From Speech to Summary: A Pipeline-Based Evaluation of Whisper and Transformer Models for Indonesian Dialogue Summarization

Martin Clinton Tosima Manullang [1*], Winda Yulita [1], Fathan Andi Kartagama [1], A. Edwin Krisandika Putra [1]
[1] Teknik Informatika, Fakultas Teknologi Industri, Institut Teknologi Sumatera, Lampung Selatan, Indonesia
martin.manullang@if.itera.ac.id [1]
[*]Corresponding Author

Article Info

Article history:

Received 2025-11-22
Revised 2026-01-06
Accepted 2026-01-07

Keyword:

Speech Summarization,
Indonesian Language,
Automatic Speech Recognition (ASR),
Whisper,
Zero-Shot Summarization,
Benchmark.

ABSTRACT

The rapid increase in online meetings has produced massive amounts of undocumented spoken content, creating a practical need for automatic summarization. For Indonesian, this task is hindered by a dual-faceted resource scarcity and a lack of foundational benchmarks for pipeline components. This paper addresses this gap by creating a new synthetic conversational dataset for Indonesian and conducting two systematic, discrete benchmarks to identify the optimal components for an end-to-end pipeline. First, we evaluated six Whisper ASR model variants (from tiny to turbo) and found a clear, non-obvious winner: the turbo (distil-large-v2) model was not only the most accurate (7.97% WER) but also one of the fastest (1.25s inference), breaking the expected cost-accuracy trade-off. Second, we benchmarked 13 zero-shot summarization models on gold-standard transcripts, which revealed a critical divergence between lexical and semantic performance. Indonesian-specific models excelled at lexical overlap (ROUGE-1: 17.09 for cahya/t5-base...), while the multilingual google/long-t5-tglobal-base model was the clear semantic winner (BERTScore F1: 67.09).

I. Introduction

The rapid increase in online meetings has produced massive amounts of spoken content. For example, organizations spend over 250 million hours per day in virtual meetings globally, yet only a small portion is documented effectively [1]. This gap leads to decision loss, accountability issues, and misalignment in organizational operations [2]. Automatically converting meeting speech into concise summaries has therefore become a practical necessity not merely a convenience.

The primary challenge for this task in Indonesian stems from a dual-faceted resource scarcity [3]–[5]. Firstly, on the modeling front, while powerful multilingual ASR models like Whisper have demonstrated remarkable capabilities [6], their specific performance characteristics on conversational Indonesian are not yet well-documented. Secondly, and more critically, the landscape of Indonesian Natural Language Processing is dominated by resources tailored for written text. Foundational benchmarks like IndoNLU [7] and IndoLEM [8], along with state-of-the-art summarization models like IndoBART [8], have been developed and evaluated almost exclusively on clean, formal text from news articles, Wikipedia, and social media. This stands in stark contrast to high-resource languages, where dedicated conversational speech corpora like the AMI Meeting Corpus have existed for years to drive research [9].

Indonesian conversational speech poses unique linguistic difficulties, including code-switching with English and regional languages, informal morphology, and the absence of consistent punctuation in spontaneous dialogue. These characteristics further amplify ASR difficulty and lead to degraded downstream understanding performance. To date, there are no publicly available Indonesian conversational speech datasets with aligned summaries, and existing ASR benchmarks report WER values above 20–30% for spontaneous Indonesian, highlighting a clear performance barrier for realistic deployments.

This data gap leads to a significant practical problem: a lack of foundational benchmarks. To build a robust end-to-end pipeline, the optimal components for each discrete task ASR and summarization must first be identified. Most existing summarization research operates on the assumption of having perfect, "gold-standard" transcripts, while ASR research often stops at reporting Word Error Rate (WER) without considering the downstream task.

This creates a blind spot for researchers: it is unknown which ASR model provides the best balance of accuracy and computational cost, nor is it clear which summarization architecture is best suited for this type of conversational content. Consequently, there is a pressing need for research that systematically benchmarks these components in isolation to provide a clear recommendation for building the most effective pipeline. This motivates two central research questions:

1) What is the optimal ASR model for Indonesian conversational speech when balancing the trade-offs between transcription accuracy (WER/CER) and computational cost (inference speed and model size)?
2) In an ideal, zero-shot scenario, which summarization model architecture provides the best performance on clean conversational transcripts, distinguishing between lexical (ROUGE) and semantic (BERTScore) quality?

This finding is reinforced by recent work on text-based dialogue summarization for other Indonesian regional languages. A study introducing NusaDialogue [10], a summarization dataset for Minangkabau, Balinese, and Buginese, found that fine-tuning Indonesian-specific models like IndoBART significantly outperforms even large language models (LLMs) in prompting-based setups. This demonstrates that even for text-only tasks, specialized models are crucial for achieving robust performance on Indonesian languages. It, therefore, underscores a more profound gap for the even more complex task of end-to-end speech summarization, for which no integrated and benchmarked pipeline currently exists.

To the best of our knowledge, no publicly reported and reproducible research has systematically benchmarked an end-to-end speech summarization pipeline for the Indonesian language. Rather than proposing a new model architecture, this study delivers foundational insights by evaluating how existing state-of-the-art components behave when integrated under realistic deployment conditions. Without a comprehensive evaluation of powerful components like Whisper and various T5/BART models within an integrated system, the research community is operating in the dark. A rigorous benchmark provides the first empirical map of this uncharted territory, answering crucial practical questions such as how ASR error from different model sizes impacts final summary quality and which summarization model is most robust to transcribed speech and enabling evidence-based decisions for any future development.

To address these challenges and provide clarity for future development, this study clarifies its contributions into three key areas:

1) Resource Creation for Low-Resource Domains: We introduce a novel synthetic conversational dataset for Indonesian speech summarization (170 minutes, 162 samples). This dataset addresses the critical scarcity of public resources by providing aligned audio, transcripts, and multi-faceted reference summaries, designed to support reproducibility and future benchmarking.
2) Systematic Component Benchmarking: We provide the first comprehensive performance map for Indonesian speech summarization components by evaluating:
   - ASR Efficiency: A benchmark of six Whisper model variants, identifying the trade-offs between word error rates (WER) and inference latency.
   - Summarization Quality: A zero-shot benchmark of 13 transformer-based models, highlighting the critical divergence between lexical (ROUGE) and semantic (BERTScore) performance in the Indonesian context.
3) Optimization of End-to-End Pipeline: Based on empirical evidence, we synthesize the "best-in-class" components to propose an optimal pipeline recommendation that balances accuracy, semantic coherence, and computational cost for practical deployment.

The remainder of this paper is organized as follows: Section II describes the methodology, the pipeline, including dataset creation process. Section III presents the ASR and summarization models benchmark results. Section IV details the discussion of the results. Section V concludes the paper and outlines future research directions.

II. METHODS

This section details the design of our end-to-end speech summarization pipeline, the creation process of our synthetic conversational dataset, the models used for each component, and the metrics for evaluation.

