

**KOMPARASI SVM, BI-LSTM, DAN DISTILBERT UNTUK
KLASIFIKASI OTOMATIS KEBUTUHAN PERANGKAT
LUNAK FUNGSIONAL DAN NON-FUNGSIONAL INDONESIA**

PROPOSAL

Oleh:



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SUMATERA
LAMPUNG SELATAN
2025**

BAB I

PENDAHULUAN

1.1 Latar Belakang

Rekayasa kebutuhan merupakan tahap fundamental dalam pengembangan perangkat lunak karena kualitas kebutuhan menentukan keberhasilan tahap desain, implementasi, dan pemeliharaan. Namun, kesalahan dalam mengidentifikasi dan mengklasifikasikan kebutuhan masih menjadi masalah klasik. Berbagai laporan industri menunjukkan bahwa kegagalan memahami kebutuhan, khususnya *Non-Functional Requirements* (NFR), menyumbang 60–80% masalah pada proyek perangkat lunak dan berdampak langsung pada biaya maupun kualitas sistem [1]. Bahkan dalam konteks *agile*, kelalaian terhadap NFR dapat meningkatkan risiko kegagalan proyek hingga 60% [2].

Sementara itu, proses klasifikasi kebutuhan secara manual masih bersifat lambat, mahal, dan rentan inkonsistensi antar-analis. Tantangan ini semakin besar karena NFR sering tersebar, implisit, dan ditulis dalam kalimat panjang yang ambigu [1]. Ketidakakuratan dalam mengklasifikasikan *Functional Requirements* (FR) dan NFR dapat berdampak fatal, seperti desain arsitektur yang salah arah, duplikasi fitur, pengujian yang tidak tepat, hingga kegagalan sistem.

Berbagai pendekatan *machine learning* telah diteliti untuk mengotomatiskan proses klasifikasi kebutuhan. Pendekatan tradisional berbasis TF-IDF, SVM, dan Naive Bayes menunjukkan performa yang cukup baik namun memiliki keterbatasan pada representasi fitur karena mengabaikan konteks linguistik dan struktur kalimat [1]. Kemunculan model berbasis *deep learning* seperti LSTM dan CNN meningkatkan akurasi, tetapi masih memiliki kelemahan pada pemahaman konteks yang lebih dalam [2].

Perkembangan terbaru menunjukkan bahwa model transformer seperti

BERT, RoBERTa, dan variannya memberikan kinerja unggul untuk klasifikasi kebutuhan karena kemampuannya memahami hubungan semantik dua-arah dalam bahasa alami. Studi internasional terbaru memperlihatkan bahwa BERT dan turunannya dapat mencapai akurasi hingga 95% dalam membedakan FR dan NFR pada bahasa Inggris dan Turki [3]. Model *hybrid* seperti BERT-BiCNN atau BERT-CNN juga terbukti meningkatkan performa klasifikasi [1].

Namun, hampir seluruh penelitian tersebut dilakukan pada bahasa Inggris, Turki, atau bahasa lain dengan sumber daya bahasa yang relatif kaya. Dalam konteks Bahasa Indonesia, penelitian NLP masih menghadapi keterbatasan dataset anotasi, kelangkaan *benchmark*, dan minimnya penelitian sistematis untuk model klasifikasi kebutuhan [4]. Meskipun telah tersedia IndoBERT sebagai model pra-latih Bahasa Indonesia, sebagian besar studi mengarah pada *dialog systems*, NER, atau *sentiment analysis*, bukan pada domain *Requirements Engineering* [5].

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan, penelitian ini merumuskan masalah berikut:

1. Bagaimana membangun dataset klasifikasi kebutuhan perangkat lunak berbahasa Indonesia yang tervalidasi, terstruktur, dan dapat direplikasi, mengingat belum adanya dataset FR/NFR berbahasa Indonesia yang tersedia secara publik?
2. Bagaimana performa metode pembelajaran mesin klasik seperti *Support Vector Machine* (SVM) dan model *deep learning* seperti *Long Short-Term Memory* (LSTM) dalam melakukan klasifikasi *Functional Requirements* (FR) dan *Non-Functional Requirements* (NFR) pada data berbahasa Indonesia?
3. Bagaimana kinerja model transformer berbahasa Indonesia, khususnya IndoBERT dan DistilBERT, dibandingkan SVM dan LSTM dalam tugas

klasifikasi FR/NFR, serta model mana yang memberikan akurasi terbaik dalam konteks sumber daya linguistik Indonesia yang terbatas?

