipeline* data yang akan dibuat.

#### Tokenization

Tokenization adalah proses pemecahan sebuah dokumen menjadi kumpulan token (Kannan, Gurusamy, Vijayarani, Ilamathi, & Nithya, 2014). Dalam studi ini, karena studi berada di tingkat *lexical*, setiap dokumen akan dipecah menjadi token-token kata (*word tokens*). Dengan mengubah sebuah dokumen menjadi sekumpulan token, beberapa proses, seperti stemming dan *lemmatizing* serta *feature extraction* akan lebih mudah dilakukan. Proses *tokenization* akan dilakukan dengan bantuan library NLP yang digunakan.

### Feature Extraction

Feature extraction adalah proses untuk mengubah setiap token yang sudah dibersihkan dengan *preprocessing* menjadi sekumpulan angka yang merepresentasikan token tersebut. Hal ini dilakukan agar token bisa menjadi input pada model *density-based clustering* yang akan digunakan. Dalam studi ini, akan dilakukan dua pendekatan *feature extraction*, yaitu *vectorization* dengan *word-* dan *sentence-embedding*, serta *topic modelling* dengan LDA. Kedua pendekatan akan diuji secara paralel untuk menentukan kinerja yang lebih baik untuk studi yang dilakukan.

#### Pendekatan Word- dan Sentence-Embedding

Pendekatan *word-embedding* digunakan untuk mengekspresikan makna semantik dan sintaktik token-token kata dalam sebuah corpus yang tidak dilabeli (Lai, Liu, He, & Zhao, 2016). Dengan pendekatan ini, setiap word token yang dihasilkan oleh proses *tokenization* akan memiliki representasi dalam bentuk sebuah vektor n-dimensi. Proses *word-embedding* akan dilakukan dengan metode *Word2Vec* yang sudah tersedia dalam *library* yang digunakan. Akan dilakukan perbandingan secara paralel pengujian beberapa variabel, diantaranya jumlah dimensi, serta model yang dibangun sendiri dan *pre-trained model*.

Adapun pendekatan *sentence-embedding* digunakan untuk menangkap makna sebuah kalimat atau dokumen yang terdiri dari sekumpulan *word token* yang sudah *di-embed* menjadi vektor. Vektor-vektor yang dihasilkan lewat *word-embedding* setiap token yang ada dalam satu dokumen akan diagregasikan menjadi satu *sentence vector* yang berdimensi sama. Akan diuji beberapa metode agregasi untuk menentukan kinerja yang paling baik. Metode-metode agregasi yang akan diuji adalah *averaging*, *sum*, *weighted-averaging*, dan *weighted-sum*; *weight* atau berat yang akan digunakan diambil dari nilai TF-IDF setiap token pada dokumen yang bersangkutan. Luaran dari proses ini adalah sebuah vektor berdimensi-n untuk setiap dokumen, dimana n adalah jumlah dimensi yang ditentukan oleh peneliti. Vektor dari setiap dokumen akan menjadi input untuk algoritma *density-based clustering* yang akan digunakan.

#### Pendekatan Topic Modelling

Metode *feature extraction* menggunakan *topic modelling* akan diuji sebagai alternatif dari pendekatan *embedding*. Dalam pendekatan ini, akan dicari pola-pola tersemat dalam token-token kata yang ada dalam sebuah dokumen (Jacobi, van Atteveldt, & Welbers, 2015), dengan asumsi jumlah topik tertentu. Pendekatan ini adalah pendekatan stokastik, sehingga dapat menghasilkan keputusan yang berbeda-beda dalam setiap operasi dengan parameter dan input yang sama. Akan diuji beberapa variasi parameter yang mempengaruhi algoritma LDA, diantaranya jumlah topik yang diasumsikan –akan diuji beberapa jumlah topik sesuai dengan studi literatur tentang LDA dan LSA (Williams & Betak, 2019). Akan diuji juga performa algoritma dengan input tambahan berupa TF-IDF sebagai *term-document matrix*. Luaran dari proses ini adalah sebuah vektor berdimensi-n, dimana n merupakan jumlah topik yang diinferensikan. Nilai dari setiap dimensi dari vektor yang terbuat melambangkan proporsi topik yang diinferensikan yang terkandung dalam dokumen yang bersangkutan. Kumpulan vektor yang didapatkan dari setiap dokumen akan menjadi input untuk algoritma *density-based clustering* yang akan digunakan.

#### TF-IDF

TF-IDF (Term Frequency-Inverse Document Frequency) adalah metode statistical untuk mendapatkan *term-document matrix* yang mewakili relevansi setiap kata pada setiap dokumen pada *corpus*. *Matrix* yang didapatkan dari proses TF-IDF untuk corpus yang menjadi bahan penelitian akan digunakan untuk augmentasi dan *weighting* metode-metode *feature extraction* yang digunakan, dengan harapan bahwa relevansi setiap token dicerminkan dari matriks yang didapatkan.

### Density-based Clustering

Setelah *feature extraction* dijalankan untuk setiap dokumen dalam *corpus*, akan dilakukan proses *density-based clustering* dengan DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*). Penggunaan metode ini bertujuan untuk mendapatkan *cluster-cluster* berdasarkan kedekatan vektor-vektor setiap dokumen yang telah didapatkan dengan metode *feature extraction* yang telah dipilih. *Feature extraction* yang baik akan menghasilkan data yang berkualitas dan merefleksikan dokumen sesungguhnya. Diharapkan akan ditemukan poin-poin data yang tidak tergabung dalam *cluster* manapun untuk diidentifikasikan sebagai *outlier*.