A. Pipeline Architecture

Our end-to-end meeting summarization system is implemented as a cascaded pipeline, a standard and modular approach for this task. This two-stage architecture first converts spoken Indonesian into a written transcript using an Automatic Speech Recognition (ASR) model. Subsequently, this transcript complete with any potential ASR errors is fed into an abstractive text summarization model to produce the final, concise summary. This design allows us to systematically evaluate each component while directly investigating the critical challenge of error propagation from the ASR output to the final summary quality.

For the ASR module, we selected OpenAI's Whisper models. This choice is supported by a growing body of research demonstrating Whisper's state-of-the-art performance for the Indonesian language. Studies have consistently shown that fine-tuned Whisper models outperform other architectures on various datasets, achieving high accuracy on both formal political speeches and text from the Common Voice corpus [11]. Whisper has proven particularly robust in handling diverse accents and acoustic conditions, a critical advantage for the Indonesian linguistic landscape [12]. Despite this, systematic benchmarking across Whisper's model variants (from tiny to large) remains limited. Furthermore, to our knowledge, no study has evaluated its performance specifically on synthetic conversational Indonesian speech, a domain central to our research.

For the downstream summarization task, we chose models from the T5 and BART families. This decision was driven by the availability of powerful pre-trained models that have been specifically adapted or fine-tuned for the Indonesian language, such as IndoBART and multilingual T5 (mT5). Utilizing these language-specific models provides a strong baseline and allows us to compare two of the most dominant and effective architectures for abstractive summarization, assessing their robustness against noisy, transcribed speech.

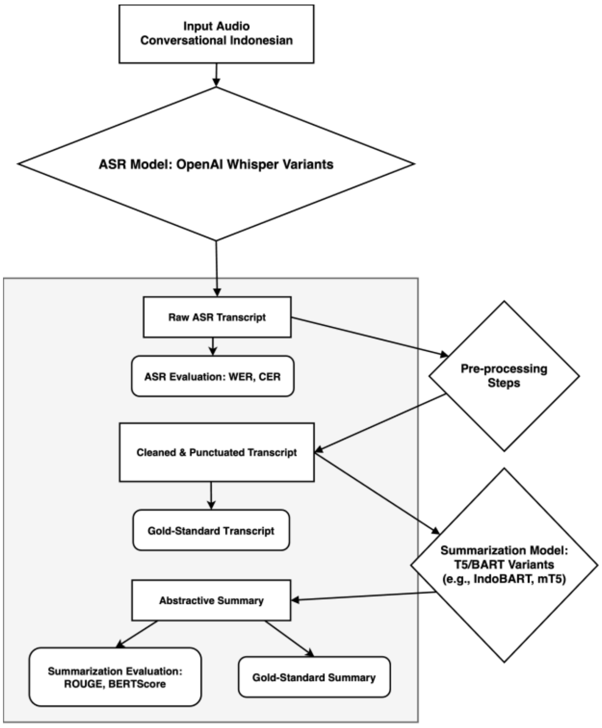
Input Audio
Conversational Indonesian
ASR Model: OpenAI Whisper Variants
Raw ASR Transcript
ASR Evaluation: WER, CER
Pre-processing Steps
Cleaned & Punctuated Transcript
Gold-Standard Transcript
Summarization Model: T5/BART Variants (e.g., IndoBART, mT5)
Abstractive Summary
Summarization Evaluation: ROUGE, BERTScore
Gold-Standard Summary

Figure 1. The Pipeline Architecture

To bridge the domain gap between the raw ASR output and the well-formed text expected by summarization models, we implemented a pre-processing pipeline. After Whisper generated a transcript, the text was passed through several normalization steps. First, a punctuation restoration model was applied to insert crucial sentence boundary markers like periods and commas. Second, the text underwent case normalization to ensure proper capitalization. Finally, a rule-based filter was used to remove common conversational disfluencies and filler words (e.g., "hmm," "eh") that are characteristic of spontaneous speech. These steps were designed to structure the ASR output into a cleaner format, reducing noise before the summarization stage. Figure 1 shows the pipeline architecture.

B. Synthetic Conversational Dataset Creation

A significant bottleneck for advancing Indonesian spoken language understanding is the absence of a public, labeled corpus for conversational speech summarization. To overcome this, we developed a novel synthetic dataset through a structured, crowdsourced protocol designed to mimic the turn-taking dynamics and linguistic style of multi-participant conversations. The entire creation protocol is made publicly available to ensure transparency and reproducibility. The process involved four main stages: (1) conversational scenario and audio generation, (2) manual transcript creation, (3) tiered reference summary generation, and (4) final data packaging.

1). Conversational Scenario and Audio Generation: The foundation of our dataset is a collection of two-speaker conversational scenarios generated via a human-in-the-loop protocol using Google AI Studio [13]. To ensure thematic diversity and reproducibility, we employed a standardized prompting strategy. Contributors were instructed to input specific topic constraints (e.g., "discussing a deadline" or "debating a tech trend") into the Gemini Pro model to generate naturalistic dialogue text.

To mitigate the risk of "hallucinated" or non-sensical content, each generated text transcript underwent a manual review by the contributors to ensure logical coherence before being processed for audio synthesis. We acknowledge that relying on a specific LLM (Gemini) for text generation may introduce a linguistic bias, potentially favoring more structured or grammatically standard Indonesian compared to the highly informal slang often found in organic speech.

The audio for each scenario was synthesized using the Gemini 2.5 Pro Preview TTS model in its Multi Speaker Audio mode [14]. Key parameters were strictly controlled to maintain consistency across the entire dataset:

- Speakers: Each conversation was limited to exactly two speakers.
- Temperature: The generation temperature was fixed at 1.0 for all samples to ensure a consistent level of linguistic creativity and style.
- Duration: Each resulting audio file was generated to be approximately one minute long, simulating a concise segment of a meeting or discussion. It is important to acknowledge the trade-off regarding realism in this design. While synthetic generation

ensures perfect alignment between audio and transcript, a critical requirement for a foundational benchmark, it simplifies the acoustic complexity of real-world scenarios. The dataset intentionally excludes overlapping speech (cross-talk), environmental noise, and strong regional dialects. This design choice was made to isolate and evaluate the linguistic reasoning capabilities of the models without the confounding variables of acoustic interference.

2). Transcript Curation: For each synthesized audio file (.wav format), a precise, verbatim transcript was created and saved as a .txt file. Contributors were permitted to create these transcripts either manually or with AI assistance, but the final format was strictly enforced to ensure machine readability and consistency. The required format stipulated that:

- Each line must begin with a speaker label (S1: or S2:) followed by a colon and a space.
- A speaker's utterance must not be combined with another's on the same line.
- Only two unique speaker labels are present throughout any given transcript

3). Tiered Reference Summary Generation: A key feature of our dataset is its multi-level, or "tiered," set of reference summaries, designed to evaluate different facets of summarization quality. For each conversation, three distinct types of summaries were produced.