4. Faktor linguistik atau pola apa saja yang menyebabkan model tertentu (SVM, LSTM, IndoBERT, DistilBERT) mengalami kesalahan klasifikasi terhadap FR dan NFR pada Bahasa Indonesia?

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah, penelitian ini memiliki tujuan sebagai berikut:

1. Menghasilkan dataset klasifikasi kebutuhan perangkat lunak berbahasa Indonesia yang tervalidasi, terstruktur, dan layak digunakan sebagai acuan penelitian lanjutan dalam domain klasifikasi FR dan NFR.
2. Mengevaluasi performa metode pembelajaran mesin klasik, khususnya *Support Vector Machine* (SVM), dalam mengklasifikasikan *Functional Requirements* (FR) dan *Non-Functional Requirements* (NFR) pada data berbahasa Indonesia.
3. Menganalisis kinerja model *deep learning* berbasis LSTM dalam mengolah karakteristik linguistik Bahasa Indonesia untuk tugas klasifikasi FR/NFR.
4. Mengevaluasi secara empiris efektivitas model transformer berbahasa Indonesia, terutama IndoBERT dan DistilBERT, dalam tugas klasifikasi FR/NFR, serta membandingkannya dengan pendekatan SVM dan LSTM.
5. Mengidentifikasi pola kesalahan klasifikasi yang muncul pada setiap model, meliputi aspek linguistik, struktur kalimat, serta karakteristik FR dan NFR yang menyebabkan model gagal membedakan kedua kategori.

1.4 Batasan Masalah

Penelitian ini dibatasi oleh ruang lingkup berikut:

1. Data yang digunakan hanya mencakup kebutuhan perangkat lunak berbahasa Indonesia, baik yang berasal dari dokumen publik, studi kasus akademik,

- maupun sumber terbuka yang telah melalui proses seleksi dan validasi.
2. Klasifikasi yang dilakukan terbatas pada dua kategori, yaitu *Functional Requirements* (FR) dan *Non-Functional Requirements* (NFR), tanpa membedakan sub-kategori NFR secara lebih rinci.
 3. Model pembelajaran yang dievaluasi dibatasi pada tiga pendekatan utama, yaitu:
 - (a) Metode pembelajaran mesin (SVM).
 - (b) Model *deep learning* berbasis LSTM.
 - (c) Model transformer IndoBERT dan DistilBERT.
 4. Eksplorasi *hyperparameter* dilakukan dalam batas wajar dan tidak mencakup *grid search* ekstensif.
 5. Penelitian tidak membahas proses pengembangan sistem pendukung keputusan, melainkan berfokus pada evaluasi model klasifikasi dan analisis hasilnya.
 6. Evaluasi performa terbatas pada metrik klasifikasi umum, seperti akurasi, presisi, *recall*, dan *F1-score*, tanpa mengevaluasi aspek efisiensi komputasi secara mendalam.

1.5 Kontribusi Penelitian

Penelitian ini memberikan kontribusi sebagai berikut:

1. Pembuatan dataset klasifikasi kebutuhan perangkat lunak berbahasa Indonesia yang tervalidasi, terstruktur, dan dapat direplikasi. Dataset ini menutup kekosongan sumber data FR/NFR berbahasa Indonesia yang sebelumnya tidak tersedia di literatur maupun repositori publik.
2. Penyusunan *baseline* model untuk klasifikasi FR dan NFR dalam Bahasa Indonesia, mencakup pendekatan pembelajaran mesin (SVM), *deep learning* (LSTM), serta dua model transformer berbahasa Indonesia (IndoBERT dan DistilBERT). *Baseline* ini memberikan landasan evaluasi bagi penelitian lanjutan di bidang *Requirements Engineering* berbasis

NLP.

3. Evaluasi komparatif yang sistematis terhadap tiga pendekatan klasifikasi (SVM, LSTM, transformer) pada dataset Bahasa Indonesia, sehingga menunjukkan model yang paling efektif dalam konteks sumber daya linguistik Indonesia yang terbatas.
4. Analisis mendalam terhadap pola kesalahan klasifikasi yang mencakup aspek linguistik, struktur kalimat, dan karakteristik FR/NFR. Analisis ini menghasilkan pemahaman baru tentang tantangan khusus Bahasa Indonesia dalam tugas klasifikasi kebutuhan, serta titik lemah dan kekuatan masing-masing model.
5. Penyediaan fondasi awal bagi pengembangan otomatisasi rekayasa kebutuhan berbahasa Indonesia, terutama pada tahap analisis awal dokumen kebutuhan, sehingga hasil penelitian dapat digunakan sebagai referensi akademik maupun aplikasi praktis di lingkungan industri.

