

BAB II

TINJAUAN PUSTAKA

Penelitian terkait klasifikasi kebutuhan perangkat lunak telah banyak dilakukan sebelumnya. Setidaknya terdapat enam studi terdahulu yang menunjukkan kemiripan dengan penelitian ini, baik dari segi tujuan klasifikasi, jenis dataset yang digunakan, maupun pendekatan metode yang diterapkan. Untuk memberikan gambaran yang lebih jelas mengenai posisi dan kontribusi penelitian ini dibandingkan dengan studi-studi sebelumnya, perbandingan tersebut disajikan dalam Tabel 2.1.

Penelitian pertama bertujuan untuk mengklasifikasikan *Non-Functional Requirements* (NFR) kedalam lima kategori, yaitu SE, US, O, MN, dan PE, menggunakan pendekatan *deep learning*. Dataset awal sebanyak 1.165 data diolah menjadi 914 data melalui proses *undersampling* untuk menyeimbangkan distribusi kelas. Data kemudian diproses melalui tahapan *case folding*, pembersihan simbol dan nama, lemmatisasi, tokenisasi, dan penanganan *Out-of-Vocabulary*. Dua model yang digunakan adalah DReqANN dan DReqBiLSTM, dengan pembagian data latih dan uji sebesar 80:20. Hasil evaluasi menunjukkan bahwa DReqANN memiliki performa lebih baik dengan *f1-score* 87% dibandingkan DReqBiLSTM yang mencapai 83%. Penelitian ini menunjukkan manfaat signifikan dalam otomatisasi klasifikasi kebutuhan NFR, yang dapat mendukung efisiensi analisis kebutuhan dalam proses pengembangan perangkat lunak [25].

Penelitian kedua menggunakan dataset PROMISE_exp yang terdiri dari 969 data perangkat lunak, yaitu 444 *Functional Requirements* (FR) dan 525 NFR [19]. Penelitian ini memanfaatkan model BERT dan variannya seperti RoBERTa dan DeBERTa untuk melakukan klasifikasi teks terhadap kebutuhan perangkat lunak, baik dalam klasifikasi biner FR dan NFR maupun 12 kelas. Meskipun hasil akurasi yang dicapai cukup tinggi, penelitian ini masih memiliki beberapa kekurangan. Proses *preprocessing* data kurang dijelaskan secara mendalam, seperti tidak adanya informasi tentang pembersihan teks, penghapusan *stopword*, atau normalisasi kata yang bisa meningkatkan kualitas data. Selain itu, evaluasi

model hanya menggunakan metrik akurasi tanpa menyertakan analisis mendalam seperti *precision*, *recall*, atau *f1-score* yang penting untuk memahami performa model secara lebih menyeluruh. Penelitian ini juga terbatas dalam eksplorasi model BERT, karena hanya menggunakan model skala besar seperti *bert-base* dan *roberta-base* tanpa mencoba varian ringan seperti Distil BERT atau Tiny BERT yang lebih efisien untuk implementasi riil. Di sisi lain, variasi dataset yang digunakan hanya terbatas pada PROMISE_exp, sehingga generalisasi model terhadap dataset lain belum dapat dibuktikan secara kuat [26].

Penelitian ketiga berfokus pada klasifikasi kebutuhan perangkat lunak (*software requirements*) kedalam 12 kategori berbeda, mencakup *requirement F* dan 11 kategori NFR, dengan menggunakan pendekatan *deep learning* berbasis model NoRBERT yang di-*fine-tune* dari BERT *base*. Dataset yang digunakan terdiri dari 969 kalimat *requirement*, data ini diproses melalui tahapan NLP lengkap seperti tokenisasi, POS *tagging*, NER, *dependency parsing*, lemmatisasi, serta penghapusan *stop words* dan kata-kata pendek. Penelitian ini menerapkan dua metode validasi silang, yakni *10-fold* dan *project-fold (p-fold) cross-validation*, untuk mengukur performa model secara menyeluruh. Hasil evaluasi menunjukkan nilai *f1-score* makro sekitar 0.50–0.51, yang menunjukkan tantangan dalam mengklasifikasi *requirement* ke banyak kelas namun tetap memberikan kontribusi penting dengan menyediakan *baseline* awal yang kuat dan metodologi yang transparan untuk pengembangan model klasifikasi kebutuhan perangkat lunak berbasis pembelajaran mendalam di masa depan [27].