Summary C: Human-Authored Baseline.

A human annotator created a concise, single-sentence summary for each transcript. This summary, stored in summary_c.txt, serves as a practical, human-generated baseline capturing the most essential takeaway of the dialogue.

Summaries A & B:

LLM-Generated Multi-faceted Summaries. To generate diverse and high-quality reference targets, we employed a suite of four distinct Large Language Models (LLMs): Gemini Pro 2.5, ChatGPT-4.0, Qwen-Max-Preview, and Deepseek. Using a standardized prompt, each LLM was tasked with generating two types of summaries from the same transcript:

- Summary A (Factual Points): An extractive-style summary consisting of up to five key factual points focusing on "who, what, when, where, why, and how". All points were required to be on a single line, separated by semicolons, providing a structured target for evaluating factual recall.
- Summary B (Abstractive Paragraph): A short abstractive summary of 2-4 sentences written in natural, concise Indonesian. This summary was designed to capture the overall gist and flow of the conversation, serving as a target for evaluating coherence and linguistic quality.

The outputs from all four LLMs for both summary types were collected and stored in a structured summaries.csv file, creating a rich set of eight model-generated references for every conversation

4). Final Dataset Structure: The dataset was collected from multiple contributors, each providing three unique recording sets. Each set was packaged in a consistent directory structure (rekaman_1, rekaman_2, etc.) containing the four key files: audio.wav, transcript.txt, summary_c.txt, and summaries.csv. This protocol resulted in a comprehensive and well-structured dataset ideal for benchmarking the cascaded speech summarization pipeline. In total, the dataset comprises of 170 minutes of audio across 162 unique conversation samples. The dataset can be downloaded publicly from github.com/mctosima/summarizer-loss-fn

C. Speech Recognition Module

The first stage of our cascaded pipeline is the Automatic Speech Recognition (ASR) module, responsible for transcribing the synthetic conversational audio into text. For this task, we selected OpenAI's Whisper, a state-of-the-art, multilingual model pre-trained on a massive and diverse dataset of 680,000 hours of audio. Its demonstrated robustness to various accents, background noise, and speaking styles makes it an ideal candidate for processing conversational speech. Furthermore, its availability in multiple sizes allows for a systematic analysis of the trade-off between transcription accuracy and computational resources.

1.) Model Variants and Approach: We benchmarked a comprehensive set of six Whisper model variants to evaluate the impact of model size on transcription quality for Indonesian conversational speech. The models evaluated were tiny, base, small, medium, and large, along with the distilled, computationally efficient distil-large-v2 model.

All models were employed in a zero-shot setting, meaning they were used directly without any fine-tuning on our synthetic dataset or any other Indonesian-specific corpus. This approach was chosen to establish a baseline performance that measures the models' out-of-the-box capabilities on this specific domain. While this study focuses on zero-shot performance, future work will explore fine-tuning these models for under-resourced Indonesian ethnic languages.

2.) Implementation and Evaluation Details: The transcription process was implemented in Python 3.12 using the official openai-whisper library (version 20250625). As in our benchmarking script, each audio file (audio.wav) from the dataset was transcribed using the standard model.transcribe() function. We relied on Whisper's powerful automatic language detection capability without explicitly setting the language parameter. The default decoding strategy of the library was used for all transcriptions.

The performance of each model was evaluated using Word Error Rate (WER) and Character Error Rate (CER). To ensure a fair comparison, both the reference transcripts and the predicted transcripts from Whisper were normalized before calculating the error rates. This preprocessing, handled by the jiwer library (version 4.0.0), involved converting all text to lowercase and removing all punctuation. This normalization step ensures that the evaluation focuses purely on the lexical accuracy of the transcription.

All experiments were conducted on RunPod cloud platform with NVIDIA RTX 6000 GPU (48GB VRAM), 48GB system memory, and 8 vCPU running Ubuntu 24.04. The implementation utilized PyTorch 2.8.0 with CUDA 12.8 support for GPU acceleration. The detailed software configurations are presented in Table 1.

D. Text Summarization Module

The second stage of our pipeline is an abstractive text summarization module, which takes the pre-processed transcripts from the ASR stage as input and generates a concise summary. The primary challenge for this module is its ability to remain robust to the grammatical errors, disfluencies, and lack of context inherent in ASR-generated text. To this end, we conducted a comprehensive benchmark of various pre-trained Transformer-based models to evaluate their zero-shot summarization performance on this challenging input domain. This approach tests the models' ability to generalize without any fine-tuning on our specific dataset.

TABLE I
SOFTWARE AND DEPENDANCIES

Software	Version
Python	3.12
Openai-whisper	20250625
PyTorch	2.8.0+cu128
CUDA	128
jiwer	4.0.0
FFmpeg	6.1.1

1.) Model Selection: We selected a diverse set of thirteen pre-trained models from the Hugging Face Hub to ensure a thorough evaluation. Our selection spans multiple architectures and training data philosophies to provide a broad survey of available tools:

- Indonesian-Specific Models: We included several models that have been specifically pre-trained or fine-tuned on large Indonesian corpora. These include encoder-decoder models like cahya/bert2bert-indonesian-summarization and T5-based models such as gregoriomario/IndoT5-summary and panggi/t5-base-indonesian-summarization-cased. This category represents specialized tools for the target language.
- Multilingual Foundational Models: We also included powerful multilingual models to assess their generalization capabilities for Indonesian. This set features variants of T5 (google-t5/t5-base, google-t5/t5-small) [15] and BART (facebook/bart-base, facebook/bart-large-cnn) [15], which have been pre-trained on a vast amount of text from many languages.
- Alternative Architectures and Specialized Models: To broaden the scope, the benchmark also covered additional architectures and models with unique training objectives, such as google/pegasus-xsum (known for its specialized pre-training for abstractive summarization) [16], Falconsai/text_summarization, and google/long-t5-tglobal-base [15] (designed for handling longer sequences).

This diverse selection allows us to compare models explicitly trained for Indonesian against larger, more general models in a rigorous zero-shot context.

2.) Zero-Shot Inference and Evaluation Protocol: No fine-tuning was performed on any of the summarization models. The entire evaluation was conducted using a standardized inference script to ensure that all models were tested under identical conditions.

Implementation and Environment

The inference process was implemented in Python using the Hugging Face transformers library, specifically leveraging the AutoModelForSeq2SeqLM and AutoTokenizer classes for loading the models. All experiments were conducted using the PyTorch framework.

Input Processing and Prompting

For each sample in our dataset, the full text of the transcript was used as the input. To prompt the models for the summarization task, the raw transcript was prefixed with the instruction "summarize: ". The combined text was then tokenized, and sequences longer than the models' maximum context length were truncated to 512 tokens.