BAB II

TINJAUAN PUSTAKA

2.1 State of the Art

Penelitian terkait klasifikasi kebutuhan perangkat lunak telah berkembang dalam empat fase besar: (1) pendekatan berbasis *Information Retrieval* (IR), (2) metode *Machine Learning* (ML) klasik, (3) arsitektur *Deep Learning* (DL), dan (4) model transformer.

2.1.1 Pendekatan IR dan Machine Learning Klasik

Pada tahap awal, klasifikasi FR dan NFR banyak mengandalkan pendekatan IR berbasis pencocokan kata. Metode ini efektif untuk kebutuhan sederhana, tetapi tidak mampu menangkap konteks kalimat yang lebih kompleks.

Perkembangan berikutnya adalah penggunaan metode ML seperti TF-IDF, Naïve Bayes, dan Support Vector Machine (SVM). Kaur dan Kaur menunjukkan bahwa pendekatan ini mendominasi penelitian klasifikasi kebutuhan sebelum munculnya model berbasis transformer [1]. Namun, metode ini bergantung pada fitur yang diekstraksi secara manual sehingga rentan kehilangan makna semantik [1].

2.1.2 Pendekatan Deep Learning

Kemunculan arsitektur DL seperti CNN dan LSTM meningkatkan kemampuan model dalam memahami konteks linguistik. Li dan Nong mengembangkan NFRNet (gabungan Bi-LSTM dan BERT) dan berhasil mencapai F1-Score 91% pada dataset yang diperluas [2]. Namun, performanya masih sangat dipengaruhi ukuran dataset dan struktur bahasa yang kompleks.

2.1.3 Model Transformer

Transformers mengubah lanskap NLP melalui mekanisme *self-attention* yang memungkinkan pemahaman konteks dua arah. Yucalar menunjukkan bahwa versi BERT khusus bahasa (misalnya BERTurk) jauh mengungguli model multilingual dalam tugas klasifikasi kebutuhan [3]. Kaur dan Kaur juga membuktikan bahwa arsitektur hibrida BERT-CNN mampu meningkatkan akurasi klasifikasi pada dataset PROMISE [1].

Dalam konteks Bahasa Indonesia, penelitian NLP masih berada pada kategori *low-resource*. Koto et al. menegaskan bahwa minimnya dataset anotasi menghambat kinerja model pra-latih [4], dan Di et al. menunjukkan bahwa model Indonesia perlu pelatihan tambahan karena perbedaan struktur morfologi [5].

2.1.4 Efektivitas Model Kecil dan ML Klasik

Studi komparatif terbaru oleh El-Hajjami et al. memperlihatkan bahwa model klasik seperti SVM tetap kompetitif terhadap LLM dalam klasifikasi kebutuhan, terutama ketika dataset kecil atau domainnya spesifik [6]. Temuan ini memperkuat relevansi penelitian yang membandingkan tiga generasi model sekaligus: SVM, arsitektur LSTM, dan model transformer Indonesia.

2.1.5 Ringkasan Penelitian Terdahulu

Tabel 2.1 Ringkasan Penelitian Terdahulu

No.	Studi	Fokus	Metode	Hasil Utama
1.	Yucalar et al. (2023) [3]	Dataset Turki	BERTurk vs ML/DL	Model spesifik bahasa unggul (F1 95%).

No.	Studi	Fokus	Metode	Hasil Utama
2.	Kaur & Kaur (2023) [1]	Fitur semantik	BERT-BiCNN	Akurasi meningkat dengan arsitektur hibrida.
3.	Li & Nong (2022) [2]	Ambiguitas NFR	NFRNet	F1 91% pada dataset diperluas.
4.	El-Hajjami et al. (2024) [6]	Efektivitas LLM	SVM vs GPT	Model klasik tetap kompetitif.
5.	Subahi (2023) [7]	Green IT NFR	Fine-tuned BERT	BERT efektif pada domain spesifik.

2.2 Kualitas Penulisan dan Sintesis Literatur

Literatur yang ada menunjukkan pola perkembangan yang jelas: setiap pendekatan lahir untuk mengatasi kelemahan pendekatan sebelumnya. Pendekatan ML klasik menawarkan efisiensi tetapi gagal memahami konteks. DL menawarkan pemrosesan sekuensial, namun tetap terbatas di bahasa dengan morfologi kompleks. Transformer hadir untuk menjawab kebutuhan pemahaman semantik yang lebih akurat.