### DBSCAN dan Parameter-Parameternya

DBSCAN adalah *algoritma density-based clustering* yang membuat *cluster-cluster* berdasarkan kedekatannya (Ester, Kriegel, Sander, & Xu, 1996). Pendekatan ini dipilih karena bisa menghasilkan *outlier* dalam bentuk poin-poin data yang terletak jauh dari *cluster-cluster* yang terbentuk. Dalam algoritma ini, terdapat dua parameter yang dapat diubah-ubah, yaitu *MinPts* –jumlah poin data minimum yang ada di sekeliling sebuah poin data, dan *ε* –radius dimana *MinPts* diperhitungkan. DBSCAN akan diterapkan dan divisualisasikan dengan *library* yang telah disediakan.

### Fungsi Jarak yang Akan Digunakan

Fungsi jarak yang digunakan dapat mempengaruhi pembentukan cluster karena perbedaan karakteristik masing-masing fungsi jarak yang dikenali. Dalam penelitian ini, akan dibandingkan secara paralel beberapa fungsi jarak, diantaranya Euclidean Distance, Cosine Distance, dan Mahalanobis Distance.

### Anomaly Detection and Validation

Dalam eksperimen yang dilakukan, diasumsikan hipotesis utama bahwa *outlier-outlier* yang ditemukan adalah *lexical anomaly* pada data, dan mencerminkan kontribusi peserta diskusi yang ganjil atau tidak sesuai topik. Hasil dari eksperimen-eksperimen ini akan divalidasi oleh tim penyelenggara diskusi Komunitas X, yang berperan sebagai pengguna dan memiliki kapasitas dalam *domain* yang bersangkutan. Akan ditentukan apakah *outlier* yang ditemukan oleh program yang dibuat mencerminkan kontribusi yang ganjil atau tidak.



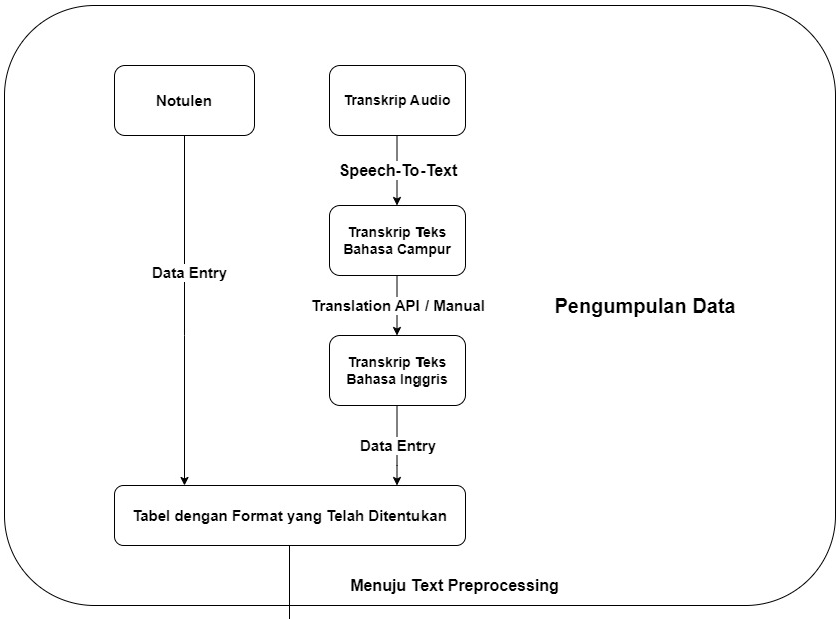
## Desain Eksperimen dan Program

Bagian ini akan menjelaskan tentang desain eksperimen untuk mendeteksi anomali, serta desain program implementasi yang akan dibuat.