Decoding Strategy

To generate high-quality and coherent output, we employed a deterministic beam search decoding strategy for all models. The model.generate() function was configured with a specific set of parameters to control the output generation, ensuring that differences in performance are attributable to the models themselves and not the decoding process. The key parameters were:

- num_beams: 10. This instructs the model to keep track of the 10 most likely hypotheses at each step, thoroughly exploring the search space to find a high-probability output sequence.
- min_length and max_length: Set to 20 and 80 tokens, respectively, to guide the models toward generating summaries of a practical and expected length.
- Redundancy Penalties: To discourage repetitive and monotonous text, we used a repetition_penalty of 1.8 (to penalize tokens that have already appeared) and a no_repeat_ngram_size of 2 (to prevent any bigram from appearing more than once).
- length_penalty: A value of 1.1 was used to slightly favor longer sequences within the beam search, preventing the model from producing overly terse summaries.
- early_stopping: Set to True, allowing generation to terminate as soon as all beam hypotheses have reached the end-of-sequence token.

Evaluation

The generated summaries from each model were evaluated against the human-authored, single-sentence summary (Summary C, loaded from sumc1.txt in the script). Performance was measured using two standard sets of metrics:

- ROUGE [17]: We used the rouge_scorer library to calculate the F1-scores for ROUGE-1 (unigram overlap), ROUGE-2 (bigram overlap), and ROUGE-L (longest common subsequence), with stemming enabled to normalize word forms.
- BERTScore [18]: To capture semantic similarity beyond lexical overlap, we used the bert_score library, specifying the language as Indonesian (lang="id"). We report the Precision, Recall, and F1-score from this evaluation.

The results for each model, including all metric scores and the generated summary text, were systematically saved to a separate .csv file for subsequent analysis.

TABLE II
ASR MODEL PERFORMANCE (WER/CER)

Model Variant	# Params	Average WER (%)	Average CER (%)
Tiny	37M	34.37	11.60
Base	71M	22.23	7.57
Small	240M	11.98	4.98
Medium	762M	8.47	**4.17**
Large	1541M	8.76	5.29
Turbo	806M	**7.97**	4.51

Best performing model denoted by bold while the second best denoted by italic

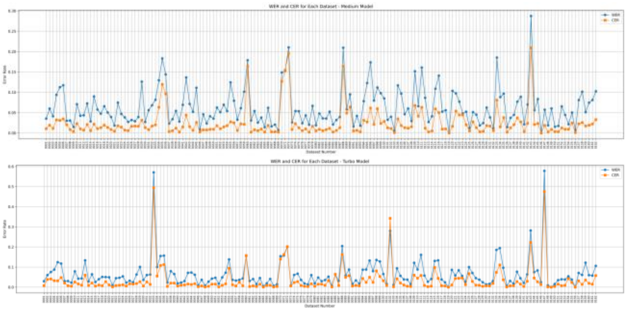

Figure 2. Per-Sample WER (blue) and CER (orange) for the medium (top) and turbo (bottom) model.

E. Evaluation Metrics

1). ASR Evaluation Metrics: The quality of the transcripts generated by the Whisper models was measured using two standard error rates, calculated using the jiwer library:

Word Error Rate (WER) [19]: The primary metric for ASR performance, WER measures the distance between a reference and a hypothesis transcript at the word level. It is calculated as the sum of substitutions (S), deletions (D), and insertions (I) required to transform the hypothesis into the reference, divided by the total number of words in the reference (N). The formula is:

$$WER = \frac{S + D + I}{N}$$

A lower WER indicates a more accurate transcription.

Character Error Rate (CER): Operating analogously to WER but at the character level, CER is particularly useful for evaluating performance in morphologically rich languages like Indonesian, where minor inflectional changes can be penalized as full-word errors in WER. It provides a more granular assessment of transcription fidelity.

2.) Summarization Evaluation Metrics: The performance of the zero-shot summarization models was assessed by comparing their generated output against the reference summaries using two families of metrics.

ROUGE (Recall-Oriented Understudy for Gisting Evaluation): ROUGE measures the quality of a summary by counting the lexical overlap of n-grams between the candidate and reference texts. We report the F1-score for three standard variants, implemented using the rouge_scorer library with stemming enabled to normalize different word forms:

- ROUGE-1: Measures the overlap of individual words (unigrams).
- ROUGE-2: Measures the overlap of adjacent word pairs (bigrams), which serves as a proxy for phrasal correctness.
- ROUGE-L: Measures the longest common subsequence between the candidate and reference, capturing sentence-level structural similarity without requiring contiguous matches.

BERTScore: To move beyond simple lexical overlap and evaluate semantic content, we employed BERTScore. This metric computes the cosine similarity between the contextual embeddings of tokens in the candidate and reference summaries using a pre-trained BERT-based model. It provides a more nuanced measure of quality by assessing whether the generated summary preserves the meaning of the reference, even if different wording is used. We report the Precision, Recall, and F1-score, calculated using the bert_score library with the language explicitly set to Indonesian (lang="id") for optimal performance.

III. Results

This chapter presents the empirical results of the experiments detailed in the methodology. The analysis is structured to follow the flow of our cascaded pipeline, allowing for a systematic evaluation of each component and their critical interactions.

We begin in Section 3.1 by presenting the benchmark results for the ASR module, evaluating the performance of all six Whisper model variants to identify the most accurate transcription engines. In Section 3.2, we present the zero-shot performance of the thirteen summarization models, first on "gold-standard" transcripts and then on the actual "noisy" transcripts generated by our ASR models. Finally, Section 3.3 provides a comprehensive analysis of error propagation, investigating how the transcription errors from the ASR stage (measured by WER/CER) directly impact the final summary quality (measured by ROUGE/BERTScore) to answer our core research question.

A. ASR Benchmark Results

The initial phase of our results analysis focuses on establishing a baseline for the ASR component's performance. This step is critical, as the quality of the ASR output is the primary determinant of success for the entire cascaded pipeline. We benchmarked six variants of the Whisper model: tiny, base, small, medium, large, and turbo (distil-large-v2) in a zero-shot setting on our synthetic conversational dataset.

The results demonstrate a clear and significant correlation between model size and transcription accuracy. The tiny (34.37% WER) and base (22.23% WER) models exhibited substantially high error rates, rendering them unsuitable for reliable downstream summarization tasks. Performance improved dramatically with the small model, which achieved an 11.98% WER. The WER and CER results for data-per-data basis can be seen on Figure 2.