Sintesis ini penting karena menggambarkan bahwa:

1. Penelitian sebelumnya tidak hanya berbeda dalam metode, tetapi dalam alasan teknis mengapa metode tersebut muncul;
2. Solusi linguistik untuk Bahasa Indonesia tidak bisa langsung mengadopsi hasil penelitian Inggris atau Turki;

3. Pemilihan model SVM, LSTM, dan DistilBERT dalam penelitian ini didasarkan pada fondasi perkembangan ilmiah, bukan arbitrer.

2.3 Research Gap

Berdasarkan tinjauan literatur, terdapat beberapa celah penelitian yang belum terisi:

1. **Belum tersedia dataset FR/NFR berbahasa Indonesia** yang terstandarisasi, sebagaimana dinyatakan dalam studi *low-resource* oleh Koto et al. [4].
 2. **Tidak ada studi yang membandingkan SVM, LSTM, dan transformer khusus Indonesia** dalam klasifikasi FR/NFR.
 3. **Minimnya analisis kesalahan berbasis linguistik Bahasa Indonesia**, padahal struktur afiks dan kalimat panjang sering menjadi sumber error.
 4. **Belum ada baseline komparatif** yang dapat dijadikan acuan bagi penelitian lanjutan dalam Kebutuhan Perangkat Lunak berbasis NLP.
- Research gap inilah yang menjadi dasar rasional penelitian ini.

2.4 Dasar Teori

2.4.1 Klasifikasi Kebutuhan

Kebutuhan perangkat lunak secara umum dibagi menjadi:

1. **Kebutuhan Fungsional (FR):** Mendefinisikan fungsi atau perilaku sistem.
2. **Kebutuhan Non-Fungsional (NFR):** Menjelaskan batasan kualitas operasi sistem seperti kinerja dan keamanan [7].

2.4.2 Algoritma Klasifikasi

2.4.2.1 Support Vector Machine (SVM)

SVM adalah algoritma *supervised learning* yang mencari *hyperplane* optimal sebagai pemisah antar kelas. Dikombinasikan dengan TF-IDF, SVM unggul dalam menangani teks berdimensi tinggi. Selain itu, SVM relatif stabil pada dataset kecil sehingga cocok sebagai *baseline* pada penelitian dengan

sumber data terbatas. Kemampuannya mengontrol margin menjadikan SVM tetap kompetitif dibandingkan model modern pada tugas klasifikasi teks yang sederhana namun padat informasi.

2.4.2.2 Bi-directional LSTM (Bi-LSTM)

Bi-LSTM memproses input dari dua arah sehingga mampu menangkap konteks panjang dalam kalimat [5]. Arsitektur dua arah ini membantu model memahami relasi kata yang saling bergantung, khususnya pada kalimat kebutuhan yang sering mengandung struktur kompleks. Meski begitu, Bi-LSTM tetap sensitif terhadap panjang sekuens dan membutuhkan pelatihan lebih lama dibandingkan metode berbasis fitur tradisional.

2.4.2.3 Transformer (DistilBERT)

DistilBERT adalah versi ringan dari BERT yang mengurangi ukuran model namun tetap mempertahankan performa tinggi melalui mekanisme *self-attention*. Mekanisme ini memungkinkan DistilBERT mempelajari hubungan semantik antar kata secara kontekstual, sebuah kemampuan yang sangat penting dalam membedakan FR dan NFR yang sering ambigu. Keunggulannya menjadikan DistilBERT lebih mudah dilatih pada dataset kecil tanpa kehilangan kualitas representasi.

2.4.3 Metrik Evaluasi

Penelitian ini menggunakan F1-Score sebagai metrik utama:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

(Rumus 2.1)

BAB III

METODE PENELITIAN

3.1 Alur Penelitian

Penelitian ini menggunakan pendekatan eksperimental komparatif untuk mengevaluasi kinerja tiga generasi model pembelajaran mesin dalam klasifikasi *Functional Requirements* (FR) dan *Non-Functional Requirements* (NFR). Proses penelitian dirancang untuk memastikan validitas internal melalui kontrol variabel yang ketat dan validitas eksternal melalui penggunaan dataset yang representatif.