### Alur Eksperimen

#### Pengumpulan Data

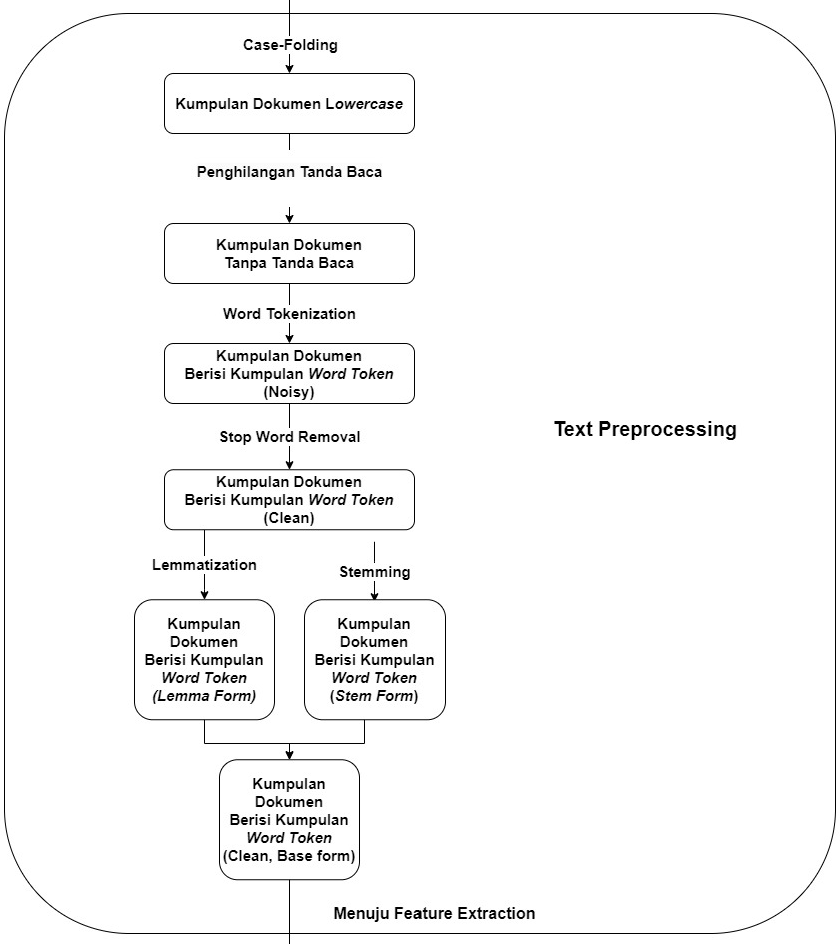
Pada tahap ini *corpus* dibangun dari notulen dan transkrip audio. Data dari Notulen akan langsung dimasukkan ke dalam tabel data dengan format yang telah ditentukan. Adapun transkrip yang berupa audio akan diubah dahulu menjadi transkrip teks melalui API Speech-to-text Google. Setelah itu, untuk input yang tidak berbahasa Inggris, akan diterjemahkan ke dalam Bahasa Inggris dengan API DeepL dan intervensi manual. Kedua proses ini akan dilakukan secara terpisah dan dijalankan secara paralel, untuk menguji efektivitas komponen API speech-to-text dan translation. Pada akhir proses ini, data yang tersimpan dalam tabel akan diakses dengan program dan dikenakan *text preprocessing*.



Gambar 3.2 Diagram Alur Eksperimen – Pengumpulan Data

#### Text Preprocessing

*Text preprocessing* dimulai dari *case-folding* untuk meratakan isi *corpus*, menjadikannya *lower case*. Setelah itu, akan dihilangkan tanda baca dan elemen-elemen nonesensial lainnya seperti angka yang tidak relevan. Dokumen-dokumen yang hanya berisi karakter-karakter alfabetik akan dijadikan kumpulan *word tokens* dengan fungsi *word tokenize*, menghasilkan dokumen-dokumen yang berisi kumpulan token. Untuk setiap dokumen, akan dijalankan proses *stop words removal*, untuk menghapus token-token yang frekuensinya tinggi namun tidak menawarkan banyak makna, untuk mengurangi *noise* dalam data. Selanjutnya, akan dijalankan secara paralel *stemming* dan *lemmatization*, untuk dibandingkan hasilnya. Kedua proses itu akan mengubah setiap dokumen menjadi kumpulan token yang berisi *stem* atau *lemma* dasar dari setiap kata, dan siap untuk memasuki tahap *feature extraction*.



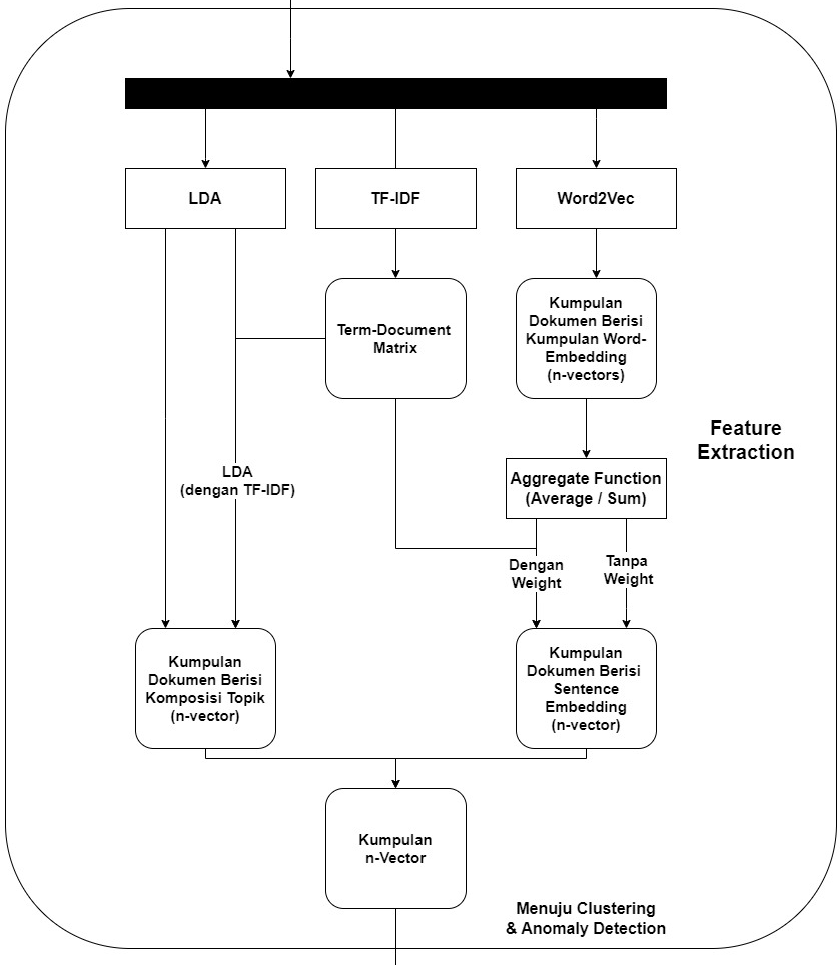
Gambar 3.3 Diagram Alur Eksperimen – Text Preprocessing