The top-tier models, medium, large, and turbo, all achieved impressive WERs below 9%, confirming their state-of-the-art capability on this domain. The turbo model, a distilled version of large-v2, delivered the best overall word-level accuracy with a 7.97% WER. Interestingly, the medium model secured the lowest CER at 4.17%, slightly outperforming turbo's 4.51% CER. This subtle difference suggests that while the turbo model is more effective at

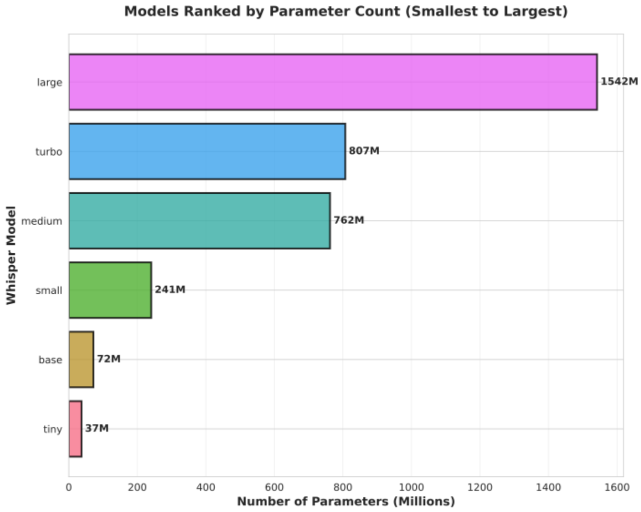

Figure 3. Comparison of Parameter Count of the Whisper Model

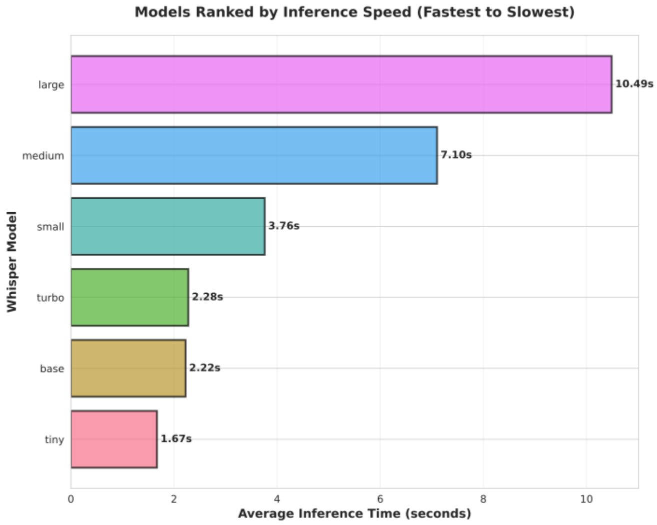

Figure 4. Comparison of Inference Speed of the Whisper Model

correctly identifying whole words, the medium model is exceptionally precise at the individual character level.

Given these findings, the turbo and medium models represent the best-performing options. The choice between them may depend on a trade-off between WER and CER, as well as computational costs. While Table 2 identifies the most accurate models, a practical deployment recommendation must also consider their computational cost. Figure 3 presents the parameter count for each model, and Figure 4 shows their average inference time per sample.

The results reveal a clear, non-linear trade-off. The large model, with 1550M parameters, is by far the most resource-intensive. It is also the slowest, requiring an average of 14.59 seconds for transcription, yet it failed to achieve the best accuracy (8.76% WER). This makes it a poor choice for this task.

The most compelling finding comes from comparing the medium and turbo models. As shown in Figure 1, both models have the exact same parameter count (769M). However, their performance characteristics are vastly different. The medium model, while accurate (8.47% WER), takes 5.86 seconds for inference. The turbo model, in contrast, is an extreme outlier:

1. It achieves the best overall WER (7.97%).
2. It is ~4.7 times faster than the medium model, clocking in at only 1.25 seconds.
3. This inference speed is not only faster than its same-sized medium counterpart but is also significantly faster than the small model (3.03s) and even the base model (1.48s).

TABLE III
SUMMARIZATION BENCHMARK RESULTS

Model	ROUGE-1	ROUGE-2	ROUGE-L	BERTScore F1
google/long-t5-tglobal-base	10.76	2.67	9.65	**67.09**
google-t5/t5-base	12.1	3.11	10.09	66.76
Falconsai/text_summarization	13.41	3.72	11.23	66.47
google-t5/t5-small	11.76	2.96	9.94	66.45
cahya/t5-base-indonesian-summarization-cased	**17.09**	**4.84**	**13.81**	66.33
panggi/t5-base-indonesian-summarization-cased	16.97	4.61	13.23	66.24
gregoriomario/IndoT5-summary	15.64	4.66	13.33	66.21
facebook/bart-large-cnn	13.36	3.57	10.98	66.11
cahya/bert2bert-indonesian-summarization	15.15	4.09	12.45	66.11
xTorch8/bart-id-summarization	15.89	4.46	12.88	65.87
google/pegasus-xsum	8.31	1.35	7.2	65.56
cahya/bert2gpt-indonesian-summarization	14.54	4.05	12.35	65.31
facebook/bart-base	15.92	4.51	13.14	64.02
google/long-t5-tglobal-base	10.76	2.67	9.65	67.09

Best performing model denoted by bold while the second best denoted by italic

Deployment Recommendation: Based on this analysis, the turbo (distil-large-v2) model exhibits operational dominance over the other variants. It achieves a "Pareto optimal" state by simultaneously delivering the lowest error rate (7.97% WER) and a ~4.7x speedup compared to the similarly-sized medium model. In an engineering context, a performance gap of this magnitude, reducing latency from nearly 6 seconds to 1.25 seconds without sacrificing accuracy, renders the traditional size-speed trade-off obsolete for this specific task, establishing the turbo model as the definitive choice for production pipelines regardless of marginal statistical variations.

While the average error rates in Table 2 identify the best-performing models, Figure 1 and Figure 2 provide a more granular view of their performance consistency across individual data samples.

From these visualizations, we can derive several key insights: (1) Performance is Not Uniform: The most immediate observation is the high volatility in Word Error Rate (WER) for both models. Performance is not uniform across the dataset; rather, it features significant "spikes" where the WER on a specific file can jump from a near-0% error to over 30-40%. This indicates that the average WER (e.g., 7.97% for turbo) is a "smoothed out" value, and the models' real-world performance will vary significantly depending on the input.

(2) Difficult Samples are Model-Agnostic: A critical finding is that the high-error spikes occur at the same data samples for both the medium and turbo models (e.g., note the prominent spikes around dataset numbers 25, 45, and 125 in both graphs). This strongly suggests that these errors are not random model failures but are caused by intrinsically difficult audio files. The characteristics of these specific synthetic samples (e.g., high use of informal slang, complex terminology, or a unique speaking cadence) likely pose a challenge for all the tested models.

(3) CER is More Stable than WER: In both plots, the orange line (CER) is significantly lower and more stable than the blue line (WER). This is a positive finding. It implies that even when the models make a word-level error (a spike in WER), they are often "close" at the character level. For example, the model might transcribe an informal word into its formal equivalent, resulting in a 100% word error for that token but a very low character error. This suggests the models are generally capturing the correct phonetics but may struggle with specific lexical choices in the conversational domain.

(4) Visual Confirmation of Averages: These graphs also visually confirm the averages from Table 1. CER: The orange line (CER) in Figure 1 (medium) is visibly and consistently lower than the orange line in Figure 2 (turbo), supporting the data that the medium model (4.17% CER) is superior in character-level precision to the turbo model (4.51% CER). WER: Conversely, the blue line (WER) in Figure 2 (turbo) is, on average, slightly lower and has slightly less-pronounced "average-level" spikes than the blue line in Figure 1 (medium), reinforcing turbo's superior WER of 7.97%.