Tahapan penelitian meliputi:

1. Konstruksi dataset mandiri dengan total 1.000 sampel.
2. Pra-pemrosesan data adaptif berdasarkan arsitektur model.
3. Rekayasa fitur: TF-IDF, *trainable embedding*, dan *contextual embedding*.
4. Pelatihan tiga model (SVM, Bi-LSTM, DistilBERT) dengan strategi penanganan ketidakseimbangan kelas.
5. Evaluasi numerik menggunakan *Stratified 5-Fold Cross Validation*.
6. Analisis kesalahan dan interpretabilitas model.

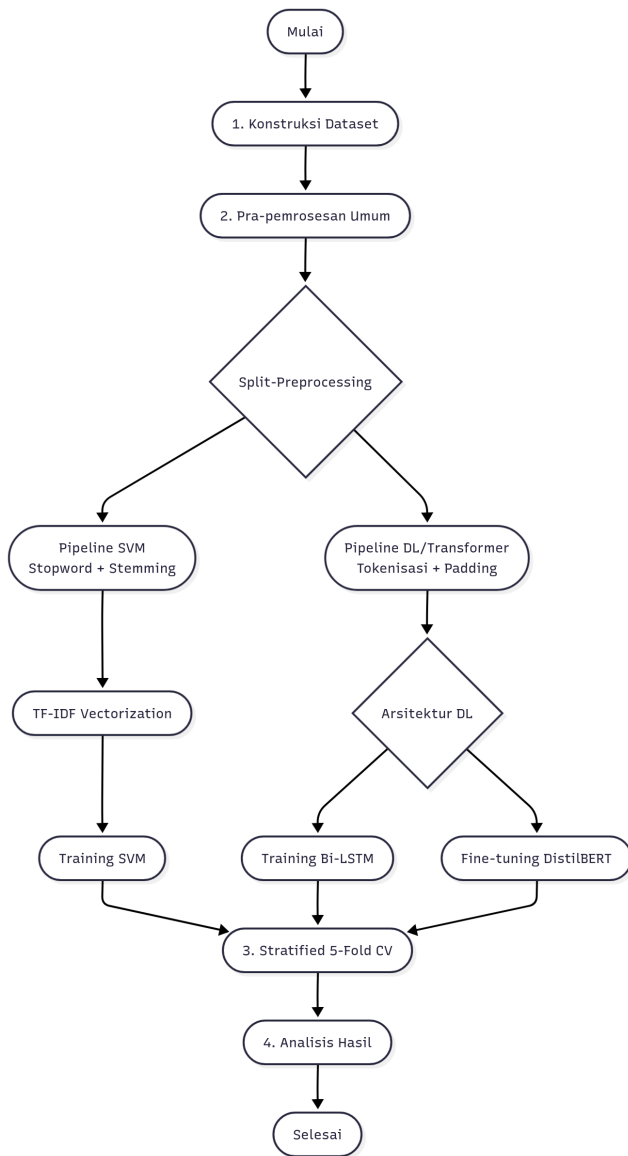
Diagram alir metodologi ditunjukkan pada Gambar 3.1.

3.2 Konstruksi Dataset

3.2.1 Sumber Data

Dataset dikumpulkan dari berbagai sumber untuk menjamin keragaman linguistik, yang meliputi:

1. Dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL).
2. Laporan Tugas Akhir mahasiswa bidang Informatika.
3. Dokumen kebutuhan proyek perangkat lunak *open-source* dari repositori publik.



Gambar 3.1 Diagram Alir Tahapan Penelitian

3.2.2 Statistik Dataset

Total dataset yang dikurasi berjumlah **1.000 kalimat**, dengan rincian distribusi kelas sebagai berikut:

1. **FR**: 580 kalimat (58%).
2. **NFR**: 420 kalimat (42%).
3. **Total**: 1.000 kalimat.
4. Rata-rata panjang kalimat: 17 token.
5. Panjang maksimum: 55 token.

Jumlah data ini (1.000 sampel) dianggap memadai untuk proses *fine-tuning* model bahasa modern dan pelatihan model *Deep Learning* sederhana tanpa risiko *overfitting* yang signifikan, asalkan disertai teknik regularisasi yang tepat. Distribusi kelas mengikuti karakteristik alami dokumen kebutuhan di mana FR cenderung lebih dominan.

3.2.3 Protokol Anotasi

Setiap kalimat dianotasi oleh dua validator dengan pengalaman minimal 3 tahun di bidang Rekayasa Perangkat Lunak (RPL). Protokol anotasi meliputi:

1. Pelabelan manual berbasis pedoman standar ISO/IEC 29148.
2. Validasi silang pada 20% sampel dataset.
3. Reliabilitas *inter-annotator* diuji menggunakan Cohen's Kappa dengan nilai 0.72 (kategori *substantial agreement*).

