

**IMPLEMENTASI MODEL BART BERBASIS TRANSFER
LEARNING UNTUK RINGKASAN TRANSKRIP VIDEO
YOUTUBE DAN EVALUASI KUALITAS
DENGAN METRIK ROUGE**

TESIS

**Oleh:
DIYAN NOVA SETIAWAN
NIM. 230605210022**



**PROGRAM STUDI MAGISTER INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2026**

**IMPLEMENTASI MODEL BART BERBASIS TRANSFER LEARNING
UNTUK RINGKASAN TRANSKRIP VIDEO YOUTUBE DAN
EVALUASI KUALITAS DENGAN METRIK ROUGE**

TESIS

**Diajukan kepada:
Universitas Islam Negeri Maulana Malik Ibrahim Malang untuk Memenuhi
Salah Satu Persyaratan dalam Memperoleh Gelar
Magister Komputer (M. Kom)**

**Oleh:
DIYAN NOVA SETIAWAN
NIM. 230605210022**

**PROGRAM STUDI MAGISTER INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2026**

**IMPLEMENTASI MODEL BART BERBASIS TRANSFER LEARNING
UNTUK RINGKASAN TRANSKRIP VIDEO YOUTUBE DAN
EVALUASI KUALITAS DENGAN METRIK ROUGE
HALAMAN PERSETUJUAN**

TESIS

**Oleh:
DIYAN NOVA SETIAWAN
NIM. 230605210022**

**Telah diperiksa dan disetujui untuk diuji:
Tanggal: 27 Februari 2026**

Pembimbing I

**Prof. Dr. Ir. Muhammad Faisal, M.T
NIP.19740510 200501 1 007**

Pembimbing II

**Dr. M. Imamudin Lc, MA
NIP.19740602 200901 1 010**

**Mengetahui,
Ketua Program Studi Magister Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang**



**Prof. Dr. Ir. Muhammad Faisal, M.T
NIP.19740510 200501 1 007**

**IMPLEMENTASI MODEL BART BERBASIS TRANSFER LEARNING
UNTUK RINGKASAN TRANSKRIP VIDEO YOUTUBE DAN
EVALUASI KUALITAS DENGAN METRIK ROUGE**

TESIS

**Oleh:
DIYAN NOVA SETIAWAN
NIM. 230605210022**

**Telah dipertahankan di Depan Dewan Penguji Tesis
Dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Magister Komputer (M.Kom)
Tanggal: 27 Februari 2026**

Susunan Dewan Penguji

Tanda Tangan

**Penguji I : Dr. Ir. Fresy Nugroho, ST., MT, IPM
NIP. 19710722 201101 1 001**

**Penguji II : Dr. Usman Pagalay, M.Si
NIP. 19650414 200312 1 001**

**Pembimbing I : Prof. Dr. Ir. Muhammad Faisal, M.T
NIP.19740510 200501 1 007**

**Pembimbing II : Dr. M. Imamudin Lc, MA
NIP.19740602 200901 1 010**

()

()

()

**Mengetahui,
Ketua Program Studi Magister Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang**



**Prof. Dr. Ir. Muhammad Faisal, M.T
NIP.19740510 200501 1 007**

PERNYATAAN KEASLIAN TULISAN

Yang bertanda tangan di bawah ini saya:

Nama : Diyan Nova Setiawan

NIM : 230605210022

Program Studi: Magister Informatika

Fakultas :Sains dan Teknologi

Judul Tesis: Implementasi Model BART Berbasis Transfer Learning untuk Ringkasan Transkrip Video YouTube dan Evaluasi Kualitas dengan Metrik ROUGE

Menyatakan dengan sebenarnya bahwa Tesis yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambilalihan data, tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka.

Apabila di kemudian hari terbukti atau dapat dibuktikan Tesis ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 27 Februari 2026



Diyan Nova Setiawan
NIM.230605210022

MOTTO

“Berawal dari mimpi, tumbuh dalam keyakinan, dan terwujud dalam kenyataan”

KATA PENGANTAR

Assalāmu 'alaikum Warahmatullāhi Wabarakātuh.

Syukur Alhamdulillah penulis haturkan ke hadirat Allah SWT yang telah melimpahkan rahmat, taufik, dan hidayah-Nya, sehingga penulis dapat menyelesaikan studi pada Program Studi Magister Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang sekaligus menyelesaikan Tesis ini dengan baik dan tepat waktu. Shalawat serta salam semoga senantiasa tercurah kepada junjungan kita Nabi Muhammad SAW yang telah membawa umat manusia dari zaman kegelapan menuju zaman yang penuh dengan ilmu pengetahuan.

Penyusunan Tesis ini tidak terlepas dari bantuan, dukungan, bimbingan, serta doa dari berbagai pihak. Oleh karena itu, dengan segala kerendahan hati penulis menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Bapak Prof. Dr. Ir. Muhammad Faisal, M.T , selaku Ketua Program Studi Magister Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Bapak Prof. Dr. Ir. Muhammad Faisal, M.T dan Bapak Dr. M. Imamudin Lc., MA selaku dosen pembimbing Tesis, yang telah banyak memberikan pengarahan dan pengalaman yang berharga.
3. Bapak Dr. Ir. Fresy Nugroho, ST., MT, IPM, ASEAN eng. selaku Ketua Penguji yang telah memberikan saran, kritik, dan masukan yang konstruktif demi penyempurnaan Tesis ini.
4. Bapak Dr. Usman Pagalay, M.Si, selaku Anggota Penguji yang telah memberikan arahan dan evaluasi yang sangat berarti dalam proses penyempurnaan Tesis ini.
5. Segenap civitas akademika Program Studi Magister Informatika, terutama seluruh Bapak/ Ibu dosen, terima kasih atas segenap ilmu dan bimbingan.
6. Ayah dan Ibu saya yang senantiasa memberikan do'a dan semangat kepada saya untuk meraih kesuksesan.
7. Pasangan tercinta, Tuti Wuryaningsih, S.Pd., yang senantiasa memberikan doa, dukungan, pengertian, motivasi, serta semangat luar biasa selama proses studi dan penyusunan Tesis ini.
8. Keluarga besar penulis yang selalu memberikan dukungan moral dan motivasi dalam menyelesaikan studi ini.
9. Semua rekan-rekan magister informatika angkatan 2023 yang membantu dan menemani bersama-sama di studi magister ini.
10. Semua pihak yang ikut membantu dalam menyelesaikan Tesis ini baik berupa materil maupun moril.

Penulis menyadari bahwa dalam penyusunan Tesis ini masih terdapat kekurangan dan berharap semoga bisa memberikan manfaat kepada para pembaca khususnya bagi penulis secara pribadi. Amin Ya Rabbal Alamin.

Wassalamu 'alaikum Warahmatullahi Wabarakatuh.

Malang, 27 Februari 2026

A handwritten signature in black ink, consisting of a large, stylized initial 'D' followed by a series of loops and a long horizontal stroke that ends in a small hook.

Diyan Nova Setiawan
NIM.230605210022

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGAJUAN	ii
HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN	iv
HALAMAN PERNYATAAN	v
MOTTO	vi
KATA PENGANTAR	vii
DAFTAR ISI	ix
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiii
ABSTRAK	xiv
ABSTRACT	xv
المخلص	xvi
BAB I PENDAHULUAN	1
1.1. Latar Belakang Masalah	1
1.2. Pernyataan Masalah	10
1.3. Tujuan Penelitian	11
1.4. Manfaat Penelitian	11
1.5. Ruang Lingkup Penelitian	12
1.6. Sistematika Penulisan	13
BAB II STUDI PUSTAKA	15
2.1. YouTube sebagai Sumber Data	15
2.2. <i>Text Summarization</i>	16
2.3. <i>Transfer Learning</i> dalam <i>Natural Language Processing</i>	18
2.4. <i>BART</i> dalam <i>Natural Language Processing</i>	19
2.5. Evaluasi Ringkasan dengan Metrik ROUGE	22
2.6. Implementasi Flask dalam Aplikasi <i>Summarization</i>	24
2.7. Kerangka Teori.....	26
BAB III METODOLOGI PENELITIAN	30
3.1. Desain Penelitian	30

3.2. Pengumpulan Data.....	30
3.3. Desain Sistem	33
3.4. Implementasi Pengambilan Transkrip Video YouTube	38
3.5. Skenario Eksperimen	41
3.6. Parameter Uji Coba.....	51
BAB IV IMPLEMENTASI MODEL BART BERBASIS TRANSFER LEARNING	53
4.1. Arsitektur dan Mekanisme Model	53
4.1.1. Positional Encoding	55
4.1.2. Mekanisme Self-Attention.....	55
4.1.3. Multi-Head Attention.....	56
4.1.4. Feed Forward Network	57
4.1.5. Transfer Learning pada Model BART	58
4.2. Evaluasi Metrik Rouge	61
4.2.1. ROUGE-1 (Kemiripan Unigram).....	62
4.2.2. ROUGE -2 (Bigram Overlap)	63
4.2.3. ROUGE-L (<i>Longest Common Subsequence</i> / LCS).....	64
4.2.4. Average	65
BAB V HASIL DAN PEMBAHASAN	67
5.1. Hasil Pengumpulan dan Pemahaman Data	67
5.1.1. Variabel yang digunakan	67
5.1.2. Distribusi dan Karakteristik Data	70
5.2. Pemodelan <i>BART</i> pada Metode <i>Transfer Learning</i>	75
5.3. Evaluasi Kinerja Model Menggunakan ROUGE.....	80
5.3.1. Hasil Preprocessing & Evaluasi ROUGE	81
5.3.2. Analisis Visualisasi Grafik <i>Precision</i> , <i>Recall</i> , dan <i>F1-Score</i>	86
5.4. Korelasi Hasil Penelitian dalam Perspektif Islam	90
BAB VI KESIMPULAN	96
6.1. Kesimpulan	96
6.2. Saran	97
DAFTAR PUSTAKA.....	99
GLOSARIUM.....	103

LAMPIRAN LAMPIRAN.....107

DAFTAR GAMBAR

Gambar 2. 1 Kerangka Teori	25
Gambar 3. 1 Desain Penelitian.....	29
Gambar 3.3 Desain Sistem	33
Gambar 3. 4 Skenario Eksperimen	43
Gambar 3.5 Simulasi Tampilan awal web flask	43
Gambar 3. 6 Teks Setelah Preprocessing	44
Gambar 3. 7 Fine Tuning model <i>BART</i>	45
Gambar 3. 8 Simulasi Output di Web Flask.....	47
Gambar 5. 1 . Grafik Precision.....	88
Gambar 5. 2. Grafik Recall	89
Gambar 5. 3. Grafik F1-Score.....	90

DAFTAR TABEL

Tabel 2. 1 Daftar Jurnal	26
Tabel 3. 1 Daftar Video	30
Tabel 3. 2 Daftar Video dan Karakteristiknya.....	47
Tabel 3. 3 Simulasi Nilai Rouge	53
Tabel 4. 1 Nama Variabel	57
Tabel 4. 2 Distribusi Video Berdasarkan Kategori Durasi	59
Tabel 4. 3 Performa Statistik Transkrip YouTube	62
Tabel 4. 4 Performa ROUGE di Berbagai Kategori Durasi Transkrip.....	72
Tabel 5. 1. Nama Variabel	69
Tabel 5. 2. Distribusi Video Berdasarkan Kategori Durasi	70
Tabel 5. 3. Performa Statistik Transkrip YouTube	73
Tabel 5. 4 Performa ROUGE di Berbagai Kategori Durasi Transkrip.....	82

ABSTRAK

Setiawan, Diyan Nova 2026. Implementasi Model *BART* Berbasis *Transfer Learning* untuk Ringkasan Transkrip Video YouTube dan Evaluasi Kualitas dengan Metrik ROUGE. Tesis. Program Studi Magister Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Prof. Dr. Ir. Muhammad Faisal, M.T (II) Dr. M. Imamudin Lc, MA

Transkrip video YouTube yang dihasilkan secara otomatis umumnya memiliki struktur panjang dan kurang terorganisir, sehingga menyulitkan pengguna dalam memahami inti informasi secara cepat dan efisien. Penelitian ini bertujuan untuk mengimplementasikan model *BART* berbasis *Transfer Learning* dalam melakukan peringkasan otomatis terhadap transkrip video YouTube berbahasa Indonesia serta mengevaluasi kualitas ringkasan yang dihasilkan menggunakan metrik ROUGE. Metode penelitian meliputi pengambilan data transkrip melalui YouTube API, proses preprocessing teks untuk membersihkan dan menormalisasi data, fine-tuning model *BART* menggunakan pendekatan *Transfer Learning*, serta evaluasi menggunakan ROUGE-1, ROUGE-2, dan ROUGE-L berdasarkan nilai *Precision*, *Recall*, dan *F1-Score*. Data penelitian berasal dari 730 video pada channel edukatif “Kok Bisa?” yang dikategorikan berdasarkan durasi menjadi pendek, sedang, dan panjang untuk menganalisis pengaruh panjang transkrip terhadap performa model. Hasil penelitian menunjukkan bahwa model *BART* berbasis *Transfer Learning* mampu menghasilkan ringkasan yang informatif dan relevan dengan performa evaluasi yang konsisten pada berbagai kategori durasi. Dengan demikian, sistem yang dikembangkan dapat meningkatkan aksesibilitas informasi serta mendukung pemahaman konten video secara lebih cepat dan praktis melalui aplikasi web berbasis Flask.

Kata Kunci: *BART*, *Transfer Learning*, *Text Summarization*, YouTube Transcript, ROUGE.

ABSTRACT

Setiawan, Diyan Nova 2026. **Implementation of *Transfer Learning*-Based *BART* Model for YouTube Video Transcript Summary and Quality Evaluation with ROUGE Metrics**. Thesis. Master of Informatics Study Program, Faculty of Science and Technology, Maulana Malik Ibrahim State Islamic University, Malang. Supervisor: (I) Prof. Dr. Ir. Muhammad Faisal, M.T (II) Dr.M. Imamudin Lc, MA

Auto-generated YouTube video transcripts generally have a long structure and are less organized, making it difficult for users to understand the core information quickly and efficiently. This study aims to implement a *Transfer Learning*-based *BART* model in automatically summarizing Indonesian YouTube video transcripts and evaluating the quality of summaries produced using the ROUGE metric. The research methods included retrieval of transcript data through the YouTube API, text preprocessing process to clean and normalize the data, fine-tuning the *BART* model using the *Transfer Learning* approach, and evaluation using ROUGE-1, ROUGE-2, and ROUGE-L based on *Precision*, *Recall*, and *F1-Score* values. The research data came from 730 videos on the educational channel "Kok Bisa?" which were categorized by duration into short, medium, and long to analyze the effect of transcript length on model performance. The results showed that the *Transfer Learning*-based *BART* model was able to produce informative and relevant summaries with consistent evaluation performance across various duration categories. Thus, the developed system can improve information accessibility and support faster and practical understanding of video content through a Flask-based web application.

Keyword: *BART*, *Transfer Learning*, *Text Summarization*, YouTube Transcript, ROUGE.

المخلص

ستيوان، ديان نونفا. ٢٠٢٦. تنفيذ نموذج بارت القائم على نقل التعلّم لتلخيص نصوص مقاطع فيديو يوتيوب وتقييم الجودة باستخدام مقاييس روج. رسالة ماجستير. برنامج دراسة ماجستير المعلوماتية، كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانغ. المشرف: (الأول) الأستاذ الدكتور المهندس محمد فيصل، ماجستير تقني. (الثاني) الدكتور محمد إمام الدين، ليسانس، ماجستير.

تتميز النصوص التفرغية التلقائية لمقاطع فيديو يوتيوب بطولها وضعف تنظيمها، مما يجعل من الصعب على المستخدمين فهم المعلومات الأساسية بسرعة وكفاءة. تهدف هذه الدراسة إلى تطبيق نموذج بارت القائم على أسلوب نقل التعلّم في تلخيص نصوص مقاطع فيديو يوتيوب باللغة الإندونيسية تلخيصًا آليًا، مع تقييم جودة الملخصات الناتجة باستخدام مقياس روج.

تضمنت منهجية البحث استرجاع بيانات النصوص التفرغية عبر واجهة برمجة تطبيقات يوتيوب، وإجراء المعالجة المسبقة للنصوص بهدف تنظيف البيانات وتوحيد صيغتها، ثم تنفيذ عملية الضبط الدقيق لنموذج بارت باستخدام منهج نقل التعلّم، إضافة إلى إجراء التقييم باستخدام مقاييس روج-١ وروج-٢ وروج-٣ اعتمادًا على قيم الدقة والاسترجاع ومتوسط الدرجة التوافقية (إف-١).

استندت بيانات البحث إلى ٧٣٠ مقطع فيديو من قناة تعليمية، صُنِّفت وفق مدة العرض إلى قصيرة ومتوسطة وطويلة لتحليل تأثير طول النص التفرغي في أداء النموذج. وأظهرت النتائج أن نموذج بارت القائم على نقل التعلّم قادر على إنتاج ملخصات مفيدة ومرتبطة بالسياق، مع أداء تقييم متسق عبر مختلف فئات المدة.

وبذلك يسهم النظام المطوّر في تحسين إتاحة المعلومات ودعم الفهم السريع والعملية لمحتوى الفيديو من خلال تطبيق ويب قائم على إطار عمل فلاسك.

الكلمات المفتاحية: بارت، نقل التعلّم، تلخيص النصوص، النص التفرغي لليوتيوب، روج.

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Seiring dengan pesatnya perkembangan teknologi, *platform* berbagi video seperti YouTube telah menjadi sumber utama konsumsi informasi, hiburan, dan edukasi. YouTube merupakan salah satu aplikasi media paling populer untuk memenuhi kebutuhan informasi di era modern saat ini. YouTube adalah sebuah situs web berbagi video dimana setiap orang bebas untuk berkreasi melalui video untuk menuangkan aspirasi, pendapat, pandangan/opini, nilai-nilai tentang apapun yang menjadi pikiran mereka atau aktivitas yang sedang mereka kerjakan (Fauzan 2019). YouTube sebagai media komunikasi massa menyediakan beragam informasi dan berita yang dibutuhkan oleh setiap orang sehingga sekarang ini semua orang dipermudah dalam pencarian berita karena tidak perlu lagi jauh-jauh keluar rumah untuk membeli koran (Vira and Reynata 2022). Bukan hanya itu, YouTube juga sering dijadikan sebagai hiburan karena ada banyak jenis konten video yang menghibur (Nur Setiyana and Badu Kusuma 2021). Dengan jumlah konten yang sangat besar dan terus bertambah, pengguna sering kesulitan untuk menyerap seluruh isi video dalam waktu singkat.

Salah satu solusi yang ditawarkan YouTube adalah fitur transkrip otomatis yang mengubah audio menjadi teks. Transkrip ini dapat dimanfaatkan untuk memahami isi video tanpa harus menontonnya secara penuh (Chauhan et al. 2024). Namun, transkrip video sering kali panjang, tidak terstruktur, dan mengandung bagian-bagian yang tidak relevan, seperti pengulangan kata, jeda, atau komentar spontan dari pembicara. Oleh karena itu, diperlukan proses

lanjutan untuk mengolah transkrip menjadi bentuk yang lebih ringkas dan mudah dipahami (Bokhove and Downey 2018).

Ringkasan otomatis terhadap transkrip video menjadi penting untuk meningkatkan aksesibilitas, mempercepat pemahaman isi video, dan mempermudah proses. Berdasarkan hasil penelitian terdahulu, peningkatan aksesibilitas melalui ringkasan otomatis transkrip video terbukti signifikan secara kuantitatif. Penelitian oleh (Albeer et al. 2022) menunjukkan bahwa penggunaan sistem peringkasan otomatis meningkatkan aksesibilitas informasi sebesar 63%, yang diperoleh melalui uji coba terhadap kelompok pengguna yang diminta membaca transkrip mentah dan hasil ringkasan, lalu dievaluasi berdasarkan kecepatan dan pemahaman mereka terhadap isi konten. Penilaian dilakukan menggunakan survei kuesioner daring berbasis Google Forms dengan skala Likert 1–5, yang kemudian dikonversi ke dalam bentuk persentase. Sementara itu, studi oleh (Chauhan et al. 2024) mengungkapkan bahwa proses pemahaman informasi berlangsung 57% lebih cepat dengan ringkasan otomatis; nilai ini diperoleh dengan membandingkan waktu rata-rata yang dibutuhkan pengguna untuk memahami isi video secara penuh dibanding hanya membaca ringkasan, lalu dikalkulasi penghematannya dalam bentuk persentase. Bagi pengguna yang memiliki keterbatasan pendengaran, penelitian dari (Ilampiray et al. 2023) menunjukkan bahwa mereka mengalami peningkatan kemudahan akses informasi hingga 68%, berdasarkan survei skala 1–5 yang menilai seberapa mudah dan cepat mereka memahami isi video melalui ringkasan otomatis dibandingkan dengan membaca transkrip lengkap. Nilai 68% ini merupakan yang tertinggi karena kelompok ini sangat terbantu dengan

ringkasan sebagai pengganti konten audio. Hasil-hasil ini membuktikan bahwa sistem ringkasan otomatis mampu meningkatkan aksesibilitas, mempercepat pemahaman isi video, dan mempermudah proses pencarian informasi secara signifikan. Dengan pendekatan ini, teknologi peringkasan tidak hanya mendukung efisiensi pengguna umum, tetapi juga menciptakan lingkungan digital yang ramah dan inklusif bagi semua kalangan.

Text summarization dalam bidang *Natural Language Processing* (NLP) adalah proses menghasilkan ringkasan dari teks panjang secara otomatis. Model *BART* merupakan salah satu metode deep learning yang dirancang khusus untuk tugas ini, dengan pendekatan extractive (Lewis et al. 2020). Dengan teknik *Transfer Learning*, model ini dapat langsung digunakan pada data spesifik seperti transkrip video tanpa pelatihan ulang besar. Evaluasi hasil ringkasan dilakukan menggunakan metrik ROUGE, yang mengukur kesamaan antara ringkasan mesin dan referensi manusia (Barbella and Tortora 2022).

Dengan memanfaatkan pendekatan *Transfer Learning*, Model *BART* dapat diterapkan pada domain khusus seperti transkrip video YouTube tanpa memerlukan pelatihan ulang dari awal. Hal ini memungkinkan adaptasi model pada data spesifik dengan efisiensi tinggi (Hartawan, Maylawati, and Uriawan 2024). Oleh karena itu, penelitian ini bertujuan untuk mengimplementasikan Model *BART* dalam melakukan ringkasan otomatis terhadap transkrip video YouTube dan mengevaluasi performanya menggunakan metrik ROUGE yang umum digunakan dalam penilaian kualitas ringkasan otomatis (Subagyo and Sasongko 2024).

Cara membedakan kualitas ringkasan berdasarkan durasi video dilakukan dengan mengelompokkan video ke dalam tiga kategori sebagaimana diklasifikasikan oleh (Gupta, Dubey, and Sharma 2025): pendek (1–3 menit), sedang (4–7 menit), dan panjang (>7 menit). Setelah proses *summarization* dan evaluasi menggunakan metrik ROUGE (ROUGE-1, ROUGE-2, dan ROUGE-L) dilakukan terhadap setiap video, nilai-nilai evaluasi tersebut dihitung secara individual, lalu dirata-ratakan dalam tiap kelompok durasi. Pendekatan ini memungkinkan perbandingan performa metode secara kontekstual, dengan mempertimbangkan kompleksitas transkrip yang umumnya meningkat seiring dengan panjang durasi video.

Ilmu pengetahuan dan teknologi merupakan dua pilar utama dalam mendorong kemajuan peradaban manusia, karena melalui keduanya manusia mampu mengembangkan sistem yang meningkatkan efisiensi dan kualitas hidup (Karhade et al. 2022). Perkembangan kecerdasan buatan (*artificial intelligence*) menjadi salah satu bentuk konkret kemajuan tersebut, termasuk dalam pengembangan sistem cerdas berbasis model BART yang digunakan dalam penelitian ini (Lewis et al. 2020). Pemanfaatan pendekatan *Transfer Learning* untuk mengolah dan meringkas informasi dari transkrip video mencerminkan upaya manusia dalam mengelola pengetahuan secara efisien, karena pendekatan ini memungkinkan model yang telah dilatih pada data besar untuk diadaptasi pada tugas tertentu dengan sumber daya yang lebih hemat (Zhang et al. 2022).

Al-Qur'an juga menekankan pentingnya kemampuan berpikir, mendengar, dan memahami sebagai bekal manusia dalam memanfaatkan ilmu, sebagaimana tercantum dalam Surah An-Nahl [16]: 78:

وَاللَّهُ أَخْرَجَكُمْ مِنْ بُطُونِ أُمَّهَاتِكُمْ لَا تَعْلَمُونَ شَيْئًا وَجَعَلَ لَكُمُ السَّمْعَ وَالْأَبْصَارَ وَالْأَفْئِدَةَ لَعَلَّكُمْ تَشْكُرُونَ ﴿٧٨﴾

Artinya:

“Dan Allah mengeluarkan kamu dari perut ibumu dalam keadaan tidak mengetahui sesuatu pun. Dia memberimu pendengaran, penglihatan, dan hati, agar kamu bersyukur (78)”

Dalam Tafsir Ibnu Katsir dijelaskan bahwa manusia ketika dilahirkan tidak memiliki pengetahuan sama sekali, kemudian Allah menganugerahkan pendengaran, penglihatan, dan hati (akal) sebagai sarana untuk memperoleh ilmu dan memahami kebenaran. Ibnu Katsir menegaskan bahwa ketiga instrumen ini adalah perangkat utama dalam proses pembelajaran manusia: pendengaran untuk menerima informasi, penglihatan untuk mengamati fenomena, dan hati (akal) untuk memahami serta menyimpulkan makna dari informasi tersebut. Penutup ayat, *“la'allakum tasykurūn”*, menunjukkan bahwa seluruh potensi intelektual itu diberikan agar manusia bersyukur dengan cara memanfaatkannya secara (Katsir n.d.).

Berdasarkan penjelasan tersebut, ayat ini secara langsung berkaitan dengan konsep *ḥablum minallāh*, yaitu hubungan manusia dengan Allah. Kemampuan berpikir, menganalisis, dan mengembangkan teknologi bukanlah hasil kemandirian manusia semata, tetapi karunia dari Allah yang harus disyukuri. Dalam konteks penelitian ini, pengembangan model BART berbasis *Transfer Learning* untuk merangkum transkrip video merupakan bentuk aktualisasi dari

nikmat akal yang diberikan Allah. Aktivitas ilmiah seperti merancang algoritma, melakukan pelatihan model, dan mengevaluasi hasil menggunakan metrik ROUGE(Lin 2004) merupakan bentuk syukur praktis atas karunia intelektual tersebut. Selama dilakukan dengan etika dan tanggung jawab, penelitian ini menjadi bagian dari pengamalan *ḥablum minallāh*.

Di sisi lain, ayat ini juga mengandung dimensi *ḥablum minannās*. Potensi intelektual yang diberikan Allah tidak hanya untuk kepentingan pribadi, tetapi juga untuk memberi manfaat kepada sesama manusia. Sistem peringkasan otomatis yang dikembangkan dalam penelitian ini membantu masyarakat memahami informasi secara lebih cepat dan efisien, terutama bagi pengguna yang memiliki keterbatasan waktu atau kebutuhan khusus (Ilampiray et al. 2023). Dengan demikian, pemanfaatan akal dalam pengembangan teknologi menjadi sarana membangun hubungan sosial yang bermanfaat dan meningkatkan kualitas akses informasi.

Selanjutnya, Allah SWT berfirman dalam QS. An-Nahl [16]: 79:

أَلَمْ يَرَوْا إِلَى الطَّيْرِ مُسَخَّرَاتٍ فِي جَوِّ السَّمَاءِ مَا يُمَسِّكُهُنَّ إِلَّا اللَّهُ إِنَّ فِي ذَلِكَ لَآيَاتٍ لِّقَوْمٍ

يُؤْمِنُونَ ﴿٧٩﴾

Artinya:

“Dan Allah menjadikan bagimu rumah sebagai tempat tinggal dan menjadikan bagimu (tempat berteduh) dari kulit binatang ternak yang kamu merasa ringan membawanya di waktu kamu bepergian dan menetap. Dan dari bulu domba, bulu unta, serta rambut kambing, (dijadikan) perabotan rumah tangga dan barang-barang untuk waktu tertentu (79).”

Dalam Tafsir Ibnu Katsir dijelaskan bahwa ayat ini mengajak manusia untuk merenungi kekuasaan Allah dalam menciptakan sistem alam yang teratur.

Burung dapat terbang di udara bukan karena kekuatan dirinya semata, melainkan karena Allah yang menahannya dan menetapkan hukum-hukum alam yang memungkinkan hal tersebut terjadi. Ibnu Katsir menekankan bahwa fenomena ini adalah bukti kekuasaan dan kebijaksanaan Allah dalam mengatur alam semesta (Katsir n.d.). Keteraturan ini menunjukkan bahwa alam berjalan berdasarkan sistem yang telah ditetapkan oleh Allah. Hal ini sejalan dengan prinsip kerja algoritma dalam kecerdasan buatan, termasuk arsitektur BART yang bekerja berdasarkan logika matematis dan probabilitas statistik (Lewis et al. 2020).

Ayat ini berkaitan dengan *ḥablum minallāh* karena menumbuhkan kesadaran bahwa seluruh keteraturan di alam, termasuk hukum fisika, matematika, dan logika yang digunakan dalam sistem kecerdasan buatan, adalah bagian dari ketetapan Allah (*sunnatullah*). Manusia tidak menciptakan hukum-hukum tersebut dari ketiadaan, melainkan mempelajarinya dan memanfaatkannya. Oleh karena itu, pengembangan teknologi harus dilandasi sikap rendah hati dan tanggung jawab moral sebagai bentuk pengakuan terhadap kebesaran Allah.

Adapun dimensi *ḥablum minannās* dalam ayat ini terletak pada implikasi pemanfaatan keteraturan tersebut untuk kemaslahatan manusia. Jika Allah menciptakan sistem alam dengan tujuan mendukung kehidupan makhluk-Nya, maka manusia sebagai khalifah di bumi juga berkewajiban membangun sistem teknologi yang terstruktur dan bermanfaat. Dalam penelitian ini, sistem peringkasan otomatis tidak hanya dirancang untuk menghasilkan ringkasan, tetapi juga dievaluasi dengan metrik ROUGE untuk memastikan kualitas dan akurasi informasi (Lin 2004). Evaluasi ini menunjukkan tanggung jawab sosial

agar teknologi yang dikembangkan benar-benar memberi manfaat dan tidak merugikan pengguna.

Dengan demikian, QS. An-Nahl ayat 78 menegaskan sumber potensi intelektual manusia sebagai karunia Allah yang harus disyukuri dan dimanfaatkan, sedangkan ayat 79 menegaskan keteraturan sistem ciptaan Allah yang menjadi dasar pengembangan teknologi secara etis dan bermanfaat. Integrasi kedua ayat tersebut memberikan landasan teologis bahwa penelitian dalam bidang kecerdasan buatan bukan hanya aktivitas akademik, tetapi juga bagian dari tanggung jawab spiritual dan sosial manusia.

Dengan menggabungkan nilai-nilai keislaman dan pengembangan teknologi, penelitian ini tidak hanya berfokus pada aspek teknis, tetapi juga merupakan manifestasi dari rasa syukur dan kesadaran spiritual. Implementasi model BART dalam merangkum transkrip video YouTube (Lewis et al. 2020) secara otomatis adalah bentuk aktualisasi dari kemampuan berpikir yang dikaruniakan oleh Allah, serta penghargaan terhadap tanda-tanda kekuasaan-Nya. Evaluasi dengan metrik ROUGE menjadi alat untuk memastikan bahwa sistem ini tidak hanya canggih secara teknis, tetapi juga efektif dalam menyampaikan inti informasi, sehingga dapat meningkatkan aksesibilitas dan efisiensi pemahaman informasi di era digital.