#### Feature Extraction

Untuk proses *feature extraction*, ada dua pendekatan utama yang akan diuji, yaitu pendekatan *topic modelling* lewat LDA, dan pendekatan *embedding* lewat *word-embedding* dan *sentence-embedding.* Akan juga dijalankan TF-IDF untuk mendapatkan *term-document matrix*, yang merupakan sebuah nilai numerik untuk setiap *word token* di setiap dokumen, sebagai variabel pengujian pada dua metode *feature extraction* yang dipakai. *Topic modelling* dan *word/sentence-embedding* akan dijalankan secara paralel untuk dibandingkan performanya.

Dalam pendekatan *embedding*, akan dilakukan *word-embedding* pada setiap token kata yang terkandung dalam setiap dokumen. Akan digunakan variasi model Word2Vec, dengan menggunakan beberapa *pre-trained* *model* yang sudah tersedia serta model yang dibangun sendiri. Akan dilakukan juga variasi jumlah dimensi vektor yang dihasilkan. Setelah setiap word-token berhasil di-embed, akan dilakukan agregasi pada setiap word-token yang dimiliki sebuah dokumen, untuk mendapatkan sebuah vektor yang merepresentasikan dokumen tersebut (*sentence-embedding*). Akan dicoba secara paralel dan dibandingkan metode *aggregation* dengan *averaging* dan *sum*, serta *weighted averaging* dan *weighted sum* dengan menggunakan *term-document matrix* sebagai *weight.* Proses agregasi ini akan menghasilkan sebuah *n-dimensional numerical vector* untuk setiap dokumen*.*

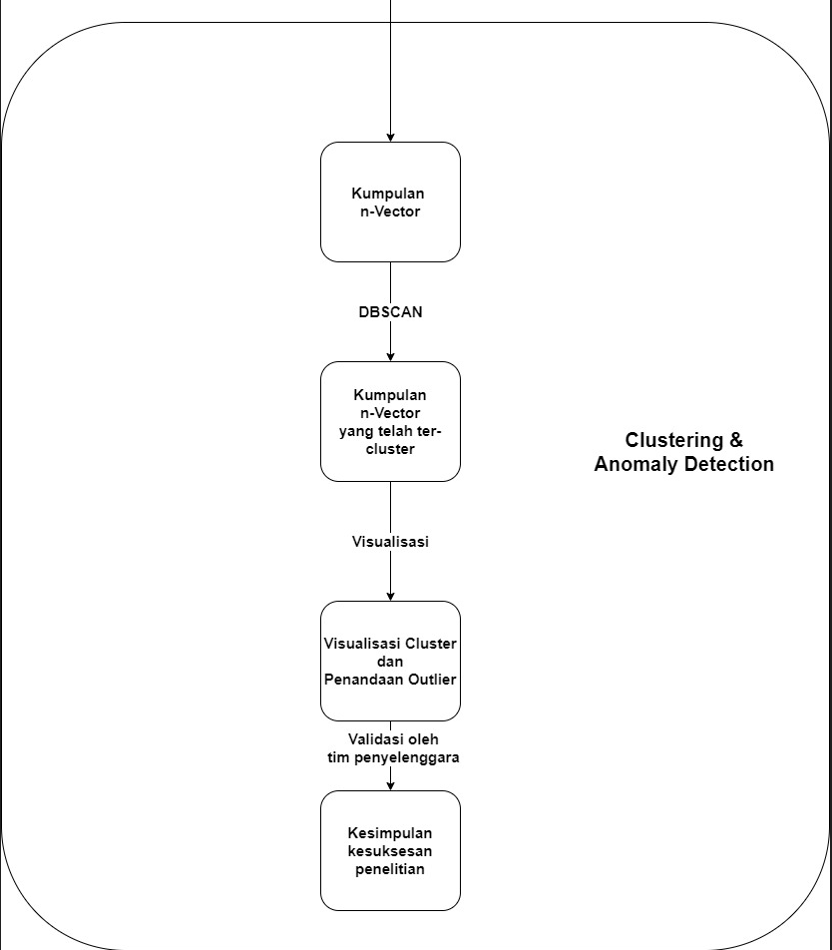
Dalam pendekatan *topic modelling* dengan LDA, akan dilakukan variasi eksperimen, yaitu proses LDA sebagaimana adanya, dibandingkan dengan proses LDA dengan input TF-IDF sebagai *term-document matrix*, dengan harapan weight yang diberikan oleh matriks tersebut mempengaruhi probabilitas stokastik. Akan juga dilakukan variasi jumlah topik yang diinferensikan, untuk mencari jumlah topik optimal yang bisa menghasilkan distribusi topik yang baik. Proses LDA akan menghasilkan sebuah *n-dimensional numerical vector* untuk setiap dokumen, dengan n merupakan jumlah topik yang diinferensikan, dan setiap dimensi vektor melambangkan proporsi topik tersebut pada dokumen yang bersangkutan.