3.3 Pra-pemrosesan Data

Penelitian menerapkan pendekatan **Split-Preprocessing** untuk menyesuaikan kebutuhan input arsitektur model yang berbeda.

3.3.1 Pra-pemrosesan Umum

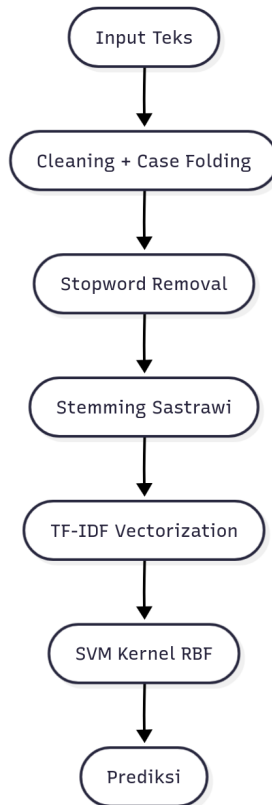
Langkah umum yang diterapkan pada seluruh data meliputi:

1. Pembersihan teks: penghapusan URL, tag HTML, karakter non-ASCII, dan simbol yang tidak relevan.
2. *Case folding* menjadi huruf kecil.
3. Normalisasi spasi dan tanda baca.

3.3.2 Pipeline A: SVM

Untuk SVM, digunakan pendekatan reduksi fitur agresif (Gambar 3.2):

1. *Stopword removal* menggunakan pustaka Sastrawi.
2. *Stemming* menggunakan algoritma Nazief–Adriani.
3. Tokenisasi berbasis *whitespace*.



Gambar 3.2 Pipeline Pra-pemrosesan dan Pelatihan SVM

3.3.3 Pipeline B: Bi-LSTM dan DistilBERT

Untuk menjaga konteks semantik yang dibutuhkan model *Deep Learning*:

1. *Stopword removal* dan *stemming* **tidak diterapkan**.

2. Tokenisasi:

(a) **Bi-LSTM**: *Word-level tokenization*.

(b) **DistilBERT**: *WordPiece tokenizer* (sub-word).

3. *Padding* dan *truncation* pada panjang tetap 128 token.

3.4 Representasi Fitur

3.4.1 TF-IDF untuk Support Vector Machine (SVM)

TF-IDF (*Term Frequency-Inverse Document Frequency*) dipilih karena efisiensi komputasi dan interpretabilitasnya [Salton & Buckley, 1988]. Metode ini menghasilkan vektor jarang (*sparse vector*) yang stabil untuk model linear.

TF-IDF untuk term t dan dokumen d didefinisikan secara matematis sebagai:

Dimana $TF(t, d) = \frac{TF(t, d)}{\sum_k n_{k,d}}$ adalah frekuensi term ternormalisasi: (Rumus 3.1)

Dan IDF mengukur informasi yang dibawa oleh kata tersebut: (Rumus 3.2)

Dimana N adalah total dokumen (4.000) dan df_t adalah jumlah dokumen yang mengandung term t . (Rumus 3.3)

Konfigurasi Parameter:

1. `max_features`: 5.000 (mengambil 5.000 kata terpenting).
2. `ngram_range`: (1, 2) (unigram dan bigram).

3.4.2 Trainable Embedding untuk Bi-LSTM

Penelitian ini menggunakan *Trainable Embedding* yang dilatih dari awal (*from scratch*). Keputusan ini didasarkan pada kebutuhan ***adaptasi domain spesifik***. Kosakata dalam dokumen **requirements** memiliki distribusi makna yang berbeda dengan korpus bahasa umum (seperti Wikipedia).

Dengan 1.000 sampel data latih, model embedding dimensi 100 ($d = 100$) masih dapat konvergen dengan baik tanpa *overfitting* berlebihan, dibandingkan menggunakan *pre-trained* embedding dimensi 300 yang terlalu kompleks untuk

tugas ini.

Representasi setiap token input w_t diperoleh melalui operasi *lookup*:

Matriks E diperbarui selama proses backpropagation*. (Rumus 3.4)

3.4.3 Contextual Embedding untuk DistilBERT

DistilBERT menggunakan mekanisme *Self-Attention* untuk menghasilkan representasi kontekstual. Vektor representasi diambil dari *hidden state* terakhir pada token spesial [CLS], yang merepresentasikan semantik kalimat secara global.