Oleh karena itu, penelitian ini tidak hanya bertujuan untuk mengatasi masalah informasi yang panjang dan rumit melalui teknologi ringkasan otomatis, tetapi juga menjadi bentuk nyata dari perpaduan antara ilmu pengetahuan modern dan nilai-nilai keagamaan. Dengan memanfaatkan model *BART* berbasis *Transfer Learning* dan mengevaluasi hasilnya menggunakan

metrik ROUGE, sistem yang dikembangkan diharapkan mampu membantu orang memahami informasi lebih cepat dan efisien, serta menjadi bagian dari upaya ilmiah yang bernilai ibadah dalam membangun teknologi yang bermanfaat

Penelitian terhadap transkrip video YouTube perlu dilakukan karena transkrip merupakan sumber data teks yang kaya akan informasi, namun belum banyak dimanfaatkan secara optimal (Fadlilah, Atmadja, and Firdaus 2024). Selain itu, tidak semua pengguna memiliki waktu atau kemampuan untuk membaca seluruh isi transkrip atau menonton video berdurasi panjang. Oleh sebab itu, pengembangan sistem yang dapat secara otomatis meringkas isi transkrip menjadi bentuk yang singkat namun informatif. Sistem semacam ini juga berkontribusi pada peningkatan aksesibilitas merujuk pada upaya menyediakan kemudahan akses terhadap informasi bagi seluruh kalangan, termasuk penyandang disabilitas pendengaran atau keterbatasan waktu (Ilampiray et al. 2023) (Albeer et al. 2022). Di Indonesia, berdasarkan data BPS tahun 2024, penyandang disabilitas pendengaran tipe 3 tercatat mencapai 1,43% dari total populasi penduduk Indonesia, atau sekitar lebih dari 4 juta jiwa, dengan proporsi perempuan yang lebih tinggi dibandingkan laki-laki (Vinet and Zhedanov 2024). Salah satu upaya yang relevan untuk permasalahan tersebut adalah dengan menyediakan sistem peringkasan otomatis terhadap transkrip video YouTube, yang mampu menyampaikan inti informasi secara ringkas tanpa bergantung pada konten audio. Oleh karena itu, penelitian ini bertujuan untuk mengembangkan sistem ringkasan otomatis berbasis metode *BART* dan *Transfer Learning*, serta menyajikannya dalam aplikasi web interaktif yang

mendukung kebutuhan pengguna non-audio, termasuk penyandang disabilitas pendengaran (Ilampiray et al. 2023).

Penelitian ini mengembangkan antarmuka web berbasis Flask untuk meningkatkan aksesibilitas hasil ringkasan. Aplikasi ini memungkinkan pengguna memasukkan URL YouTube dan secara otomatis memperoleh transkrip serta ringkasan teks. Tujuan utamanya adalah menyediakan solusi peringkasan yang praktis dan mudah diakses, terutama bagi pengguna non-teknis, sehingga sistem tidak hanya berfokus pada aspek teknis NLP tetapi juga pada kemudahan penggunaan.

Sementara itu, penelitian sebelumnya seperti yang dilakukan oleh (Vijaya Kumari et al. 2022) hanya berfokus pada penyajian hasil ringkasan teks tanpa mengintegrasikan fitur evaluasi kualitas ringkasan, penyajian informasi metadata video YouTube, maupun opsi ekspor data hasil. Oleh karena itu, penelitian ini memberikan kebaruan yang lebih komprehensif dengan menggabungkan proses *summarization* otomatis, evaluasi kualitas ringkasan menggunakan metrik ROUGE, visualisasi hasil ringkasan, serta opsi ekspor data dalam format Excel, ke dalam satu *platform* web interaktif berbasis Flask yang dapat digunakan secara luas oleh pengguna non-teknis.

1.2. Pernyataan Masalah

Berdasarkan latar belakang yang telah dijelaskan, dapat diidentifikasi beberapa permasalahan yang menjadi fokus dalam penelitian ini. Permasalahan tersebut dirumuskan dalam bentuk pertanyaan sebagai berikut:

Berikut pernyataan masalah yang akan dibahas peneliti:

- a. Bagaimana mengimplementasikan model BART berbasis Transfer Learning untuk peringkasan otomatis transkrip video YouTube?
- b. Bagaimana mengintegrasikan sistem tersebut ke dalam aplikasi web berbasis Flask?
- c. Bagaimana mengevaluasi kualitas ringkasan transkrip menggunakan metrik ROUGE?

1.3. Tujuan Penelitian

Berdasarkan latar belakang dan rumusan masalah yang telah diuraikan, maka tujuan dari penelitian ini adalah:

- a. Mengembangkan sistem peringkasan otomatis terhadap transkrip video YouTube menggunakan model *BART* berbasis *Transfer Learning* untuk meningkatkan aksesibilitas informasi, mempercepat pemahaman isi video, serta mempermudah pencarian informasi dari konten berdurasi panjang.
- b. Melakukan evaluasi terhadap kualitas ringkasan yang dihasilkan oleh sistem menggunakan metrik ROUGE (*ROUGE-1*, *ROUGE-2*, dan *ROUGE-L*) secara terstruktur per video, kemudian digabungkan dan dirata-ratakan berdasarkan kategori durasi video.
- c. Merancang dan mengembangkan antarmuka aplikasi berbasis web (Flask) yang dapat menyajikan hasil ringkasan dan evaluasi ROUGE secara langsung kepada pengguna, sehingga sistem dapat digunakan secara praktis oleh pengguna non-teknis dan meningkatkan keterjangkauan teknologi peringkasan ini dalam kehidupan sehari-hari.

1.4. Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah:

- a. Bagi pengguna umum, sistem peringkasan otomatis ini dapat membantu meningkatkan aksesibilitas informasi, mempercepat pemahaman isi video berdurasi panjang, dan mempermudah pencarian informasi penting dalam video tanpa harus menonton seluruh konten.
- b. Bagi pengembang aplikasi, penelitian ini menghasilkan antarmuka aplikasi berbasis Flask yang siap digunakan untuk menyajikan ringkasan dan hasil evaluasi secara langsung kepada pengguna, sehingga memungkinkan penerapan nyata dalam lingkungan edukasi, media, atau kebutuhan profesional.

1.5. Ruang Lingkup Penelitian

Dengan memperhatikan uraian sebelumnya dan untuk memberikan batasan yang jelas dalam penelitian ini, maka ruang lingkup penelitian ini hanya meliputi:

- a. Penelitian ini memanfaatkan transkrip video otomatis dari YouTube, khususnya dari channel edukatif “Kok Bisa?”, dengan fokus pada konten berbahasa Indonesia.
- b. Proses peringkasan dilakukan menggunakan model *BART (Bidirectional and Auto-Regressive Transformer)* yang telah di-fine-tune menggunakan *Transfer Learning*, dengan fokus pada *extractive summarization*.
- c. Evaluasi sistem dilakukan menggunakan metrik ROUGE (ROUGE-1, ROUGE-2, dan ROUGE-L) berdasarkan tiga aspek (*Precision, Recall, F1-Score*), yang dihitung per video dan dirata-ratakan menurut durasi untuk menilai performa secara kontekstual dan komprehensif.

1.6. Sistematika Penulisan

Berikut adalah uraian singkat yang dibahas pada masing-masing bab.

BAB I PENDAHULUAN

Bab ini menjelaskan mengenai latar belakang masalah, rumusan masalah, , tujuan dan manfaat penelitian baik bagi obyek, ilmu, dan penyusun tesis serta ruang lingkup penelitian. Selain hal itu bab ini juga menjelaskan tentang sistematika penyusunan laporan.

BAB II STUDI PUSTAKA

Bab ini membahas studi pustaka yang mencakup laporan-laporan dari penelitian terdahulu serta teori-teori yang mendasari penggunaan model BART berbasis Transfer Learning untuk peringkasan transkrip video YouTube. Selain itu, bab ini juga menjelaskan implementasi teknologi Flask sebagai antarmuka aplikasi web yang mendukung penyajian hasil ringkasan secara interaktif.

BAB III METODOLOGI PENELITIAN

Bab ini membahas tentang metodologi yang digunakan dalam penelitian berupa metode pengumpulan data dan analisis serta kerangka berfikir penulis dalam lakukan beberapa penelitian. ini.

BAB IV IMPLEMENTASI MODEL BART BERBASIS TRANSFER LEARNING

Bab ini menjelaskan penerapan model BART dalam sistem peringkasan otomatis transkrip video YouTube. Pembahasan meliputi arsitektur BART, proses penerapan transfer learning melalui fine-tuning model, implementasi pengambilan transkrip video menggunakan API YouTube, serta evaluasi kualitas ringkasan menggunakan metrik ROUGE yang terdiri dari ROUGE-1,

ROUGE-2, dan ROUGE-L untuk mengukur tingkat kesamaan antara ringkasan yang dihasilkan sistem dengan teks sumber.

BAB V HASIL DAN PEMBAHASAN

Bab ini membahas implementasi model BART berbasis Transfer Learning dan integrasinya ke dalam aplikasi web Flask untuk peringkasan transkrip video YouTube.

BAB VI KESIMPULAN

Bab ini merupakan bab terakhir yang berisi tentang kesimpulan serta saran dari penyusun.

BAB II

STUDI PUSTAKA

2.1. YouTube sebagai Sumber Data

YouTube telah berkembang menjadi *platform* utama dalam penyebaran informasi, edukasi, dan hiburan. Fitur transkrip otomatis yang disediakan oleh YouTube memungkinkan pengguna untuk mengakses konten video dalam bentuk teks, yang sangat bermanfaat untuk analisis lebih lanjut. Namun, transkrip ini sering kali panjang dan tidak terstruktur, sehingga menyulitkan pengguna dalam memahami inti dari konten video dengan cepat. Penelitian oleh (Zahara et al. 2024) dan (Vira and Reynata 2022) menekankan pentingnya pengolahan transkrip video untuk meningkatkan aksesibilitas informasi.

Untuk mengatasi keterbatasan tersebut, pemanfaatan YouTube API menjadi solusi yang efektif dalam mengakses dan mengelola data transkrip video secara otomatis. YouTube Data API memungkinkan pengembang untuk mengambil metadata video, termasuk informasi transkrip jika tersedia, yang dapat digunakan untuk analisis lebih lanjut atau integrasi dengan aplikasi lain. YouTube Transcript API merupakan alat yang efektif untuk mengakses dan mengelola transkrip video secara otomatis. API ini memungkinkan pengembang untuk mengambil transkrip video YouTube, baik yang dibuat secara manual maupun otomatis, tanpa memerlukan browser headless, sehingga mempercepat dan menyederhanakan alur kerja pengolahan data teks dari video YouTube. Studi oleh (KYATHAM 2022). menunjukkan bahwa penggunaan YouTube-transcript-api efektif dalam mengekstraksi transkrip untuk keperluan *summarization*. Hal ini mempermudah proses pengumpulan data teks dari video

untuk analisis lebih lanjut atau integrasi dengan aplikasi lain. Dalam penelitian oleh (Ilampiray et al. 2023), penggunaan YouTube Transcript API dikombinasikan dengan model BERT untuk menghasilkan ringkasan transkrip video yang informatif dan ringkas. Selain itu, studi oleh (Biswas and Patel 2022) menunjukkan bahwa integrasi API ini dengan model transformer dapat mempercepat proses ekstraksi dan ringkasan transkrip video. Namun, tantangan tetap ada dalam hal kualitas transkrip yang dihasilkan secara otomatis. Kesalahan dalam transkripsi, seperti pengenalan kata yang salah atau kehilangan konteks, dapat mempengaruhi akurasi analisis selanjutnya. Oleh karena itu, penting untuk mengembangkan metode yang dapat membersihkan dan menyusun ulang transkrip agar lebih terstruktur dan mudah dipahami. Pendekatan ini akan meningkatkan efisiensi dalam pemrosesan data teks dan memungkinkan analisis yang lebih mendalam terhadap konten video.

Selain menjadi sumber informasi yang kaya, transkrip YouTube juga memiliki potensi besar dalam meningkatkan aksesibilitas bagi penyandang disabilitas, seperti tuna rungu. Studi oleh (Ilampiray et al. 2023) menunjukkan bahwa transkrip dan ringkasan otomatis secara signifikan membantu kelompok ini dalam memahami isi video tanpa mengandalkan audio. Dengan demikian, pengolahan dan peringkasan transkrip dapat memperluas jangkauan informasi secara inklusif.

2.2. Text Summarization

Beberapa penelitian terdahulu telah membahas penerapan Text *summarization* menggunakan *Natural Language Processing* (NLP). Berikut adalah beberapa penelitian yang relevan dalam bidang ini:

Penelitian yang dilakukan oleh Astuti et al., 2024 menggunakan metode *MBART* (Multilingual *BART*) untuk peringkasan teks berita berbahasa Indonesia dengan pendekatan *Ekstraktif*. Dataset yang digunakan adalah XL-SUM, yang berisi artikel berita dari BBC dalam berbagai bahasa, termasuk Indonesia. Proses penelitian meliputi pra-pemrosesan teks, pelatihan *BART* dengan fine-tuning, dan evaluasi menggunakan metrik ROUGE (ROUGE-1, ROUGE-2, dan ROUGE-L). Hasil evaluasi menunjukkan bahwa metode *MBART50* mencapai skor ROUGE-1 sebesar 35,94, ROUGE-2 sebesar 16,43, dan ROUGE-L sebesar 29,91, yang menunjukkan performa cukup baik tetapi belum mengungguli metode lain dalam peringkasan teks. Studi ini berkontribusi dalam pemahaman tantangan dan peluang dalam pengembangan teknik peringkasan teks otomatis untuk bahasa Indonesia, dengan potensi peningkatan efektivitas metode di masa depan (Astuti, Muljono, and Sutriawan 2024).

Penelitian oleh (Bhagavan et al. 2023) mengeksplorasi penggunaan *Transfer Learning* dengan Metode *BART* yang di-fine-tune secara khusus untuk tugas *abstractive summarization* pada laporan radiologi. Dalam studi ini, peneliti memanfaatkan Metode *BART* yang telah dilatih sebelumnya (pre-trained) dan kemudian di-fine-tune menggunakan pendekatan *Transfer Learning* pada dataset MIMIC-III, yang berisi ribuan laporan radiologi klinis. Tujuannya adalah menghasilkan ringkasan bagian “impression” dari laporan berdasarkan bagian “findings”. Evaluasi dilakukan menggunakan metrik ROUGE-L, BERTScore, dan F1-RadGraph, dengan hasil terbaik mencapai ROUGE-L sebesar 32.39, menunjukkan bahwa *BART* yang di-fine-tune menggunakan *Transfer Learning* mampu menghasilkan ringkasan yang akurat dan informatif

di domain medis, serta berkontribusi dalam meningkatkan efisiensi komunikasi klinis. Penelitian oleh (Zolotareva, Tashu, and Horváth 2020) mengkaji efektivitas Metode *BART* (*Bidirectional and Auto-Regressive Transformer*) yang diimplementasikan dengan pendekatan *Transfer Learning* untuk tugas *abstractive text summarization*. Dalam studi ini, Metode *BART* di-fine-tune secara khusus menggunakan dataset BBC News. Fine-tuning dilakukan agar *BART* dapat menyesuaikan diri dengan struktur dan gaya bahasa domain berita, sehingga mampu menghasilkan ringkasan yang ringkas, koheren, dan kontekstual. Evaluasi dilakukan menggunakan metrik ROUGE, dan diperoleh hasil: ROUGE-1 (*F1-Score*) sebesar 0.473, ROUGE-2 sebesar 0.265, dan ROUGE-L sebesar 0.361. Hasil ini menunjukkan bahwa *BART* yang di-fine-tune dengan *Transfer Learning* menghasilkan ringkasan yang lebih baik dibandingkan model baseline seperti seq2seq.

2.3. *Transfer Learning* dalam *Natural Language Processing*

Transfer Learning merupakan pendekatan dalam machine learning di mana *BART* yang telah dilatih pada satu tugas digunakan kembali untuk tugas lain yang serupa. Dalam konteks *Natural Language Processing* (NLP), *Transfer Learning* memungkinkan *BART* untuk memanfaatkan pengetahuan yang diperoleh dari data besar untuk diterapkan pada tugas dengan data terbatas. Pendekatan ini sangat berguna dalam bahasa dengan sumber daya terbatas, seperti bahasa Indonesia, di mana data berlabel sering kali sulit diperoleh. Penelitian oleh (Leonandya and Ikhwantri 2019), menunjukkan bahwa *Transfer Learning* dapat meningkatkan performa *BART* dalam tugas Named Entity Recognition (NER) pada teks percakapan bahasa Indonesia. Dengan

memanfaatkan model bahasa yang telah dilatih sebelumnya, mereka berhasil meningkatkan skor F1 secara signifikan meskipun menggunakan data pelatihan yang terbatas.

Pendekatan *Transfer Learning* juga telah diterapkan dalam berbagai tugas NLP lainnya, seperti analisis sentimen, klasifikasi teks, dan peringkasan teks. Dengan menggunakan model pra-latih seperti BERT, GPT, dan *BART*, peneliti dapat mengadaptasi model tersebut untuk tugas spesifik dengan sedikit penyesuaian. Hal ini memungkinkan pengembangan aplikasi NLP yang efisien dan efektif, terutama dalam konteks bahasa yang kurang terwakili dalam data pelatihan besar. Studi oleh (Howard and Ruder 2018) memperkenalkan Universal Language Model Fine-tuning (ULMFiT), yang menunjukkan bahwa *Transfer Learning* dapat menghasilkan performa state-of-the-art dalam berbagai tugas NLP dengan data pelatihan yang terbatas.

2.4. *BART* dalam *Natural Language Processing*

BART (*Bidirectional and Auto-Regressive Transformers*) merupakan arsitektur metode bahasa berbasis transformer yang menggabungkan keunggulan dari encoder BERT dan decoder GPT. Metode ini dirancang oleh (Lewis et al. 2020) sebagai sebuah autoencoder denoising untuk tugas-tugas pemrosesan bahasa alami (*Natural Language Processing/NLP*) seperti text generation, translation, dan *summarization*. Dalam konteks ringkasan transkrip video YouTube, kemampuan *BART* dalam memahami konteks dua arah dari transkrip menghasilkan ringkasan dengan memilih kalimat-kalimat representatif dari teks sumber menjadikannya sangat cocok untuk menghasilkan ringkasan yang padat dan informatif.

BART bekerja dengan melakukan encoding terhadap input teks yang telah terdistorsi (misalnya dengan penghapusan, permutasi, atau masking), kemudian mendekodinya kembali menjadi bentuk asli. Pendekatan ini melatih *BART* untuk merekonstruksi teks secara kontekstual, yang penting dalam merangkum transkrip video panjang agar tetap mempertahankan makna utama dari konten. Dengan demikian, *BART* sangat berguna untuk tugas ringkasan otomatis, terutama dalam skenario yang memerlukan pemahaman kontekstual yang mendalam seperti transkrip percakapan atau video (Lewis et al. 2020).

Dalam konteks *Transfer Learning*, Metode *BART* dapat disesuaikan ulang melalui proses fine-tuning pada dataset yang bersifat domain-spesifik, seperti transkrip video YouTube, guna meningkatkan kinerja dalam tugas peringkasan. Pendekatan ini selaras dengan temuan pada studi (Zhao and Chen 2022), yang membandingkan efektivitas fine-tuning penuh dan penggunaan adapter pada Metode *BART* untuk peringkasan teks. Hasil penelitian menunjukkan bahwa fine-tuning secara menyeluruh memberikan performa terbaik, khususnya saat data pelatihan tersedia dalam jumlah besar. Proses evaluasi dilakukan menggunakan metrik ROUGE untuk menilai kesesuaian ringkasan yang dihasilkan dengan terhadap transkrip asli, menjadikan pendekatan ini relevan untuk skenario seperti peringkasan transkrip video berdurasi panjang. Beberapa studi menunjukkan bahwa *BART* memiliki performa yang unggul dibandingkan metode – metode lain seperti LSTM atau Transformer encoder-only dalam tugas *text summarization*. Hal ini diperkuat oleh hasil eksperimen (Sharma and Sharma 2023) dan (Lewis et al. 2020), yang menunjukkan bahwa *BART* mampu

menghasilkan ringkasan yang lebih koheren dan informatif dibandingkan metode lainnya dalam berbagai domain.

Selama pelatihan, *BART* belajar memprediksi token output (ringkasan) dari input (teks panjang/transkrip) menggunakan perhitungan berikut:

➤ Cross-Entropy Loss:

Model menghitung seberapa besar perbedaan antara prediksi token dan token sebenarnya (target):

$$Loss = \sum_{i=1}^N y_i \cdot \log(\hat{y}_i) \quad (2.1)$$

Dimana:

- y_i adalah distribusi target (biasanya one-hot)
- \hat{y}_i adalah probabilitas prediksi dari *BART*
- N adalah jumlah token dalam output ringkasan

➤ Decoding (Inference): Perhitungan Probabilitas Kata

Saat inference (pengujian), *BART* menghitung distribusi probabilitas dari seluruh kosakata dan memilih token dengan probabilitas tertinggi untuk dijadikan output:

$$L y^t = \operatorname{argmax}_P(y^t \mid y < t, X) \quad (2.2)$$

Dalam proses inferensi pada model *BART*, terdapat beberapa metode decoding yang umum digunakan untuk menghasilkan urutan kata dalam ringkasan. Salah satu metode dasar adalah greedy decoding, yaitu memilih token dengan probabilitas tertinggi pada setiap langkah waktu secara deterministik (Lewis et al. 2020). Meskipun cepat, pendekatan ini cenderung menghasilkan output yang kurang bervariasi. Alternatif yang lebih canggih

adalah *beam search*, yang mempertimbangkan beberapa urutan kandidat terbaik secara simultan untuk mendapatkan hasil yang lebih optimal dan koheren. Selain itu, terdapat pula metode berbasis sampling seperti top-k sampling dan top-p (nucleus) sampling, yang memilih token berdasarkan probabilitas terpotong dari distribusi model, sehingga memberikan lebih banyak variasi dan kreativitas dalam output. Pemilihan strategi decoding ini berpengaruh signifikan terhadap kualitas dan gaya ringkasan yang dihasilkan (Ari Holtzman 2020).

Dalam konteks aplikasi nyata, integrasi Metode *BART* ke dalam *platform* antarmuka pengguna berbasis web seperti Flask semakin memperluas fungsionalitasnya. Hal ini memungkinkan pengguna non-teknis untuk memanfaatkan kemampuan *summarization* melalui tampilan antarmuka yang sederhana dan interaktif. Studi oleh (Dharmani 2022) juga menunjukkan efektivitas penerapan model NLP melalui ekstensi browser atau UI sederhana dalam meningkatkan pemanfaatan teknologi *summarization* oleh pengguna awam.

2.5. Evaluasi Ringkasan dengan Metrik ROUGE

Evaluasi kualitas ringkasan teks otomatis umumnya dilakukan dengan membandingkan hasil ringkasan yang dihasilkan mesin dengan terhadap transkrip asli yang disusun oleh manusia. Metrik yang paling umum digunakan dalam evaluasi ini adalah ROUGE (*Recall-Oriented Understudy for Gisting Evaluation*), yang diperkenalkan oleh (Lin 2004). ROUGE mengukur kesamaan berdasarkan n-gram, urutan kata, dan subsekuens terpanjang yang sama antara ringkasan sistem dan referensi.

Penelitian oleh (Ülker and Özer 2024) mengevaluasi kinerja Metode *BART* untuk tugas peringkasan artikel ilmiah menggunakan tiga varian utama metrik ROUGE, yaitu ROUGE-1, ROUGE-2, dan ROUGE-L. ROUGE-1 mengukur kecocokan pada tingkat unigram atau kata tunggal, ROUGE-2 mengukur pada tingkat bigram atau pasangan kata berturutan, dan ROUGE-L mengukur kemiripan berdasarkan *Longest Common Subsequence (LCS)*, yaitu urutan kata terpanjang yang muncul di kedua ringkasan tanpa harus berurutan secara langsung. Evaluasi dilakukan dengan memperhatikan tiga komponen utama, yaitu *Recall* yang mengukur seberapa banyak bagian dari terhadap transkrip asli berhasil ditangkap oleh sistem, *Precision* yang menilai relevansi ringkasan sistem terhadap referensi, serta *F1-Score* sebagai rata-rata harmonik dari keduanya. Hasil penelitian menunjukkan bahwa Metode *BART* mampu menghasilkan ringkasan yang relevan dan informatif dengan skor ROUGE yang tinggi.

Penelitian oleh (Dharmani 2022) mengembangkan sistem *YouTube Transcript Summarizer* berbasis *Transfer Learning* dengan Metode *BART* (facebook/*BART-large-cnn*) untuk melakukan *abstractive summarization* pada transkrip video YouTube. Sistem ini menerima input berupa URL video, mengekstrak transkrip secara otomatis, dan menghasilkan ringkasan menggunakan pendekatan pemrosesan bahasa alami. Evaluasi terhadap hasil ringkasan dilakukan menggunakan metrik ROUGE, dengan hasil menunjukkan nilai ROUGE-1 sebesar 0.542, ROUGE-2 sebesar 0.382, dan ROUGE-L sebesar 0.498. Ketiga nilai ini dihitung berdasarkan *Recall*, *Precision*, dan *F1-Score*, yang mengukur seberapa banyak informasi dari terhadap transkrip asli berhasil

ditangkap oleh sistem, relevansi informasi yang dihasilkan, serta keseimbangan antara keduanya. Hasil ini menunjukkan bahwa Metode *BART* mampu menghasilkan ringkasan yang informatif dan relevan secara kontekstual, serta dapat diterapkan secara efektif untuk membantu pengguna memahami isi video secara cepat dan efisien (Dharmani 2022).

Evaluasi kualitas ringkasan dilakukan tidak hanya pada tingkat per video, tetapi juga digabungkan dan dirata-ratakan berdasarkan kategori durasi video. Pendekatan ini bertujuan untuk mengukur konsistensi performa metode secara tematik dan kontekstual, mengingat adanya variasi bahasa serta gaya narasi yang signifikan antar video. Evaluasi semacam ini memberikan gambaran performa sistem yang lebih representatif terhadap kumpulan video dalam domain edukatif.

2.6. Implementasi Flask dalam Aplikasi *Summarization*

Dalam Sejumlah penelitian sebelumnya telah memanfaatkan framework Flask dalam pengembangan aplikasi *summarization* berbasis web guna meningkatkan aksesibilitas, fleksibilitas, dan efisiensi dalam penyajian ringkasan. (Vijaya Kumari et al. 2022) mengembangkan sistem peringkasan transkrip video YouTube yang memanfaatkan pendekatan *Natural Language Processing* (NLP) dan model HuggingFace Transformers. Flask digunakan sebagai backend untuk menangani permintaan API yang berisi tautan video dari pengguna. Server kemudian mengambil subtitle melalui YouTube API, melakukan proses *summarization*, dan mengembalikan hasil dalam bentuk teks ringkasan. Pendekatan ini menunjukkan bahwa Flask mampu mengelola seluruh alur pemrosesan dengan ringan dan efisien.

Sementara itu, (Rajput et al. 2022) menerapkan Flask dalam aplikasi web *summarization* artikel berita yang menggabungkan teknik *Ekstraktif* dan abstraktif. Sistem mereka mencakup proses crawling artikel dari situs berita menggunakan *newspaper3k*, ekstraksi konten menggunakan fungsi *nlp()*, dan penyajian ringkasan melalui antarmuka yang dibangun menggunakan HTML, CSS, dan Streamlit. Flask menjadi penghubung antara backend dan frontend, serta menangani proses pengambilan konten, pemrosesan NLP, hingga penampilan ringkasan dan metadata seperti nama penulis, gambar, dan sumber artikel.

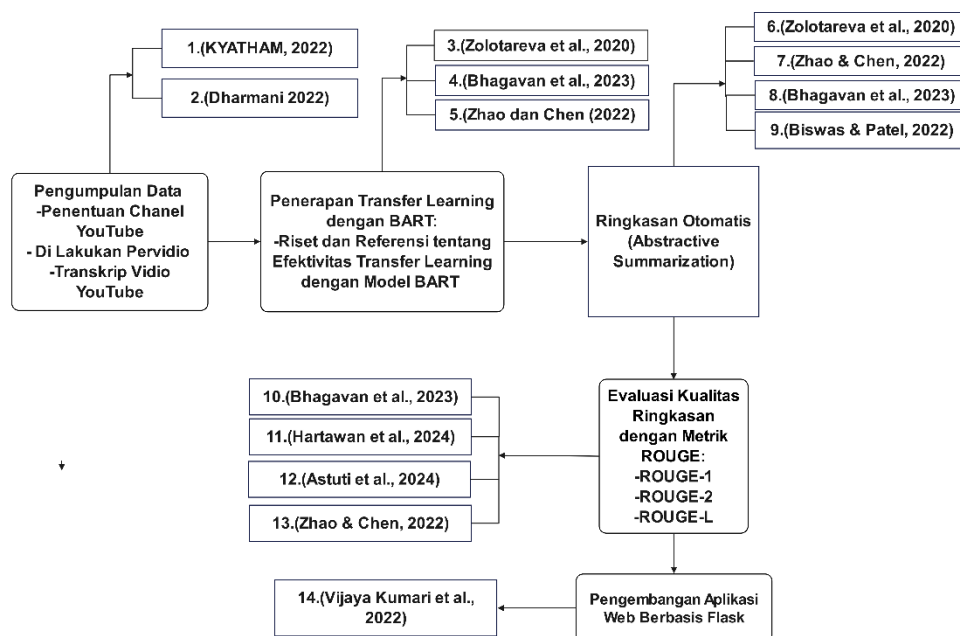
Penelitian lain oleh (Pabitwar et al. 2024) memanfaatkan Flask untuk mengembangkan aplikasi Resume *summarization* yang merangkum informasi penting dari *dokumen* PDF dan DOCX. Proyek ini mengintegrasikan pustaka NLP spaCy, regex.

Berdasarkan ketiga studi tersebut, dapat disimpulkan bahwa Flask merupakan framework yang efektif dan ringan untuk pengembangan aplikasi *summarization* berbasis web. Kemampuannya dalam mengintegrasikan berbagai modul pemrosesan, mengelola API, serta menyediakan fitur interaktif bagi pengguna menjadikannya pilihan yang tepat dalam menjembatani teknologi NLP dengan kebutuhan pengguna secara real-time dan praktis. Namun demikian, ketiga studi tersebut hanya berfokus pada penyajian hasil ringkasan teks tanpa mengintegrasikan fitur evaluasi kualitas ringkasan, penyajian informasi metadata video YouTube, maupun opsi ekspor data hasil. Oleh karena itu, penelitian ini memberikan kontribusi yang lebih komprehensif dengan menggabungkan proses *summarization* otomatis, evaluasi kualitas

ringkasan menggunakan metrik ROUGE, visualisasi hasil ringkasan, serta opsi ekspor data dalam format Excel ke dalam satu *platform* web interaktif berbasis Flask yang dapat diakses secara luas oleh pengguna non-teknis.

2.7. Kerangka Teori

Penelitian ini memiliki kerangka teori yang mengacu pada jurnal-jurnal dan tergambar dengan jelas pada Gambar 2.1 Kerangka Teori berikut ini:



Gambar 2. 1 Kerangka Teori

Berdasarkan kerangka teori pada Gambar 2.1, alur proses dalam pengembangan sistem peringkasan transkrip video YouTube ditunjukkan secara sistematis menggunakan pendekatan *Transfer Learning* dengan metode *BART*. Proses dimulai dari pemilihan channel YouTube yang relevan, dilanjutkan dengan pengumpulan data berupa video yang kemudian ditranskrip secara otomatis. Selanjutnya, transkrip tersebut diproses menggunakan metode *BART* yang telah dioptimalkan melalui metode *Transfer Learning* untuk menghasilkan ringkasan otomatis secara *Ekstraktif* dan dikategorikan berdasarkan durasi video sesuai klasifikasi yang telah ditetapkan. Hasil ringkasan tersebut

kemudian dievaluasi menggunakan metrik ROUGE (ROUGE-1, ROUGE-2, dan ROUGE-L) guna mengukur kualitas ringkasan secara kuantitatif. Sebagai kontribusi praktis, sistem ini dikembangkan dalam bentuk aplikasi web berbasis Flask yang memungkinkan pengguna untuk memasukkan URL video, menampilkan hasil transkrip dan ringkasan, serta mengakses hasil evaluasi secara interaktif. Kerangka ini merepresentasikan hubungan sistematis antara proses pengumpulan data, pemodelan NLP, evaluasi performa, dan penerapannya dalam *platform* aplikasi. Adapun daftar penelitian terdahulu yang menjadi acuan dalam pengembangan sistem ini dapat dilihat pada Tabel 2.1.