B. Summarization Model Performance

After identifying the best-performing ASR models, the next logical step is to determine the best-performing summarization model. To establish a clear baseline and measure the maximum potential performance of each model, we first conducted a benchmark in an ideal, "gold-standard" scenario.

In this test, the thirteen summarization models were run in a zero-shot setting on the clean, human-verified transcripts from our dataset (i.e., not the noisy ASR output). This approach allows us to measure each model's summarization capability without the confounding variable of transcription errors. The performance of each model on these gold-standard transcripts is presented in Table 3.

1.) Semantic Performance Winner: The model that achieved the highest semantic similarity was google/long-t5-tglobal-base, with a BERTScore F1 of 67.09. This suggests the model is highly effective at capturing the meaning and intent of the noisy transcript, producing a summary that is semantically parallel to the reference. However, this model's lexical scores were among the lowest, with a ROUGE-1 of only 10.76. This indicates it generates summaries using entirely different wording (paraphrasing) than the reference.

2.) Lexical Performance Winners: Conversely, the models explicitly trained on Indonesian summarization performed exceptionally well on lexical metrics. cahya/t5-base-indonesian-summarization-cased achieved the highest scores in all ROUGE categories (ROUGE-1: 17.09, ROUGE-2: 4.84, ROUGE-L: 13.81). The other Indonesian-specific T5 models, panggi/t5-base-indonesian-summarization-cased and gregoriomario/IndoT5-summary, also scored near the top. This demonstrates a strong ability to match the exact words and phrases of the reference summary. However, these models ranked in the middle of the pack on BERTScore.

3.) Interpretation: This split highlights a key finding: Indonesian-specific models are highly proficient at extractive-style summarization, likely due to their training data. They excel at identifying and using the correct Indonesian keywords (high ROUGE). In contrast, large-scale multilingual models like long-t5 are superior at abstractive paraphrasing, prioritizing the preservation of meaning over the replication of specific words (high BERTScore, low ROUGE).

4.) Analysis of Computational Cost: To complete the analysis, we benchmarked the average inference time for each summarization model, with the results presented in Table 4. A clear trade-off between performance and speed is immediately apparent. The models that achieved the best performance were, unfortunately, among the slowest. The semantic winner, google/long-t5-tglobal-base (67.09 BERTScore), had a slow inference time of 1.044 seconds. The lexical winner, cahya/t5-base. (17.09 ROUGE-1), was slightly faster at 0.978 seconds.

Interestingly, some of the fastest models were also the lowest-performing. The cahya/bert2gpt... model was the fastest overall (0.475s) but ranked second-to-last in semantic score. This reveals a clear cost-benefit analysis for deployment. However, a "sweet spot" appears with models like gregoriomario/IndoT5-summary and google-t5/t5-small. These models are both very fast (0.551s and 0.633s, respectively) while also placing in the top 7 for BERTScore. This makes them excellent candidates for a practical pipeline where speed is a critical factor, representing a modest trade-off in semantic quality for a nearly 2x gain in inference speed over the top-performing long-t5 model.

C. Human Evaluation Validation

To complement the automated metrics and address the limitations of ROUGE, we conducted a human evaluation to assess the practical utility of the summaries. We selected a random subset of 30 samples from the dataset and recruited three native Indonesian speakers to act as annotators. They scored the summaries generated by the "Lexical Winner" (cahya/t5-base) and the "Semantic Winner" (google/long-t5-tglobal-base) on a 1-5 Likert scale focusing on Fluency, Coherence, and Informativeness.

TABLE IV
ASR MODEL PERFORMANCE (WER/CER)

Model	Average Inference Time (s)
cahya/bert2gpt-indonesian-summarization	**0.475**
cahya/bert2bert-indonesian-summarization	0.546
gregoriomario/IndoT5-summary	0.551
google-t5/t5-small	0.633
google/pegasus-xsum	0.648
facebook/bart-base	0.686
Falconsai/text_summarization	0.766
cahya/t5-base-indonesian-summarization-cased	0.978
google/long-t5-tglobal-base	1.044
facebook/bart-large-cnn	1.119
google-t5/t5-base	1.155
panggi/t5-base-indonesian-summarization-cased	1.217
xTorch8/bart-id-summarization	1.279

Best performing model denoted by bold while the second best denoted by italic

The results reveal a clear dichotomy between linguistic form and semantic content. In terms of fluency, the lexical model (cahya/t5-base) scored slightly higher with an average of 4.62 out of 5, compared to 4.48 for the semantic model (Long-T5), likely due to its formal text training. However, for the more critical metrics of coherence and informativeness, the Long-T5 model demonstrated a significant lead. It achieved a Coherence score of 4.55 (vs. 3.85 for cahya/t5) and an Informativeness score of 4.71 (vs. 3.92). This human preference data consistently favors the Long-T5 summaries for their ability to capture the "gist" of the conversation, confirming our hypothesis that semantic metrics (BERTScore) are better predictors of human preference than lexical metrics (ROUGE) for this task.

IV. DISCUSSIONS

This chapter discusses the key findings presented in Chapter 3. We will now interpret the results of our two foundational benchmarks, synthesizing them to answer our core research questions and build toward a final pipeline recommendation. The discussion is structured as follows:

First, we analyze the profound implications of the ASR benchmark, which revealed a clear and unexpected winner. Second, we explore the critical "semantic vs. lexical" dilemma uncovered in the summarization benchmark, making a case for why semantic quality is the more important metric for this task. Finally, we combine these two findings to propose a state-of-the-art optimal pipeline for Indonesian speech summarization and frankly address this study's limitations.

A. A Clear Choice for ASR

The most significant finding from our ASR benchmark was not just a winner, but an anomaly. The turbo (distil-large-v2) model decisively broke the expected trade-off between model size and performance. While the massive large model (1550M parameters) was not only the slowest (14.59s) but also failed to achieve top accuracy, the turbo model (769M parameters) was an extreme outlier. It achieved:

1. The best-in-class accuracy (7.97% WER).
2. An exceptional inference speed (1.25s), which was ~4.7 times faster than the similarly-sized medium model.

This finding is a powerful conclusion for the first stage of the pipeline: there is no "trade-off" to be made. The turbo model is unequivocally the best choice, providing both the highest accuracy and a low-latency speed suitable for practical deployment. This makes it the clear, non-negotiable first component for any recommended pipeline.

Furthermore, a qualitative breakdown of the error patterns provides deeper linguistic insight beyond raw metrics. The observed gap between WER (7.97%) and CER (4.51%) for the turbo model suggests that errors are predominantly morphological rather than semantic. We identified three primary linguistic error types:

1. *Morphological Normalization:* The model occasionally standardizes informal Indonesian affixes (e.g., transcribing the informal suffix '-in' as the formal '-kan'), which penalizes WER despite preserving meaning.
2. *Loan Word Transliteration:* English technical terms are sometimes phonetically transliterated into Indonesian (e.g., 'device' transcribed as 'divais') or vice versa, creating mismatches with the ground truth.
3. *Disfluency Removal:* Whisper models exhibit an aggressive tendency to filter out conversational fillers (-hmm, anu-), resulting in deletion errors in the transcript but cleaner output for summarization.