3.5 Arsitektur dan Implementasi Model

3.5.1 Support Vector Machine (SVM)

SVM bekerja dengan mencari *hyperplane* optimal yang memisahkan kelas FR dan NFR. Formulasi optimasi primal SVM [Vapnik, 1995] adalah:

Dengan batasan:

$$\min_{\mathbf{w}, \mathbf{b}} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \xi_i$$

$$\mathbf{w}^T \mathbf{x}_i + \mathbf{b} \geq 1 + \xi_i$$

$$\mathbf{w}^T \mathbf{x}_i + \mathbf{b} \leq -1 - \xi_i$$

$$\xi_i \geq 0$$

(Rumus 3.5)

(Rumus 3.6)

(Rumus 3.7)

Tabel 3.1 Konfigurasi Hyperparameter SVM

Parameter	Nilai
Kernel	RBF
C	1.0
Gamma	Scale
Class Weight	Balanced

Penggunaan `class_weight='balanced'` secara otomatis menyesuaikan bobot penalti berbanding terbalik dengan frekuensi kelas untuk mengatasi ketidakseimbangan data (FR 58% vs NFR 42%).

3.5.2 Bidirectional LSTM (Bi-LSTM)

Bi-LSTM memproses sekuens input dari dua arah menggunakan mekanisme *gating* [Hochreiter & Schmidhuber, 1997]. Untuk setiap langkah waktu t , persamaan pembaruan sel adalah sebagai berikut:

$$\begin{aligned} f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) && \text{(Rumus 3.8)} \\ \tilde{C}_t &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) && \text{(Rumus 3.10)} \\ o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) && \text{(Rumus 3.12)} \end{aligned}$$

Keluaran akhir diperoleh dengan menggabungkan (*concatenate*) state

terakhir dari arah maju (\rightarrow) dan mundur (\leftarrow):

Penanganan Ketidakseimbangan Data: Berbeda dengan SVM yang menggunakan parameter bobot, pada Bi-LSTM diterapkan **Weighted Binary Cross-Entropy Loss**. Fungsi *loss* dimodifikasi dengan bobot w_j yang diberikan lebih besar untuk kelas minoritas (NFR) agar model tidak bias ke kelas mayoritas.

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^N w_{y_i} [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)]$$

Konfigurasi Hyperparameter: Nilai hyperparameter ditentukan melalui

(Rumus 3.15)

studi empiris awal (*preliminary experiments*) pada data validasi untuk mencapai konvergensi yang stabil.

Tabel 3.2 Hyperparameter Bi-LSTM

Parameter	Nilai
Embedding Dim	100
LSTM Units	64 (Bidirectional)
Dropout Rate	0.5
Optimizer	Adam ($lr = 1e - 3$)
Batch Size	32
Epochs	10 (Early Stopping)
L2 Regularization	0.001

3.5.3 DistilBERT (Fine-Tuning)

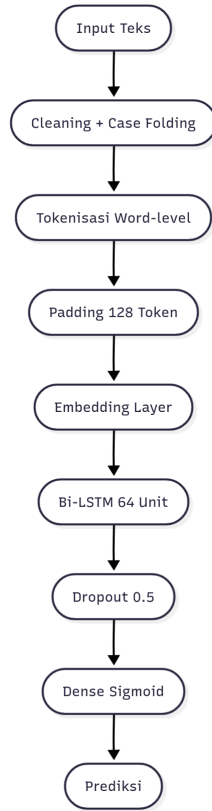
DistilBERT adalah varian ringan dari BERT yang menggunakan mekanisme *Knowledge Distillation* [Sanh et al., 2019]. Inti dari model ini adalah *Multi-Head Self-Attention*:

Proses *fine-tuning* dilakukan dengan menambahkan satu lapisan klasifikasi (*classification head*) di atas output token [CLS].

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

(Rumus 3.16)

Penanganan Ketidakseimbangan Data: Sama halnya dengan Bi-LSTM,



Gambar 3.3 Arsitektur Bi-LSTM

pelatihan DistilBERT menggunakan **WeightedLoss** pada objek *Trainer* dari pustaka HuggingFace, untuk menyeimbangkan kontribusi gradien dari kelas NFR.