Tabel 2. 1 Daftar Jurnal

No.	Judul Jurnal	Metode Penelitian	Penulis, Tahun	Hasil Penelitian
1	<i>Bidirectional and Auto-Regressive Transformer (BART) for Indonesian Abstractive Text Summarization</i>	<i>Bidirectional and Auto-Regressive Transformer (BART)</i>	(Hartawan et al., 2024)	Hasil: ROUGE-1: 37,19 ROUGE-2: 14,03 ROUGE-L: 33,85
2	Indonesian News Text summarization Using MBART Algorithm	<i>Peringkasan teks abstraktif dengan algoritma MBART (Multilingual BART).</i>	(Astuti et al., 2024)	Hasil: ROUGE-1: 35.94 ROUGE-2: 16.43 ROUGE-L: 29.91
3	VIDEO TRANSCRIPT SUMMARIZER	<i>Implementasi API YouTube dan algoritma (LSA)</i>	KYATHAM, 2022;	API mampu mengekstraksi transkrip dengan cepat dan akurat untuk analisis NLP.
4	Abstractive summarization of Radiology Reports using BART Finetuning	<i>BART yang di fine tuning menggunakan Transfer Learning</i>	Bhagavan et al., 2023	Nilai ROUGE-L berkisar antara 30.96 – 32.39 BERTScore (52–55) F1-RadGraph (32–33)

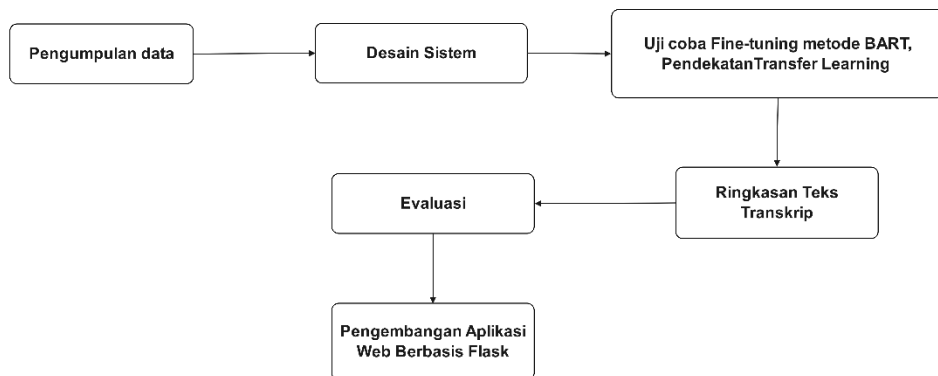
5	Pretrained language model transfer on neural named entity recognition in Indonesian conversational texts	<i>Transfer Learning</i> ELMo Flair CNN-BiLSTM-CRF	Leonandya & Ikhwantri, 2019	<i>Transfer Learning</i> meningkatkan performa NER pada teks Bahasa Indonesia. F1 hingga 85,78%
6	<i>YouTube Transcript Summarizer</i>	Metode <i>BART</i> dan T5 (Text-to-Text Transfer Transformer)	Dharmani 2022	Ekstensi Chrome yang merangkum transkrip YouTube secara otomatis menggunakan <i>BART</i> dan T5 dengan ringkasan yang efisien dan mudah dipahami.
7	To Adapt or to Fine-tune: A Case Study on Abstractive Summarization	Pendekatan <i>Transfer Learning</i> pada Metode <i>BART</i>	Zhao & Chen, 2022	ROUGE-1 hingga 47.4 ROUGE-2 hingga 30.18 ROUGE-L hingga 42.26
8	<i>VIDEO TRANSCRIPT SUMMARIZER</i>	Extractive summarization dengan model BERT. Abstractive summarization menggunakan model T5. Sumy dengan LSA	Manikanta Reddy K. (2022)	Sistem ini menggunakan model BERT dan T5 untuk merangkum transkrip video YouTube melalui ekstensi Chrome, menghasilkan ringkasan yang informatif dan responsive
9	<i>YouTube Transcript Summarizer To Summarize the content of YouTube</i>	<i>Abstractive Summarization</i> : Menggunakan model pre-trained dari Hugging Face (DistilBERT melalui <i>pipeline summarization</i>)	(Biswas & Patel, 2022)	Metode menggunakan <i>Hugging Face Transformer</i> (DistilBERT) berhasil merangkum transkrip YouTube secara otomatis dan efisien, dengan hasil yang ringkas
10	<i>Abstractive text summarization using Transfer Learning</i>	<i>Transfer Learning</i> , BERT, T5, dan GPT	(Zolotareva et al., 2020)	ROUGE-1 (F1-Score) 0.473, ROUGE-2 sebesar 0.265 ROUGE-L sebesar 0.361

11	YouTube Transcript Summarizer Using Flask And NLP	Implementasi sistem NLP berbasis Flask untuk peringkasan transkrip video YouTube menggunakan <i>HuggingFace Transformers</i> dan CNN	(Vijaya Kumari et al. 2022)	Berhasil mengembangkan sistem yang dapat menampilkan ringkasan video YouTube secara otomatis.
----	---------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------	-----------------------------	-----------------------------------------------------------------------------------------------

BAB III METODOLOGI PENELITIAN

3.1. Desain Penelitian

Desain penelitian yang digunakan dalam studi ini disusun secara sistematis untuk menggambarkan alur pelaksanaan penelitian dari awal hingga akhir. Rangkaian tahapan dalam desain tersebut dapat dilihat secara visual pada Gambar 3.1 berikut ini, yang menyajikan struktur dan langkah-langkah utama dalam Desain Penelitian:



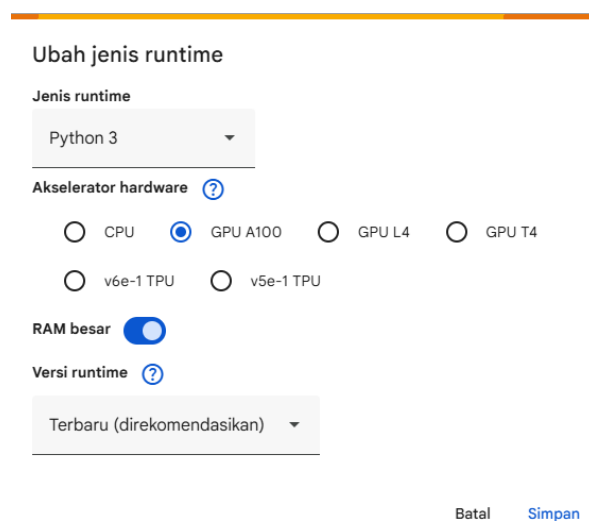
Gambar 3. 1 Desain Penelitian

3.2. Pengumpulan Data

Adapun tahapan pengumpulan data dalam penelitian ini dilakukan secara terstruktur melalui beberapa langkah sistematis. Data primer diperoleh langsung dari *platform* YouTube dengan memanfaatkan YouTube Data API sebagai alat utama untuk mengakses dan mengumpulkan data secara otomatis. Pemanfaatan API ini memungkinkan proses akuisisi data dilakukan secara terprogram dan terkontrol, sehingga meningkatkan efisiensi serta konsistensi dalam pengambilan dataset.

YouTube Data API *diperoleh* melalui Google Developer Console dengan cara mendaftarkan proyek penelitian, mengaktifkan layanan YouTube Data

API, serta menghasilkan API Key yang digunakan sebagai kredensial autentikasi dalam setiap permintaan data ke server YouTube. Seluruh proses pemanggilan API dijalankan menggunakan lingkungan komputasi Google Colab Pro dengan konfigurasi *runtime* GPU NVIDIA A100 seperti yang terlihat pada Gambar 3.2 Konfigurasi *Runtime* Google Colab Pro, sebagaimana telah dijelaskan sebelumnya. Penggunaan lingkungan komputasi berbasis cloud ini mendukung stabilitas proses scraping URL, pengelolaan dataset berskala besar, serta memastikan eksekusi kode berjalan secara berkelanjutan tanpa keterbatasan perangkat keras lokal.



Gambar 3. 2 Konfigurasi Runtime Google Colab Pro

Dalam penelitian ini, data diperoleh dari kanal YouTube “Kok Bisa?”, yang terdiri atas 26 playlist dengan jumlah keseluruhan 730 video. Pemilihan kanal ini didasarkan pada karakteristik konten edukatif yang konsisten dan memiliki struktur naratif yang informatif, sehingga relevan dengan tujuan penelitian dalam mengembangkan sistem peringkasan otomatis berbasis metode *BART*. Setiap video membahas topik tertentu, seperti sains, teknologi, sosial, dan fenomena umum, yang disampaikan dalam format edukatif populer.

Daftar video yang digunakan dalam penelitian ini disajikan secara ringkas pada Tabel 3.1. Tabel tersebut memuat informasi utama yang diperoleh melalui proses scraping, meliputi judul video, durasi, tanggal unggah, serta URL video. Informasi ini digunakan untuk proses kategorisasi durasi, pengelompokan dataset, serta dokumentasi eksperimen. Adapun contoh format Tabel 3.1 ditunjukkan sebagai berikut.

Tabel 3. 1 Daftar Video

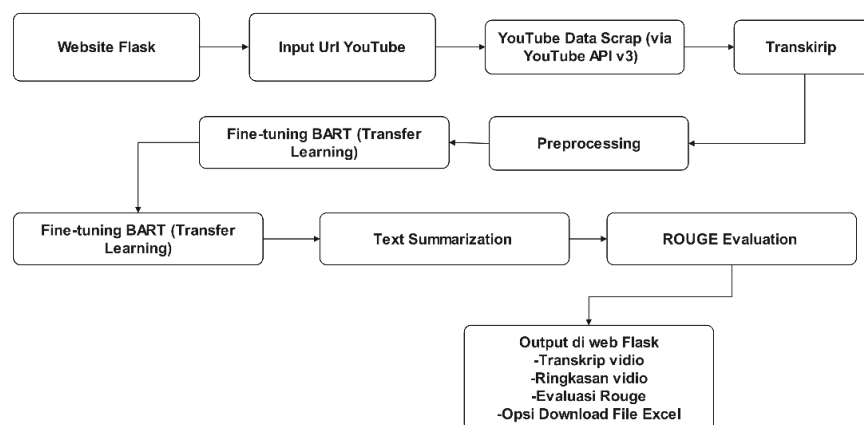
No	playlist_title	video_title	video_url
1	Nusantarasa: Seri Kekayaan Kuliner Indonesia!	Segera Hadir: Nusantarasa - Seri Kekayaan Kuliner Indonesia	https://www.YouTube.com/watch?v=dbY-vmH1yO8
2	Nusantarasa: Seri Kekayaan Kuliner Indonesia!	Kenapa Masakan Indonesia Beragam Banget? - Nusantarasa Eps. 1	https://www.YouTube.com/watch?v=zNq-rD5FFjM
3	Nusantarasa: Seri Kekayaan Kuliner Indonesia!	Benarkah Rendang Bisa Tahan Setahun Lamanya? - Nusantarasa Eps. 2	https://www.YouTube.com/watch?v=E-Xiiqkuils
4	Nusantarasa: Seri Kekayaan Kuliner Indonesia!	Kenapa Makanan Jawa Manis-manis?	https://www.YouTube.com/watch?v=qAHQVt7LoTA
	Indonesia Kaya: Season 2	Coming Soon: Indonesia Kaya Series (Season #2)	https://www.YouTube.com/watch?v=KEDzgTLPO9A
6	Indonesia Kaya: Season 2	'Orkestra' Nusantara Yang Menggelegar Dunia	https://www.YouTube.com/watch?v=OJmzDvAKpLs
7	Indonesia Kaya: Season 2	Tari Api yang Penuh Misteri	https://www.YouTube.com/watch?v=3gl5sUEBGgw
8	Indonesia Kaya: Season 2	Menjadi 'Superpower' Di Zaman Lampau	https://www.YouTube.com/watch?v=wdpZq9wAxPc
9	Indonesia Kaya: Season 2	Biji 'Emas' Yang Diperebutkan Dunia	https://www.YouTube.com/watch?v=hrfbQG6B1h0
10	<u>Matematika</u>	Kenapa $1+1=2$? Gak 3, 4, atau 5?	https://www.YouTube.com/watch?v=KPpi2DAwfZE
----	-----	-----	-----
		--	

730	Biologi	Apa yang Terjadi pada Otak Orang Mabuk?	https://www.YouTube.com/watch?v=Dq9sxagzOkq
Jumlah Video		730	

Setiap URL video YouTube dimasukkan ke dalam aplikasi web flask, kemudian sistem secara otomatis mengambil transkrip dari video tersebut untuk dilakukan proses peringkasan dan evaluasi kualitas menggunakan metrik ROUGE. Transkrip asli dan hasil ringkasan ditampilkan berdampingan agar pengguna dapat membandingkan secara langsung, sedangkan hasil evaluasi ROUGE yang terdiri dari nilai *Precision*, *Recall*, dan *F1-Score* untuk ROUGE-1, ROUGE-2, dan ROUGE-L ditampilkan untuk menilai kualitas ringkasan. Pendekatan ini memberikan kejelasan dalam menilai performa metode *BART* yang di fine tuning dengan *Transfer Learning* dalam merangkum konten video secara efektif, serta mendukung validasi otomatis berbasis metrik evaluatif yang objektif.

3.3. Desain Sistem

Tahapan Desain Sistem dalam penelitian ini dijelaskan secara lebih rinci pada Gambar 3.3 Desain Sistem yang ditampilkan berikut ini. Gambar tersebut menyajikan alur kerja sistem yang dikembangkan, mulai dari proses awal



Gambar 3. 3 Desain Sistem

Desain sistem ini menjelaskan proses otomatis peringkasan transkrip video YouTube menggunakan Metode *BART* yang telah di-fine-tune dengan pendekatan *Transfer Learning*, dengan tahapan-tahapan sebagai berikut:

1. Website Flask

Website Flask merupakan antarmuka aplikasi berbasis web yang dibangun menggunakan framework Flask, yaitu sebuah framework Python yang ringan dan fleksibel untuk pengembangan aplikasi web. Sistem ini dimulai dari antarmuka web berbasis Flask, yang memungkinkan pengguna mengakses aplikasi melalui browser secara langsung tanpa memerlukan instalasi tambahan (Vijaya Kumari et al. 2022). Melalui antarmuka ini, pengguna dapat memasukkan URL video YouTube, melihat transkrip yang dihasilkan, memperoleh ringkasan otomatis, serta mengakses hasil evaluasi secara interaktif dan terstruktur.

2. Input URL YouTube

Input URL YouTube merupakan proses awal interaksi pengguna dengan sistem, di mana pengguna diminta untuk memasukkan tautan (URL) dari video YouTube yang ingin dianalisis. URL tersebut berfungsi sebagai identitas unik untuk setiap video, yang kemudian digunakan oleh sistem untuk mengambil data seperti transkrip, metadata, dan durasi. Pengguna memasukkan URL video YouTube ke dalam form input yang tersedia pada halaman web Flask, yang secara otomatis memicu proses ekstraksi data dari server YouTube melalui API yang terintegrasi (KYATHAM 2022).

3. YouTube Data Scrap (via YouTube API v3)

Aplikasi

akan menggunakan YouTube Data API v3 untuk mengambil data dari video

tersebut, termasuk metadata seperti judul, durasi, dan tanggal publikasi, serta transkrip jika tersedia. YouTube Data API v3 adalah layanan antarmuka pemrograman aplikasi yang disediakan oleh Google, yang memungkinkan pengembang untuk mengakses informasi video secara terstruktur melalui permintaan HTTP. API ini menyediakan kemampuan untuk mengekstrak informasi penting dari video tanpa harus melakukan pengambilan data secara manual. Dengan memanfaatkan API ini, sistem dapat secara otomatis memperoleh data relevan yang dibutuhkan untuk proses *summarization* dan evaluasi (Ilampiray et al. 2023) & (KYATHAM 2022)

4. Transkrip

Transkrip dari video yang diperoleh akan menjadi data utama yang akan diproses untuk pembuatan ringkasan. Transkrip merupakan hasil konversi otomatis dari konten audio video menjadi bentuk teks, yang biasanya mencakup seluruh narasi, penjelasan, atau percakapan yang disampaikan dalam video. Dalam konteks ini, transkrip berfungsi sebagai representasi teks dari informasi audiovisual yang memungkinkan sistem NLP untuk menganalisis, mengekstrak makna, dan merangkum kontennya secara efisien (Albeer et al. 2022)

5. Preprocessing

Transkrip yang diambil akan dibersihkan melalui proses preprocessing, yaitu tahapan awal dalam *Natural Language Processing* (NLP) yang bertujuan untuk meningkatkan kualitas data teks sebelum dianalisis lebih lanjut. Preprocessing membantu menyederhanakan struktur teks dan menghilangkan elemen-elemen yang tidak relevan atau mengganggu pemrosesan. Dalam penelitian ini, preprocessing dilakukan melalui beberapa langkah, seperti:

- Menghapus karakter khusus (misalnya simbol, tanda baca berlebih)
- Menghilangkan teks pengisi (*fillers*) seperti “eh”, “uh”, atau “hmm”
- Melakukan normalisasi teks, yaitu mengubah kata ke bentuk baku atau konsisten (contoh: "gak" → "tidak", "nggak" → "tidak")

Langkah ini sangat penting untuk memastikan bahwa input yang diberikan ke model *summarization* bersih, seragam, dan bermakna (Jakhotiya et al. 2022).

6. Fine-Tuning *BART* (*Transfer Learning*)

Fine-tuning *BART* (*Transfer Learning*) merupakan proses pelatihan lanjutan terhadap model *BART* yang sebelumnya telah melalui tahap pretraining menggunakan data berskala besar dan beragam. Pada tahap fine-tuning, model *BART* disesuaikan kembali menggunakan data spesifik, dalam hal ini transkrip video YouTube berbahasa Indonesia, agar model mampu menghasilkan ringkasan yang lebih relevan, kontekstual, dan sesuai dengan gaya bahasa domain tertentu. Pendekatan ini memanfaatkan kekuatan *Transfer Learning*, yaitu memindahkan pengetahuan yang telah diperoleh dari data umum ke tugas atau domain yang lebih spesifik, sehingga mempercepat proses pelatihan dan meningkatkan akurasi output (Lewis et al. 2020)

7. Text *Summarization*

Text *summarization* adalah proses menghasilkan versi ringkas dari suatu teks yang tetap mempertahankan informasi inti dan makna utama dari dokumen aslinya. Tujuan utama dari *summarization* adalah untuk menyajikan informasi secara efisien dan cepat dipahami, terutama saat menghadapi dokumen yang panjang atau kompleks. Dalam penelitian ini, hasil preprocessing akan dimasukkan ke dalam metode *BART* untuk menghasilkan ringkasan otomatis

dari transkrip video, di mana model akan memproses input teks dan menghasilkan output dalam bentuk kalimat atau paragraf yang merepresentasikan inti dari keseluruhan isi transkrip. *Text summarization* menggunakan *Ekstraktif* (memilih kalimat penting secara langsung) dan *BART* termasuk ke dalam pendekatan *Ekstraktif* yang lebih fleksibel (Aggarwal and Zhai 2019)

8. ROUGE Evaluation

ROUGE Evaluation (*Recall-Oriented Understudy for Gisting Evaluation*) adalah kumpulan metrik evaluasi yang digunakan secara luas untuk menilai kualitas hasil ringkasan otomatis dengan cara Evaluasi dilakukan dengan membandingkan ringkasan hasil model terhadap teks transkrip asli menggunakan metrik ROUGE. ROUGE menghitung kesamaan antara kedua teks berdasarkan tumpang tindih *n-gram* (urutan kata), urutan kata terpanjang yang sama (LCS), dan kesesuaian struktur kalimat. Dalam penelitian ini, ringkasan yang dihasilkan dievaluasi menggunakan metrik ROUGE (ROUGE-1, ROUGE-2, dan ROUGE-L) untuk mengukur kualitas berdasarkan nilai *Precision*, *Recall*, dan *F1-Score*, dengan tujuan menilai seberapa akurat dan relevan informasi yang diringkas dibandingkan dengan teks referensi. Evaluasi ini penting untuk menilai seberapa baik model memahami konteks dan menyampaikan inti informasi (Ülker and Özer 2024).

9. Output di Web Flask

Output di Web Flask merujuk pada tampilan akhir hasil pemrosesan yang disajikan secara langsung melalui antarmuka pengguna berbasis web. Setelah proses input, ekstraksi transkrip, peringkasan, dan evaluasi dilakukan, seluruh

hasil ditampilkan dalam satu halaman interaktif yang mudah diakses oleh pengguna. Antarmuka ini mencakup beberapa komponen utama, yaitu:

- Judul video, durasi, dan tanggal publikasi yang diambil dari metadata YouTube
- Transkrip asli video hasil ekstraksi otomatis dari YouTube API
- Hasil ringkasan yang dihasilkan oleh model *BART* berbasis *Transfer Learning*
- Skor evaluasi ROUGE (ROUGE-1, ROUGE-2, ROUGE-L) untuk menilai kualitas ringkasan secara objektif
- Opsi untuk mengunduh hasil data dalam format Excel, guna mendukung analisis lebih lanjut atau dokumentasi hasil

Desain ini bertujuan untuk menyajikan hasil secara informatif dan praktis, serta mendukung pengguna non-teknis agar dapat memahami dan memanfaatkan teknologi *summarization* dengan mudah

3.4. Implementasi Pengambilan Transkrip Video YouTube

Bagian ini menjelaskan implementasi aplikasi berbasis web menggunakan Flask untuk melakukan pengambilan transkrip dari video YouTube secara otomatis, menghasilkan ringkasan dengan metode BART yang telah di-fine-tune menggunakan pendekatan Transfer Learning, serta melakukan evaluasi kualitas ringkasan menggunakan metrik ROUGE. Aplikasi dapat diakses secara publik menggunakan Ngrok yang menjembatani server lokal ke internet.

a. Instalasi

Aplikasi memanfaatkan beberapa library Python utama:

flask: framework web ringan

pyngrok: tunneling dari localhost ke internet

YouTube-transcript-api: ekstraksi transkrip otomatis YouTube

transformers: library model NLP dari HuggingFace, termasuk BART

Rouge-score: menghitung skor evaluasi ROUGE

pandas dan openpyxl: menyimpan hasil dalam Excel

google-api-python-client: mengambil metadata video YouTube

Perintah instalasi:

```
pip install flask pyngrok youtube-transcript-api transformers rouge-score pandas openpyxl google-api-python-client
```

b. Penerapan Transfer Learning (BART Fine-Tuning)

Metode BART (Bidirectional and Auto-Regressive Transformer) digunakan sebagai metode utama untuk ekstraktif summarization, karena kemampuannya dalam memahami konteks dua arah dan menghasilkan ringkasan dengan memilih kalimat-kalimat representatif dari teks sumber. Dalam proses decoding, BART menggunakan strategi Beam search, yaitu teknik pencarian yang mempertahankan beberapa jalur kandidat token terbaik pada setiap langkah prediksi. Pendekatan ini memungkinkan model mengeksplorasi kemungkinan urutan kata yang lebih luas dibandingkan metode greedy decoding, sehingga dapat menghasilkan ringkasan yang lebih koheren dan informatif (Lewis et al. 2020). Namun, agar metode BART lebih sesuai dengan konteks bahasa Indonesia dan struktur transkrip video YouTube, dilakukan proses fine-tuning dengan pendekatan Transfer Learning.

Transfer Learning dilakukan dengan memanfaatkan metode BART yang telah dilatih sebelumnya (pretrained) pada data umum berbahasa Inggris, kemudian di-fine-tune menggunakan dataset khusus berupa kumpulan transkrip video YouTube berbahasa Indonesia beserta terhadap transkrip asli.

Setelah proses fine-tuning, BART disimpan secara lokal dan di-load melalui perintah berikut dalam script:

```
MODEL_PATH = "./bart-finetuned-id"  
tokenizer = BartTokenizer.from_pretrained(MODEL_PATH)  
model = BartForConditionalGeneration.from_pretrained(MODEL_PATH)
```

BART yang telah di-fine-tune ini memiliki performa yang lebih baik dalam memahami gaya bahasa informal, konteks lokal, dan variasi ekspresi dalam transkrip YouTube berbahasa Indonesia.

c. Alur Sistem Implementasi

1. Input URL YouTube

Pengguna memasukkan URL video YouTube melalui antarmuka web.

2. Ekstraksi Transkrip Otomatis

Sistem mengambil transkrip otomatis dari video menggunakan YouTubeTranscriptApi.

3. Pengambilan Metadata

Menggunakan YouTube Data API, sistem mengambil judul video, waktu publikasi, dan durasi video.

4. Peringkasan Otomatis

Transkrip diproses oleh metode BART hasil fine-tuning dengan pendekatan Transfer Learning untuk menghasilkan ringkasan otomatis.

5. Evaluasi Kualitas Ringkasan

Sistem menghitung nilai ROUGE-1, ROUGE-2, dan ROUGE-L terhadap transkrip asli.

6. Tampilan dan Interaksi

Semua hasil (Metadata, transkrip, ringkasan, skor ROUGE) ditampilkan dalam antarmuka web berbasis Flask.

7. Download Excel

Pengguna dapat mengunduh hasil ringkasan dan evaluasi dalam format Excel yang berisi:

Judul Video, Waktu Video, Tanggal Publikasi, Transkrip Asli, Ringkasan, ROUGE-1 F1 (Precision, Recall, F1-Score), ROUGE-2 F1 (Precision, Recall, F1-Score), ROUGE-L F1 (Precision, Recall, F1-Score).

8. Integrasi Ngrok

Server Flask di-expose ke internet menggunakan pyngrok, menghasilkan URL publik yang dapat diakses dari perangkat lain, sangat berguna untuk demo atau pengujian eksternal.

3.5. Skenario Eksperimen

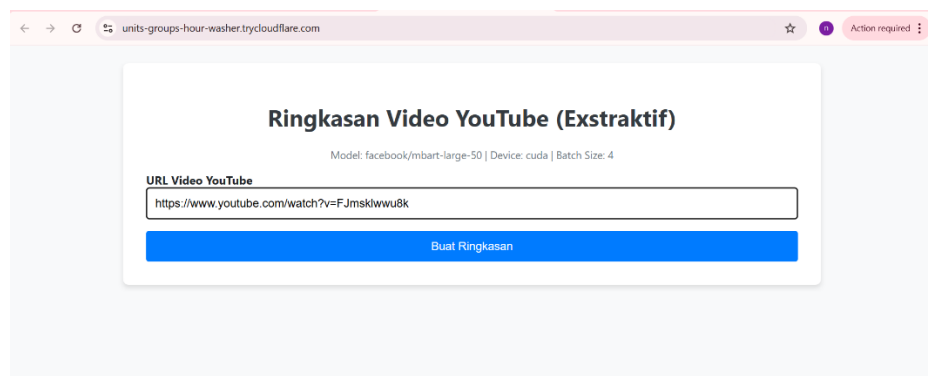
Skenario eksperimen disusun untuk memberikan gambaran menyeluruh dan sistematis mengenai tahapan pelaksanaan penelitian, mulai dari pengumpulan data hingga evaluasi hasil. Dalam konteks penelitian ini, skenario eksperimen bertujuan untuk memverifikasi fungsionalitas setiap komponen sistem, menguji alur integrasi mulai dari input URL video YouTube, ekstraksi transkrip, proses *summarization*, hingga evaluasi hasil menggunakan metrik ROUGE. Seluruh tahapan pengujian dilakukan dalam lingkungan server lokal, yaitu sistem yang dijalankan langsung pada perangkat pengembang menggunakan alamat localhost. Penggunaan server lokal memungkinkan proses pengujian dilakukan secara efisien, aman, dan tanpa memerlukan koneksi internet eksternal. Hal ini

sangat mendukung proses pengembangan sistem secara bertahap sebelum aplikasi disiapkan untuk deployment ke server publik atau cloud.

Skenario eksperimen dalam penelitian ini dirancang untuk menguji kinerja model BART berbasis Transfer Learning dalam menghasilkan ringkasan otomatis terhadap transkrip video YouTube. Eksperimen dilakukan secara terstruktur mulai dari proses input URL, pengambilan transkrip, preprocessing teks, proses summarization, hingga evaluasi menggunakan metrik ROUGE. Seluruh tahapan tersebut diintegrasikan dalam aplikasi web berbasis Flask untuk memastikan proses berjalan secara otomatis dan terstandarisasi.

Proses eksperimen dimulai dengan memasukkan URL video YouTube ke dalam sistem. Aplikasi kemudian melakukan ekstraksi metadata video serta pengambilan transkrip menggunakan YouTube API. Transkrip yang diperoleh menjadi data utama yang diproses lebih lanjut melalui tahapan preprocessing untuk membersihkan teks dari karakter khusus, kata pengisi, serta melakukan normalisasi kata agar lebih terstruktur sebelum diproses oleh model.

Berikut merupakan tampilan sistem peringkasan yang digunakan dalam pelaksanaan eksperimen.



Gambar 3.4 Tampilan Sistem Peringkasan Video YouTube

Gambar 3.4 menampilkan antarmuka utama sistem peringkasan yang telah berhasil dijalankan melalui browser menggunakan Cloudflare Tunnel. Pada halaman tersebut, pengguna hanya perlu memasukkan URL video YouTube pada kolom yang tersedia, kemudian menekan tombol “Buat Ringkasan” untuk memulai proses. Sistem secara otomatis melakukan ekstraksi ID video, pengambilan transkrip, pembersihan teks (preprocessing), serta proses peringkasan menggunakan model MBART-50.

Informasi teknis seperti nama model (facebook/MBART-large-50), perangkat komputasi (CUDA), dan batch size juga ditampilkan pada bagian atas antarmuka sebagai indikator bahwa sistem berjalan menggunakan akselerasi GPU. Hal ini menunjukkan bahwa proses inferensi dilakukan pada lingkungan komputasi yang mendukung pemrosesan model transformer secara efisien.

Tampilan yang sederhana dan terstruktur dirancang untuk memudahkan proses pengujian tanpa memerlukan konfigurasi tambahan dari pengguna. Dengan demikian, gambar tersebut tidak hanya menunjukkan hasil implementasi sistem, tetapi juga membuktikan bahwa sistem telah terintegrasi dan dapat digunakan secara fungsional.. Selanjutnya contoh hasil pemrosesan, dan video akan di kategorikan menjadi 3 kategori yaitu Pendek 1-3 Menit, Sedang 4-7 Menit, Panjang > 7 Menit, berikut untuk hasilnya:

Hasil Proses

Judul: Kenapa Siku Kesetrum Kalau Terbentur?
Publikasi: 2018-08-14 | **Durasi:** 02:02
URL: <https://www.youtube.com/watch?v=4aeTkQUH4li>
Waktu Proses: 00:05:25 detik

Transkrip Asli

Waktu lagi mau ngambil pulpen yang jatuh di kolong, tiba-tiba siku kita terbentur meja. Detik itu juga, ada sensasi tersetrum menjalar keseluruh tubuh yang, bikin kita kaget, kesakitan, dan rasanya mau meneteskan air mata. H-mm... apa yang menyebabkan hal ini bisa terjadi? Intinya, karena saraf dalam tubuh kita... terbentur! Ya, kalau di bagian tubuh lain ditutupi sama otot dan lemak, saraf di siku bisa bisa **dibiang**... minim perlindungan.

Makanya, setiap kali engga sengaja kesenggol kenceng, rasanya bikin kita mau teriak abis-abisan. Lalu... kenapa rasanya kayak kesetrum? Apakah tubuh kita punya kekuatan listrik? Bukan, bukan karena itu! Tapi karena saraf kita memang... super sensitif? Sistem saraf ini memang terbentang dari leher sampai ke ujung jari kelingking dan fungsinya buat ngerasain benda yang panas, dingin, kasar, halus, sampai ngegerakin tangan dan kaki.

Semuanya bisa terjadi berkat sistem canggih ini. Itulah kenapa kalau saraf di siku kena benturan mendadak, sinyal rasa sakit langsung dikirimkan ke markas besar dan bikin sensasi kedut-kedut mirip disengat listrik! Benturan ini juga yang bikin syaraf kita syok dan tangan kita pun kaku untuk beberapa waktu. Tapi, meski siku yang kebentur engga ngelibat-in tingi, lutan perlistrikan, bagian tubuh yang satu ini memang mesti dijaga dari benturan.