Gambar 3.4 Diagram Alur Eksperimen – *Feature Extraction*

#### Clustering & Anomaly Detection

Vektor-vektor n-dimensi yang terbentuk akan di-*fit* dalam model DBSCAN dan dikelompokkan dalam *cluster-cluster* menurut algoritmanya. Akan dilakukan *tuning* terhadap parameter berupa MinPts dan ε untuk mendapatkan *cluster-cluster* yang berkualitas, dan mendapatkan *outlier* yang benar. Setiap iterasi dari eksperimen akan direpresentasikan secara visual dalam *point graph*, dan akan divalidasi oleh penyelenggara diskusi Komunitas X sebagai pengguna, untuk menentukan keakuratan *cluster* yang terbentuk dan validasi *outlier-outlier* yang teridentifikasi sebagai anomali yang dicari dalam data.



Gambar 3.5 Diagram Alur Eksperimen – *Clustering* & *Anomaly Detection*

### Alur Program





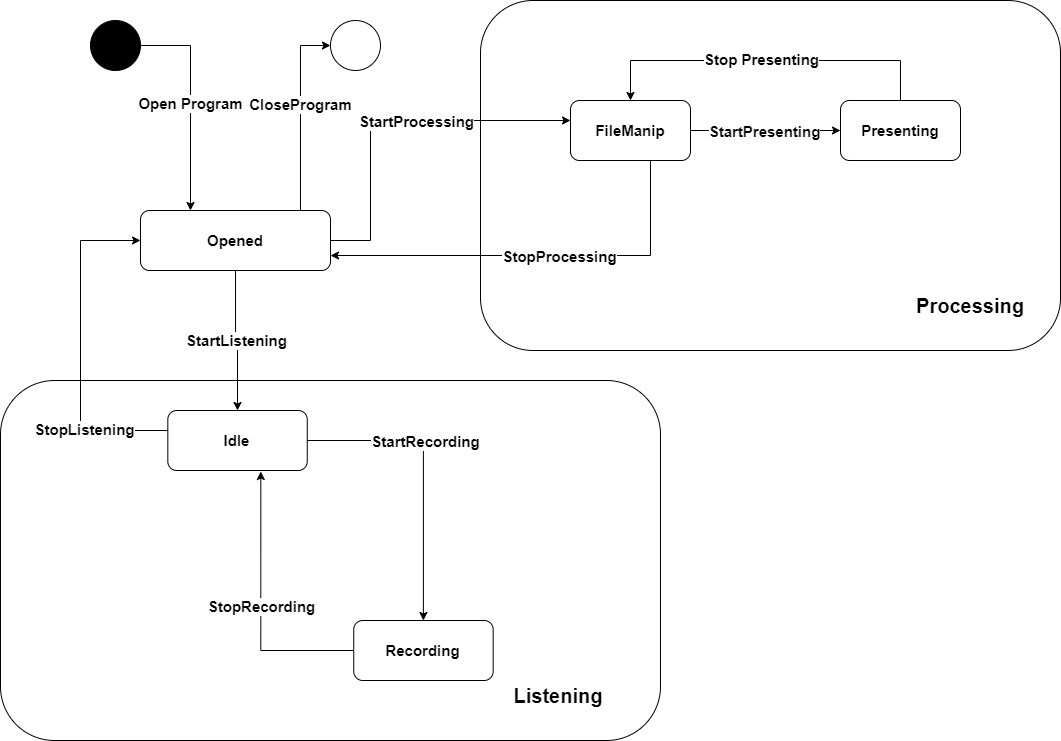




#### Alur Keseluruhan Program

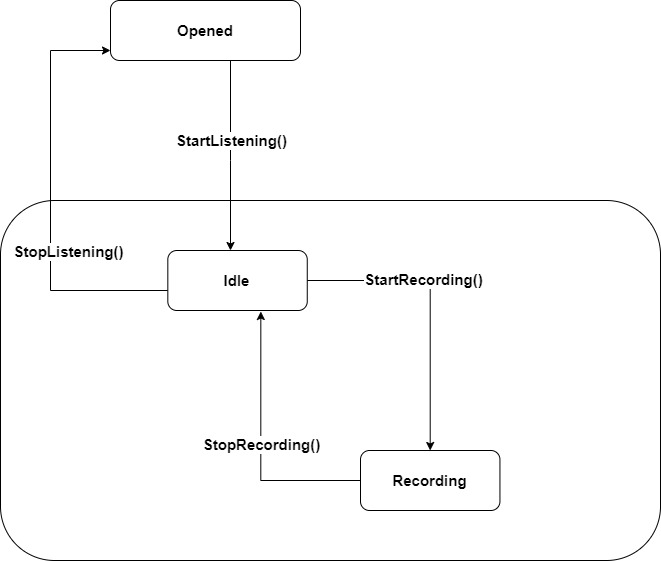
Program akan memiliki tiga *state* inti, yaitu Opened, Listening, dan Processing. Pada saat program dijalankan, program akan memasuki *state* Opened. Akan disajikan sebuah menu bagi pengguna untuk memilih aksi selanjutnya, yaitu StartProcessing, StartListening, atau CloseProgram. Ketika pengguna memilih aksi StartListening, program akan memasuki *state* Listening, dimana program akan mendengarkan dan merekam suara yang ada di komputer pengguna, untuk menangkap jalannya diskusi yang sedang berlangsung. Dalam *state* ini, pengguna dapat kembali ke *state* Opened dengan aksi StopListening.