This error profile reflects the unique linguistic challenges of Indonesian conversational speech. Specifically, the agglutinative nature of the language means that root words are often modified by complex affixation (prefixes and suffixes). We observed that ASR models frequently struggle with informal affixes (e.g., the suffix -in in bikinin), often hallucinating them into their formal counterparts (-kan in buatkan). Furthermore, the prevalence of code-switching (Indonesian-English mixing) creates phonetic ambiguities. English technical terms are often transcribed as Indonesian homophones (e.g., file transcribed as fail), generating "out-of-vocabulary" tokens that disrupt the semantic coherence required by the subsequent summarization models.

B. The Semantic vs. Lexical Dilemma in Meeting Summarization

Our summarization benchmark (Table 3) revealed a critical divergence in performance, clearly splitting the models into two distinct groups: "Lexical winners" and "Semantic winners." The Indonesian-specific models, such as cahya/t5-base-indonesian-summarization-cased, dominated all ROUGE metrics (e.g., 17.09 ROUGE-1). This indicates they are exceptionally good at lexical overlap,finding and repeating the exact keywords and phrases from the reference summary. This suggests they are highly proficient at extractive-style summarization.

In stark contrast, the multilingual model google/long-t5-tglobal-base "won" on semantic similarity (67.09 BERTScore) while performing poorly on ROUGE scores. This means it is highly abstractive, prioritizing the meaning of the conversation rather than matching specific keywords. For a task like meeting summarization, this distinction is paramount. A user is less concerned with what was said (lexical) and more concerned with what was meant (semantic). For example, a high ROUGE model might struggle if the ASR transcript is noisy, as the exact keywords may be misspelled or lost. A high BERTScore model, however, is more likely to understand the underlying intent and still produce a high-quality summary.

Therefore, we argue that BERTScore is the more important metric for this task. The google/long-t5-tglobal-base model is our "best-case" summarizer, precisely because its low ROUGE and high BERTScore prove it is a powerful abstractive engine. However, its high computational cost (1.044s) makes it a "quality-first" option, in contrast to faster models like gregoriomario/IndoT5-summary (0.551s) which offers a more balanced "sweet spot" of good speed and good (though not the best) semantic performance.

This divergence dictates a clear strategy for practical deployment. We posit that model selection should be task-dependent: for institutional archiving or verbatim transcription where preserving exact terminology is paramount, high-ROUGE models (like Cahya/T5) are preferable despite their lower coherence. However, for automated meeting minutes and executive summaries where the goal is to capture the "gist" and actionable insights efficiently high-BERTScore models (like Long-T5) are the superior choice. This distinction allows practitioners to select the component that best fits their specific operational requirements rather than relying on a single "one-size-fits-all" metric.

C. Proposed Optimal Pipeline and Deployment Recommendations

By synthesizing the findings from our discrete benchmarks, we can now propose a state-of-the-art optimal pipeline for end-to-end Indonesian speech summarization.

This pipeline consists of:

1. Stage 1 (ASR): The Whisper turbo model. As our results showed, it is the undisputed optimal choice, providing the best accuracy (7.97% WER) and the fastest inference speed (1.25s) by a large margin.
2. Stage 2 (Summarization): The google/long-t5-tglobal-base model. As argued in the previous section, its top-ranking semantic score (67.09 BERTScore) makes it the most effective abstractive summarizer for capturing the meaning of a conversation, which we deem the most critical quality for this task.
3. Crucially, this pairing addresses the challenge of error propagation. As noted in Section IV.A, the ASR errors are primarily morphological (e.g., informal affixes) rather than completely semantic failures. A lexical summarizer (high ROUGE) would likely penalize these mismatches heavily. However, an abstractive, semantically-oriented summarizer like Long-T5 (high BERTScore) is theoretically more resilient to such "surface-level" noise, as it focuses on the underlying intent rather than exact word matching. Thus, this pipeline is designed to be robust against the specific types of errors inherent in Indonesian ASR.

This recommended "dream team" combines the most accurate and efficient ASR component with the most semantically-proficient summarization component. However, we also identified a "balanced" recommendation for more resource-constrained environments:

- Balanced Pipeline: Whisper turbo (ASR) + gregoriomario/IndoT5-summary (Summarization).
- Justification: While the IndoT5-summary model's semantic score is slightly lower (66.21 BERTScore), its inference speed is nearly twice as fast (0.551s vs 1.044s). This represents an excellent, high-speed alternative for applications where latency is a primary concern.

D. Limitations and Future Work

This study provides the first foundational benchmark for this task, but it is not without limitations. The primary limitation is that our two core components,ASR and summarization,were benchmarked in isolation. Our summarization results (Section 3.2) were measured on "gold-standard" transcripts, which allowed us to identify the best-performing models in an ideal scenario.

Secondly, the reliance on synthetic data introduces a limitation regarding ecological validity. While the dataset mimics conversational turn-taking, it lacks the chaotic elements of spontaneous real-world Indonesian meetings, such as simultaneous speech (overlaps), inconsistent volume levels, and heavy code-mixing with regional languages. Consequently, the performance metrics reported in this study likely represent an upper-bound "best-case" scenario, and performance may degrade in noisy, real-world deployments.

However, this study did not complete the final step: measuring the performance degradation by feeding the noisy transcripts from the turbo model into the summarization models. The 7.97% WER from the ASR stage will undoubtedly cause a drop in final summary quality. The key unanswered question, which forms the basis for our future work, is: how much?

We hypothesize that the abstractive google/long-t5 model will be more resilient to ASR errors than the extractive, ROUGE-focused models, but this must be empirically verified. Therefore, the critical next steps for this research are:

(1) Implement the proposed pipeline (Whisper-turbo + google/long-t5) and run the end-to-end experiment to quantify the drop in ROUGE and BERTScore caused by ASR error. (2) Explore fine-tuning the summarization models on ASR-generated transcripts (domain adaptation) to make them more robust to noisy, unpunctuated text. (3) Conduct a human evaluation of the final summaries to confirm whether the semantically-rich summaries from the google/long-t5 model are, in fact, preferred by users over the lexically-precise summaries from models like cahya/t5-base.

Thirdly, our benchmarking protocol is restricted to a zero-shot setting. We deliberately chose this approach to establish a fundamental baseline of how "off-the-shelf" models perform on this new dataset without the computational and data overhead of training. Consequently, this study does not capture the potential performance gains achievable through fine-tuning or instruction-tuning, which represent the current state-of-the-art. Future iterations of this benchmark should investigate how much performance lift can be gained by fine-tuning the best-performing models (e.g., Long-T5) on the training split of our dataset.