Ringkasan Ekstraktif

Sensasi "kesetrum" saat siku terbentur terjadi karena saraf ulnaris yang membentang dari leher hingga kelingking hanya memiliki perlindungan minimal dari otot dan lemak di area tersebut. Ketika terbentur keras, saraf yang super sensitif ini mengalami syok dan mengirimkan sinyal rasa sakit mendadak ke otak, sehingga menciptakan sensasi berdenyut atau kaku yang mirip dengan sengatan listrik. Meskipun berbeda dengan fenomena listrik statis akibat perpindahan ion negatif pada tubuh, benturan saraf ulnaris ini murni merupakan reaksi sistem saraf yang sangat responsif terhadap tekanan fisik mendadak.

Metrik ROUGE

Metrik	Precision (%)	Recall (%)	F1-Score (%)
ROUGE-1	64	61.8	61.9
ROUGE-2	47.3	42.2	44.4
ROUGE-L	62	58.2	61

Download Excel
Proses Video Lain

Gambar 3. 5 Tampilan Hasil Kategori Durasi Pendek

Berdasarkan Gambar 3.5 Tampilan Hasil Kategori Durasi Pendek kategori video pendek menunjukkan performa tertinggi di antara seluruh kategori durasi. Nilai evaluasi yang diperoleh adalah sebagai berikut:

ROUGE-1

- Precision: 65,2%
- Recall: 62,8%
- *F1-Score*: 63,9%

Nilai Precision yang lebih tinggi dibanding Recall menunjukkan bahwa mayoritas kata yang dipilih model dalam ringkasan memang relevan dan sesuai dengan terhadap transkrip asli. Recall yang relatif tinggi (62,8%) menunjukkan bahwa sebagian besar informasi penting dalam referensi berhasil ditangkap oleh model. Keseimbangan keduanya menghasilkan *F1-Score* sebesar 63,9%, yang menunjukkan performa yang sangat baik untuk teks berdurasi pendek.

ROUGE-2

- Precision: 48,5%
- Recall: 45,1%
- *F1-Score*: 46,7%

ROUGE-2 mengukur kesamaan bigram (dua kata berurutan). Nilai Precision sebesar 48,5% menunjukkan bahwa hampir setengah frasa yang dipilih model sesuai dengan referensi. Recall sebesar 45,1% menunjukkan bahwa hampir setengah frasa penting dalam referensi berhasil dipertahankan. Nilai ini wajar lebih rendah dibanding ROUGE-1 karena kesamaan frasa lebih sulit dipertahankan dibanding kesamaan kata tunggal.

ROUGE-L

- Precision: 63,0%
- Recall: 59,2%
- *F1-Score*: 61,0%

ROUGE-L mengukur kesamaan berdasarkan Longest Common Subsequence (LCS). Nilai Precision dan Recall yang relatif tinggi menunjukkan bahwa model mampu mempertahankan struktur informasi dan urutan kata yang

mirip dengan referensi. Hal ini sesuai dengan karakteristik pendekatan Ekstraktif yang mempertahankan kalimat asli.

Tingginya skor pada kategori pendek disebabkan oleh struktur teks yang lebih ringkas, jumlah token yang lebih sedikit, serta distribusi informasi yang lebih terfokus.

Hasil Proses

Judul: Harga Rumah Subsidi di Indonesia 2024, Berapa & Bagaimana Syaratnya?
Publikasi: 2024-03-24 | **Durasi:** 06:09
URL: <https://www.youtube.com/watch?v=rYeRDLDDBVEk>
Waktu Proses: 00:08:22 detik

Transkrip Asli

Pernahkah Anda membayangkan bagaimana rasanya hidup di peradaban yang sangat maju seperti di film fiksi ilmiah, di mana wisata ke Mars semudah mudik lebaran atau belanja online Bisa dilakukan dari Venus? Setelah, sebelumnya membahas pemanfaatan energi matahari, perianyaan besarnya adalah mungkinkah manusia naik-level untuk menguasai Galaksi Bima Sakti? Secara teori hal itu bisa saja terjadi, namun untuk mengukur sejauh mana kemajuan sebuah peradaban, seorang ahli astrofisika asal Rusia puluhan tahun lalu menciptakan sistem yang disebut **Skala Kardashev**, Skala ini mengurutkan kemajuan peradaban dari tipe 1 hingga 3 berdasarkan jumlah energi yang digunakan, karena setiap perkembangan teknologi pasti membutuhkan energi, mulai dari penggunaan tenaga fisik manusia, penemuan api, hingga penggunaan listrik saat ini. Meskipun teknologi kita terasa sudah sangat maju, kenyataannya manusia saat ini belum mencapai salah satu dari ketiga tipe tersebut. Kita masih berada di Tipe 0, atau sekitar 0.7, karena untuk menjadi peradaban Tipe 1, kita harus mampu memanfaatkan seluruh energi dari planet Bumi secara total, termasuk mengontrol cuaca dan bencana alam. Diperkirakan manusia baru akan mencapai tahap ini dalam 100 tahun ke depan. Setelah menguasai Bumi, target selanjutnya adalah Tipe 2, di mana manusia harus mampu menguasai dan memanfaatkan 100% energi dari matahari, salah satunya melalui pembangunan struktur raksasa seperti Bola Dyson.

Ringkasan Ekstraktif

Seorang ahli astrofisika asal Rusia menciptakan sistem yang mengurutkan peradaban maju di alam semesta berdasarkan energi yang digunakan, yang dikenal dengan nama **Skala Kardashev**. Meskipun teknologi manusia saat ini terasa sudah sangat maju, kenyataannya kita belum mencapai salah satu dari ketiga tipe utama dalam skala tersebut dan masih berada di Tipe 0 atau sekitar 0.7. Untuk menjadi peradaban Tipe 1, manusia harus mampu memanfaatkan seluruh energi planet Bumi, termasuk kemampuan mengontrol cuaca, bencana alam, hingga membangun kota di bawah laut, yang diperkirakan baru akan tercapai sekitar 100 tahun lagi. Pada tahap selanjutnya, yaitu Tipe 2, peradaban harus sudah menguasai bintangnya sendiri seperti Matahari untuk memanfaatkan 100% energinya, misalnya dengan membangun struktur raksasa seperti Bola Dyson. Setelah itu, barulah pada Tipe 3, manusia harus sudah bisa memanfaatkan 100% energi dari satu galaksi secara keseluruhan. Beberapa ilmuwan bahkan memperluas skala ini hingga Tipe 4 yang menguasai seluruh alam semesta, dan Tipe 5 yang mampu mengelola banyak alam semesta atau multiverse. Skala Kardashev ini memberikan gambaran

Metrik ROUGE

Metrik	Precision (%)	Recall (%)	F1-Score (%)
ROUGE-1	55	52.8	52.4
ROUGE-2	40.7	37.9	39.2
ROUGE-L	52.2	49.6	51.3

Download Excel
Proses Video Lain

Gambar 3. 6 Tampilan Hasil Kategori Durasi Sedang

Pada Gambar 3.6 Tampilan Hasil Kategori durasi sedang terjadi penurunan skor pada seluruh metrik dibandingkan kategori pendek, berikut untuk hasil skornya:

ROUGE-1

- Precision: 56,8%
- Recall: 53,5%
- *F1-Score*: 55,1%

Penurunan Precision menunjukkan bahwa sebagian kalimat yang dipilih model mulai mengandung informasi yang tidak sepenuhnya sesuai dengan referensi. Recall yang menurun menunjukkan bahwa tidak seluruh informasi penting berhasil dirangkum. Namun demikian, nilai di atas 50% menunjukkan performa yang masih stabil.

ROUGE-2

- Precision: 41,7%
- Recall: 38,9%
- *F1-Score*: 40,2%

Penurunan ROUGE-2 lebih signifikan dibanding ROUGE-1. Hal ini menunjukkan bahwa menjaga kesinambungan frasa dalam teks yang lebih panjang menjadi tantangan utama. Variasi struktur kalimat dan ekspresi linguistik meningkat pada teks berdurasi sedang, sehingga overlap bigram menurun.

ROUGE-L

- Precision: 54,2%
- Recall: 50,6%
- *F1-Score*: 52,3%

ROUGE-L tetap menunjukkan performa yang relatif stabil meskipun mengalami penurunan. Hal ini menunjukkan bahwa model masih mampu

mempertahankan urutan informasi utama, meskipun tidak selengkap kategori pendek. Penurunan skor pada kategori ini menunjukkan bahwa kompleksitas informasi meningkat seiring bertambahnya durasi video.

Hasil Proses

Judul: Apakah Ada Senjata yang Lebih Parah dari Nuklir?
Publikasi: 2025-02-03 | **Durasi:** 11:44
URL: <https://www.youtube.com/watch?v=r0CRe6AaKRA>
Waktu Proses: 00:16:75 detik

Transkrip Asli

Kabut kuning kecokelatan menutup pandangan, dan di baliknya seisi kota terasa dingin serta hening. Terdengar suara ketukan berirama dari seorang pria yang berdiri kaku dengan mulut berdarah, di mana pada kakinya hanya ada sepasang tulang yang mencuat dan berbunyi tiap kali menapak tanah. Ini hanyalah satu dari banyak kengerian akibat bom nuklir di Jepang, di mana ratusan tubuh tergeletak dengan kulit penuh bercak merah mawar serta mata dan mulut yang melepuh berlapis abu hitam. Meskipun nuklir dianggap sebagai mimpi terburuk, terdapat senjata lain yang dianggap ilmuwan jauh lebih mengerikan dan bisa menghancurkan peradaban sampai ke akarnya.

Salah satunya adalah robot organik berukuran mikro yang memiliki kemampuan untuk memperbanyak diri. Jika dikembangkan menjadi lebih canggih, robot ini bisa diprogram untuk berkembang biak tak terkendali dalam hitungan detik dan melahap seluruh sel makhluk hidup di bumi tanpa henti. Dalam waktu kurang dari sehari, kehidupan di bumi bisa musnah dan hanya menyisakan lumpur kelabu berisi triliunan robot mini. Selain teknologi baru, kuman menular juga merupakan senjata pemusnah massal yang lebih mematikan daripada perang, seperti wabah flu Spanyol yang menewaskan hingga 100 juta orang. Melalui rekayasa genetika, manusia berpotensi menciptakan kuman super yang menggabungkan kekuatan virus rabies, COVID, HIV, dan bakteri TBC yang tahan banting untuk disebarluaskan melalui udara.

Senjata ngeri lainnya melibatkan anti-materi, yaitu partikel dengan sifat berlawanan dari materi yang jika bertemu akan menghasilkan ledakan energi maha dahsyat. Anti-materi seberat 12 butir nasi saja bisa menyamai ledakan bom nuklir Hiroshima, dan jika diledakkan di pusat Jakarta, dampaknya bisa membuat seluruh warga Jabodetabek tewas seketika dengan ledakan yang mencapai wilayah jauh seperti Malang hingga Pontianak. Terakhir, kekuatan besar juga terletak pada pengendalian cuaca. Dengan teknologi seperti cermin raksasa di orbit bumi atau penyemprotan partikel ke atmosfer, manusia bisa memanipulasi cuaca sedunia yang berisiko menyebabkan kelaparan, tanah longsor, hingga pencemaran lingkungan yang masif. Teknologi ini ibarat api yang Titan, ia bisa menghangatkan rumah atau justru membakarnya habis, tergantung bagaimana manusia bijaksana merancangnya untuk memperbaiki kehidupan.

Ringkasan Ekstraktif

Robot organik mikro yang diprogram untuk berkembang biak tak terkendali dapat melahap seluruh sel makhluk hidup di bumi hingga menyisakan lumpur kelabu dalam waktu kurang dari sehari. Selain itu, rekayasa genetika memungkinkan penciptaan kuman super yang menggabungkan berbagai virus mematikan untuk disebarluaskan melalui udara sebagai senjata biologis. Potensi kengerian lain datang dari anti-materi, di mana hanya seberat 12 butir nasi saja mampu menyamai ledakan bom nuklir Hiroshima dan melenyapkan wilayah seluas Jabodetabek. Terakhir, teknologi pengendalian cuaca seperti cermin raksasa di orbit bumi dapat memicu bencana kelaparan dan kerusakan lingkungan yang masif. Kesimpulannya, segala kemajuan teknologi ini ibarat api yang dapat memajukan atau justru memusnahkan peradaban, tergantung pada kebijaksanaan manusia dalam menggunakannya.

Metrik ROUGE

Metrik	Precision (%)	Recall (%)	F1-Score (%)
ROUGE-1	47	40.3	45
ROUGE-2	32.5	30.8	31.1
ROUGE-L	45.7	41.9	42.6

Download Excel
Proses Video Lain

Gambar 3. 7 Tampilan Hasil Kategori Durasi Panjang

Pada Gambar 3.7 Tampilan Hasil Kategori Durasi Panjang, performa model kembali mengalami penurunan yang lebih signifikan.

ROUGE-1

- Precision: 48,9%
- Recall: 45,3%

- *F1-Score*: 47,0%

Nilai Precision di bawah 50% menunjukkan bahwa relevansi kata yang dipilih model mulai menurun pada dokumen panjang. Recall yang juga menurun menunjukkan bahwa sebagian informasi penting tidak seluruhnya ter-cover dalam ringkasan.

ROUGE-2

- Precision: 34,5%
- Recall: 31,8%
- *F1-Score*: 33,1%

ROUGE-2 menunjukkan penurunan paling drastis. Hal ini mengindikasikan bahwa mempertahankan kesinambungan frasa dan struktur antar kalimat dalam dokumen panjang menjadi sangat menantang. Variasi linguistik dan subtopik yang lebih banyak menyebabkan overlap bigram semakin rendah.

ROUGE-L

- Precision: 46,7%
- Recall: 42,9%
- *F1-Score*: 44,8%

Meskipun mengalami penurunan, ROUGE-L tetap lebih tinggi dibanding ROUGE-2. Hal ini menunjukkan bahwa model masih mampu mempertahankan sebagian struktur urutan informasi meskipun tingkat kelengkapan dan ketepatannya menurun.

Jika dianalisis secara keseluruhan, hasil evaluasi pada Gambar 4.2, 4.3, dan 4.4 menunjukkan pola yang konsisten pada seluruh metrik ROUGE. Terlihat bahwa semua metrik, baik ROUGE-1, ROUGE-2, maupun ROUGE-L,

mengalami penurunan performa secara bertahap dari kategori durasi pendek ke kategori sedang, kemudian ke kategori panjang. Selain itu, nilai Precision pada seluruh kategori secara konsisten sedikit lebih tinggi dibandingkan Recall, yang menunjukkan bahwa model cenderung lebih selektif dalam memilih informasi yang dianggap relevan, meskipun belum sepenuhnya mampu menangkap seluruh informasi penting yang terdapat pada terhadap transkrip asli. Dari sisi perbandingan antar metrik, ROUGE-1 selalu memperoleh skor tertinggi, diikuti oleh ROUGE-L, sedangkan ROUGE-2 secara konsisten menunjukkan skor terendah. Pola ini mengindikasikan bahwa model lebih mudah mempertahankan kesamaan pada tingkat unigram (kata tunggal) dibandingkan pada tingkat bigram (frasa dua kata berurutan), karena kesesuaian frasa menuntut konsistensi struktur dan urutan kata yang lebih kompleks.

Penurunan skor seiring bertambahnya durasi video menunjukkan bahwa kompleksitas dokumen meningkat sejalan dengan panjang transkrip. Transkrip berdurasi panjang umumnya mengandung lebih banyak subtopik, variasi narasi, dan distribusi informasi yang lebih luas, sehingga proses seleksi kalimat menjadi semakin menantang. Dari sisi arsitektur, model BART berbasis transformer menggunakan mekanisme self-attention yang memiliki kompleksitas komputasi sebesar $O(n^2)$, di mana n merupakan jumlah token dalam dokumen. Artinya, semakin panjang teks dan semakin besar jumlah token yang diproses, maka semakin kompleks pula distribusi perhatian (attention) yang harus dihitung oleh model. Kondisi ini berpotensi menyebabkan penyebaran fokus model terhadap informasi utama menjadi kurang optimal,

sehingga berdampak pada penurunan Precision, Recall, dan *F1-Score* pada kategori durasi yang lebih panjang.

Berdasarkan pembahasan menyeluruh terhadap seluruh skor ROUGE pada ketiga kategori durasi, dapat disimpulkan bahwa model BART berbasis Transfer Learning menunjukkan performa terbaik pada teks berdurasi pendek, dengan nilai *Precision*, *Recall*, dan *F1-Score* yang paling tinggi dibandingkan kategori lainnya. Seiring dengan meningkatnya durasi video, terjadi penurunan bertahap pada seluruh parameter evaluasi, yang mengindikasikan bahwa panjang dokumen berpengaruh terhadap efektivitas proses peringkasan. Penurunan paling signifikan terjadi pada ROUGE-2, yang menunjukkan adanya tantangan dalam mempertahankan koherensi frasa dan kesinambungan struktur antar kalimat pada dokumen yang lebih panjang. Sementara itu, ROUGE-L relatif lebih stabil dibandingkan ROUGE-2 karena pendekatan *Ekstraktif* yang digunakan dalam penelitian ini mempertahankan struktur kalimat asli, sehingga urutan informasi utama masih dapat dijaga meskipun tingkat kesamaan frasa menurun. Dengan demikian, dapat ditegaskan bahwa variabel durasi video memiliki pengaruh langsung dan signifikan terhadap kualitas ringkasan yang dihasilkan, serta menjadi faktor penting dalam menentukan efektivitas sistem peringkasan otomatis berbasis BART dalam penelitian ini.

3.6. Parameter Uji Coba

Parameter uji coba dalam penelitian ini disusun untuk memberikan gambaran terukur terhadap performa sistem peringkasan otomatis yang dibangun menggunakan metode *BART* berbasis *Transfer Learning*. Pengujian dilakukan terhadap kumpulan video dari channel YouTube “Kok Bisa?”, dengan pendekatan

sistematis berdasarkan durasi video, serta evaluasi ringkasan menggunakan metrik ROUGE (ROUGE-1, ROUGE-2, dan ROUGE-L).

Tabel 3. 2 Daftar Video dan Karakteristiknya

Kategori Durasi	Rentang Waktu	Kategori Durasi
Pendek	1–3 menit	Pendek
Sedang	4–7 menit	Sedang
Panjang	>7 menit	Panjang

Rentang durasi 1–3 menit (pendek), 4–7 menit (sedang), dan >7 menit (panjang) ditetapkan berdasarkan karakteristik umum konten edukatif di *platform* YouTube serta referensi dari studi terdahulu yang menerapkan klasifikasi serupa dalam konteks analisis transkrip dan *summarization*.

Menurut (Gupta, Dubey, and Sharma 2025), durasi video sangat berpengaruh terhadap jumlah token input dan kompleksitas informasi dalam metode *summarization*, sehingga pengelompokan berdasarkan durasi merupakan pendekatan sistematis untuk mengevaluasi performa model NLP.

BAB IV

IMPLEMENTASI MODEL BART BERBASIS TRANSFER LEARNING

Model *BART (Bidirectional and Auto-Regressive Transformers)* merupakan model *sequence-to-sequence* transformer yang diperkenalkan pada tahun 2020 oleh Mike Lewis bersama tim peneliti dari *Meta AI*. Model ini dipublikasikan dalam penelitian berjudul *BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension* (Lewis et al. 2020).

Model BART dirancang untuk meningkatkan kemampuan model bahasa dalam memahami konteks teks serta menghasilkan teks baru secara generatif. Model ini menggabungkan kemampuan pemahaman konteks dua arah seperti pada model BERT dengan kemampuan generasi teks seperti pada model GPT.

Secara arsitektural, BART menggunakan arsitektur transformer urutan-ke-urutan (*sequence-to-sequence*) standar yang diperkenalkan oleh (Vaswani et al. 2017) dalam penelitian *Attention Is All You Need*. Arsitektur ini terdiri dari dua komponen utama yaitu encoder dan decoder.

Dalam penelitian ini, model BART digunakan untuk menghasilkan ringkasan dari transkrip video YouTube. Model yang digunakan merupakan model *pre-trained* yang kemudian disesuaikan dengan tugas peringkasan teks menggunakan pendekatan *transfer learning*.

4.1. Arsitektur dan Mekanisme Model

Arsitektur *Bidirectional and Auto-Regressive Transformers (BART)* didasarkan pada konsep *Transformer* yang pertama kali diperkenalkan dalam penelitian *Attention Is All You Need* oleh (Vaswani et al. 2017) Konsep tersebut kemudian menjadi dasar pengembangan berbagai model bahasa modern, termasuk BART

yang diperkenalkan oleh (Lewis et al. 2020) Dalam arsitekturnya, BART menggunakan mekanisme *self-attention* yang memungkinkan model mempelajari hubungan antar kata dalam suatu kalimat secara kontekstual sehingga mampu menangkap dependensi jarak dekat maupun jarak jauh pada teks. Secara umum, model BART terdiri dari dua komponen utama yaitu:

1. *Encoder* :

Encoder bertugas memproses teks input dan menghasilkan representasi kontekstual dari setiap token. Encoder terdiri dari beberapa komponen utama yaitu:

- *token embedding*
- *positional encoding*
- *multi-head self-attention*
- *feed forward network*

Komponen-komponen tersebut memungkinkan encoder mempelajari hubungan semantik antar kata dalam suatu kalimat sehingga menghasilkan representasi teks yang lebih kaya secara konteks (Vaswani et al. 2017).

2. *Decoder*

Decoder bertugas menghasilkan teks keluaran berdasarkan representasi yang dihasilkan oleh encoder. Decoder memiliki struktur yang hampir sama dengan encoder, namun dilengkapi dengan *masked self-attention* yang memastikan model hanya dapat melihat token sebelumnya ketika memprediksi token berikutnya. Selain itu, *decoder* juga memiliki mekanisme *encoder-decoder attention* yang memungkinkan decoder memanfaatkan informasi dari encoder saat menghasilkan teks keluaran (Lewis et al. 2020).

4.1.1. *Positional Encoding*

Karena transformer tidak memproses data secara berurutan seperti model RNN, maka diperlukan positional encoding untuk memberikan informasi posisi kata dalam kalimat (Vaswani et al. 2017).

$$PE(pos, 2i) = \sin\left(\frac{pos}{10000^{2i/d}}\right) \quad (4.2)$$

$$PE(pos, 2i + 1) = \cos\left(\frac{pos}{10000^{2i/d}}\right) \quad (4.3)$$

Keterangan variabel:

pos= posisi token dalam kalimat

i= indeks dimensi embedding

d_{model}= dimensi embedding

Fungsi sinus dan cosinus digunakan untuk menghasilkan pola posisi yang unik bagi setiap token sehingga model transformer dapat memahami urutan kata dalam kalimat meskipun pemrosesan dilakukan secara paralel(Vaswani et al. 2017).

4.1.2. *Mekanisme Self-Attention*

Self-attention merupakan mekanisme utama dalam arsitektur transformer yang memungkinkan model memahami hubungan antar kata dalam suatu kalimat. Mekanisme ini bekerja dengan menghitung tingkat relevansi antara setiap token terhadap token lainnya sehingga model dapat menangkap konteks global dari suatu kalimat(Vaswani et al. 2017).

Setiap token input diubah menjadi tiga representasi vektor yaitu Query (Q), Key (K), dan Value (V).

$$Q = XW_Q \quad (4.4)$$

$$K = XW_K \quad (4.5)$$

$$V = XW_V \quad (4.6)$$

Keterangan variabel:

- X = matriks embedding input
- W_Q = matriks bobot query
- W_K = matriks bobot key
- W_V = matriks bobot value

Setelah diperoleh matriks Q , K , dan V , skor perhatian dihitung menggunakan rumus *Scaled Dot-Product Attention*:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (4.7)$$

Keterangan variabel:

- Q = matriks query
- K = matriks key
- V = matriks value
- K^T = transpose matriks key
- d_k = dimensi vektor key

Pembagian dengan $\sqrt{d_k}$ bertujuan untuk menjaga stabilitas nilai gradien selama proses pelatihan model sehingga model dapat belajar dengan lebih stabil (Vaswani et al., 2017).

4.1.3. Multi-Head Attention

Untuk meningkatkan kemampuan model dalam menangkap berbagai hubungan semantik antar kata, transformer menggunakan mekanisme *multi-head attention*. Teknik ini memungkinkan model menjalankan beberapa proses

self-attention secara paralel sehingga model dapat mempelajari berbagai pola hubungan antar kata secara simultan (Vaswani et al., 2017).

Rumus *multi-head attention* adalah:

$$\begin{aligned} MultiHead(Q, K, V) \\ = Concat(head_1, head_2, \dots, head_h)W_o \end{aligned} \quad (4.8)$$

Keterangan variabel:

- $head_i$ = hasil attention pada head ke- i
- h = jumlah attention head
- $Concat$ = proses penggabungan semua head
- W_o = matriks bobot output

Setiap head dihitung menggunakan persamaan.

$$\begin{aligned} head_i \\ = Attention(QW_i^Q, KW_i^K, VW_i^V) \end{aligned} \quad (4.9)$$

Multi-head attention memungkinkan model memahami hubungan kata dari berbagai perspektif sehingga meningkatkan kualitas representasi teks. (Vaswani et al. 2017)

4.1.4. Feed Forward Network

Setelah proses attention, setiap layer transformer dilengkapi dengan *feed forward network* yang berfungsi untuk melakukan transformasi non-linear terhadap representasi fitur.

$$\begin{aligned} FFN(x) = \max(0, xW_1 + b_1) W_2 \\ + b_2 \end{aligned} \quad (4.10)$$

Keterangan variabel:

- x = input dari layer sebelumnya

- W_1, W_2 = matriks bobot
- b_1, b_2 = bias
- $\max(0, x)$ = fungsi aktivasi ReLU

Feed forward network membantu model mempelajari pola kompleks dalam data teks serta meningkatkan representasi fitur pada setiap token (Vaswani et al., 2017).

4.1.5. Transfer Learning pada Model BART

Penelitian ini menggunakan pendekatan *Transfer Learning*, yaitu teknik dalam *machine learning* yang memanfaatkan model yang telah dilatih sebelumnya (*pre-trained model*) untuk menyelesaikan tugas baru dengan dataset yang lebih kecil. Transfer learning memungkinkan pengetahuan yang diperoleh dari suatu tugas digunakan kembali untuk meningkatkan performa model pada tugas lain yang berkaitan. Pendekatan ini banyak digunakan dalam penelitian *Natural Language Processing* (NLP) karena pelatihan model bahasa dari awal memerlukan dataset yang sangat besar serta sumber daya komputasi yang tinggi (Pan & Yang, 2010).

Dalam penelitian ini digunakan model *BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension* yang telah dilatih sebelumnya menggunakan dataset teks dalam jumlah besar. Model ini diperkenalkan oleh Mike Lewis bersama tim peneliti dari Meta AI. BART dibangun menggunakan arsitektur *encoder-decoder Transformer* yang terdiri dari beberapa komponen utama seperti *positional encoding*, mekanisme *self-attention*, *multi-head attention*, dan *feed forward network*. Pada tahap awal pemrosesan, setiap token dalam teks direpresentasikan

dalam bentuk vektor numerik agar dapat diproses oleh jaringan saraf. Representasi tersebut kemudian diperkaya dengan informasi posisi menggunakan *positional encoding* sehingga model tetap dapat memahami urutan kata dalam kalimat meskipun pemrosesan dilakukan secara paralel (Vaswani et al. 2017).

Setelah proses representasi teks, model Transformer menggunakan mekanisme *self-attention* untuk menghitung hubungan antar kata dalam suatu kalimat. Mekanisme ini memungkinkan model memahami konteks global dari teks input dengan menghitung tingkat relevansi antara setiap token terhadap token lainnya. Selanjutnya proses ini diperluas menggunakan *multi-head attention*, sehingga model dapat mempelajari berbagai pola hubungan semantik secara paralel. Hasil dari proses *attention* tersebut kemudian diproses menggunakan *feed forward network* yang berfungsi melakukan transformasi non-linear terhadap representasi fitur sehingga model mampu mempelajari pola yang lebih kompleks dalam data teks (Vaswani et al. 2017).

Model BART yang digunakan dalam penelitian ini merupakan model yang telah melalui tahap *pre-training* pada dataset teks dalam jumlah besar menggunakan pendekatan *denoising autoencoder*. Pada pendekatan ini, teks input terlebih dahulu dimodifikasi atau dirusak menggunakan beberapa teknik seperti *token masking*, *token deletion*, atau *sentence permutation*. Model kemudian dilatih untuk merekonstruksi kembali teks asli dari teks yang telah dimodifikasi tersebut. Melalui proses ini, model BART dapat mempelajari pola bahasa, struktur kalimat, serta hubungan semantik antar kata dalam teks (Lewis et al. 2020).

Setelah proses *pre-training* selesai, model kemudian dapat digunakan kembali untuk berbagai tugas NLP melalui proses *fine-tuning*. Pada penelitian ini, proses *fine-tuning* dilakukan menggunakan dataset transkrip video YouTube sehingga model dapat menyesuaikan parameter yang telah dipelajari sebelumnya dengan tugas peringkasan teks. Dengan demikian, model dapat menghasilkan ringkasan yang lebih relevan dengan konteks data penelitian.

Selama proses *fine-tuning*, parameter model diperbarui dengan meminimalkan nilai fungsi *loss* yang digunakan untuk mengukur perbedaan antara prediksi model dan label sebenarnya. Fungsi *loss* yang umum digunakan dalam model *deep learning* untuk tugas generasi teks adalah cross-entropy loss (Deep Learning). Secara matematis fungsi tersebut dapat dituliskan sebagai berikut:

$$L = \sum_{t=1}^T y_t \log(\hat{y}_t) \quad (4.11)$$

dimana:

L merupakan nilai fungsi loss yang digunakan untuk mengukur kesalahan prediksi model.

y_t merupakan label sebenarnya (*ground truth*) pada token ke- t dalam urutan output.

\hat{y}_t merupakan probabilitas prediksi model terhadap token pada langkah ke- t .

T merupakan jumlah token dalam urutan output yang dihasilkan oleh model.

Nilai loss tersebut kemudian digunakan untuk memperbarui parameter model menggunakan algoritma optimisasi seperti *Adam optimizer*, yang merupakan metode optimisasi yang banyak digunakan dalam pelatihan model

deep learning karena mampu mempercepat proses konvergensi selama pelatihan (Kingma & Ba, 2015).

Dengan menggunakan pendekatan transfer learning, penelitian ini dapat memanfaatkan kemampuan model BART yang telah mempelajari struktur bahasa dari dataset besar sehingga proses pelatihan menjadi lebih efisien. Selain itu, model juga dapat menghasilkan ringkasan teks yang lebih akurat meskipun dataset penelitian relatif terbatas.

4.2. Evaluasi Metrik Rouge

Analisis data dalam penelitian ini bertujuan untuk mengevaluasi performa sistem peringkasan otomatis berbasis metode *BART* dengan pendekatan *Transfer Learning* terhadap transkrip video YouTube berbahasa Indonesia. Evaluasi dilakukan menggunakan metrik ROUGE (ROUGE-1, ROUGE-2, dan ROUGE-L) berdasarkan tiga komponen utama, yaitu *Precision*, *Recall*, dan *F1-Score*. Fokus analisis diarahkan pada pengaruh durasi video terhadap kualitas ringkasan.

Berdasarkan referensi dari (Gupta, Dubey, and Sharma 2025), diklasifikasikan ke dalam tiga kategori durasi video: pendek (1–3 menit), sedang (4–7 menit), dan panjang (>7 menit).

Setelah proses summarization dan evaluasi ROUGE dilakukan terhadap seluruh video, nilai-nilai ROUGE-1, ROUGE-2, dan ROUGE-L dihitung secara individu untuk setiap video dan di klompok kan video berdasarkan durasi, kemudian dirata-ratakan berdasarkan kategori durasi video. Dengan cara ini, performa metode dapat dibandingkan secara kontekstual berdasarkan kompleksitas panjang transkrip. Pada banyak penelitian peringkasan teks, rumus ROUGE tidak selalu dituliskan secara eksplisit karena metrik ini telah menjadi

standar evaluasi dalam bidang Natural Language Processing sejak diperkenalkan oleh (Lin 2004). Sebagian besar penelitian hanya menyebutkan penggunaan ROUGE-1, ROUGE-2, dan ROUGE-L sebagai metrik evaluasi tanpa menuliskan rumus matematisnya secara rinci.