Konfigurasi Hyperparameter:

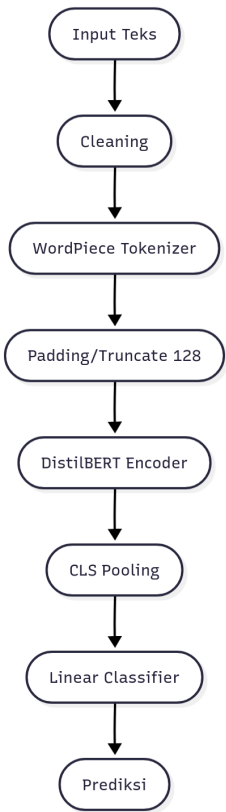
3.6 Skenario Pengujian

3.6.1 Pembagian Data

1. Rasio Data Latih/Uji: 80:20.
2. Metode: *Stratified Sampling* (mempertahankan rasio kelas).
3. *Random Seed*: 42 (untuk reproduktifitas).

Tabel 3.3 Konfigurasi Fine-tuning DistilBERT

Parameter	Nilai
Base Model	distilbert-base-multilingual-cased
Learning Rate	2×10^{-5}
Batch Size	16
Epochs	3 – 5
Optimizer	AdamW
Weight Decay	0.01
Warmup Steps	100



Gambar 3.4 Alur Fine-tuning DistilBERT

3.6.2 Stratified 5-Fold Cross Validation

Pengujian dilakukan menggunakan validasi silang 5-lipatan yang terstratifikasi. Hasil akhir merupakan rata-rata dari 5 iterasi pengujian untuk mengurangi bias varians pada estimasi performa.

3.7 Evaluasi dan Validasi

3.7.1 Metrik Performa

Evaluasi didasarkan pada elemen *Confusion Matrix* (TP, TN, FP, FN).

Precision:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (\text{Rumus 3.17})$$

Recall:

$$\text{Recall} = \frac{TP}{TP + FN} \quad (\text{Rumus 3.18})$$

F1-Score: Metrik utama yang digunakan adalah F1-Score karena memberikan

keseimbangan antara presisi dan recall, terutama pada dataset yang tidak seimbang sempurna.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (\text{Rumus 3.19})$$

3.8 Analisis Kesalahan

Analisis kesalahan (*Error Analysis*) dilakukan secara kualitatif dengan memeriksa sampel *False Positive* dan *False Negative*. Tujuannya adalah mengidentifikasi pola linguistik spesifik (seperti ambiguitas, negasi, atau struktur kalimat kompleks) yang gagal diprediksi dengan benar oleh model.

3.9 Lingkungan Pengembangan

Seluruh eksperimen dijalankan pada lingkungan komputasi awan Google Colab dengan spesifikasi:

1. **GPU:** NVIDIA Tesla T4 (16GB VRAM) untuk akselerasi pelatihan Deep Learning.
2. **RAM:** 12 GB.
3. **Framework:** TensorFlow 2.15, PyTorch 2.1, HuggingFace Transformers.
4. **Bahasa:** Python 3.10.

Daftar Pustaka

DAFTAR PUSTAKA

- [1] Kamaljit Kaur and Parminder Kaur. “Improving BERT model for requirements classification by bidirectional LSTM-CNN deep model”. *Computers and Electrical Engineering* 108 (2023), p. 108699. 0045-7906.
- [2] Bing Li and Xiuwen Nong. “Automatically classifying non-functional requirements using deep neural network”. *Pattern Recognition* 132 (2022), p. 108948. 0031-3203.
- [3] Fatih Yucalar. “Developing an Advanced Software Requirements Classification Model Using BERT: An Empirical Evaluation Study on Newly Generated Turkish Data”. *Applied Sciences* 13.20 (2023), p. 11127. 2076-3417.
- [4] Fajri Koto et al. “IndoLEM and IndoBERT: A Benchmark Dataset and Pre-trained Language Model for Indonesian NLP”. *Proceedings of the 28th International Conference on Computational Linguistics (COLING)*. 2020, pp. 757–770.
- [5] Donglin Di et al. “Building Dialogue Understanding Models for Low-resource Language Indonesian from Scratch”. *arXiv preprint arXiv:2410.18430* (2024).
- [6] Abdelkarim El-Hajjami, Nicolas Fafin, and Camille Salinesi. “Which AI Technique Is Better to Classify Requirements? An Experiment with SVM, LSTM, and ChatGPT”. *Joint Proceedings of REFSQ-2024 Workshops, Doctoral Symposium, Posters & Tools Track, and Education and Training Track*. Co-located with REFSQ 2024, Winterthur, Switzerland. 2024.
- [7] Ahmad F. Subahi. “BERT-Based Approach for Greening Software Requirements Engineering Through Non-Functional Requirements”. *IEEE Access* 11 (2023), pp. 103001–103012.