Pada penelitian keempat memiliki perkembangan dengan pendekatan klasifikasi satu fase serta dua fase. Metode dua fase yang memisahkan FR dan NFR sebelum klasifikasi lebih lanjut menghasilkan akurasi yang lebih tinggi, terutama dengan teknik *ensemble learning*. Penggunaan model LSTM, BiLSTM, GRU, dan CNN menunjukkan perbandingan performa yang lebih komprehensif. Meskipun pendekatan ini lebih kompleks, hasilnya menunjukkan bahwa klasifikasi bertahap dapat meningkatkan kinerja model secara signifikan, khususnya dengan pendekatan *ensemble* yang memperhitungkan akurasi per kelas [28].

Penelitian kelima bertujuan mengklasifikasikan kebutuhan perangkat lunak ke dalam berbagai kategori menggunakan BERT (*base* dan *large*) yang di-*fine-tune* tanpa *preprocessing* manual pada 625 data (255 FR dan 370 NFR). Terdapat empat klasifikasi: (1) FR vs NFR, (2) NFR ke dalam empat subkategori (US, SE, O, PE), (3) NFR ke dalam 10 subkategori, dan (4) berdasarkan dimensi *Function* (F), *Quality* (Q), F+Q, dan *Neither*. Validasi dilakukan dengan *10-fold*, *split* 75/25, *p-fold*, dan *leave-one-project-out*. Hasil terbaik diperoleh dari NoRBERT *large* (*p-fold*) dengan *F1-score* 0.94 (NFR) dan 0.91 (FR); klasifikasi kedua mencapai F1 0.87; klasifikasi ketiga akurasi 0.82; dan klasifikasi keempat untuk kategori OnlyF mencapai *F1-score* 0.98. Penelitian ini menunjukkan efektivitas BERT dalam mengotomatisasi klasifikasi kebutuhan perangkat lunak secara akurat [29].

Penelitian keenam berfokus pada klasifikasi NFR dengan pemrosesan awal yang mencakup *stop word removal*, *tokenization*, *stemming*, dan *padding*. Model ANN dan CNN digunakan untuk klasifikasi, dengan CNN menunjukkan kinerja lebih baik (*f1-score* 92%). Meskipun jumlah data lebih sedikit dan hanya mengklasifikasikan 5 kelas NFR dari sebelum dilakukan pemangkasan dataset berjumlah 1.165 data, pendekatan ini menunjukkan bahwa CNN memiliki keunggulan dalam menangani teks terstruktur dengan baik. Namun, penelitian ini belum mempertimbangkan metode *embedding* yang lebih canggih atau teknik klasifikasi yang lebih kompleks [30].

Tabel 2.1 Perbandingan Penelitian Terdahulu

Keterangan	Penelitian						
	Penulis	[25]	[26]	[27]	[28]	[29]	[30]
	2025	2024	2024	2022	2021	2020	2019
Jumlah Dataset	Sebelum augmentasi data berjumlah 3.887, setelah augmentasi data berjumlah 2.920.	Sebelum <i>undersampling</i> NFR data berjumlah 1.165, setelah <i>undersampling</i> NFR berjumlah 914 dengan lima kategori (SE, US, O, MN, dan PE).	969 kalimat <i>requirement</i> terdiri dari 12 kelas (F dan 11 NFR)	969 kalimat <i>requirement</i> terdiri dari 12 kelas (F dan 11 NFR)	266 data FR dan 393 data NFR, tertotal 659 data.	255 data FR dan data 370 NFR, tertotal 625 data.	Sebelum pemangkasan data berjumlah 1.165 dan terdiri dari 10 kelas, setelah pemangkasan data berjumlah 914 terdiri dari 5 kelas.