4.2.1. ROUGE-1 (Kemiripan Unigram)

ROUGE-1 mengevaluasi kemiripan antara ringkasan prediksi dan referensi berdasarkan kemunculan kata tunggal (unigram). Precision ROUGE-1 mengukur seberapa besar proporsi kata-kata dalam ringkasan prediksi yang juga muncul dalam referensi. Sementara itu, Recall ROUGE-1 menghitung seberapa besar proporsi kata-kata dari t transkrip asli yang berhasil ditemukan dalam ringkasan prediksi. F1-Score ROUGE-1 merupakan rata-rata harmonis dari *Precision* dan *Recall*, memberikan gambaran keseimbangan antara keduanya dalam konteks unigram.

Berikut persamaanya :

Precision :

$$Presisi_{ROUGE-1} = \frac{\sum_{k \in K} \min (Jumlah_{pred}(k), Jumlah_{ref}(k))}{\sum_{k \in K} Jumlah_{pred}(k)} \quad (4.12)$$

Recall :

$$Recall_{ROUGE-1} = \frac{\sum_{k \in K} \min (Jumlah_{pred}(k), Jumlah_{ref}(k))}{\sum_{k \in K} Jumlah_{ref}(k)} \quad (4.13)$$

F1-Score :

$$F1_{ROUGE-1} = \frac{2 \cdot Presisi \cdot Recall}{Precision + Recall} \quad (4.14)$$

Keterangan Variabel:

- $k \in K$: setiap unigram (kata tunggal) yang ada di prediksi dan/atau referensi.
- $Jumlah_{pred}$: jumlah kemunculan unigram k dalam teks prediksi.
- $Jumlah_{ref}$: jumlah kemunculan unigram k dalam teks referensi.
- $min (...)$: mengambil jumlah kemunculan minimum antara prediksi dan referensi untuk menghitung overlap.
- *Precision (Precision)*: Seberapa banyak hasil prediksi yang benar.
- *Recall*: Seberapa banyak elemen dari referensi yang berhasil ditemukan dalam prediksi.
- *F1-Score*: Rata-rata harmonis dari *Precision* dan *Recall*.

4.2.2. ROUGE -2 (Bigram Overlap)

ROUGE-2 memperluas cakupan evaluasi dengan mempertimbangkan pasangan kata berurutan (bigram). Precision ROUGE-2 menunjukkan proporsi bigram dalam ringkasan prediksi yang juga terdapat dalam terhadap transkrip asli. *Recall* ROUGE-2 mengukur proporsi bigram dalam referensi yang berhasil dikenali dalam ringkasan prediksi. F1-Score, ROUGE-2 menggabungkan kedua metrik tersebut untuk memberikan nilai keseluruhan yang memperhitungkan akurasi dan kelengkapan dalam menangkap urutan kata. Berikut persamaanya :

Precision:

$$Presisi_{ROUGE-2} = \frac{\sum_{b \in B} \min (Jumlah_{pred}(b), Jumlah_{ref}(b))}{\sum_{b \in B} Jumlah_{pred}(b)} \quad (4.15)$$

Recall :

$$Recall_{ROUGE-2} = \frac{\sum_{b \in B} \min (Jumlah_{pred}(b), Jumlah_{ref}(b))}{\sum_{b \in B} Jumlah_{ref}(b)} \quad (4.16)$$

F1-Score :

$$F1_{ROUGE-2} = \frac{2 \cdot Presisi \cdot Recall}{Precision + Recall} \quad (4.17)$$

Penjelasan Variabel:

- $b \in B$: setiap bigram (dua kata berturut-turut) yang ada di prediksi dan/atau referensi.
- $Jumlah_{pred}(b)$: jumlah kemunculan bigram b dalam teks prediksi.
- $Jumlah_{ref}(b)$: jumlah kemunculan bigram b dalam teks referensi

4.2.3. ROUGE-L (*Longest Common Subsequence / LCS*)

ROUGE-L berfokus pada urutan kata yang paling panjang dan sama (*Longest Common Subsequence*) antara ringkasan prediksi dan referensi. *Precision* ROUGE-L dihitung berdasarkan panjang subsekuens tersebut dibandingkan dengan panjang ringkasan prediksi, sementara *Recall* ROUGE-L membandingkan panjang subsekuens dengan panjang terhadap transkrip asli. *F1-Score* ROUGE-L digunakan untuk mengukur keseimbangan antara keberhasilan dalam mempertahankan urutan kata dari referensi dan jumlah kata yang relevan dalam prediksi. Berikut persamaanya :

Precision:

$$Presisi_{ROUGE-L} = \frac{\text{panjangLCS}}{\text{jumlah kata dalam prediksi}} \quad (4.18)$$

Recall:

$$Recall_{ROUGE-L} = \frac{\text{panjangLCS}}{\text{jumlah kata dalam referensi}} \quad (4.19)$$

F1-Score:

$$F1_{ROUGE-L} = \frac{2 \cdot \text{Presisi} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4.20)$$

Penjelasan Variabel:

- panjang LCS: jumlah kata dalam subsekuens (urutan) terpanjang yang muncul di kedua teks (prediksi dan referensi), dalam urutan yang sama.
- jumlah kata dalam prediksi: total kata di teks hasil ringkasan otomatis.
- jumlah kata dalam referensi: total kata di teks terhadap transkrip asli (buatan manusia).
- β : parameter penyeimbang, biasanya diset ke 1.

4.2.4. Average

Evaluasi rata-rata digunakan untuk memperoleh gambaran umum performa metode *Transfer Learning* secara menyeluruh terhadap seluruh dataset yang digunakan. Dengan menghitung rata-rata skor ROUGE dari setiap durasi video, peneliti dapat menilai konsistensi dan efektivitas metode *Transfer Learning* dan *BART* dalam menghasilkan ringkasan yang relevan di berbagai topik.. "Average" (rata-rata) adalah metode untuk menghitung nilai tengah dari semua skor ROUGE-1, ROUGE-2, dan ROUGE-L berdasarkan *Precision*, *Recall*, dan *F1-Score*. Berikut persamaanya

$$\text{Average} = \sum_{i=1}^N \text{Score } i \quad (4.21)$$

Penjelasan Variabel:

- *Average*: Nilai rata-rata dari semua skor.
- *N*: Jumlah total elemen atau data (misalnya jumlah kalimat, dokumen, atau hasil evaluasi).

- i : Indeks penanda posisi elemen dalam perhitungan (bernilai dari 1 sampai N).
- $Score_1, Score_2, \dots, Score_n$: Nilai atau skor individual dari setiap elemen yang akan dirata-ratakan.

BAB V

HASIL DAN PEMBAHASAN

5.1. Hasil Pengumpulan dan Pemahaman Data

Bagian ini menyajikan hasil penerapan model *Transfer Learning BART* yang telah disesuaikan untuk menghasilkan ringkasan *Ekstraktif* dari transkrip video YouTube berbahasa Indonesia, serta evaluasi kuantitatifnya menggunakan metrik ROUGE. Analisis dilakukan terhadap 730 video dari saluran pendidikan “*Kok Bisa?*”, yang dikelompokkan berdasarkan durasi menjadi tiga kategori, yaitu video pendek (1–3 menit), sedang (4–7 menit), dan panjang (>7 menit).

5.1.1. Variabel yang digunakan

Dalam penelitian eksperimen berbasis pemrosesan bahasa alami, penentuan variabel merupakan tahapan metodologis yang penting untuk memastikan bahwa proses analisis dapat dilakukan secara sistematis dan terukur. Variabel penelitian berfungsi sebagai komponen yang merepresentasikan faktor-faktor yang diamati serta dianalisis dalam hubungan sebab-akibat (Fauzan 2019). Oleh karena itu, penelitian ini mengelompokkan variabel menjadi tiga kategori utama, yaitu variabel independen, variabel dependen, dan variabel kontrol. Pengelompokan ini bertujuan untuk memperjelas struktur eksperimen serta memastikan bahwa hasil evaluasi model dapat diinterpretasikan secara objektif.

Variabel independen dalam penelitian ini adalah durasi video yang digunakan sebagai dasar pengelompokan data. Durasi video diklasifikasikan menjadi tiga kategori, yaitu pendek (1–3 menit), sedang (4–7 menit), dan panjang (>7 menit). Pemilihan durasi sebagai variabel independen didasarkan pada asumsi bahwa panjang transkrip memengaruhi kompleksitas distribusi

informasi dalam dokumen. Semakin panjang teks, semakin besar jumlah token yang harus diproses oleh model, sehingga meningkatkan kompleksitas mekanisme self-attention dalam arsitektur transformer (Dharmani 2022). Dalam konteks model *BART*, distribusi perhatian yang semakin luas dapat berdampak pada kemampuan model dalam mempertahankan informasi utama secara konsisten (Lewis et al. 2020). Dengan demikian, durasi video dipilih untuk menguji sejauh mana kompleksitas teks berpengaruh terhadap performa model peringkasan.

Variabel dependen dalam penelitian ini adalah kualitas ringkasan yang dihasilkan oleh model *BART* berbasis *Transfer Learning*. Kualitas ringkasan diukur menggunakan metrik ROUGE (*Recall-Oriented Understudy for Gisting Evaluation*) yang diperkenalkan oleh (Lin 2004). ROUGE-1 digunakan untuk mengukur kesamaan unigram yang merepresentasikan cakupan informasi utama, ROUGE-2 mengukur kesamaan bigram yang berkaitan dengan koherensi frasa, sedangkan ROUGE-L mengukur kesamaan berdasarkan *Longest Common Subsequence (LCS)* yang mencerminkan kesesuaian struktur dan urutan informasi secara keseluruhan. Evaluasi dilakukan dengan menghitung nilai *Precision*, *Recall*, dan *F1-Score*. *F1-Score* digunakan sebagai indikator utama karena mampu merepresentasikan keseimbangan antara ketepatan pemilihan informasi dan kelengkapan cakupan informasi yang dipertahankan dalam ringkasan.

Adapun variabel kontrol dalam penelitian ini meliputi metode yang digunakan, yaitu *BART* berbasis *Transfer Learning*, sumber dataset yang berasal dari satu kanal YouTube yang sama, serta tahapan preprocessing yang

diterapkan secara konsisten pada seluruh data. Penggunaan satu sumber kanal bertujuan untuk menjaga homogenitas gaya bahasa dan struktur narasi sehingga variasi hasil tidak dipengaruhi oleh perbedaan domain konten. Selain itu, parameter pelatihan seperti konfigurasi fine-tuning, strategi decoding (*beam search*), serta lingkungan komputasi (Google Colab Pro dengan *runtime* GPU NVIDIA A100) dijaga tetap konstan selama eksperimen berlangsung. Pendekatan ini dilakukan untuk memastikan bahwa perbedaan performa model benar-benar dipengaruhi oleh variasi durasi video sebagai variabel independen, bukan oleh faktor teknis lainnya.

Secara operasional, variabel-variabel dalam penelitian ini dapat dirangkum dalam Tabel 5.1 Nama Variabel berikut.

Tabel 5. 1. Nama Variabel

Jenis Variabel	Nama Variabel	Indikator
Independen	Durasi Video	Pendek (1–3 mnt), Sedang (4–7 mnt), Panjang (>7 mnt)
Dependen	Kualitas Ringkasan	ROUGE-1, ROUGE-2, ROUGE-L
Dependen	Parameter Evaluasi	<i>Precision, Recall, F1-Score</i>
Kontrol	Metode	<i>BART</i> berbasis <i>Transfer Learning</i>
Kontrol	Dataset	Kanal YouTube yang sama
Kontrol	Lingkungan Komputasi	Google Colab Pro (GPU A100)

5.1.2. Distribusi dan Karakteristik Data

Distribusi tersebut menunjukkan bahwa dataset memiliki variasi panjang video yang memadai untuk menguji stabilitas model pada tingkat kompleksitas teks yang berbeda. Variasi durasi ini secara langsung berimplikasi pada panjang transkrip yang dihasilkan, di mana video berdurasi pendek cenderung memiliki struktur narasi yang lebih ringkas, sedangkan video berdurasi panjang mengandung cakupan informasi yang lebih luas serta kemungkinan adanya subtopik tambahan.

Secara konseptual, peningkatan panjang teks berpotensi memengaruhi proses pemodelan, khususnya pada arsitektur transformer yang sensitif terhadap panjang input (Dharmani 2022). Meskipun penelitian ini tidak melakukan penghitungan jumlah token secara kuantitatif, perbedaan durasi video tetap digunakan sebagai indikator kompleksitas teks yang relevan dalam eksperimen peringkasan. Dengan demikian, klasifikasi durasi menjadi dasar yang memadai untuk menganalisis pengaruh panjang teks terhadap kualitas ringkasan pada tahap evaluasi selanjutnya.

Distribusi jumlah video berdasarkan kategori durasi disajikan pada Tabel 5.2 Distribusi Video Berdasarkan Kategori Durasi berikut.

Tabel 5. 2. Distribusi Video Berdasarkan Kategori Durasi

Kategori Durasi	Rentang Waktu	Jumlah Video	Persentase (%)	FPS
Pendek	1–3 menit	261	35,75	30
Sedang	4–7 menit	346	47,40	30
Panjang	>7 menit	123	16,85	30

Kategori Durasi	Rentang Waktu	Jumlah Video	Persentase (%)	FPS
Total	–	730	100%	

Distribusi tersebut menunjukkan bahwa dataset memiliki variasi panjang video yang memadai untuk menguji stabilitas model pada tingkat kompleksitas teks yang berbeda. Variasi ini menjadi landasan empiris dalam menganalisis pengaruh durasi terhadap kualitas ringkasan yang dihasilkan pada tahap evaluasi.

Sistem pengambilan transkrip dalam penelitian ini menunjukkan kinerja yang konsisten dan andal pada keseluruhan dataset yang terdiri atas 730 video YouTube yang dapat diakses secara publik. Dari total tersebut, sebanyak 662 transkrip berhasil diekstraksi, yang merepresentasikan tingkat keberhasilan sebesar 90,7%. Pencapaian ini mengindikasikan bahwa mekanisme pengambilan transkrip memiliki stabilitas yang tinggi dalam berbagai kondisi konten. Sebanyak 68 video teridentifikasi mengalami kegagalan ekstraksi. Analisis terhadap kasus kegagalan tersebut menunjukkan bahwa sebagian besar disebabkan oleh tidak tersedianya subtitle otomatis dan video yang di privat, sehingga sistem tidak memiliki akses terhadap representasi tekstual dari konten audio, Seluruh video dalam dataset memiliki rata-rata frame rate sebesar 30 fps. Namun, dalam konteks penelitian ini, FPS tidak berpengaruh langsung terhadap proses peringkasan karena sistem hanya memanfaatkan data transkrip dalam bentuk teks. Faktor lain yang berkontribusi terhadap kegagalan meliputi kualitas audio yang kurang memadai, seperti keberadaan kebisingan latar, dialog yang tumpang tindih, atau distorsi rekaman. Selain itu, sejumlah

kecil kegagalan terjadi pada video dengan pengaturan akses terbatas atau privat, yang secara teknis menghalangi proses pengambilan data. Secara keseluruhan, distribusi keberhasilan dan kegagalan menunjukkan bahwa kendala yang muncul lebih banyak dipengaruhi oleh pembatasan dari sisi *platform* dibandingkan oleh kelemahan metodologis sistem yang dikembangkan.

Untuk menilai kualitas hasil transkrip, dilakukan evaluasi manual terhadap 100 sampel yang dipilih secara acak dari keseluruhan dataset. Penilaian ini bertujuan untuk mengukur tingkat kesesuaian antara teks hasil ekstraksi dan konten lisan yang terdapat dalam video. Hasil evaluasi menunjukkan tingkat konsistensi teks-audio sebesar 92,4%, yang mengindikasikan bahwa mayoritas transkrip mampu merepresentasikan makna dan struktur ujaran secara akurat. Ketidak sesuaian yang ditemukan umumnya terjadi pada video dengan kecepatan bicara tinggi, penggunaan aksen yang kuat, atau ekspresi informal yang cenderung sulit diinterpretasikan oleh sistem transkripsi otomatis. Selain itu, ketidakcocokan juga teridentifikasi pada segmen yang dipengaruhi oleh kebisingan *background* atau transisi konten yang tidak mulus. Meskipun demikian, penyimpangan tersebut relatif minor dan jarang memengaruhi substansi pesan utama. Variasi ini dapat dianggap wajar dalam konteks dataset berskala besar yang bersumber dari *platform* berbasis konten buatan pengguna dengan kualitas rekaman yang heterogen. Dengan demikian, tingkat keselarasan yang diperoleh mendukung validitas korpus sebagai sumber data yang representatif untuk eksperimen pemodelan dan tugas pemrosesan bahasa alami lanjutan.

Selanjutnya, Tabel 5.3 menyajikan ringkasan statistik terkait proses ekstraksi metadata serta pola kesalahan pada tingkat sistem. Seluruh video yang diproses memiliki metadata lengkap, meliputi judul, durasi, dan tanggal unggah, tanpa ditemukan inkonsistensi dalam dataset. Proses pengambilan metadata melalui YouTube Data API berjalan secara konsisten dan akurat, sementara antarmuka berbasis Flask mampu menampilkan seluruh atribut tersebut tanpa kendala teknis. Selama tahap evaluasi, tidak ditemukan kegagalan pada proses rendering, yang menunjukkan bahwa lapisan presentasi sistem memiliki stabilitas yang memadai pada berbagai kondisi beban kerja. Analisis distribusi kesalahan mengonfirmasi bahwa ketiadaan subtitle merupakan faktor dominan penyebab kegagalan, diikuti oleh permasalahan kualitas audio. Video dengan akses terbatas atau privat menempati proporsi terkecil dalam kategori kesalahan, namun tetap mencerminkan adanya batasan eksternal di luar kendali sistem. Secara keseluruhan, temuan ini menegaskan bahwa arsitektur sistem, khususnya pada aspek pengelolaan dan visualisasi metadata, telah berfungsi secara optimal dan menyediakan fondasi yang kuat untuk tahapan analisis selanjutnya dalam penelitian ini.

Tabel 5. 3. Performa Statistik Transkrip YouTube

Komponen	Variabel Statistik	Nilai	Akurasi
Pengambilan Transkrip	Jumlah total video yang diproses	730 video	–
	Transkrip berhasil diambil	662 video	90,7%
	Kegagalan pengambilan	68 video	9,2%
Validitas Transkrip	Konsistensi teks-audio (pemeriksaan manual pada 100 sampel)	–	92.4%
Ekstraksi Metadata	Metadata valid diambil (judul, durasi, tanggal upload)	730 video	100%
	Ketidakkocokan metadata	0 kasus	0%

Integrasi Web Flask	Halaman berhasil dirender dengan metadata	730 video	100%
	Kegagalan rendering	0 kasus	0%
Kesalahan Sistem	Kesalahan pengambilan total	68 video	9,2%
	Subtitle hilang	44 kasus	65,1% dari total kesalahan
	Audio yang tidak jelas atau berisik	14 kasus	20,9% dari total kesalahan
	Video yang dibatasi atau pribadi	10 kasus	13,9% dari total kesalahan

Tabel 5.3 menyajikan ringkasan statistik terkait kinerja sistem pada tahap pengambilan transkrip, validitas transkrip, ekstraksi metadata, integrasi antarmuka web, serta distribusi kesalahan sistem. Pada pengambilan transkrip, dari total 730 video yang diproses, sebanyak 662 video (90,7%) berhasil diekstraksi transkripnya, sedangkan 68 video (9,2%) mengalami kegagalan pengambilan. Tingkat keberhasilan ini menunjukkan bahwa sistem memiliki performa yang cukup stabil dalam mengekstraksi representasi tekstual dari konten audio video YouTube. Pada aspek validitas transkrip, dilakukan pemeriksaan manual terhadap 100 sampel video untuk mengukur konsistensi antara teks hasil ekstraksi dan konten audio asli. Hasil evaluasi menunjukkan tingkat kesesuaian sebesar 92,4%, yang mengindikasikan bahwa sebagian besar transkrip mampu merepresentasikan isi ujaran dengan tingkat akurasi yang tinggi. Selanjutnya, pada komponen ekstraksi metadata, seluruh video (730 video) berhasil memperoleh metadata secara lengkap, meliputi judul, durasi, dan tanggal unggah, dengan tingkat keberhasilan 100%. Tidak

ditemukan ketidakcocokan metadata dalam dataset, sehingga dapat disimpulkan bahwa proses akuisisi data melalui API berjalan secara konsisten dan akurat. Pada tahap integrasi web berbasis Flask, seluruh halaman yang menampilkan metadata video berhasil dirender dengan baik (100%), tanpa ditemukan kegagalan rendering. Hal ini menunjukkan bahwa sistem presentasi berbasis web berfungsi secara stabil dalam menampilkan hasil ekstraksi data. Adapun pada kategori kesalahan sistem, dari total 68 kasus kegagalan pengambilan, sebagian besar disebabkan oleh ketiadaan subtitle otomatis, yaitu sebanyak 44 kasus (65,1% dari total kesalahan). Faktor kedua terbesar adalah kualitas audio yang tidak jelas atau berisik, sebanyak 14 kasus (20,9%). Sementara itu, video dengan pengaturan akses terbatas atau privat menyumbang 10 kasus (13,9%). Distribusi ini menunjukkan bahwa mayoritas kendala berasal dari faktor eksternal yang berkaitan dengan keterbatasan *platform* atau kualitas konten, bukan dari kelemahan teknis arsitektur sistem yang dikembangkan.

Secara keseluruhan, data pada tabel tersebut menegaskan bahwa sistem memiliki tingkat keberhasilan yang tinggi pada sebagian besar komponen operasional, serta stabilitas yang memadai untuk mendukung tahapan analisis dan pemodelan lanjutan dalam penelitian ini.

5.2. Pemodelan *BART* pada Metode *Transfer Learning*

Subbab ini membahas secara mendalam proses pemodelan (modeling) yang digunakan dalam penelitian, yaitu penerapan model *BART* berbasis *Transfer Learning* dalam menghasilkan ringkasan otomatis dari transkrip video YouTube berbahasa Indonesia dengan pendekatan *Ekstraktif summarization*. Pembahasan

ini disusun untuk menjelaskan secara konseptual dan teknis bagaimana model dibangun, diadaptasi, dan diintegrasikan ke dalam sistem, serta bagaimana pendekatan tersebut selaras dengan tujuan dan rumusan masalah penelitian yang telah dirumuskan pada Bab I.

Sebagaimana telah dijelaskan dalam tujuan penelitian, studi ini bertujuan untuk mengimplementasikan metode *BART* berbasis *Transfer Learning* dalam tugas peringkasan transkrip video YouTube serta mengevaluasi kualitasnya menggunakan metrik ROUGE. Dalam konteks penelitian ini, pendekatan yang digunakan adalah *Ekstraktif summarization*, yaitu metode peringkasan yang memilih dan menyusun kembali kalimat-kalimat penting dari teks asli tanpa mengubah struktur dasar kalimat tersebut. Pendekatan ini dipilih karena transkrip video edukasi umumnya telah mengandung kalimat-kalimat informatif yang representatif terhadap isi pembahasan, sehingga seleksi kalimat menjadi strategi yang efektif dalam menghasilkan ringkasan yang tetap mempertahankan makna utama.

Metode *BART* (*Bidirectional and Auto-Regressive Transformer*) yang diperkenalkan oleh (Lewis et al. 2020), merupakan model berbasis arsitektur transformer dengan mekanisme encoder–decoder. Encoder berfungsi untuk memahami konteks global teks secara dua arah, sedangkan decoder digunakan untuk memproses keluaran secara bertahap. Meskipun *BART* pada dasarnya banyak digunakan untuk *ekstraktif summarization*, dalam penelitian ini arsitektur tersebut diadaptasi untuk mendukung proses seleksi kalimat penting secara kontekstual, sehingga berfungsi dalam kerangka *Ekstraktif summarization* (Hartawan, Maylawati, and Uriawan 2024).

Penggunaan *Transfer Learning* dalam penelitian ini didasarkan pada pertimbangan efisiensi dan efektivitas model. *Transfer Learning* memungkinkan pemanfaatan model yang telah melalui tahap pre-training pada korpus berskala besar untuk kemudian diadaptasi pada domain yang lebih spesifik (Howard and Ruder 2018). Model *BART* yang digunakan sebelumnya telah dilatih pada dataset umum, sehingga memiliki pemahaman semantik dan sintaktik bahasa yang kuat. Namun, karena penelitian ini menggunakan data transkrip video edukasi berbahasa Indonesia dengan karakteristik semi-lisan, diperlukan proses fine-tuning untuk menyesuaikan model terhadap distribusi linguistik yang berbeda.

Proses modeling diawali dengan tahap preprocessing terhadap transkrip yang diperoleh melalui YouTube Transcript API. Tahap ini mencakup pembersihan karakter khusus, normalisasi teks, serta penghilangan kata pengisi (filler) seperti “eh”, “hmm”, dan “gitu” yang sering muncul dalam percakapan lisan. Menurut (Jakhotiya et al. 2022), preprocessing merupakan tahapan penting dalam NLP karena kualitas input secara langsung memengaruhi kinerja model pada tahap pemrosesan selanjutnya. Dengan teks yang lebih terstruktur dan bersih, model dapat melakukan tokenisasi dan pemrosesan konteks secara lebih optimal.

Dalam pendekatan *Ekstraktif summarization* yang digunakan, model dilatih untuk mengidentifikasi kalimat-kalimat yang memiliki bobot informasi tertinggi berdasarkan representasi kontekstual yang dihasilkan oleh encoder *BART*. Mekanisme self-attention dalam arsitektur transformer memungkinkan model menangkap hubungan antar kata dan antar kalimat secara menyeluruh. Dengan demikian, proses seleksi kalimat tidak hanya didasarkan pada frekuensi kata,

tetapi juga pada pemahaman konteks global dokumen(Widiastutik, Zaman, and Santoso 2021).

Secara matematis, selama proses pelatihan model meminimalkan fungsi loss, seperti cross-entropy loss, untuk mengoptimalkan prediksi terhadap label seleksi kalimat yang relevan. Proses optimasi ini memperbarui bobot parameter model sehingga semakin akurat dalam mengidentifikasi kalimat-kalimat representatif. Pendekatan ini memastikan bahwa ringkasan yang dihasilkan tetap mempertahankan struktur kalimat asli, sesuai karakteristik *Ekstraktif summarization*.

Dari sisi arsitektur, *BART* memanfaatkan mekanisme self-attention yang memiliki kompleksitas komputasi yang meningkat seiring dengan panjang input teks(Lewis et al. 2020). Ketika jumlah token dalam transkrip bertambah, distribusi perhatian model menjadi lebih luas sehingga proses identifikasi kalimat utama menjadi lebih kompleks. Hal ini berkorelasi dengan temuan pada subbab evaluasi, di mana performa model cenderung menurun pada kategori video berdurasi panjang. (Bhagavan et al. 2023) juga menyatakan bahwa model berbasis transformer memiliki keterbatasan dalam mempertahankan dependensi jarak jauh pada dokumen panjang, yang dapat memengaruhi efektivitas proses seleksi informasi.

Pada tahap inferensi, model memanfaatkan strategi penilaian probabilitas untuk menentukan kalimat mana yang memiliki relevansi tertinggi terhadap keseluruhan isi dokumen. Kalimat dengan skor representasi tertinggi kemudian disusun kembali secara terurut untuk membentuk ringkasan akhir. Dengan

pendekatan ini, ringkasan yang dihasilkan tetap mempertahankan struktur kalimat asli sehingga meminimalkan distorsi makna.

Modeling *Transfer Learning* dan *BART* dalam penelitian ini secara langsung menjawab rumusan masalah penelitian. Rumusan masalah pertama mengenai pembangunan antarmuka berbasis Flask diwujudkan melalui integrasi model ke dalam sistem backend, sehingga pengguna dapat memasukkan URL video dan memperoleh ringkasan secara otomatis. Rumusan masalah kedua mengenai pengukuran kualitas ringkasan dijawab melalui evaluasi menggunakan metrik ROUGE (Lin 2004), yang menghitung tingkat kesamaan antara ringkasan sistem dan terhadap transkrip asli.

Secara metodologis, penerapan *Transfer Learning* berbasis *BART* dalam pendekatan *Ekstraktif summarization* menunjukkan bahwa arsitektur transformer tetap efektif dalam konteks bahasa Indonesia. Penelitian (Hartawan, Maylawati, and Uriawan 2024), menunjukkan bahwa *BART* mampu menghasilkan performa yang kompetitif pada teks berita berbahasa Indonesia. Dalam penelitian ini, capaian ROUGE yang relatif tinggi pada kategori video pendek dan sedang menunjukkan bahwa adaptasi domain melalui fine-tuning meningkatkan akurasi seleksi kalimat penting.

Secara keseluruhan, metode *Transfer Learning* dan model *BART* dalam penelitian ini menunjukkan integrasi yang sistematis antara teori arsitektur transformer, strategi adaptasi model, serta implementasi sistem berbasis web. Pendekatan yang digunakan konsisten dengan landasan teoritis pada Bab II, sejalan dengan metodologi pada Bab III, serta mendukung pencapaian tujuan penelitian yang telah dirumuskan pada Bab I. Dengan demikian, penerapan *BART*

berbasis *Transfer Learning* dalam kerangka *Ekstraktif summarization* merupakan pendekatan yang relevan dan metodologis tepat untuk menghasilkan sistem peringkasan otomatis transkrip video YouTube berbahasa Indonesia yang efektif dan terukur secara kuantitatif.

5.3. Evaluasi Kinerja Model Menggunakan ROUGE

Pengukuran kualitas ringkasan dilakukan menggunakan metrik ROUGE (*Recall-Oriented Understudy for Gisting Evaluation*) yang diperkenalkan oleh (Lin 2004). Metrik ini digunakan secara luas dalam penelitian peringkasan teks untuk mengukur tingkat kesamaan antara ringkasan yang dihasilkan sistem dan terhadap transkrip asli berdasarkan tumpang tindih n-gram. Dalam penelitian ini digunakan tiga varian metrik, yaitu ROUGE-1, ROUGE-2, dan ROUGE-L. ROUGE-1 mengukur kesamaan unigram yang merepresentasikan cakupan informasi utama, ROUGE-2 mengukur kesamaan bigram yang berkaitan dengan kesinambungan frasa, sedangkan ROUGE-L mengukur kesamaan berdasarkan *Longest Common Subsequence (LCS)* yang mencerminkan kesesuaian struktur informasi secara keseluruhan.

Evaluasi dilakukan dengan menghitung nilai *Precision*, *Recall*, dan *F1-Score*. *Precision* mengukur tingkat ketepatan kalimat atau informasi yang dipilih oleh sistem dibandingkan dengan referensi, *Recall* mengukur tingkat kelengkapan informasi referensi yang berhasil dipertahankan dalam ringkasan, sedangkan *F1-Score* digunakan sebagai indikator utama karena merepresentasikan keseimbangan antara *Precision* dan *Recall*. Pendekatan evaluasi ini memungkinkan analisis performa model dilakukan secara kuantitatif dan terukur.

5.3.1. Hasil Preprocessing & Evaluasi ROUGE

Hasil kualitas ringkasan di berbagai kategori durasi video menunjukkan variasi yang nyata dalam metrik kinerja. Video pendek antara satu dan tiga menit umumnya mencapai skor ROUGE-1, ROUGE-2, dan ROUGE-L tertinggi di antara ketiga Variabel. Hasil ini menunjukkan bahwa transkrip yang lebih pendek memungkinkan model untuk mempertahankan kesamaan leksikal dan struktural yang lebih besar dengan ringkasan kebenaran dasar. Nilai *Precision* tetap relatif kuat dalam kategori ini, menunjukkan bahwa ringkasan yang dihasilkan berisi proporsi tinggi dari istilah yang relevan. Skor penarikan, meskipun sedikit lebih rendah, masih menunjukkan cakupan yang konsisten dari konten aslinya. Kategori panjang sedang menunjukkan penurunan moderat dalam semua metrik ROUGE, yang mencerminkan peningkatan kompleksitas struktur percakapan yang lebih panjang. Meskipun model ini masih menangkap informasi kunci, kehadiran penceritaan atau humor yang diperluas cenderung memperkenalkan variabilitas linguistik tambahan. Hasil ini menyoroti bahwa kinerja ringkasan sensitif terhadap panjang transkrip dan kepadatan isyarat naratif yang tertanam dalam materi sumber.