Ketika pengguna memilih aksi StartProcessing, pengguna akan memasuki state Processing. Dalam state ini, pengguna bisa membuka file database untuk mengakses data rekaman terstruktur yang telah dibuat, lalu menjalankan anomaly detection dan mendapatkan visualisasi data. Pengguna dapat kembali ke state Opened dengan aksi StopProcessing. Ketika di state Opened, pengguna dapat mejalankan aksi CloseProgram untuk menghentikan jalan program.



Gambar 3.6 State Diagram Program yang Akan Dibuat

#### Alur Pengumpulan Klip Suara



Gambar 3.7 *State Diagram* Alur Pengumpulan Klip Suara

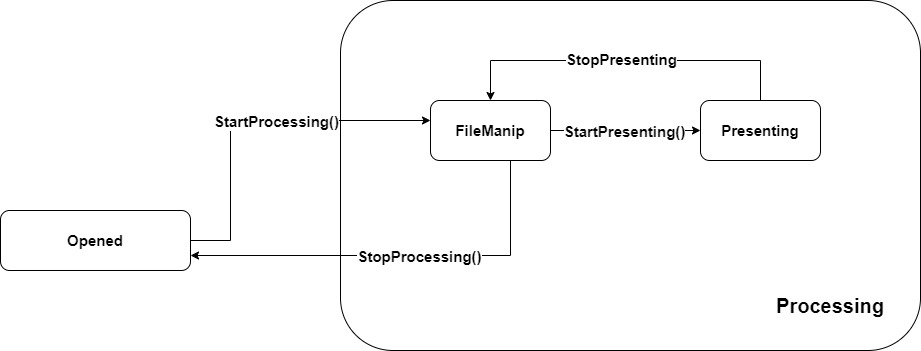


Gambar 3.8 Pseudocode Alur Pengumpulan Klip Suara

Berikut penjelasan setiap perintah yang ada dalam bagan yang ditunjukkan:

* StartListening(): Ketika pengguna memilih perintah StartListening(), program akan membuka koneksi ke tabel dalam *database* untuk menyimpan data rekaman suara, lalu memasuki *state* Idle. Dalam *state* Idle, pengguna dapat memulai proses rekaman dengan perintah StartRecording().
* StartRecording(): Ketika perintah ini diberikan, akan disiapkan sebuah *struct* untuk menampung data yang akan disimpan dalam *database*. Data mencakup ID rekaman, tanggal rekaman, dan *file* suara yang akan disimpan. Proses rekaman suara akan dimulai sampai perintah StopRecording() diberikan.
* StopRecording(): Ketika perintah ini diberikan, proses rekaman akan dihentikan. *File* rekaman yang terbentuk akan disimpan dalam *database* berdasarkan *struct* yang telah didefinisikan. *State* program akan kembali ke Idle.
* StopListening(): Ketika perintah ini diberikan, program akan menutup koneksi ke *database* dan kembali ke *state* Opened.

#### Alur Proses dan Penyajian Data



Gambar 3.9 State Diagram Alur Proses dan Penyajian Data



Gambar 3.10 Pseudocode Alur Proses dan Penyajian Data

Berikut penjelasan setiap perintah yang ada dalam bagan yang ditunjukkan:

* StartProcessing(): Pada saat perintah ini diberikan, program akan memasuki *state* FileManip. Dalam *state* ini, dibuka koneksi ke tabel-tabel dalam *database* yang menyimpan transkrip audio dan transkrip teks. Pengguna dapat memilih untuk mengakses rekor transkrip audio yang diinginkan, untuk diproses menjadi transkrip teks dan dijadikan input proses *anomaly detection*.
* StartPresenting(recordID): Perintah ini dijalankan dengan parameter ID dari salah satu rekor data rekaman audio. Data akan diambil dari tabel yang bersangkutan dan diproses dengan komponen-komponen API *speech-to-text* dan *translator* untuk menghasilkan transkrip berupa teks berbahasa Inggris. Kumpulan transkrip teks akan dijadikan input untuk *pipeline* data yang melibatkan *preprocessing*, *tokenization*, *feature extraction*, *clustering*, dan *anomaly detection*. Akhirnya, hasil dari *anomaly detection* akan dipresentasikan. Ketika perintah ini dijalankan, program akan memasuki *state* Presenting, dimana pengguna dapat melihat visualisasi analisa data.
* StopPresenting(): Ketika perintah ini dijalankan, hasil *anomaly detection* akan disimpan dalam sebuah format tertentu dalam sebuah tabel pada *database*. Program akan kembali ke *state* FileManip dimana pengguna bisa memilih rekor-rekor data audio yang akan diproses.
* StopProcessing(): Ketika perintah ini dijalankan, koneksi *database* ke tabel yang bersangkutan akan ditutup dan *state* akan kembali ke Opened.

# IMPLEMENTASI SISTEM

Bab ini akan menjabarkan implementasi desain sistem yang telah dirumuskan dalam Bab sebelumnya, serta mendokumentasikan code dan tools yang dipakai untuk merancang program dan eksperimen.