Finally, regarding generalizability, it is important to emphasize that this study evaluates the pipeline within a controlled, synthetic environment. While this approach is necessary to establish a reproducible baseline, it does not fully guarantee performance in "in-the-wild" scenarios. Real-world meetings often contain environmental factors such as background noise, reverberation, and non-collaborative overlaps that are absent in our dataset. Therefore, the practical generalizability of the proposed pipeline remains to be empirically tested. The logical next step for this research line is to deploy the recommended pipeline (Whisper Turbo + Long-T5) on a corpus of recorded real-world Indonesian meetings to quantify the "reality gap" between our benchmark results and actual operational performance.

Lastly, the transition from synthetic to real-world application necessitates a critical discussion on ethics and privacy. While our synthetic dataset circumvents privacy concerns, processing real-world meeting recordings involves handling sensitive biometric data (voice) and potentially confidential information. Future research must prioritize the development of privacy-preserving protocols, such as speaker de-identification and local-only processing, to ensure that the convenience of automated summarization does not come at the cost of user privacy or data security.

V. Conclusion

This research addressed a significant gap in Indonesian Natural Language Processing: the absence of a foundational benchmark for an end-to-end speech summarization pipeline. The primary goal of this study was to systematically evaluate the two core, discrete components,Automatic Speech Recognition (ASR) and Text Summarization,to identify the "best-in-class" models and propose an optimal, state-of-the-art pipeline.

Our contributions are twofold and provide clear, actionable recommendations. (1) For the ASR component, our benchmark of six Whisper variants revealed a definitive and non-obvious winner. The turbo (distil-large-v2) model was not only the most accurate (7.97% WER) but also one of the fastest (1.25s), decisively breaking the expected trade-off between model size and performance.

(2) For the summarization component, our zero-shot benchmark of 13 models on gold-standard transcripts uncovered a critical divergence between semantic (BERTScore) and lexical (ROUGE) performance. We demonstrated that Indonesian-specific models (e.g., cahya/t5-base...) excel at lexical matching, while large multilingual models like google/long-t5-tglobal-base are superior at capturing abstractive meaning.

Based on these findings, we conclude that BERTScore is the more critical metric for the task of meeting summarization, as it prioritizes semantic intent over simple keyword matching.

Therefore, this paper recommends an optimal pipeline composed of the Whisper turbo model for transcription and the google/long-t5-tglobal-base model for semantic summarization. This combination represents the most powerful and promising, state-of-the-art configuration for future development in this domain.

The critical next step for this research is to implement this recommended pipeline and measure the end-to-end performance degradation. This future work will empirically quantify the impact of ASR errors on final summary quality and test the hypothesis that abstractive, semantically-focused models are more resilient to real-world transcription noise.

AUTHOR CONTRIBUTION

Conceptualization: MCTM; Data curation: MCTM, AEKP; Funding acquisition: WY; Methodology: MCTM; Project administration: WY; Validation: FAK, AEKP; Visualization: FAK, AEKP; Software: FAK, AEKP; Writing: MCTM; Review: WY, FAK, AEKP.

ACKNOWLEDGEMENT

This project is fully funded by BIMA Kemdiktisaintek under grants number: 018/C3/DT.05.00/PL/2025 and 1483ca/IT9.2.1/PT.01.03/2025.

REFERENCES

[1] S. Adnams, "The distributed workplace of the future is now," *Gartner, Report G00726412*, 2020.

[2] J. A. Allen and S. G. Rogelberg, "Manager-led group meetings: A context for promoting employee engagement," *Group Organ. Manag.*, vol. 38, no. 5, pp. 543–569, Oct. 2013.

[3] A. F. Hidayatullah, R. A. Apong, D. T. C. Lai, and A. Qazi, "Word level language identification in Indonesian-Javanese-English code-mixed text," *Procedia Comput. Sci.*, vol. 244, pp. 105–112, 2024.

[4] A. F. Hidayatullah, R. Apong, D. Lai, and A. Qazi, "Corpus creation and language identification for code-mixed Indonesian-Javanese-English Tweets," *PeerJ Comput. Sci.*, vol. 9, June 2023.

[5] G. Winata, A. F. Aji, Z. X. Yong, and T. Solorio, "The decades progress on code-switching research in NLP: A systematic survey on trends and challenges," in *Findings of the Association for Computational Linguistics: ACL 2023*, Toronto, Canada, 2023, pp. 2936–2978.

[6] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," *arXiv [eess.AS]*, 06-Dec-2022.

[7] B. Wilie *et al.*, "IndoNLU: Benchmark and Resources for Evaluating Indonesian Natural Language Understanding," in *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, 2020.

[8] F. Koto, A. Rahimi, J. H. Lau, and T. Baldwin, "IndoLEM and IndoBERT: A Benchmark Dataset and Pre-trained Language Model for Indonesian NLP," *arXiv [cs.CL]*, 01-Nov-2020.

[9] I. McCowan *et al.*, "The AMI meeting corpus," pp. 137–140, Aug. 2005.

[10] A. Purwarianti *et al.*, "NusaDialogue: Dialogue summarization and generation for underrepresented and extremely low-resource languages," pp. 82–100, 2025.

[11] R. F. Khoiroh, E. Julianto, S. A. Ardiyansa, H. A. Fajri, A. A. R. Yasa, and B. Sangapta, "Implementasi Speech Recognition Whisper pada Debat Calon Wakil Presiden Republik Indonesia," *Ex*, vol. 14, no. 2, pp. 67–74, July 2024.

[12] A. Aulia, L. Dessi, P. Ayu, T. Dipta, A. Kurniawati, and S. Sakriani, "Enhancing Indonesian automatic speech recognition: Evaluating multilingual models with diverse speech variabilities," *arXiv [cs.CL]*, 11-Oct-2024.

[13] Z. Aljneibi, S. Almenhali, and L. Lanca, "Convolutional neural network application for automated lung cancer detection on chest CT using Google AI Studio," *Radiography (Lond.)*, no. 103152, p. 103152, Sept. 2025.

[14] G. Comanici *et al.*, "Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities," *arXiv [cs.CL]*, 16-Oct-2025.

[15] C. Raffel *et al.*, "Exploring the limits of transfer learning with a unified text-to-text transformer," *arXiv [cs.LG]*, 23-Oct-2019.

[16] G. E. Abdul, I. A. Ali, and C. Megha, "Fine-tuned T5 for abstractive summarization," *Int. J. Perform. Eng.*, vol. 17, no. 10, p. 900, 2021.

[17] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," *Annu Meet Assoc Comput Linguistics*, pp. 74–81, July 2004.

[18] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "BERTScore: Evaluating Text Generation with BERT," *arXiv [cs.CL]*, 21-Apr-2019.

[19] D. Klakow and J. Peters, "Testing the correlation of word error rate and perplexity," *Speech Commun.*, vol. 38, no. 1–2, pp. 19–28, Sept. 2002.