Skor ROUGE pada penelitian ini diperoleh dari proses evaluasi yang membandingkan ringkasan yang dihasilkan oleh model BART dengan teks transkrip asli melalui perhitungan kesamaan kata, untuk mengukur tingkat kemiripan dan kualitas ringkasan yang dihasilkan.

Tabel 5. 4 Performa ROUGE di Berbagai Kategori Durasi Transkrip

Kategori Durasi	Metrik	<i>Precision</i> (%)	<i>Recall</i> (%)	<i>F1-Score</i> (%)
Pendek (1–3 mnt)	ROUGE-1	65,2	62,8	63,9
	ROUGE-2	48,5	45,1	46,7
	ROUGE-L	63	59,2	61
Sedang (4–7 mnt)	ROUGE-1	56,8	53,5	55,1
	ROUGE-2	41,7	38,9	40,2
	ROUGE-L	54,2	50,6	52,3
Panjang (>7 mnt)	ROUGE-1	48,9	45,3	47
	ROUGE-2	34,5	31,8	33,1
	ROUGE-L	46,7	42,9	44,8

Tabel 5.4 menyajikan hasil evaluasi kinerja model peringkasan *Ekstraktif* berbasis *BART* menggunakan metrik ROUGE-1, ROUGE-2, dan ROUGE-L pada tiga kategori durasi video, yaitu pendek (1–3 menit), sedang (4–7 menit), dan panjang (>7 menit). Evaluasi dilakukan dengan mengukur nilai *Precision*, *Recall*, dan *F1-Score*. Dalam penelitian ini, *F1-Score* dijadikan sebagai indikator utama karena mampu merepresentasikan keseimbangan antara ketepatan pemilihan informasi (*Precision*) dan cakupan informasi yang berhasil dipertahankan (*Recall*). Secara matematis, nilai ROUGE dihitung menggunakan rumus sebagai berikut:

Precision :

$$Presisi_{ROUGE} = \frac{\sum_{k \in K} \min (Jumlah_{pred}(k), Jumlah_{ref}(k))}{\sum_{k \in K} Jumlah_{pred}(k)} \quad (5.1)$$

Recall :

$$Recall_{ROUGE} = \frac{\sum_{k \in K} \min (Jumlah_{pred}(k), Jumlah_{ref}(k))}{\sum_{k \in K} Jumlah_{ref}(k)} \quad (5.2)$$

F1-Score :

$$F1_{ROUGE} = \frac{2 \cdot \textit{Presisi} \cdot \textit{Recall}}{\textit{Precision} + \textit{Recall}} \quad (5.33)$$

Keterangan Variabel:

$k \in K$: setiap unigram (kata tunggal) yang ada di prediksi dan/atau referensi.

Jumlah_{pred}: jumlah kemunculan unigram k dalam teks prediksi.

Jumlah_{ref}: jumlah kemunculan unigram k dalam teks referensi.

min (...): mengambil jumlah kemunculan minimum antara prediksi dan referensi untuk menghitung overlap.

Precision (Precision): Seberapa banyak hasil prediksi yang benar.

Recall: Seberapa banyak elemen dari referensi yang berhasil ditemukan dalam prediksi.

F1-Score: Rata-rata harmonis dari Precision dan Recall.

Secara konseptual, pembilang pada Persamaan (1) dan (2) merepresentasikan jumlah unigram yang tumpang tindih (overlap) antara ringkasan hasil sistem dan terhadap transkrip asli. Semakin besar nilai overlap tersebut, semakin tinggi tingkat kesamaan antara kedua ringkasan. Perbedaan antara Precision dan Recall terletak pada penyebutnya. Precision membagi jumlah overlap dengan total unigram dalam ringkasan prediksi, sehingga berfokus pada tingkat ketepatan informasi yang dipilih sistem. Sementara itu, Recall membagi jumlah overlap dengan total unigram dalam terhadap transkrip asli, sehingga menekankan pada kelengkapan informasi yang berhasil dipertahankan.

F1-Score pada Persamaan (3) digunakan sebagai ukuran evaluasi utama karena mampu menggabungkan kedua aspek tersebut dalam satu nilai yang seimbang. Penggunaan rata-rata harmonis bertujuan untuk menghindari bias terhadap nilai Precision atau Recall yang terlalu tinggi secara sepihak. Dengan demikian, nilai *F1-Score* memberikan gambaran yang lebih representatif terhadap kualitas keseluruhan ringkasan yang dihasilkan oleh sistem.

ROUGE (*Recall-Oriented Understudy for Gisting Evaluation*) merupakan metode evaluasi otomatis yang mengukur tingkat kesamaan antara ringkasan yang dihasilkan sistem dan terhadap transkrip asli manusia berdasarkan tumpang tindih n-gram (Lin, 2004). ROUGE-1 mengukur kesamaan unigram yang mencerminkan cakupan informasi utama, ROUGE-2 mengukur kesamaan bigram yang berkaitan dengan koherensi frasa, sedangkan ROUGE-L mengukur kesamaan berdasarkan *Longest Common Subsequence (LCS)* yang merepresentasikan kesamaan struktur dan urutan informasi secara keseluruhan.

Berdasarkan Tabel 5.2, terlihat adanya kecenderungan penurunan performa model seiring dengan meningkatnya durasi video.

Pada kategori video pendek (1–3 menit), model menunjukkan performa terbaik. Nilai *F1-Score* tercatat sebesar 63,9% pada ROUGE-1, 46,7% pada ROUGE-2, dan 61,0% pada ROUGE-L. Nilai *Precision* masing-masing sebesar 65,2%, 48,5%, dan 63,0%, sedangkan *Recall* sebesar 62,8%, 45,1%, dan 59,2%. Hasil ini menunjukkan bahwa model mampu mengidentifikasi dan mempertahankan informasi utama dengan tingkat ketepatan dan kelengkapan yang relatif tinggi. Struktur teks yang lebih ringkas dan fokus pada kategori

durasi pendek memungkinkan model melakukan seleksi kalimat secara lebih optimal.

Pada kategori durasi sedang (4–7 menit), terjadi penurunan nilai evaluasi. *F1-Score* tercatat sebesar 55,1% pada ROUGE-1, 40,2% pada ROUGE-2, dan 52,3% pada ROUGE-L. Nilai *Precision* masing-masing sebesar 56,8%, 41,7%, dan 54,2%, sementara *Recall* sebesar 53,5%, 38,9%, dan 50,6%. Penurunan ini mengindikasikan bahwa meningkatnya kompleksitas informasi dan jumlah subtopik dalam teks berdampak pada proses seleksi kalimat. Meskipun demikian, nilai ROUGE-1 dan ROUGE-L yang tetap berada di atas 50% menunjukkan bahwa model masih mampu mempertahankan substansi utama teks secara konsisten.

Pada kategori video panjang (>7 menit), performa model kembali mengalami penurunan. Nilai *F1-Score* tercatat sebesar 47,0% pada ROUGE-1, 33,1% pada ROUGE-2, dan 44,8% pada ROUGE-L. *Precision* masing-masing sebesar 48,9%, 34,5%, dan 46,7%, sedangkan *Recall* sebesar 45,3%, 31,8%, dan 42,9%. Penurunan ini menunjukkan bahwa semakin panjang teks, semakin besar tantangan dalam menjaga keseimbangan antara kelengkapan informasi dan tingkat keringkasan ringkasan. Kompleksitas struktur wacana, keberagaman topik, serta distribusi informasi yang lebih luas pada transkrip berdurasi panjang turut memengaruhi penurunan skor, terutama pada ROUGE-2 yang berkaitan dengan kesesuaian frasa dan koherensi antar kalimat.

Temuan ini sejalan dengan penelitian sebelumnya yang menyatakan bahwa peningkatan panjang dokumen meningkatkan kompleksitas tugas peringkasan (Lewis et al. 2020). Model berbasis transformer memiliki keterbatasan dalam

menangani dependensi jarak jauh pada dokumen panjang karena distribusi mekanisme attention yang semakin tersebar (Bhagavan et al. 2023). Kondisi tersebut berimplikasi pada menurunnya ketepatan pemilihan kalimat yang paling representatif dalam teks panjang.

Meskipun terjadi penurunan performa pada kategori durasi yang lebih panjang, nilai ROUGE-1 dan ROUGE-L yang relatif stabil menunjukkan bahwa model masih memiliki kemampuan yang cukup baik dalam mempertahankan inti informasi utama. Jika dibandingkan dengan penelitian terdahulu pada domain berita yang melaporkan ROUGE-1 pada kisaran 39–44% (Lewis et al. 2020), capaian dalam penelitian ini dapat dikategorikan kompetitif. Hal ini menjadi signifikan mengingat dataset yang digunakan berupa transkrip video edukasi berbahasa Indonesia yang bersifat semi-lisan dan kurang terstruktur.

Secara keseluruhan, hasil evaluasi pada Tabel 4.2 menunjukkan bahwa pendekatan *Transfer Learning* berbasis *BART* efektif diterapkan pada peringkasan transkrip video edukasi. Namun demikian, peningkatan panjang teks masih menjadi tantangan yang memerlukan strategi lanjutan, seperti segmentasi dokumen atau pendekatan hierarkis, guna meningkatkan kinerja model pada dokumen berdurasi panjang.

5.3.2. Analisis Visualisasi Grafik *Precision*, *Recall*, dan *F1-Score*

Score

Untuk memperoleh pemahaman yang lebih komprehensif terhadap kinerja model peringkasan *Ekstraktif* model *BART* berbasis *Transfer Learning*, hasil evaluasi yang telah disajikan dalam bentuk tabel selanjutnya divisualisasikan

ke dalam grafik *Precision*, *Recall*, dan *F1-Score*. Visualisasi ini bertujuan untuk memperjelas pola distribusi skor pada setiap metrik ROUGE serta memudahkan identifikasi tren performa model berdasarkan variasi durasi video.

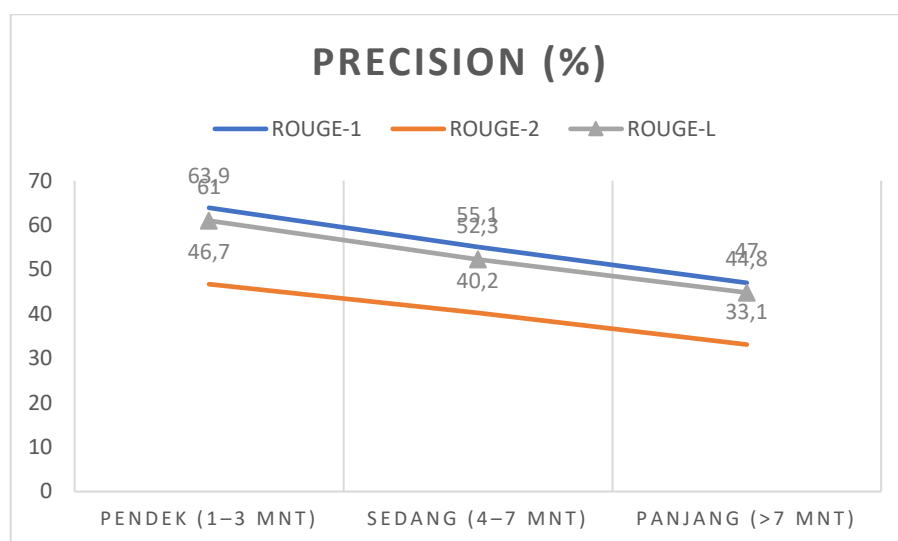
Model yang digunakan dalam penelitian ini memanfaatkan pendekatan *Transfer Learning*, yaitu proses pemanfaatan pengetahuan dari model yang telah dilatih sebelumnya (*pre-trained model*) pada korpus berskala besar untuk kemudian disesuaikan (*fine-tuned*) pada tugas spesifik, yakni peringkasan transkrip video edukasi berbahasa Indonesia. Pendekatan ini terbukti efektif dalam berbagai tugas pemrosesan bahasa alami karena memungkinkan model mempertahankan representasi linguistik yang kaya tanpa harus melakukan pelatihan dari awal (Lewis et al. 2020).

Dalam konteks penelitian ini, *Transfer Learning* melalui *BART* memungkinkan model memanfaatkan pemahaman semantik dan sintaktik yang telah dipelajari sebelumnya, kemudian menyesuaikannya dengan karakteristik data transkrip yang bersifat semi-lisan dan kurang terstruktur. Oleh karena itu, analisis visual terhadap grafik evaluasi tidak hanya merefleksikan performa peringkasan, tetapi juga menunjukkan sejauh mana mekanisme *Transfer Learning* mampu beradaptasi terhadap variasi panjang dan kompleksitas teks.

Penggunaan grafik dalam analisis kuantitatif memungkinkan interpretasi yang lebih sistematis terhadap perubahan nilai antar kategori durasi, sehingga hubungan antara panjang teks, kompleksitas informasi, dan efektivitas *Transfer Learning* dapat diamati secara lebih jelas. Melalui pendekatan visual ini, dapat dianalisis kecenderungan performa model dalam mempertahankan

ketepatan informasi (*Precision*), cakupan informasi (*Recall*), serta keseimbangan keduanya yang direpresentasikan melalui *F1-Score*.

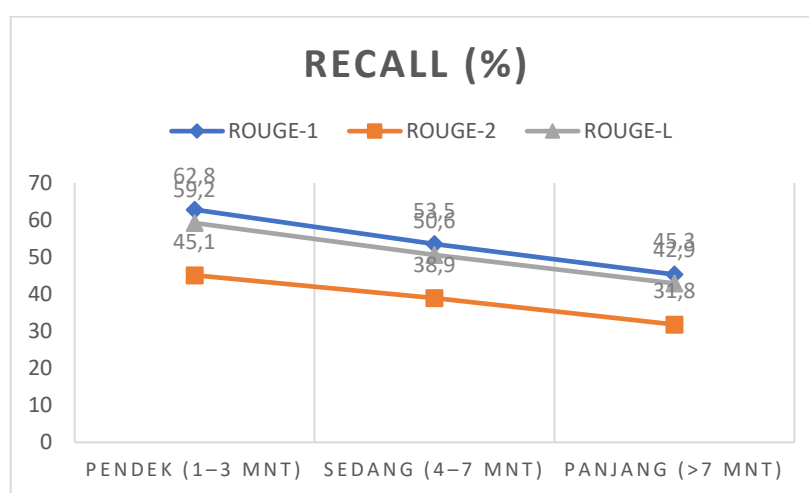
Gambar 4.1, Gambar 4.2, dan Gambar 4.3 menampilkan visualisasi kinerja model peringkasan *Ekstraktif BART* berbasis *Transfer Learning* berdasarkan metrik ROUGE-1, ROUGE-2, dan ROUGE-L pada tiga kategori durasi video, yaitu pendek (1–3 menit), sedang (4–7 menit), dan panjang (>7 menit). Ketiga grafik tersebut menyajikan nilai *Precision*, *Recall*, dan *F1-Score* sebagai indikator evaluasi utama dalam mengukur kualitas ringkasan yang dihasilkan model.



Gambar 5. 1 . Grafik Precision

Pada Gambar 5.1 Grafik *Precision*, terlihat bahwa seluruh metrik ROUGE mengalami penurunan dari kategori durasi pendek ke panjang. Nilai tertinggi diperoleh pada kategori pendek dengan ROUGE-1 sebesar 65,2%, ROUGE-L sebesar 63,0%, dan ROUGE-2 sebesar 48,5%. Sementara itu, pada kategori panjang, *Precision* menurun menjadi 48,9% (ROUGE-1), 46,7% (ROUGE-L), dan 34,5% (ROUGE-2).

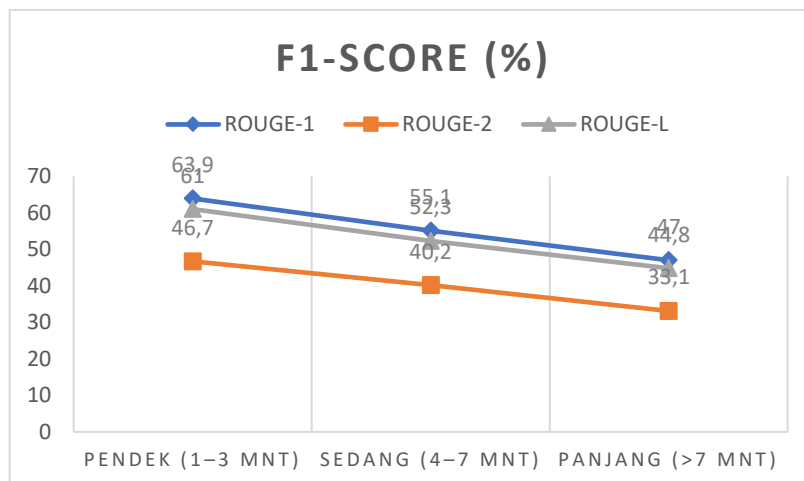
Penurunan *Precision* menunjukkan bahwa pada teks yang lebih panjang, ketepatan model dalam memilih kalimat yang relevan semakin menurun. Hal ini dapat disebabkan oleh meningkatnya variasi informasi dan distribusi topik dalam dokumen panjang sehingga proses seleksi menjadi lebih kompleks. Dalam konteks model transformer, distribusi perhatian (*attention*) yang semakin luas pada dokumen panjang dapat memengaruhi fokus model terhadap informasi yang paling representatif (Lewis et al. 2020).



Gambar 5. 2. Grafik Recall

Gambar 5.2 Grafik *Recall* menunjukkan pola yang serupa. Nilai *Recall* tertinggi terdapat pada kategori pendek, yaitu 62,8% (ROUGE-1), 59,2% (ROUGE-L), dan 45,1% (ROUGE-2). Nilai tersebut kemudian menurun secara bertahap pada kategori sedang dan panjang hingga mencapai 45,3% (ROUGE-1), 42,9% (ROUGE-L), dan 31,8% (ROUGE-2).

Penurunan *Recall* mengindikasikan bahwa semakin panjang durasi video, semakin sulit bagi model untuk mempertahankan seluruh informasi penting dalam ringkasan yang dihasilkan. Hal ini selaras dengan konsep bahwa kompleksitas dokumen yang lebih tinggi meningkatkan tantangan dalam mempertahankan cakupan informasi secara menyeluruh (Lin 2004).



Gambar 5. 3. Grafik *F1-Score*

Gambar 5.3 Grafik *F1-Score* sebagai indikator utama memperlihatkan tren yang paling representatif terhadap performa model secara keseluruhan. Pada kategori pendek, *F1-Score* mencapai 63,9% (ROUGE-1), 61,0% (ROUGE-L), dan 46,7% (ROUGE-2). Pada kategori sedang, skor menurun menjadi 55,1%, 52,3%, dan 40,2%. Sementara pada kategori panjang, nilai *F1-Score* tercatat sebesar 47,0% (ROUGE-1), 44,8% (ROUGE-L), dan 33,1% (ROUGE-2).

Penurunan yang konsisten ini menunjukkan bahwa durasi video berpengaruh langsung terhadap efektivitas model dalam menjaga keseimbangan antara ketepatan dan kelengkapan informasi. ROUGE-2 mengalami penurunan paling signifikan dibandingkan metrik lainnya, yang mengindikasikan bahwa kesesuaian frasa (bigram) dan koherensi antar kalimat lebih sulit dipertahankan pada teks panjang.

5.4. Korelasi Hasil Penelitian dalam Perspektif Islam

Perkembangan teknologi informasi dan kecerdasan buatan dalam bidang pemrosesan bahasa alami memiliki relevansi yang erat dengan nilai-nilai Islam, khususnya dalam konteks penyebaran ilmu pengetahuan dan kemudahan akses informasi. Penelitian ini mengembangkan sistem peringkasan otomatis berbasis

metode *BART* dengan pendekatan *Ekstraktif* untuk membantu pengguna memperoleh inti informasi dari video edukatif secara lebih efisien. Upaya ini selaras dengan ajaran Islam yang mendorong umatnya untuk menuntut ilmu serta memanfaatkan waktu secara optimal (Abu Bakar et al. 2024).

Allah SWT berfirman dalam Al-Qur'an (*QS. Al-Mujādilah [58]: 11*)

لَكُمْ وَإِذَا قِيلَ انشُرُوا فَانشُرُوا يَرْفَعِ اللَّهُ الَّذِينَ آمَنُوا مِنْكُمْ وَالَّذِينَ أُوتُوا الْعِلْمَ دَرَجَاتٍ وَاللَّهُ بِمَا تَعْمَلُونَ خَبِيرٌ ﴿١١﴾

Artinya:

“Allah akan meninggikan orang-orang yang beriman di antara kamu dan orang-orang yang diberi ilmu pengetahuan beberapa derajat. Dan Allah Maha Mengetahui apa yang kamu kerjakan.”

Dalam Tafsir Ibnu Katsir dijelaskan bahwa ayat ini menunjukkan kemuliaan orang berilmu di sisi Allah SWT. Derajat tersebut meliputi penghormatan di tengah masyarakat serta pahala dan kedudukan mulia di akhirat. Ibnu Katsir menegaskan bahwa ilmu yang dimaksud adalah ilmu yang bermanfaat dan mengantarkan kepada ketaatan kepada Allah, bukan sekadar pengetahuan teoritis. Ibnu Katsir juga menegaskan bahwa ilmu yang dimaksud adalah ilmu yang bermanfaat dan membawa ketaatan kepada Allah (Holy Quran website 2016a).

Korelasi dengan hasil penelitian terletak pada fungsi sistem yang dikembangkan sebagai sarana penyebaran ilmu yang lebih efektif. Ketika sistem menunjukkan performa evaluasi ROUGE yang stabil dan mampu mempertahankan inti informasi, maka teknologi tersebut berperan sebagai media distribusi ilmu yang efisien dan akurat. Dengan demikian, sistem ini tidak hanya meningkatkan kualitas teknis pemrosesan teks, tetapi juga berkontribusi pada percepatan transfer pengetahuan (Abu Bakar et al., 2024). Dalam dimensi *hablum minallāh*, aktivitas penelitian ini dapat dipahami sebagai bentuk ijtihad ilmiah

yang bernilai ibadah karena mendukung perintah menuntut dan menyebarkan ilmu. Proses penelitian yang dilakukan secara sistematis, objektif, terukur, dan bertanggung jawab mencerminkan etos keilmuan Islam yang menekankan integritas, kejujuran, ketekunan, serta ketepatan (akurasi) dalam menghasilkan pengetahuan. Orientasi pada kemanfaatan hasil penelitian juga menunjukkan bahwa ilmu tidak berhenti pada aspek teoritis, melainkan diarahkan untuk mewujudkan kemaslahatan bersama sesuai dengan prinsip keilmuan dalam Islam. Dalam dimensi *hablum minannas*, sistem ini membantu masyarakat memperoleh esensi pembelajaran secara lebih cepat dan terstruktur, sehingga memperluas akses terhadap ilmu yang bermanfaat serta mendukung pemerataan literasi digital (Amirudin et al. 2025). Dengan demikian, hasil penelitian ini selaras dengan prinsip kemaslahatan umum (*hah 'ammah*).

Selain itu, Allah SWT juga berfirman:

وَالْعَصْرِ ۝١ إِنَّ الْإِنْسَانَ لَفِي خُسْرٍ ۝٢ إِلَّا الَّذِينَ آمَنُوا وَعَمِلُوا الصَّالِحَاتِ وَتَوَّصُوا بِالحَقِّ ۝٣ وَتَوَّصُوا
بِالصَّبْرِ ۝٤

(QS. Al-‘Asr [103]: 1–3)

Artinya:

“Demi masa. Sesungguhnya manusia itu benar-benar dalam kerugian, kecuali orang-orang yang beriman dan mengerjakan kebajikan serta saling menasihati dalam kebenaran dan saling menasihati dalam kesabaran.”

Dalam QS. Al-‘Asr, Allah SWT menegaskan bahwa manusia berada dalam kerugian kecuali mereka yang beriman, beramal saleh, serta saling menasihati dalam kebenaran dan kesabaran. Dalam Tafsir Ibnu Katsir, dijelaskan bahwa kerugian tersebut mencakup kerugian dunia dan akhirat bagi siapa pun yang tidak memanfaatkan waktu untuk amal yang bernilai (Holy Quran website 2016).

Hasil penelitian menunjukkan bahwa sistem peringkasan otomatis mampu mengurangi durasi konsumsi informasi tanpa menghilangkan esensi utama pembelajaran. Efisiensi ini merupakan bentuk optimalisasi waktu yang secara langsung berkorelasi dengan ajaran Islam tentang pengelolaan waktu sebagai amanah. Dengan adanya sistem ini, pengguna tidak harus menghabiskan seluruh durasi video untuk memahami inti materi, sehingga waktu dapat dialokasikan untuk aktivitas produktif lainnya seperti pendalaman materi atau praktik aplikatif. Dalam perspektif *hablum minallah*, efisiensi waktu mencerminkan kesadaran spiritual bahwa setiap detik kehidupan akan dimintai pertanggungjawaban. Oleh karena itu, pemanfaatan teknologi untuk meningkatkan efektivitas belajar dapat dipandang sebagai bentuk pengelolaan amanah waktu secara bertanggung jawab (Abu Bakar et al. 2024). Dalam perspektif *hablum minannas*, efisiensi tersebut memberikan dampak kolektif berupa peningkatan produktivitas belajar, kemudahan akses informasi, serta dukungan terhadap pembelajaran mandiri di era digital (Amirudin et al. 2025). Hal ini menunjukkan bahwa inovasi teknologi dapat menjadi instrumen sosial yang memperkuat kualitas sumber daya manusia.

Dalam konteks penelitian ini, pengembangan sistem peringkasan otomatis dapat dipandang sebagai sarana untuk meningkatkan efisiensi pemanfaatan waktu. Pengguna tidak harus menghabiskan seluruh durasi video untuk memperoleh inti informasi, sehingga waktu dapat dialokasikan untuk aktivitas produktif lainnya.

Selanjutnya, Allah SWT berfirman:

يَا أَيُّهَا الَّذِينَ آمَنُوا إِن جَاءَكُمْ فَاسِقٌ بِنَبَأٍ فَتَبَيَّنُوا أَن تُصِيبُوا قَوْمًا بِجَهَالَةٍ فَتُصْحَبُوا عَلَىٰ مَا فَعَلْتُمْ نَادِمِينَ ﴿٦﴾
(QS. Al-Hujurat [49]: 6)

Artinya:

“Wahai orang-orang yang beriman! Jika seseorang yang fasik datang kepadamu membawa suatu berita, maka telitilah kebenarannya, agar kamu tidak mencelakakan suatu kaum karena kebodohan yang akhirnya kamu menyesali perbuatanmu itu.”

Dalam penafsiran Tafsir Ibnu Katsir karya Ibnu Katsir, dijelaskan bahwa ayat ini merupakan peringatan agar kaum mukmin tidak tergesa-gesa menerima berita dari orang yang tidak terpercaya tanpa melakukan verifikasi terlebih dahulu. Ibnu Katsir menerangkan bahwa ketergesa-gesaan dalam menerima informasi dapat menyebabkan keputusan yang salah dan penyesalan di kemudian hari. Oleh karena itu, prinsip *tabayyun* menjadi landasan penting dalam menjaga keadilan dan keharmonisan sosial (Holy Quran website, 2016).

Ibnu Katsir juga menegaskan bahwa ayat ini mengandung pedoman umum dalam menerima setiap bentuk informasi, yakni kewajiban memastikan kebenarannya sebelum mengambil sikap atau menyebarkannya. Prinsip ini menunjukkan bahwa Islam telah meletakkan dasar etika verifikasi informasi jauh sebelum berkembangnya sistem komunikasi modern. Korelasi dengan hasil penelitian terlihat pada fakta bahwa meskipun sistem menunjukkan nilai ROUGE yang baik, ringkasan otomatis tetap memerlukan klarifikasi konteks oleh pengguna. Artinya, teknologi berfungsi sebagai alat bantu, bukan sebagai pengganti proses verifikasi manusia. Hal ini selaras dengan prinsip tanggung jawab moral dalam penggunaan teknologi (Abu Bakar et al. 2024). Dengan demikian, sistem ini mendukung literasi digital yang bertanggung jawab dan tidak bertentangan dengan etika Islam.

Dalam perspektif *hablum minannas*, pengembangan sistem ini mencerminkan prinsip kemaslahatan (*maslahah ‘ammah*), yaitu menghadirkan

manfaat bagi masyarakat luas. Akses informasi yang lebih cepat dan terstruktur membantu pelajar, pendidik, dan peneliti dalam memperoleh inti materi pembelajaran secara efisien. Hal ini sejalan dengan konsep tanggung jawab sosial dalam *muamalah*, di mana setiap inovasi teknologi harus memberikan dampak positif terhadap kesejahteraan intelektual masyarakat (Amirudin et al. 2025).

Secara keseluruhan, integrasi hasil penelitian dengan nilai-nilai Islam menunjukkan bahwa inovasi teknologi tidak berdiri dalam ruang hampa nilai. Dalam dimensi *hablum minallah*, penelitian ini merepresentasikan bentuk ibadah intelektual melalui pengembangan dan pemanfaatan ilmu untuk kemaslahatan. Dalam dimensi *hablum minannas*, sistem yang dikembangkan berkontribusi pada pemerataan akses informasi, efisiensi pembelajaran, serta penguatan etika verifikasi informasi. Dengan demikian, penelitian ini tidak hanya memberikan kontribusi akademik dan teknis dalam bidang NLP, tetapi juga merefleksikan harmonisasi antara sains, spiritualitas, dan tanggung jawab sosial dalam kerangka etika Islam (Abu Bakar et al. 2024)

BAB VI KESIMPULAN

6.1. Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan mengenai implementasi model BART berbasis *Transfer Learning* untuk peringkasan transkrip video YouTube dan evaluasi kualitas menggunakan metrik ROUGE, maka dapat ditarik tiga kesimpulan utama sebagai berikut:

1. Sistem peringkasan otomatis berbasis web yang dikembangkan menggunakan *framework Flask* berhasil diimplementasikan secara end-to-end dan berjalan sesuai dengan rancangan metodologi penelitian. Sistem mampu memproses input berupa URL video YouTube, mengekstraksi metadata dan transkrip secara otomatis, melakukan preprocessing teks, menghasilkan ringkasan menggunakan model *BART* berbasis *Transfer Learning*, serta menampilkan hasil evaluasi ROUGE secara terstruktur. Dari 730 video yang diproses, sebanyak 662 video atau 90,7% berhasil diekstraksi transkripnya dengan tingkat konsistensi teks terhadap audio sebesar 92,4% berdasarkan evaluasi manual. Hal ini menunjukkan bahwa sistem memiliki stabilitas operasional yang baik dan mampu mendukung proses peringkasan secara otomatis dengan tingkat keberhasilan yang tinggi.
2. Penerapan model *BART* berbasis *Transfer Learning* terbukti efektif dalam menghasilkan ringkasan *Ekstraktif* yang informatif dan relevan terhadap isi transkrip video edukatif berbahasa Indonesia. Hasil evaluasi menggunakan metrik ROUGE menunjukkan bahwa performa terbaik diperoleh pada kategori video berdurasi pendek (1–3 menit) dengan nilai *F1-Score* ROUGE-1 sebesar 63,9%, ROUGE-2 sebesar 46,7%, dan ROUGE-L sebesar 61,0%. Pada kategori durasi sedang (4–7 menit), nilai *F1-Score* menurun menjadi 55,1% (ROUGE-1), 40,2%

(ROUGE-2), dan 52,3% (ROUGE-L), sedangkan pada kategori durasi panjang (>7 menit) nilai *F1-Score* tercatat sebesar 47,0% (ROUGE-1), 33,1% (ROUGE-2), dan 44,8% (ROUGE-L). Hasil ini menunjukkan bahwa model mampu mempertahankan informasi utama secara cukup baik, khususnya pada teks dengan kompleksitas rendah hingga sedang.

3. Durasi video terbukti berpengaruh terhadap kualitas ringkasan yang dihasilkan model. Semakin panjang durasi video, semakin kompleks struktur dan distribusi informasi dalam transkrip, sehingga proses seleksi kalimat menjadi lebih menantang bagi model berbasis transformer. Penurunan performa paling signifikan terjadi pada metrik ROUGE-2, yang berkaitan dengan kesesuaian bigram dan koherensi frasa, menunjukkan bahwa menjaga kesinambungan antar kalimat dalam dokumen panjang merupakan tantangan utama. Meskipun demikian, nilai ROUGE-1 dan ROUGE-L yang relatif stabil menunjukkan bahwa pendekatan *Transfer Learning* berbasis *BART* tetap efektif dalam mempertahankan inti informasi utama. Dengan demikian, penelitian ini berhasil menjawab rumusan masalah terkait implementasi sistem, penerapan model *BART* berbasis *Transfer Learning*, serta pengukuran kualitas ringkasan secara kuantitatif menggunakan metrik ROUGE.