## Implementasi Umum

### Tools dan IDE

Implementasi eksperimen dan program ditulis dengan Bahasa Python

### Mapping Konsep dan Implementasi

## Implementasi Eksperimen

### Tools dan IDE

### Library

## Implementasi Program

### Perancangan Front-End Program

#### Perancangan GUI

import sys

import wave

import pyaudio as pa

from PyQt6.QtCore import QThread, pyqtSignal, pyqtSlot

from PyQt6.QtWidgets import \*

class RecordingThread(QThread):

    stopped = False

    sig\_started = pyqtSignal()

    sig\_stopped = pyqtSignal()

    def \_\_init\_\_(self, target\_file):

*self*.target\_file = target\_file

        super().\_\_init\_\_()

    def run(self) -> None:

        print("rt.run()")

        audio = pa.PyAudio()

        frames = []

        stream = audio.open(format=pa.paInt16, channels=1, rate=44100, input=True, frames\_per\_buffer=1024)

*self*.stopped = False

*self*.sig\_started.emit("bruhlord")

        while not *self*.stopped:

            data = stream.read(1024)

            frames.append(data)

        stream.close()

        print("you tryna be on worldstar???")

        wf = wave.open(*self*.target\_file, 'wb')

        wf.setnchannels(1)

        wf.setsampwidth(audio.get\_sample\_size(pa.paInt16))

        wf.setframerate(44100)

        wf.writeframes(b''.join(frames))

        wf.close()

*self*.sig\_stopped.emit()

    @pyqtSlot()

    def stop(self):

        print("we slotting and ending")

*self*.stopped = True

class Window(QWidget):

    def \_\_init\_\_(self):

        super().\_\_init\_\_()

*self*.setWindowTitle("Rec Audio")

*# Create recording thread and attach slots to its signals*

*self*.recording\_thread = RecordingThread(target\_file='test\_recording.wav')

*self*.recording\_thread.sig\_started.connect(*self*.recording\_started)

*self*.recording\_thread.sig\_stopped.connect(*self*.recording\_stopped)

        vbox = QVBoxLayout()

*self*.labelRec = QLabel('')

*self*.labelRec.setFixedSize(130, 15)

        hbox = QHBoxLayout()

*self*.recbtn = QPushButton('▶ record')

*self*.recbtn.setFixedSize(90, 30)

*# Connect signal "recbtn.clicked" to the slot "recording\_thread.start" of our QThread*

*# Never connect directly to the run, always to start!*

*self*.recbtn.clicked.connect(*self*.recording\_thread.start)

*self*.stopbtn = QPushButton('▪')

*self*.stopbtn.setDisabled(True)

*self*.stopbtn.setFixedSize(40, 30)

*# Connect signal "stopbtn.clicked" to the slot "recording\_thread.stop" of our QThread*

*self*.stopbtn.clicked.connect(*self*.recording\_thread.stop)

        hbox.addWidget(*self*.recbtn)

        hbox.addWidget(*self*.stopbtn)

        vbox.addWidget(*self*.labelRec)

        vbox.addLayout(hbox)

*self*.setLayout(vbox)

    @pyqtSlot(str)

    def recording\_started(self):

        """This slot is called when recording starts"""

        print("recording\_started entered")

*self*.labelRec.setText('◉ recording...')

*self*.stopbtn.setDisabled(False)

*self*.recbtn.setDisabled(True)

    @pyqtSlot()

    def recording\_stopped(self):

        """This slot is called when recording stops"""

        print("we ending!!")

*self*.labelRec.setText('recording stopped')

*self*.recbtn.setDisabled(False)

*self*.stopbtn.setDisabled(True)

if \_\_name\_\_ == "\_\_main\_\_":

    app = QApplication(sys.argv)

    win = Window()

    win.show()

    app.exec()

#### Coding GUI

### Perancangan Back-End Program

#### Skema Database

#### Code Database

# DAFTAR PUSTAKA

*"Forum"*. (2022). Retrieved from Oxford Learner’s Dictionaries.: https://www.oxfordlearnersdictionaries.com/definition/english/forum?q=forum

Aghammadzada, E. (2020). *kaggle.com*. Retrieved from Stemming vs Lemmatization NLP [Image]: https://www.kaggle.com/getting-started/186152

Blei, D., Ng, A., & Jordan, M. (2001). Latent Dirichlet Allocation. *Advances in Neural Information Processing Systems 14 (NIPS 2001).* Vancouver. Retrieved from https://proceedings.neurips.cc/paper/2001/file/296472c9542ad4d4788d543508116cbc-Paper.pdf

Brownlee, J. (2020, April 6). *10 Clustering Algorithms With Python*. Retrieved from machinelearningmastery.com: https://machinelearningmastery.com/clustering-algorithms-with-python/

Chandola, V., Banerjee, A., & Kumar, V. (2009, July). Anomaly Detection: A Survey. *ACM Computing Surveys*, Vol 41, No. 3, Article 15. Retrieved from https://dl.acm.org/doi/10.1145/1541880.1541882

Chowdhary, K. (2020). Fundamentals of Artificial Intelligence. Springer. Retrieved from https://dokumen.pub/qdownload/fundamentals-of-artificial-intelligence-1nbsped-8132239709-9788132239703.html

Curiskis, S., & al., e. (2019). An evaluation of document clustering and topic modelling in two online social networks: Twitter and Reddit. *Information Processing and Management 57(2)*. Retrieved from https://doi.org/10.1016/j.ipm.2019.04.002

Danziger, S., Levav, J., & Avnaim-Pesso, L. (2011). Extraneous factors in judicial decisions. *Proceedings of the National Academy of Sciences*, (pp. 6889-6892). doi:10.1073/pnas.1018033108