Keterangan	Penelitian						
	Penulis	[25]	[26]	[27]	[28]	[29]	[30]
	2025	2024	2024	2022	2021	2020	2019
Pembagian Dataset	Training: 70%, Validation: 15%, Testing: 15%	Training: 80%, Testing: 20%	-	10-fold Cross-Validation, Project-fold (p-fold) Cross-Validation.	Training: 35%, Validation: 35%, Testing: 30%	10-fold stratified cross-validation, 75/25 split, Project-fold (p-fold), Leave-one-project-out (loPo)	-
Klasifikasi	Klasifikasi 12 kelas, kategori FR dan 11 kategori NFR (<i>Portability, Legal, Fault Tolerance, Maintainability,</i>	Lima kategori NFR yaitu, SE, US, O, MN, dan PE.	Klasifikasi biner kategori F dan NFR. Klasifikasi 12 kelas, kategori F dan 11 kategori NFR	Klasifikasi 12 kelas, kategori F dan 11 kategori NFR (A, L, LF, MN, PE, SC, SE, PO, US, O, dan FT).	Klasifikasi Satu Fase (17 Kelas), Klasifikasi Dua Fase: (A) Klasifikasi biner FR dan	Klasifikasi Pertama, FR dan NFR. Klasifikasi Kedua, US, SE, O, dan PE.	Klasifikasi lima kelas kategori NFR (MN, O, PE, SE, dan US)

Keterangan	Penelitian						
	Penulis	[25]	[26]	[27]	[28]	[29]	[30]
	2025	2024	2024	2022	2021	2020	2019
	<i>Availability, Scalability, Look and Feel, Performance, Security, Operability, Usability).</i>		(A, L, LF, MN, PE, SC, SE, PO, US, O, dan FT).		NFR, (B) Klasifikasi (FR dan NFR ke 17 Kelas); FRs (6 Kelas), NFRs (11 Kelas).	Klasifikasi Ketiga, 10 kelas yaitu L, FT, MN, A, SC, LF, PE, SE, O, dan US. Klasifikasi Keempat, <i>Function</i> (F), <i>Quality</i> (Q), F+Q, dan <i>Neither</i> .	
Data Preprocessing	<i>Removal NFR, Replace F to FR, Drop</i>	<i>Case folding, Removal (punctuation</i>	Tokenisasi (BertTokenizer)	<i>Tokenization, POS tagging, NER,</i>	<i>Case Folding, Tokenization, Lemmatization,</i>	<i>Tokenizer BERT, tanpa preprocessing</i>	<i>Removing Stop Words, Tokenization,</i>

Keterangan	Penelitian						
	Penulis	[25]	[26]	[27]	[28]	[29]	[30]
	2025	2024	2024	2022	2021	2020	2019
	<i>Duplicate, Deduplication by Normalized, Text Normalization, Back-translation (Augmentation), Tokenization Distil BERT & BERT.</i>	<i>marks, company names, hashtags, and other symbols), Lemmatization (WordNet), Tokenization, Out-of-Vocabulary handling dengan token</i>		<i>Dependency parsing, Lowercasing, Lemmatization, Removal of stop words & short words</i>	<i>Padding</i>	<i>manual, sampling (OS/US)</i>	<i>Lowercasing, Stemming, Vectorization, Padding (CNN)</i>
Model Deep Learning	Distil BERT, BERT Base, dan BERT Large.	DReqANN (Deep Requirement ANN),	BERT, RoBERTa, dan DeBERTa.	NoRBERT (berbasis BERT base, <i>fine-tuned</i>).	LSTM, BiLSTM, GRU, CNN, Mean	BERT (<i>base & large</i>), <i>fine-tuned</i> (NoRBERT).	ANN, dan CNN.