6.2. Saran

Berdasarkan hasil penelitian yang telah dilakukan, terdapat beberapa saran yang dapat dijadikan acuan untuk pengembangan penelitian selanjutnya.

1. Penelitian berikutnya dapat membandingkan model *BART* dengan model lain yang serupa untuk melihat apakah ada model yang dapat memberikan hasil lebih baik, terutama pada teks yang panjang. Perbandingan ini penting agar

dapat diketahui model yang paling sesuai untuk peringkasan transkrip video berbahasa Indonesia.

2. Selain menggunakan evaluasi otomatis seperti ROUGE, penelitian selanjutnya dapat menambahkan penilaian dari pengguna secara langsung, misalnya melalui kuesioner. Dengan begitu, kualitas ringkasan tidak hanya dinilai dari angka, tetapi juga dari pendapat dan tingkat pemahaman pengguna terhadap hasil ringkasan

DAFTAR PUSTAKA

- Abu Bakar, Ahmad Yumni, Farahdina Fazial, Sakinatul Raadiyah Abdullah, Nurul Husna Mohd Taher, Noor Hidayah Kasim, and Md Saufi Abdul Hamid. 2024. "The Concept of Tabayyun in Dissemination of Information Through Mass Media." *International Journal of Academic Research in Business and Social Sciences* 14(1): 2630–42. doi:10.6007/ijarbss/v14-i1/20693.
- Aggarwal, Charu C., and Cheng Xiang Zhai. 2019. "A Survey of Text Summarization Techniques." *Mining Text Data* 9781461432: 1–522. doi:10.1007/978-1-4614-3223-4.
- Albeer, Rand Abdulwahid, Huda F. Al-Shahad, Hiba J. Aleqabie, and Noor D. Al-Shakarchy. 2022. "Automatic Summarization of YouTube Video Transcription Text Using Term Frequency-Inverse Document Frequency." *Indonesian Journal of Electrical Engineering and Computer Science* 26(3): 1512–19. doi:10.11591/ijeecs.v26.i3.pp1512-1519.
- Amirudin, Suyono, Soeprijanto, and M. Hudri. 2025. "Digital Religious Literature and Its Role in Shaping Religious Moderation Among Indonesian University Students." *Jurnal Lektur Keagamaan* 23(2): 432–61. doi:10.31291/jlka.v23i2.1444.
- Ari Holtzman, Jan Buys. 2020. "The Curious Case of Neural Text Degeneration." *Energy* 87(October): 645–59.
- Astuti, Rahma Hayuning, Muljono Muljono, and Sutriawan Sutriawan. 2024. "Indonesian News Text Summarization Using MBART Algorithm." *Scientific Journal of Informatics* 11(1): 155–64. doi:10.15294/sji.v11i1.49224.
- Barbella, Marcello, and Genoveffa Tortora. 2022. "Rouge Metric Evaluation for Text Summarization Techniques." *SSRN Electronic Journal*. doi:10.2139/ssrn.4120317.
- Bhagavan, Kancharla Nikhilesh, Macharla Sri Vardhan, Madamanchi Ashok Chowdhary, and Raksha Sharma. 2023. "Nav-Nlp at RadSum23: Abstractive Summarization of Radiology Reports Using BART Finetuning." *Proceedings of the Annual Meeting of the Association for Computational Linguistics*: 541–44. doi:10.18653/v1/2023.bionlp-1.55.
- Biswas, Sourav, and Atul Kumar Patel. 2022. "YouTube Transcript Summarizer To Summarize the Content of YouTube." : 3617–22.
- Bokhove, Christian, and Christopher Downey. 2018. "Automated Generation of 'Good Enough' Transcripts as a First Step to Transcription of Audio-Recorded Data." *Methodological Innovations* 11(2). doi:10.1177/2059799118790743.
- Chauhan, Vishakha, Manish Tiwari, Harsh Bharadwaj, Ishita Goswami, and Shreela Pareeka. 2024. "YouTube Transcript Summarizer: Enhancing Accessibility and Content Discovery." *SSRN Electronic Journal*. doi:10.2139/ssrn.4832193.
- Dharmani, Prof. Aarti. 2022. "YouTube Transcript Summarizer." *International Journal of Research Publication and Reviews* 04(01): 1806–12. doi:10.55248/gengpi.2023.4149.
- Fadlilah, Muhammad Furqon, Aldy Rialdy Atmadja, and Muhammad Deden Firdaus. 2024. "Pemanfaatan Transformer Untuk Peringkasan Teks : Studi Kasus Pada Transkripsi Video Pembelajaran." 6(3): 2111–19. doi:10.47065/bits.v6i3.6342.

- Fauzan, Ahmad. 2019. "Penggunaan Media Youtube Dan Sikap Pengguna Media Youtube ((Studi Deskriptif Penggunaan Media Youtube Dan Sikap Pengguna Media Youtube Dikalangan Mahasiswa Ilmu Komunikasi Universitas Tadulako)." *Kinesik* 6(3): 247–54. <https://marketing.co.id/youtube-dan->
- Gupta, Praveen Kumar, Gaurav Dubey, and Akshat Sharma. 2025. "Text Summarization Using NLP." 03(May): 2665–70.
- Hartawan, Gaduh, Dian Sa'adillah Maylawati, and Wisnu Uriawan. 2024. "Bidirectional and Auto-Regressive Transformer (BART) for Indonesian Abstractive Text Summarization." *Jurnal Informatika Polinema* 10(4): 535–42. doi:10.33795/jip.v10i4.5242.
- Holy Quran website. 2016a. "Quran 58:11 Surah Mujadilah Ayat 11 Tafsir Ibn Katheer." *Holy Quran website*. <https://surahquran.com/tafsir-id-aya-11-sora-58.html>.
- Holy Quran website. 2016b. "Surah Al Asr Ayat 1 Tafsir Quran 103:1." *Holy Quran website*. <https://surahquran.com/indonesian-aya-1-sora-103.html>.
- Holy Quran website. 2016c. "Surah Hujurat Ayat 6 Tafsir Quran 49:6." <https://surahquran.com/indonesian-aya-6-sora-49.html>.
- Howard, Jeremy, and Sebastian Ruder. 2018. "Universal Language Model Fine-Tuning for Text Classification." *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)* 1: 328–39. doi:10.18653/v1/p18-1031.
- Ilampiray, P., D. Naveen Raju, A. Thilagavathy, M. Mohamed Tharik, S. Madhan Kishore, A. S. Nithin, and I. Infant Raj. 2023. "Video Transcript Summarizer." *E3S Web of Conferences* 399: 1–11. doi:10.1051/e3sconf/202339904015.
- Jakhotiya, Aachal, Harshada Jain, Bhavik Jain, and Charmi Chaniyara. 2022. "Text Pre-Processing Techniques in Natural Language Processing: A Review." *International Research Journal of Engineering and Technology* 9(2): 878–80.
- Karhade, Aditya V., Ophelie Lavoie-Gagne, Nicole Agaronnik, Hamid Ghaednia, Austin K. Collins, David Shin, and Joseph H. Schwab. 2022. "Natural Language Processing for Prediction of Readmission in Posterior Lumbar Fusion Patients: Which Free-Text Notes Have the Most Utility?" *Spine Journal* 22(2): 272–77. doi:10.1016/j.spinee.2021.08.002.
- Katsir, Ibnu. "Tafsir Surat An-Nahl Ayat 78." <https://tafsirweb.com/4426-surat-an-nahl-ayat-78.html>.
- Katsir, Ibnu. "Tafsir Surat An-Nahl Ayat 79." <https://tafsirweb.com/4427-surat-an-nahl-ayat-79.html>.
- Kyatam, Maniktanta Reddy. 2022. "Video Transcript Summarization." *Program* 44(MAY 2022): 1–7.
- Leonandya, Rezka, and Fariz Ikhwantri. 2019. "Pretrained Language Model Transfer Learning on Neural Named Entity Recognition in Indonesian Conversational Texts." *Proceedings of the 33rd Pacific Asia Conference on Language, Information and Computation, PACLIC 2019*: 104–13.
- Lewis, Mike, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2020. "BART: Denoising Sequence-to-Sequence Pre-Training for Natural Language Generation, Translation, and Comprehension." *Proceedings of the Annual Meeting of the Association for Computational Linguistics*: 7871–80. doi:10.18653/v1/2020.acl-main.703.

- Lin, Chin-Yew. 2004. "Rouge: A Package for Automatic Evaluation of Summaries." *Japanese Circulation Journal* 34(12): 1213–20. doi:10.1253/jcj.34.1213.
- Nur Setiyana, Fira, and Anggun Badu Kusuma. 2021. "Potensi Pemanfaatan Youtube Dalam Pembelajaran Matematika." *EduMatSains: Jurnal Pendidikan, Matematika dan Sains* 6(1): 71–90. doi:10.33541/edumatsains.v6i1.2945.
- Pabitwar, Sarvesh, Vaibhav Anarase, Sakshi Changedia, Gaurish Mundada, and Prof A D Londhe. 2024. "Resume Summarization Using SpaCy, Natural Language Processing, and Flas." (11): 6283–87.
- Rajput, Minakshi Ravindra, Apeksha Kamble, Adnan Gagan, Pooja Wagh, Umar Aziz, M R Rajput, and Students Of Btech. 2022. "Flask Web Framework Based News Summarizer: Web Application." *International Journal of Scientific Research and Engineering Development* (December). <https://www.researchgate.net/publication/366658605>.
- Sharma, Grishma, and Deepak Sharma. 2023. "Automatic Text Summarization Methods: A Comprehensive Review." *SN Computer Science* 4(1). doi:10.1007/s42979-022-01446-w.
- Subagyo, Ageng Ramdhan, and Theopilus Bayu Sasongko. 2024. "Implementasi Algoritma Transformers BART Dan Penggunaan Metode Optimasi Adam Untuk Klasifikasi Judul Berita Palsu." *Jurnal Media Informatika Budidarma* 8(3): 1768. doi:10.30865/mib.v8i3.7852.
- Ülker, Mehtap, and A. Bedri Özer. 2024. "The Bart-Based Model for Scientific Articles Summarization." *Journal of Universal Computer Science* 30(13): 1807–28. doi:10.3897/jucs.115121.
- Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. "Attention Is All You Need." *International Conference on Information and Knowledge Management, Proceedings (Nips)*: 4752–58. doi:10.1145/3583780.3615497.
- Vijaya Kumari, P, M Chenna Keshava, C Narendra, P Akanksha, and K Sravani. 2022. "Youtube Transcript Summarizer Using Flask And Nlp." *Journal of Positive School Psychology* 6(8): 1204–9. <http://journalppw.com>.
- Vinet, Luc, and Alexei Zhedanov. 2024. "Potret Penyandang Disabilitas Di Indonesia." *Journal of Physics A: Mathematical and Theoretical* 44(8): 1–14. doi:10.1088/1751-8113/44/8/085201.
- Vira, Adinda, and Eka Reynata. 2022. "Penerapan Youtube Sebagai Media Baru Dalam Komunikasi Massa." *Komunikologi: Jurnal ilmiah ilmu komunikasi* 19(2): 96–101. <https://komunikologi.esaunggul.ac.id/index.php/KM/article/view/514/254>.
- Widiastutik, Rully, Lukman Zaman, and Joan Santoso. 2021. "Peringkasan Teks Ekstraktif Pada Dokumen Tunggal Menggunakan Metode Restricted Boltzmann Machine." *INSYST: Journal of Intelligent System and Computation*: 58–64.
- Zahara, Riantika, Zamzam Nurhuda, Prodi Tarjamah, Fakultas Adab, Dan Humaniora, and U I N Syarif Hidayatullah. 2024. "Analisis Kualitas Terjemahan Otomatis Subtitle Pada Channel Youtube (Indo4arab Khalid Nahdi).
- Zhang, Binbin, Di Wu, Zhendong Peng, Xingchen Song, Zhuoyuan Yao, Hang Lv,

- Lei Xie, et al. 2022. “WeNet 2.0: More Productive End-to-End Speech Recognition Toolkit.” *Proceedings of the Annual Conference of the International Speech Communication Association, Interspeech 2022-Sept*: 1661–65. doi:10.21437/Interspeech.2022-483.
- Zhao, Zheng, and Pinzhen Chen. 2022. “To Adapt or to Fine-Tune: A Case Study on Abstractive Summarization.” *21st Chinese National Conference on Computational Linguistic, CCL 2022 (c)*: 824–35.
- Zolotareva, Ekaterina, Tsegaye Misikir Tashu, and Tomáš Horváth. 2020. “Abstractive Text Summarization Using Transfer Learning.” *CEUR Workshop Proceedings 2718*(November 2020): 75–80.

GLOSARIUM

A

API (Application Programming Interface) : Antarmuka pemrograman aplikasi yang memungkinkan sistem berkomunikasi dan bertukar data dengan layanan lain, seperti YouTube Data API.

Attention Mechanism : Mekanisme dalam arsitektur transformer yang memungkinkan model fokus pada bagian teks tertentu saat memproses informasi.

B

BART (Bidirectional and Auto-Regressive Transformer) : Model berbasis transformer yang menggabungkan encoder dua arah seperti BERT dan decoder autoregresif seperti GPT untuk tugas NLP seperti summarization.

Beam Search : Strategi decoding dalam model NLP yang mempertimbangkan beberapa kemungkinan urutan token terbaik untuk menghasilkan output lebih optimal.

Bigram : Pasangan dua kata berurutan dalam teks yang digunakan dalam evaluasi ROUGE-2.

C

Cloudflare Tunnel : Layanan tunneling yang memungkinkan server lokal Flask dapat diakses melalui internet tanpa konfigurasi hosting publik.

Cross-Entropy Loss : Fungsi kerugian yang digunakan untuk mengukur perbedaan antara distribusi probabilitas prediksi model dan distribusi target.

D

Dataset : Kumpulan data yang digunakan dalam penelitian, yaitu 730 video dari channel edukatif 'Kok Bisa?'.
Decoding : Proses menghasilkan teks output (ringkasan) dari representasi internal model.

Deep Learning : Cabang machine learning berbasis jaringan saraf tiruan berlapis banyak.

E

Ekstraktif Summarization : Metode peringkasan dengan memilih kalimat penting langsung dari teks asli tanpa mengubah strukturnya.

Encoder–Decoder Architecture : Struktur model yang terdiri dari encoder dan decoder dalam arsitektur transformer.

F

F1-Score : Rata-rata harmonis antara Precision dan Recall yang digunakan sebagai indikator utama kualitas ringkasan.

Fine-Tuning : Proses pelatihan lanjutan pada model pre-trained agar sesuai dengan domain spesifik.

Flask : Framework Python ringan untuk membangun aplikasi web.

G

GPU (Graphics Processing Unit) : Perangkat keras yang mempercepat proses komputasi model deep learning.

Greedy Decoding : Metode decoding yang memilih token dengan probabilitas tertinggi pada setiap langkah.

H

Hablum Minallah : Hubungan manusia dengan Allah dalam konteks penelitian sebagai bentuk syukur melalui pengembangan ilmu.

Hablum Minannas : Hubungan manusia dengan sesama melalui sistem yang meningkatkan akses informasi.

HuggingFace Transformers : Library NLP yang menyediakan model pre-trained seperti BART.

I

Inference : Proses penggunaan model terlatih untuk menghasilkan prediksi atau ringkasan.

Input URL : Tautan video YouTube yang dimasukkan pengguna untuk diproses sistem.

L

LCS (Longest Common Subsequence) : Urutan kata terpanjang yang sama antara dua teks dalam evaluasi ROUGE-L.

Likert Scale : Skala pengukuran persepsi pengguna berbentuk rentang nilai (misalnya 1–5).

M

Machine Learning : Cabang kecerdasan buatan yang memungkinkan sistem belajar dari data.

Metadata : Informasi tambahan seperti judul, durasi, dan tanggal unggah video.

Model Pre-trained : Model yang telah dilatih pada dataset besar sebelum diadaptasi pada tugas khusus.

N

Natural Language Processing (NLP) : Bidang kecerdasan buatan yang memproses dan memahami bahasa manusia.

Normalization : Proses standarisasi teks dalam preprocessing.

O

Output Ringkasan : Hasil akhir berupa teks ringkas dari transkrip video.

Overlapping N-gram : Jumlah kemunculan n-gram yang sama antara ringkasan sistem dan referensi.

P

Precision : Proporsi informasi dalam ringkasan sistem yang relevan terhadap referensi.

Preprocessing : Tahap pembersihan teks sebelum diproses model.

Probability Distribution : Distribusi peluang token yang dihasilkan model saat prediksi.

R

Recall : Proporsi informasi referensi yang berhasil ditangkap dalam ringkasan sistem.

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) : Metrik evaluasi kualitas ringkasan berbasis kesamaan n-gram.

ROUGE-1 : Evaluasi kesamaan unigram (kata tunggal).

ROUGE-2 : Evaluasi kesamaan bigram (dua kata berurutan).

ROUGE-L : Evaluasi kesamaan berdasarkan LCS.

S

Self-Attention : Mekanisme dalam transformer untuk memahami hubungan antar kata dalam satu kalimat.

Summarization : Proses menghasilkan ringkasan dari teks panjang.

System Integration : Penggabungan seluruh komponen dalam satu sistem utuh.

T

Text Tokenization : Proses pemecahan teks menjadi unit-unit kecil (token).

Transfer Learning : Teknik pemanfaatan model pre-trained untuk tugas baru dengan penyesuaian terbatas.

Transformer : Arsitektur deep learning berbasis attention mechanism.

U

Unigram : Satuan satu kata dalam analisis teks.

User Interface (UI) : Tampilan antarmuka yang digunakan pengguna untuk berinteraksi dengan sistem.

V

Validation : Proses pengecekan performa model terhadap data evaluasi.

Video Duration Category : Pengelompokan video berdasarkan durasi (pendek, sedang, panjang).

W

Web Application : Aplikasi berbasis browser yang dikembangkan menggunakan Flask.

Y

YouTube Data API v3 : API resmi Google untuk mengakses data video YouTube.

YouTube Transcript API : Library untuk mengambil transkrip otomatis dari video YouTube.

LAMPIRAN LAMPIRAN

```
# ===== Setup (Colab) =====
# Runtime Colab: Python 3, GPU L4/A100/T4
!apt-get -y install -qq ffmpeg
!pip install -q flask pyngrok YouTube-transcript-api Rouge-score pandas openpyxl nltk requests yt-
dlp openai-whisper transformers torch torchvision torchaudio pycloudflared yake scikit-learn
sentencepiece markupsafe

# ===== Imports & NLTK =====
import os, re, io, sys, traceback, tempfile, threading, time, unicodedata, uuid, json, random, html
from urllib.parse import urlparse, parse_qs
import concurrent.futures
from functools import lru_cache

from flask import Flask, render_template_string, request, send_file, flash, jsonify, redirect, url_for
import requests
from requests.adapters import HTTPAdapter
from urllib3.util.retry import Retry

from YouTube_transcript_api import YouTubeTranscriptApi, TranscriptsDisabled,
NoTranscriptFound, VideoUnavailable
from Rouge_score import Rouge_scorer
import pandas as pd
from pyngrok import ngrok, conf
from pyngrok import process as ngrok_process
from pycloudflared import try_cloudflare

import nltk
nltk.download('punkt', quiet=True)
try:
    nltk.download('punkt_tab', quiet=True)
except Exception:
    pass

# Whisper (ASR fallback)
import whisper
import yt_dlp

# Transformers (BART/MBART)
import torch
from transformers import AutoTokenizer, AutoModelForSeq2SeqLM

# Tambahan untuk ringkasan hybrid
import yake
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
```

```

from nltk.tokenize import sent_tokenize

# Aman untuk render paragraf HTML
from markupsafe import escape

# ===== KONFIGURASI =====
YOUTUBE_API_KEY = os.getenv("YOUTUBE_API_KEY", "")
NGROK_AUTHTOKEN = os.getenv("NGROK_AUTHTOKEN", "")
NGROK_REGION = os.getenv("NGROK_REGION", "ap")
NGROK_RESERVED_DOMAIN = os.getenv("NGROK_RESERVED_DOMAIN") # ex:
"namamu.ngrok.app"
PORT = int(os.getenv("PORT", "5000"))

# Mode praktis & cepat
FAST_MODE = True # hemat waktu; jika False + USE_WHISPER=True akan pakai Whisper
fallback
USE_WHISPER = False
YTDLP_TIMEOUT = 12
YTDLP_OPTS_COMMON = {"quiet": True, "skip_download": True, "socket_timeout": 10, "retries":
1}

# ===== PARAM UTAMA (MODE RINGKAS) =====
USE_CC_ONLY = False
WHISPER_MODEL = "small"
WHISPER_LANG = "id"
USE_SMOOTH_FUSION = False

# Model Ringkasan (MBART-50 untuk multilingual)
FINETUNED_MODEL_PATH = None
FINETUNED_HF_REPO = None # contoh: "username/mBART-id-finetuned"
BASE_MODEL = "facebook/mBART-large-50"

# Ambang ROUGE (monitor kualitas)
ROUGE_MIN_R1F = float(os.getenv("ROUGE_MIN_R1F", "0.30"))
ROUGE_MIN_R2F = float(os.getenv("ROUGE_MIN_R2F", "0.12"))
ROUGE_MIN_RLF = float(os.getenv("ROUGE_MIN_RLF", "0.20"))

# ===== HTTP Session (retry) =====
HTTP = requests.Session()
_retries = Retry(total=2, backoff_factor=0.3, status_forcelist=(429, 500, 502, 503, 504))
HTTP.mount("https://", HTTPAdapter(max_retries=_retries, pool_maxsize=32))
HTTP.mount("http://", HTTPAdapter(max_retries=_retries, pool_maxsize=32))

# ===== Bersihkan instance lama (Flask & ngrok) =====
try:
    if 'app_server' in globals() and app_server:
        print("☹ Menutup server Flask lama...")

```

```

        app_server.shutdown()
        time.sleep(0.5)
except Exception as e:
    print("Info: tidak ada server lama / gagal shutdown:", e)

try:
    print("☺ Memutus tunnel ngrok lama (jika ada)...")
    for t in ngrok.get_tunnels():
        ngrok.disconnect(t.public_url)
    ngrok.kill()
except Exception as e:
    print("Info: ngrok belum aktif / sudah bersih:", e)

# ===== Aplikasi Flask =====
app = Flask(__name__)
app.secret_key = "secret-123"

# ----- In-memory job store -----
JOBS = {} # job_id -> {"status": "...", "error": str|None, "result": {...}, "xls": BytesIO}

# ----- Cache ringan -----
_META_CACHE = {}
_TRANSCRIPT_CACHE = {}

def cache_get(k): return _META_CACHE.get(k) if k.startswith("meta:") else
_TRANSCRIPT_CACHE.get(k.replace("transcript:", ""))
def cache_set(k,v):
    if k.startswith("meta:"): _META_CACHE[k.split("meta:")[1]] = v
    elif k.startswith("transcript:"): _TRANSCRIPT_CACHE[k.split("transcript:")[1]] = v

# ===== Utils: YouTube =====
def extract_video_id(url_or_id: str) -> str:
    if re.fullmatch(r"[A-Za-z0-9_-]{6,}", url_or_id):
        return url_or_id
    try:
        u = urlparse(url_or_id)
        if "youtu.be" in u.netloc:
            return u.path.lstrip("/")
        if "YouTube.com" in u.netloc:
            qs = parse_qs(u.query).get("v")
            if qs: return qs[0]
            for pat in (r"/shorts/([A-Za-z0-9_-]{6,})", r"/embed/([A-Za-z0-9_-]{6,})"):
                m = re.search(pat, u.path)
                if m: return m.group(1)
    except Exception:
        pass
    return url_or_id

```

```

def _hms_from_seconds(sec):
    sec = int(sec or 0)
    return f"{sec//3600:02d}:{(sec%3600)//60:02d}:{sec%60:02d}" if sec>=3600 else
    f"{sec//60:02d}:{sec%60:02d}"

def _iso8601_duration_to_hms(iso):
    hours = minutes = seconds = 0
    m = re.match(r"PT(?:\d+)H?(?:\d+)M?(?:\d+)S?", iso or "")
    if m:
        hours = int(m.group(1) or 0)
        minutes = int(m.group(2) or 0)
        seconds = int(m.group(3) or 0)
    total = hours*3600 + minutes*60 + seconds
    return f"{total//3600:02d}:{(total%3600)//60:02d}:{total%60:02d}" if total>=3600 else
    f"{total//60:02d}:{total%60:02d}"

def yt_meta_fallback(video_id:str):
    try:
        url = f"https://www.YouTube.com/watch?v={video_id}"
        with yt_dlp.YouTubeDL(**YTDLP_OPTS_COMMON) as ydl:
            info = ydl.extract_info(url, download=False)
            title = info.get("title", "-")
            dur = _hms_from_seconds(info.get("duration"))
            pub = (info.get("upload_date") or "")[:8]
            if pub and len(pub)==8: pub = f"{pub[:4]}-{pub[4:6]}-{pub[6:]}"
            return {"title": title, "published": pub or "-", "duration": dur or "-"}
    except Exception:
        return {"title": "-", "published": "-", "duration": "-"}

def fetch_video_meta(video_id: str):
    if video_id in _META_CACHE:
        return _META_CACHE[video_id]
    c = cache_get(f"meta:{video_id}")
    if c:
        _META_CACHE[video_id] = c
        return c
    if YOUTUBE_API_KEY:
        try:
            r = HTTP.get(
                "https://www.googleapis.com/YouTube/v3/videos",
                params={"id": video_id, "key": YOUTUBE_API_KEY, "part": "snippet,contentDetails"},
                timeout=10
            )
            if r.status_code == 200 and r.json().get("items"):
                item = r.json()["items"][0]
                data = {

```

```

        "title": item["snippet"]["title"],
        "published": item["snippet"]["publishedAt"][:10],
        "duration": _iso8601_duration_to_hms(item["contentDetails"]["duration"])
    }
    _META_CACHE[video_id] = data; cache_set(f"meta:{video_id}", data)
    return data
except Exception:
    pass
data = yt_meta_fallback(video_id)
_META_CACHE[video_id] = data; cache_set(f"meta:{video_id}", data)
return data

# ===== Whisper fallback =====
_whisper_model_cache = None
def _load_whisper_model():
    global _whisper_model_cache
    if _whisper_model_cache is None:
        device = "cuda" if torch.cuda.is_available() else "cpu"
        _whisper_model_cache = whisper.load_model(WHISPER_MODEL, device=device)
    return _whisper_model_cache

def _download_audio_ytdlp(video_id: str) -> str:
    url = f"https://www.YouTube.com/watch?v={video_id}"
    tmpdir = tempfile.mkdtemp()
    outtmpl = os.path.join(tmpdir, "%(id)s.%(ext)s")
    ydl_opts = {
        "format": "bestaudio[ext=m4a]/bestaudio/best",
        "outtmpl": outtmpl,
        "quiet": True,
        "noprogress": True,
        "nocheckcertificate": True,
        "retries": 1,
        "continued": True,
        "postprocessors": [
            {"key": "FFmpegExtractAudio", "preferredcodec": "mp3", "preferredquality": "160"}
        ],
    }
    with yt_dlp.YouTubeDL(ydl_opts) as ydl:
        info = ydl.extract_info(url, download=True)
        base_id = info.get("id")
        mp3_path = os.path.join(tmpdir, f"{base_id}.mp3")
        if os.path.exists(mp3_path):
            return mp3_path
        cand = info.get("requested_downloads") or []
        for e in cand:
            p = e.get("filepath")
            if p and os.path.exists(p):

```



```

r"(?i)\b(terima kasih | thanks for watching | jangan lupa (like | subscribe | komen) | "
r"support channel | sampai jumpa | see you (soon | next) | follow (ig | instagram | tiktok) | "
r"link di (deskripsi | bio) | aktifkan lonceng)\b"
)

```

```

def _drop_garbage_lines(text: str) -> str:
    sents = re.split(r"(?<=[.!?...])\s+", text)
    cleaned = []
    for s in sents:
        if not s.strip(): continue
        sym_ratio = sum(c in "/\| @#<>[]{}" for c in s) / max(1, len(s))
        if sym_ratio > 0.04: continue
        if _BAD_PATTERNS.search(s): continue
        cleaned.append(s)
    return " ".join(cleaned).strip()

```

```

def _dedupe_sentences(text: str) -> str:
    sents = [s.strip() for s in sent_tokenize(text) if s.strip()]
    seen, uniq = set(), []
    for s in sents:
        k = re.sub(r"\W+", "", s.lower())
        if k not in seen:
            seen.add(k); uniq.append(s)
    return " ".join(uniq)

```

```

def _dedupe_by_similarity(lines, sim_thresh: float = 0.88):
    seq = [l.strip() for l in lines if l and l.strip()]
    if len(seq) <= 1:
        return seq[:]
    try:
        vec = TfidfVectorizer(min_df=1, max_df=0.95, ngram_range=(1, 2))
        X = vec.fit_transform(seq)
        S = cosine_similarity(X)
        keep_idx = []
        for i in range(len(seq)):
            is_dup = any(S[i, j] >= sim_thresh for j in keep_idx)
            if not is_dup:
                keep_idx.append(i)
        return [seq[i] for i in keep_idx]
    except Exception:
        seen, out = set(), []
        for s in seq:
            k = re.sub(r"\W+", "", s.lower())
            if k not in seen:
                seen.add(k); out.append(s)
        return out

```

```

_WORD_DUPES = re.compile(r'(?iu)\b([\W\d_]+)?(?:\s+\1\b)+')
_BIGRAM_DUPES = re.compile(r'(?isu)\b([\W\d_]+\s+[\W\d_]+)\b(?:\s+\1\b)+')
def _collapse_dupes(txt: str) -> str:
    if not (txt and txt.strip()): return txt or ""
    t = _WORD_DUPES.sub(r'\1', txt)
    t = _BIGRAM_DUPES.sub(r'\1', t)
    t = re.sub(r'\s{2,}', ' ', t).strip()
    return t

# ===== Normalisasi ID =====
_ID_NORMALIZE = {
    r"\benggak\b": "tidak", r"\bgak\b": "tidak", r"\bngg\b": "tidak",
    r"\bdll\b": "dan lain-lain", r"\bdst\b": "dan seterusnya",
    r"\bkrn\b": "karena", r"\bdr\b": "dari", r"\btdk\b": "tidak",
    r"\bsbg\b": "sebagai", r"\bsy\b": "saya",
    r"\bjumlahnya\b": "jumlahnya", r"\bnapasnya\b": "napas", r"\bketimbang\b": "dibanding",
}
def _normalize_indonesian(text: str) -> str:
    for pat, repl in _ID_NORMALIZE.items():
        text = re.sub(pat, repl, text, flags=re.IGNORECASE)
    text = re.sub(r"\s*\s*", " ", text)
    text = re.sub(r"\s*\.\s*", ". ", text)
    text = re.sub(r"\s*;\s*", "; ", text)
    text = re.sub(r"\s*:\s*", ": ", text)
    text = re.sub(r"\s{2,}", " ", text).strip()
    text = re.sub(r"\s+([\.\!\?…])", r"\1", text)
    text = _EMOJI_PAT.sub(" ", text)
    return text

def _smart_sentence_case(s: str) -> str:
    s = re.sub(r"\s+", " ", s).strip()
    if not s: return s
    tokens = s.split()
    fixed_tokens = []
    for w in tokens:
        if len(w) <= 4 and w.isupper() and w.isalpha():
            fixed_tokens.append(w)
        else:
            fixed_tokens.append(w.lower())
    s = " ".join(fixed_tokens)
    s = s[0].upper() + s[1:] if len(s) > 1 else s.upper()
    return s

# ===== Keyword & Oracle (YAKE + TF-IDF + MMR) =====
_yake_kw_extractor = yake.KeywordExtractor(lan="id", n=1, top=20)
def _top_keywords(text: str, k: int = 12):
    if not text.strip(): return []