Dictionary, C. (n.d.). *Lemmatisation.* Retrieved from https://www.collinsdictionary.com/dictionary/english/lemmatize

Ester, M., Kriegel, H., Sander, J., & Xu, X. (1996). A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. *KDD-96* (pp. 226-231). Munich: AAAI. Retrieved from https://dl.acm.org/doi/10.5555/3001460.3001507

Firdaus, A. (2019, Aug 13). *Cara Kerja Word2Vec*. Retrieved from https://medium.com/@afrizalfir/mengenal-word2vec-af4758da6b5d

Gautam, H. (2020, Mar 1). *Word Embedding: Basics*. Retrieved from https://medium.com/@hari4om/word-embedding-d816f643140

GeekForGeeks. (2022, December 11). *Removing stop words with NLTK in Python*. Retrieved from www.geeksforgeeks.org: https://www.geeksforgeeks.org/removing-stop-words-nltk-python/

Google. (2022, July 19). *Clustering Algorithms*. Retrieved from https://developers.google.com/: https://developers.google.com/machine-learning/clustering/clustering-algorithms

Hvitfeld, E., & Silge, J. (2022, May 11). *Supervised Machine Learning for Text Analysis in R*. Retrieved from https://smltar.com: https://smltar.com/tokenization.html

Jacobi, C., van Atteveldt, W., & Welbers, K. (2015). Quantitative analysis of large amounts of journalistic texts using topic modelling. *Digital Journalism*, 89–106. Retrieved from https://doi.org/10.1080/21670811.2015.1093271

Kalepalli, Y., Tasneem, S., Phani Teja, P. D., & Manne, S. (2020). Control Systems (ICICCS). *Effective Comparison of LDA with LSA for Topic Modelling*. doi:10.1109/iciccs48265.2020.9120888

Kannan, S., Gurusamy, V., Vijayarani, S., Ilamathi, J., & Nithya, M. (2014). Preprocessing Techniques for Text Mining. *International Journal of Computer Science & Communication Networks*, 7-16. Retrieved from https://www.academia.edu/35015140/Preprocessing\_Techniques\_for\_Text\_Mining

Katryn, R. (2020, September 28). *Text Preprocessing: Tahap Awal dalam Natural Language Processing (NLP)*. Retrieved from https://medium.com/: https://medium.com/mandiri-engineering/text-preprocessing-tahap-awal-dalam-natural-language-processing-nlp-bc5fbb6606a

Kienle, A., & Ritterskamp, C. (2007). Facilitating asynchronous discussions in learning communities: The impact of moderation strategies. *Behaviour and Information Technology, 26*(1), 73-80. doi:10.1080/01449290600811594

Kumar, E. (2013). *Natural Language Processing.* Retrieved from https://books.google.com/books/about/Natural\_Language\_Processing.html?id=FpUBFNFuKWgC

Lai, S., Liu, K., He, S., & Zhao, J. (2016). How to Generate a Good Word Embedding. *IEEE Intelligent Systems*, 5-14. Retrieved from https://arxiv.org/abs/1507.05523

Lee, J., & et.al. (2018). Ensemble Modeling for Sustainable Technology Transfer. *Sustainability, X*(7), 2278-. doi:10.3390/su10072278

Mohammed, S., & Al-augby, S. (2020). LSA & LDA Topic Modeling Classification : Comparison study on E-books. *Indonesian Journal of Electrical Engineering and Computer Science*, 353-362. Retrieved from https://ijeecs.iaescore.com/index.php/IJEECS/article/view/20547

NLTK. (2022, December 12). *Documentation*. Retrieved from https://www.nltk.org: https://www.nltk.org/howto/stem.html

Pai, A. (2020, May 26). *What is Tokenization in NLP? Here’s All You Need To Know*. Retrieved from www.analyticsvidhya.com: https://www.analyticsvidhya.com/blog/2020/05/what-is-tokenization-nlp/

Porter, M. F. (1980). An algorithm for suffix stripping. *Program: electronic library and information systems*, 130-137. doi:10.1108/eb046814

Ribeiro, J. (2021, Feb 18). *What is Natural Language Processing (NLP), and why does it matter to you?* Retrieved from https://towardsdatascience.com/what-is-natural-language-processing-nlp-and-why-does-it-matter-to-you-3cc4fb003940

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management, XXIV*(5), 513-523. doi:doi:10.1016/0306-4573(88)90021-0

Scikit Learn. (n.d.). *Demo of DBSCAN clustering algorithm.* Retrieved from scikit-learn.org: https://scikit-learn.org/stable/auto\_examples/cluster/plot\_cluster\_comparison.html

Williams, T., & Betak, J. (2019). A Comparison of LSA and LDA for the Analysis of Railroad Accident. *Journal of Ubiquitous Systems & Pervasive Networks*, 11-15. Retrieved from https://iasks.org/articles/juspn-v11-i1-pp-11-15.pdf

Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). *Data Mining: Practical Machine Learning Tools and Techniques (Morgan Kaufmann Series in Data Management Systems).* Massachusetts: Morgan Kaupfmann.

Xu, D., & Tian, Y. (2015). A Comprehensive Survey of Clustering Algorithms. *Annals of Data Science*, 165-193. Retrieved from https://link.springer.com/article/10.1007/s40745-015-0040-1