Keterangan	Penelitian						
	Penulis	[25]	[26]	[27]	[28]	[29]	[30]
	2025	2024	2024	2022	2021	2020	2019
		DReqBiLSTM (Deep Requirement Bi-directional LSTM).			ensemble, Accuracy as a weight ensemble, Accuracy per class as a weight.		
Metrik Evaluasi	Distil BERT: Accuracy: ?, Precision: ?, Recall: ?, F1-Score: ? BERT Base: Accuracy: ?, Precision: ?,	<i>Weighted Avg;</i> DReqANN (Precision: 90%, Recall: 85%, F1-Score: 87%), DReqBiLSTM	Klasifikasi biner FR dan NFR: BERT: 88,6%; RoBERTa: 88,6%; DeBERTa: 79,3%.	10-fold Cross Validation; Precision: 0,61; Recall: 0,48; F1-Score (Macro): 0,51, P-Fold Cross Validation:	Klasifikasi Satu Fase: Accuracy per class as a weight (Precision: 0,93; Recall: 0,93; F1-Score: 0,93),	Klasifikasi Pertama: NoRBERT (large, ep.=10) dengan p-fold, F1-score NFR: 0.94, F1-score F: 0.91	ANN Model (Precision: 82%-90%, Recall: 78%-85%, F1-Score: 84%), CNN Model (Precision:

Keterangan	Penelitian						
	Penulis	[25]	[26]	[27]	[28]	[29]	[30]
	2025	2024	2024	2022	2021	2020	2019
	<i>Recall: ?, F1-Score: ?</i> BERT Large: <i>Accuracy: ?, Precision: ?, Recall: ?, F1-Score: ?</i>	<i>(Precision: 85%, Recall: 82%, F1-Score: 83%)</i>	Klasifikasi 12 kelas: BERT: 76,2; RoBERTa: 71,71%; DeBERTa: 57,7%.	<i>Precision: 0,61; Recall: 0,48; F1-Score (Macro): 0,50.</i>	Klasifikasi Dua Fase: (A) <i>Accuracy per class as a weight ensemble</i> <i>(Precision: 0,96; Recall: 0,96; F1-Score: 0,96);</i> (B) <i>Accuracy per class as a weight ensemble</i>	Klasifikasi Kedua: NoRBERTmulti all (large, ep.=50), RF1: 0.87. Klasifikasi Ketiga: multiL50 (10-fold cross-validation, Accuracy: 0.82. Klasifikasi Keempat:	82%-94%, <i>Recall: 76%-97%, F1-Score: 92%)</i>

Keterangan	Penelitian						
	Penulis	[25]	[26]	[27]	[28]	[29]	[30]
	2025	2024	2024	2022	2021	2020	2019
					(Precision: 0,98; Recall: 0,98; F1-Score: 0,98);	NoRBERT (base, ep.=10) kategori OnlyF, F1-score: 0.98.	

Berdasarkan kajian terhadap berbagai penelitian terdahulu, dapat disimpulkan bahwa penelitian ini memberikan kontribusi yang signifikan dengan menggabungkan keunggulan pendekatan klasifikasi banyak kelas serta pemanfaatan berbagai model *deep learning* variasi BERT yaitu Distil BERT, BERT *Base*, dan BERT *Large* sebagai representasi model ringan, berat, dan besar. Penelitian ini menggunakan dua dataset dengan perbedaan tidak seimbang dan seimbang. Cakupan kelas meliputi 12 kategori (FR dan 11 kategori NFR), menjadikannya lebih komprehensif dibandingkan studi sebelumnya yang umumnya terbatas pada jumlah kelas yang lebih sedikit serta hanya berasal dari dataset PROMISE_Exp. Proses *preprocessing* minimalis yang dilakukan secara menyeluruh meliputi penghapusan kategori tidak relevan, duplikasi data, penyeragaman data, dan tokenisasi juga turut meningkatkan kualitas data masukan kedalam model.

Penelitian ini menetapkan LLM GPT-4o sebagai *baseline* dalam proses klasifikasi, untuk kemudian membandingkan dan mengevaluasi performa berbagai arsitektur *deep learning* guna menentukan model terbaik dalam konteks klasifikasi kebutuhan perangkat lunak. Dengan cakupan kategori yang lebih lengkap, jumlah data yang lebih besar, serta pendekatan evaluasi yang luas dan objektif, penelitian ini memiliki keunggulan dalam menghasilkan sistem klasifikasi yang lebih akurat, efisien, dan aplikatif dengan antarmuka berbasis *website* di lingkungan pengembangan perangkat lunak modern.