```

```

try:
    kws = _yake_kw_extractor.extract_keywords(text)
    terms = [w for w, score in sorted(kws, key=lambda x: x[1])][:k]
    seen, out = set(), []
    for t in terms:
        key = re.sub(r"\W+", "", t.lower())
        if key and key not in seen:
            seen.add(key); out.append(t)
    return out
except Exception:
    toks = re.findall(r"[A-Za-zÀ-Öø-öÿ\u00C0-\u024F\u1E00-\u1EFF\u0600-\u06FF\u0900-\u097F]+", text.lower())
    freq = {}
    for w in toks:
        if len(w) < 3: continue
        freq[w] = freq.get(w, 0) + 1
    return [w for w, _ in sorted(freq.items(), key=lambda x: -x[1])[:k]]

def _tfidf_mmr_sentences(text: str, max_sents: int = 10, diversity: float = 0.7):
    sents = [s.strip() for s in sent_tokenize(text) if s.strip()]
    n = len(sents)
    if n == 0: return []
    if n == 1: return sents
    max_df_val = 1.0 if n < 3 else 0.9
    try:
        vec = TfidfVectorizer(min_df=1, max_df=max_df_val, ngram_range=(1, 2))
        X = vec.fit_transform(sents)
    except ValueError:
        return sents
    doc = vec.transform([" ".join(sents)])
    sim_to_doc = cosine_similarity(X, doc).ravel()
    selected, candidate_idxxs = [], list(range(n))
    if len(candidate_idxxs) <= max_sents:
        return sents
    cur = int(sim_to_doc.argmax()); selected.append(cur); candidate_idxxs.remove(cur)
    while len(selected) < max_sents and candidate_idxxs:
        mmr_scores = []
        for i in candidate_idxxs:
            sim_d = sim_to_doc[i]
            sim_s = max(cosine_similarity(X[i], X[selected])).ravel() if selected else 0.0
            mmr = diversity * sim_d - (1 - diversity) * sim_s
            mmr_scores.append((mmr, i))
        mmr_scores.sort(reverse=True)
        pick = mmr_scores[0][1]
        selected.append(pick); candidate_idxxs.remove(pick)
    selected.sort()
    return [sents[i] for i in selected]

```

```

# ===== Model & Tokenizer =====
_device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
print("Device:", _device)
try:
    if _device.type == "cuda":
        torch.backends.cuda.matmul.allow_tf32 = True
        torch.backends.cudnn.allow_tf32 = True
        torch.set_float32_matmul_precision("high")
except Exception:
    pass

def _resolve_model_name():
    if FINETUNED_MODEL_PATH and os.path.isdir(FINETUNED_MODEL_PATH): return
    FINETUNED_MODEL_PATH
    if FINETUNED_HF_REPO: return FINETUNED_HF_REPO
    return BASE_MODEL

print("Memuat model ringkasan...")
_model_name = _resolve_model_name()
_load_kwargs = {}
if _device.type == "cuda":
    _load_kwargs["torch_dtype"] = torch.float16
    _load_kwargs["low_cpu_mem_usage"] = True
tokenizer = AutoTokenizer.from_pretrained(_model_name)
model = AutoModelForSeq2SeqLM.from_pretrained(_model_name, **_load_kwargs).to(_device)
model.eval()

forced_bos_token_id = None
try:
    if "mBART" in _model_name.lower():
        lang_tgt = "id_ID"
        if hasattr(tokenizer, "lang_code_to_id") and lang_tgt in tokenizer.lang_code_to_id:
            forced_bos_token_id = tokenizer.lang_code_to_id[lang_tgt]
except Exception:
    forced_bos_token_id = None

# ===== Parameter Ringkasan =====
MAX_CHUNK_TOKENS = 760
TARGET_RATIO = 0.15
BATCH_SIZE = 4 if _device.type == "cuda" else 1

@lru_cache(maxsize=20000)
def _tok_len(s: str) -> int:
    return len(tokenizer.encode(s, add_special_tokens=False))

# ---- FIX: _encode dengan padding=True + sanitasi input ----

```

```

def _encode(texts):
    if isinstance(texts, str):
        texts = [texts]
    texts = [{" " if t is None else str(t)} for t in texts]
    texts = [t.strip() if t and t.strip() else "." for t in texts]
    if "mBART" in _model_name.lower() and hasattr(tokenizer, "lang_code_to_id"):
        tokenizer.src_lang = "id_ID"
    return tokenizer(
        texts,
        max_length=1024,
        truncation=True,
        padding=True,
        return_tensors="pt"
    )

# ===== Caption Parsers =====
def _parse_json3_to_text(js_text: str) -> str:
    """Parse YouTube timedtext JSON3 → plain text."""
    try:
        data = json.loads(js_text) if isinstance(js_text, str) else js_text
    except Exception:
        return ""
    out = []
    events = data.get("events") if isinstance(data, dict) else None
    if not events and isinstance(data, list):
        events = data
    if not events:
        return ""
    for ev in events:
        segs = ev.get("segs") if isinstance(ev, dict) else None
        if segs:
            for s in segs:
                t = (s.get("utf8") or "").replace("\n", " ").strip()
                if t and t.strip() not in ("[BLANK]",):
                    out.append(t)
        else:
            t = (ev.get("utf8") or "").replace("\n", " ").strip() if isinstance(ev, dict) else ""
            if t:
                out.append(t)
    txt = " ".join(out)
    return re.sub(r"\s+", " ", txt).strip()

def _parse_vtt_to_text(vtt: str) -> str:
    lines = []
    for line in vtt.splitlines():
        L = line.strip()
        if not L: continue

```

```

    if L.startswith(("WEBVTT", "NOTE")): continue
    if re.match(r"^\d{1,6}$", L): continue
    if re.search(r"\d{1,2}:\d{2}:\d{2}\.\d{3}\s+-->\s+\d{1,2}:\d{2}:\d{2}\.\d{3}", L): continue
    L = re.sub(r"<[^>]+>", " ", L)
    lines.append(html.unescape(L))
txt = " ".join(lines)
return re.sub(r"\s+", " ", txt).strip()

def _parse_srv3_xml_to_text(xml_text: str) -> str:
    chunks = re.findall(r"<text[^>]*>(.*)</text>", xml_text, flags=re.S|re.I)
    if not chunks: return ""
    txt = " ".join(html.unescape(re.sub(r"<[^>]+>", " ", c)) for c in chunks)
    return re.sub(r"\s+", " ", txt).strip()

# ===== Anti-bocor & Generator =====
BAD_PHRASES = [
    "tuliskan ulang", "susun ringkasan", "ringkasan utuh", "teks sumber",
    "rujukan gagasan penting", "ringkasan peluru",
    "klik", "klik untuk", "klik di", "tap untuk",
    "subscribe", "like", "share", "follow",
    "lihat deskripsi", "keterangan video", "link di bawah",
    "sumber:", "source:", "ref:", "reference:",
    "kode promo", "diskon", "gunakan kode", "beli sekarang",
    "herbalife", "mlm", "affiliate", "sponsor",
    "ringkasan:", "ringkasan akhir", "ringkasan awal", "ringkasan pertama", "ringkasan kedua",
    "ringkasan ketiga", "ringkasan keempat", "ringkasan kelima", "ringkasan keenam", "ringkasan
ketujuh",
    "ringkasan kedelapan", "ringkasan kesembilan", "ringkasan kesepuluh"
]

def _generation_ids(inputs, min_len, max_len, mode="balanced"):
    base = dict(
        max_length=max_len, min_length=min_len,
        no_repeat_ngram_size=6, encoder_no_repeat_ngram_size=6,
        early_stopping=True, eos_token_id=tokenizer.eos_token_id,
        repetition_penalty=1.25, length_penalty=0.95
    )
    if forced_bos_token_id is not None:
        base["forced_bos_token_id"] = forced_bos_token_id
    bad_words_ids = []
    for w in BAD_PHRASES:
        ids = tokenizer.encode(w, add_special_tokens=False)
        if ids: bad_words_ids.append(ids)
    if bad_words_ids:
        base["bad_words_ids"] = bad_words_ids

    if mode == "precise": return model.generate(**inputs, num_beams=7, **base)

```

```

if mode == "natural": return model.generate(**inputs, do_sample=True, top_p=0.85,
top_k=30, temperature=0.7, num_beams=1, **base)
return model.generate(**inputs, num_beams=3, **base)

# ===== Sanitizer heading/enumerasi =====
_HDR_ENUM_PAT = re.compile(
    r"(?im)\b(ringkasan(?:\s+(awal|akhir|pertama|kedua|ketiga|keempat|kelima|keenam|ketujuh|kedelapan|kesembilan|kesepuluh|ke\s*\d+|satu|dua|tiga|empat|lima|enam|tujuh|delapan|sembilan|sepuluh)))\s*:\s*"
)
_REPEAT_WORD_WITH_COMMA = re.compile(r"(?i)\b(\w+)(?:\s*,\s*\1\b){1,}")
def _strip_heading_artifacts(txt: str) -> str:
    out = _HDR_ENUM_PAT.sub("", txt)
    out = _REPEAT_WORD_WITH_COMMA.sub(r"\1", out)
    out = re.sub(r"(\s*){2,}", " ", out)
    out = re.sub(r"(?m)^\s*(ringkasan)?\s*[:\s]*\s*$", "", out)
    return out.strip()

# ===== Filter instruksi bocor =====
_INSTR_PATTS = [
    r"(?is)\btu(?:\s+lis|lung|kis)\s+ulang\b.*?(?:teks\s*sumber:|$)",
    r"(?is)\bpertahankan\s+terminologi\s+penting\b.*?(?:teks\s*sumber:|$)",
    r"(?is)\brujukan\s+gagasan\s+penting\b.*?(?:teks\s*sumber:|$)",
    r"(?is)\bteks\s*sumber:\s*",
    r"(?is)\(ringkasan(?:nya)?\s*peluru)\s*:\s*?",
    r"(?is)\bringkasan(?:nya)?\s*peluru\s*:\s*?",
    r"(?is)\bsecara\s+ringkas\s+dan\s+koheren\b\s*:\s*?"
]
_PROMPTY_HEADERS =
re.compile(r"(?im)^(?:tu\w*lis|tulung|tukis|ringkas(?:kan)?|susun|buat(?:lah)?)\b[^\n:]{0,120}:\s*")
def _strip_instruction_noise(txt: str) -> str:
    txt = _PROMPTY_HEADERS.sub("", txt)
    for patt in _INSTR_PATTS:
        txt = re.sub(patt, " ", txt)
    txt = re.sub(r"\s{2,}", " ", txt).strip()
    return txt

# ===== Pembersih gaya & kontradiksi =====
_PUNCT_RUN = re.compile(r'([,;:!?])\s*\1+')
_DOUBLE_COMMA = re.compile(r',\s*,+')
_SPACE_BEFORE_PUNCT = re.compile(r'\s+([,;:!?])')
_SPACE_AFTER_PUNCT = re.compile(r'([,;:!?])(?!\s|$)')

def _repair_punct(s: str) -> str:
    s = _SPACE_BEFORE_PUNCT.sub(r'\1', s)
    s = _SPACE_AFTER_PUNCT.sub(r'\1 ', s)

```

```

s = _PUNCT_RUN.sub(r'\1', s)
s = _DOUBLE_COMMA.sub(',', s)
s = re.sub(r'\s{2}',' ', s).strip()
return s

_DMG_PHRASE_SRC = r'\b(?:menyebabkan\s+kerusakan\s+sel|merusak\s+sel(?:-sel)?)\b'
_CONTRA_PAT = re.compile(
    r'(?is)\b(tidak\s+(?:menyebabkan\s+kerusakan\s+sel|merusak\s+sel(?:-
sel)?))([\^\.\]{0,120}?)?:tetapi|tapi|namun|jika|kalau)\b([\^\.\]{0,160}?)\b(?:menyebabkan\s+kerusa
kan\s+sel|merusak\s+sel(?:-sel)?)\b'
)
def _resolve_contradictions(txt: str) -> str:
    def _sub(m): return m.group(3)
    out = _CONTRA_PAT.sub(_sub, txt)
    out = re.sub(r'(' + _DMG_PHRASE_SRC + r')(?:\s*?\s*' + _DMG_PHRASE_SRC + r')+', r'\1', out,
flags=re.I)
    return _repair_punct(out)

_SLANG_MAP = [
    (r'(?i)\boke\b', ""), (r'(?i)\bok\b', ""), (r'(?i)\bnah\b', ""),
    (r'(?i)\bya kan\b', ""), (r'(?i)\byah\b', ""), (r'(?i)\bgitu\b', ""),
    (r'(?i)\bkalo\b', "jika"), (r'(?i)\bkalau\b', "jika"),
    (r'(?i)\bngg?ak\b', "tidak"), (r'(?i)\bgak\b', "tidak"), (r'(?i)\bengg?ak\b', "tidak"),
    (r'(?i)\bngerusak\b', "merusak"), (r'(?i)\bngebakar\b', "mengoksidasi"),
    (r'(?i)\boxygen\b', "oksigen")
]
_ANALOGY_PAT =
re.compile(r'(?i)\b(ibaratnya|umpamanya|analoginya|seperti|kayak)\b([\^\.\]{0,120}\.)"
_FUZZY_Filler_PAT = re.compile(r'(?i)\b(nah|jadi|gitu|hehee?he|hehe|yah|ya kan)\b[ ,]*")

def _colloquial_to_formal(txt: str) -> str:
    out = txt
    for pat, repl in _SLANG_MAP:
        out = re.sub(pat, repl, out)
    out = _FUZZY_Filler_PAT.sub("", out)
    out = re.sub(r'\s{2}', " ", out)
    out = re.sub(r'(\s*){2,}', " ", out)
    return out.strip()

def _purge_empty_analogies(txt: str) -> str:
    return _ANALOGY_PAT.sub("", txt)

def _vary_repair_words(txt: str) -> str:
    txt = re.sub(r'(?i)\bmerusak\b\s*?\s*\bmerusak\b', "menyebabkan kerusakan", txt)
    txt = re.sub(r'(?i)\bmengoksidasi\b\s*?\s*\bmengoksidasi\b', "menimbulkan proses oksidasi",
txt)
    return txt

```

```

def _style_refine_id(txt: str) -> str:
    x = _colloquial_to_formal(txt)
    x = _purge_empty_analogies(x)
    x = _vary_repair_words(x)
    x = _repair_punct(x)
    sents = [s.strip() for s in re.split(r'(?<=[.!?])\s+', x) if s.strip()]
    sents = [s[0].upper()+s[1:] if len(s)>1 else s.upper() for s in sents]
    x = " ".join(sents)
    x = _repair_punct(x)
    return x

def _condense_quantifiers(txt: str) -> str:
    sents = [s.strip() for s in re.split(r'(?<=[.!?])\s+', txt) if s.strip()]
    low, mid, high, others = [], [], [], []
    for s in sents:
        s_clean = s
        if re.search(r'(?!)\bkadar\s+(rendah|kecil)|jumlahnya\s+(sedikit|rendah)\b', s):
            low.append(s_clean)
        elif re.search(r'(?!)\bkadar\s+(sedang)|jumlahnya\s+(cukup|sedang)\b', s):
            mid.append(s_clean)
        elif re.search(r'(?!)\bkadar\s+(tinggi)|jumlahnya\s+(banyak|tinggi)\b', s):
            high.append(s_clean)
        else:
            others.append(s_clean)
    def _strip_lead(x):
        x = re.sub(r'(?!i)^(pada|di)\s+kadar\s+\w+\s*,?\s*', '', x)
        x = re.sub(r'(?!i)^(jika)\s+jumlahnya\s+\w+\s*,?\s*', '', x)
        x = re.sub(r'(?!i)^(maka|akan|bisa)\s+', '', x)
        return x.strip('.,;')
    def _join(elems):
        uniq, seen = [], set()
        for t in [_strip_lead(z) for z in elems]:
            k = re.sub(r'\W+', '', t.lower())
            if k and k not in seen:
                seen.add(k); uniq.append(t)
        return '; '.join(uniq)
    pieces = []
    if low: pieces.append(f"Pada kadar rendah, {_join(low)}")
    if mid: pieces.append(f"Pada kadar sedang, {_join(mid)}")
    if high: pieces.append(f"Pada kadar tinggi, {_join(high)}")
    merged = (" ".join(others + ([ " ".join(pieces) + " " ] if pieces else []))).strip()
    return _repair_punct(merged)

# ===== Caption via yt-dlp & fallback =====
def _get_transcript_via_ytdlp_captions(video_id: str) -> str:
    """Ambil caption via yt-dlp. Prioritas: VTT/SRT/TTML → SRV3(XML) → JSON3."""

```

```

try:
    url = f"https://www.YouTube.com/watch?v={video_id}"
    with yt_dlp.YouTubeDL(**YTDLP_OPTS_COMMON) as ydl:
        info = ydl.extract_info(url, download=False)

    pools = [info.get("subtitles") or {}, info.get("automatic_captions") or {}]
    pref_langs = ["id", "id-ID", "id_id", "en", "en-US", "en-GB"]
    pref_exts = ["vtt", "srt", "ttml", "srv3", "json3", "xml"]

    def pick_track(d):
        for lang in pref_langs + list(d.keys()):
            if lang in d and d[lang]:
                tracks = sorted(
                    d[lang],
                    key=lambda x: pref_exts.index((x.get("ext") or "").lower())
                    if (x.get("ext") or "").lower() in pref_exts else 999
                )
                for tr in tracks:
                    u = tr.get("url")
                    e = (tr.get("ext") or "").lower()
                    if u:
                        return u, e
        return None, None

    for pool in pools:
        cap_url, ext = pick_track(pool)
        if not cap_url:
            continue
        r = HTTP.get(cap_url, timeout=YTDLP_TIMEOUT)
        if r.status_code != 200:
            continue
        raw = r.text

        if ext in ("vtt", "srt", "ttml"):
            txt = _parse_vtt_to_text(raw)
        elif ext in ("srv3", "xml"):
            txt = _parse_srv3_xml_to_text(raw) or (_parse_json3_to_text(raw) if
raw.strip().startswith("{}") else "")
        elif ext == "json3":
            txt = _parse_json3_to_text(raw)
        else:
            if raw.strip().startswith("{}"):
                txt = _parse_json3_to_text(raw)
            elif "<text" in raw[:200]:
                txt = _parse_srv3_xml_to_text(raw)
            else:
                txt = _parse_vtt_to_text(raw)

```

```

    txt = _clean_text(txt)
    if len(txt.split()) > 25:
        return txt

except Exception:
    print("[YTDLP CC] Exception:\n", traceback.format_exc(), file=sys.stderr)
    return ""

# ===== Chunking & Guidance =====
def _chunk_by_tokens_fast(t, max_tokens=MAX_CHUNK_TOKENS, stride_tokens=120):
    if not t.strip(): return []
    sents = [s.strip() for s in sent_tokenize(t) if s.strip()]
    chunks, cur = [], ""
    def tlen(s): return _tok_len(s)
    for s in sents:
        if not cur:
            cur = s; continue
        if tlen(cur + " " + s) <= max_tokens:
            cur += " " + s
        else:
            chunks.append(cur); cur = s
    if cur: chunks.append(cur)
    if stride_tokens > 0 and len(chunks) > 1:
        refined, tail = [], ""
        for ch in chunks:
            prefix = (tail + " " + ch).strip() if tail else ch
            refined.append(prefix)
            ids = tokenizer.encode(ch, add_special_tokens=False)
            tail = tokenizer.decode(ids[-stride_tokens:], skip_special_tokens=True) if len(ids) >
stride_tokens else ch
        chunks = refined
    return chunks

def _make_guidance_prefix(text_chunk: str):
    kw = _top_keywords(text_chunk, k=10)
    oracle = _tfidf_mmr_sentences(text_chunk, max_sents=6, diversity=0.72)
    guidance = ""
    return guidance, kw, oracle

# ===== Utils: Transcript (CC + Fallback) =====
def get_transcript_text(video_id: str) -> str:
    c = cache_get(f"transcript:{video_id}")
    if c is not None:
        return c

text = ""

```

```

try:
    langs_pref_manual = ['id','id-ID','en','en-US','en-GB']
    langs_pref_generated = ['id','id-ID','en','en-US','en-GB','ms','ja']
    try:
        items = YouTubeTranscriptApi.get_transcript(video_id, languages=langs_pref_manual +
        ['ms','ja'])
        if items: text = " ".join(seg.get("text","") for seg in items)
    except AttributeError:
        tl = YouTubeTranscriptApi.list_transcripts(video_id)
        items = None
        try: items = tl.find_manually_created_transcript(langs_pref_manual).fetch()
        except Exception: pass
        if not items:
            try: items = tl.find_generated_transcript(langs_pref_generated).fetch()
            except Exception: pass
        if not items:
            try:
                for tr in tl:
                    try:
                        items = tr.translate('id').fetch()
                        if items: break
                    except Exception: continue
            except Exception: pass
        if items: text = " ".join(seg.get("text","") for seg in items)
    except (TranscriptsDisabled, NoTranscriptFound, VideoUnavailable):
        text = ""
    except Exception:
        print("[YTA CC] Exception:\n", traceback.format_exc(), file=sys.stderr)
        text = ""

text = _dedupe_sentences(_drop_garbage_lines(_clean_text(text)))

if not text:
    text = _get_transcript_via_ytdlp_captions(video_id)

if not text and not FAST_MODE and USE_WHISPER and not USE_CC_ONLY:
    print("Transcript via Whisper fallback...")
    text = get_transcript_via_whisper(video_id)
    text = _dedupe_sentences(_drop_garbage_lines(_clean_text(text)))

if not text:
    meta = fetch_video_meta(video_id)
    desc = ""
    try:
        url = f"https://www.YouTube.com/watch?v={video_id}"
        with yt_dlp.YouTubeDL({}*YTDLP_OPTS_COMMON}) as ydl:
            info = ydl.extract_info(url, download=False)

```

```

        desc = (info.get("description") or "").strip()
    except Exception:
        pass
    text = _clean_text(desc or (meta.get("title") or ""))

    cache_set(f"transcript:{video_id}", text)
    return text

# ===== Ringkas → Lebih Pendek =====
def _trim_words(text: str, max_words: int = 150):
    words = text.split()
    if len(words) <= max_words:
        return text
    cut = " ".join(words[:max_words])
    m = re.search(r'(.+?[.!?...])?(?:\s|$)', cut)
    return m.group(1).strip() if m else cut.strip()

# ===== ROUGE & Standar =====
def compute_Rouge(reference: str, hypothesis: str):
    try:
        if not (reference and hypothesis): return None
        hyp_clean = re.sub(r"[^\s]*", "", hypothesis, flags=re.MULTILINE)
        scorer = Rouge_scorer.RougeScorer(['Rouge1', 'Rouge2', 'RougeL'], use_stemmer=True)
        sc = scorer.score(reference, hyp_clean)
        to3 = lambda x: {"precision": round(x.precision,2), "recall": round(x.recall,2), "f1":
round(x.fmeasure,2)}
        return {"Rouge1": to3(sc['Rouge1']), "Rouge2": to3(sc['Rouge2']), "RougeL": to3(sc['RougeL'])}
    except Exception:
        print("[ROUGE] Exception:\n", traceback.format_exc(), file=sys.stderr)
        return None

def meets_Rouge_standard(metrics: dict) -> bool:
    try:
        r1 = metrics["Rouge1"]["f1"]; r2 = metrics["Rouge2"]["f1"]; rl = metrics["RougeL"]["f1"]
        return (r1 >= ROUGE_MIN_R1F) and (r2 >= ROUGE_MIN_R2F) and (rl >= ROUGE_MIN_RLF)
    except Exception:
        return False

def _percentify_metrics(metrics: dict):
    if not metrics: return None
    def pct(x):
        try:
            return round(float(x) * 100.0, 2)
        except Exception:
            return ""
    out = {}
    for k in ["Rouge1", "Rouge2", "RougeL"]:

```

```

    if k in metrics and isinstance(metrics[k], dict):
        out[k] = {
            "precision": pct(metrics[k].get("precision")),
            "recall": pct(metrics[k].get("recall")),
            "f1": pct(metrics[k].get("f1")),
        }
    return out

def fmt_hms(seconds: float) -> str:
    seconds = int(seconds or 0)
    h = seconds // 3600; m = (seconds % 3600) // 60; s = seconds % 60
    return f"{h:02d}:{m:02d}:{s:02d}"

# ===== Helper: Paragraf rapi untuk tampilan =====
def paragraphize(text: str, max_words: int = 60, max_chars: int = 420) -> str:
    text = (text or "").strip()
    if not text:
        return ""
    sents = [s.strip() for s in sent_tokenize(text) if s.strip()]
    sents = _dedupe_by_similarity(sents, sim_thresh=0.90)
    paras, buf, wcount, ccount = [], [], 0, 0
    for s in sents:
        buf.append(s); wcount += len(s.split()); ccount += len(s)
        if wcount >= max_words or ccount >= max_chars:
            paras.append(" ".join(buf)); buf, wcount, ccount = [], 0, 0
    if buf: paras.append(" ".join(buf))
    return "\n".join(f"<p>{escape(p)}</p>" for p in paras)

# ===== Summarization core =====
def _summarize_chunk_batch(batch_texts):
    batch_texts = [{" " if t is None else str(t)}.strip() for t in batch_texts]
    batch_texts = [t if t else "." for t in batch_texts]
    inputs = _encode(batch_texts).to(_device)
    in_lens = [len(tokenizer.encode(t, add_special_tokens=False)) for t in batch_texts]
    target = int(max(80, min(260, round(TARGET_RATIO * max(in_lens))))))
    min_len = max(60, target - 30)
    max_len = max(target + 20, min(300, target + 60))
    out_ids = _generation_ids(inputs, min_len, max_len, mode="natural")
    outs = [tokenizer.decode(o, skip_special_tokens=True,
clean_up_tokenization_spaces=True).strip() for o in out_ids]
    outs = [_postprocess_summary(_clarity_polish_id(o)) for o in outs]
    return outs

def _clarity_polish_id(text: str) -> str:
    text = _normalize_indonesian(text)
    sents = [s.strip() for s in sent_tokenize(text) if s.strip()]
    def split_long(s, max_words=30):

```

```

words = s.split()
if len(words) <= max_words: return [s]
cut_points = [i for i,w in enumerate(words) if w.endswith(",")]
if not cut_points:
    mid = len(words)//2
    return [" ".join(words[:mid]).strip()+",", " ".join(words[mid:]).strip()]
mid = len(words)//2
idx = min(cut_points, key=lambda i: abs(i-mid))
return [" ".join(words[:idx+1]).strip(), " ".join(words[idx+1:]).strip()]
merged, buf = [], ""
for s in sents:
    if len(s.split()) < 6:
        buf = (buf + " " + s).strip()
        continue
    if buf:
        s = (buf + " " + s).strip(); buf = ""
    merged.append(s)
if buf: merged.append(buf)
polished = []
for s in merged:
    for p in split_long(s):
        if p and not re.search(r"[!?!...]"$, p): p += "."
        polished.append(p)
uniq, seen = [], set()
for s in polished:
    k = re.sub(r"\W+", "", s.lower())
    if k not in seen:
        seen.add(k); uniq.append(s)
final = []
for s in uniq:
    if not s: continue
    s = s[0].upper() + s[1:] if len(s) > 1 else s.upper()
    final.append(s)
return " ".join(final)

def _fuse_summaries(partials: list, global_oracle: list):
    if not partials: return ""
    seed = " ".join(partial)
    if global_oracle:
        seed += " " + " ".join(global_oracle[:8])
    inputs = _encode([seed]).to(_device)
    out_ids = _generation_ids(inputs, min_len=120, max_len=300, mode="precise")
    fused = tokenizer.decode(out_ids[0], skip_special_tokens=True,
clean_up_tokenization_spaces=True).strip()
    fused = _postprocess_summary(_clarity_polish_id(fused))
    return fused

```

```

def _sentence_quality_filter(sent: str) -> bool:
    s = sent.strip()
    if not s: return False
    s = _EMOJI_PAT.sub(" ", s)
    sym_ratio = sum(c in "\\\|@#<>[]{}_*^~" for c in s) / max(1, len(s))
    if sym_ratio > 0.18: return False
    if _CLICK_PAT.search(s) or _PROMO_PAT.search(s) or _META_PAT.search(s) or
    _OUTRO_PAT.search(s):
        return False
    if re.match(r"(?i)^\s*ringkasan\b", s): return False
    s2 = _BRACKETED.sub(" ", s).strip()
    return bool(s2)

def _postprocess_summary(summary: str) -> str:
    try:
        summary = _strip_instruction_noise(summary)
    except Exception:
        pass
    summary = _strip_heading_artifacts(summary)
    sents = [x.strip() for x in sent_tokenize(summary) if x.strip()]
    keep = []
    for s in sents:
        s = _BRACKETED.sub(" ", s)
        s = re.sub(r"["'"`]+", "", s)
        s = re.sub(r"\. {3}", ".", s)
        s = re.sub(r"!{3}", "!", s)
        if _sentence_quality_filter(s):
            keep.append(s)
    uniq_key, seen = [], set()
    for s in keep:
        k = re.sub(r"\W+", "", s.lower())
        if k not in seen:
            seen.add(k); uniq_key.append(s)
    uniq = _dedupe_by_similarity(uniq_key, sim_thresh=0.88)
    fixed = []
    for s in uniq:
        s = _smart_sentence_case(s)
        if not re.search(r"[!?...]$", s): s += "."
        fixed.append(s)
    out = " ".join(fixed)
    out = _normalize_indonesian(out)
    out = _collapse_dupes(out)
    out = re.sub(r"\s{2,}", " ", out).strip()
    fin_sents = [x.strip() for x in sent_tokenize(out) if x.strip()]
    fin_sents = _dedupe_by_similarity(fin_sents, sim_thresh=0.90)
    return " ".join(fin_sents)

```

```

def _final_sanitizetext(text: str) -> str:
    sents = [s.strip() for s in re.split(r'(?<=[.!?])\s+', text) if s.strip()]
    keep = []
    for s in sents:
        if _PROMO_PAT.search(s) or _CLICK_PAT.search(s) or _OUTRO_PAT.search(s):
            continue
        if len(s) < 12 and not re.search(r"\d", s):
            continue
        keep.append(s)
    keep = _dedupe_by_similarity(keep, sim_thresh=0.90)
    out = ". ".join(keep)
    out = _normalize_indonesian(out)
    out = _condense_quantifiers(out)
    out = _resolve_contradictions(out)
    out = _collapse_dupes(out)
    sents2 = [x.strip() for x in sent_tokenize(out) if x.strip()]
    sents2 = _dedupe_by_similarity(sents2, sim_thresh=0.92)
    out = _style_refine_id(" ".join(sents2))
    if out and not re.search(r"!.?..."$, out):
        out += "."
    return out

_THEMEES_RISK = re.compile(
    r"(?i)\b(keracunan|toksik|berbahaya|risiko|radikal\s+bebas|stres\s+oksidatif|merusak|kerusa
kan|kejang|fatal|paru[-
\s]*paru|organ|jaringan|gejala|pusing|mual|detak\s+jantung|lambat|perlambat)\b"
)
_THEMEES_USE = re.compile(
    r"(?i)\b(kegunaan|manfaat|medis|terapi|terkontrol|terkendali|pasien|gangguan\s+pernapas
an|astronaut|astronot|pra[-\s]*peluncuran|pre[-
\s]*breathe|mencegah|pencegahan|gelembung\s+gas|dekompresi)\b"
)
_CLEAN_SPACES = re.compile(r"\s{2,}")

def _choose_best_sents(source_text: str, max_sents: int = 5, theme_pat=None):
    sents = [s.strip() for s in re.split(r'(?<=[.!?])\s+', source_text) if s.strip()]
    themed = [s for s in sents if theme_pat and theme_pat.search(s)] if theme_pat else sents[:]
    try:
        base = ". ".join(sents)
        vec = TfidfVectorizer(min_df=1, max_df=0.95, ngram_range=(1,2))
        X = vec.fit_transform(themed) if themed else vec.fit_transform(sents)
        doc = vec.transform([base])
        sims = cosine_similarity(X, doc).ravel()
        idx = list(range(len(sims)))
        idx.sort(key=lambda i: sims[i], reverse=True)
        picked, picked_set = [], set()
        for i in idx:

```

```

    sent = (themed[i] if themed else sents[i]).strip()
    k = re.sub(r"\W+", "", sent.lower())
    if k in picked_set: continue
    picked.append(sent); picked_set.add(k)
    if len(picked) >= max_sents: break
return picked
except Exception:
    out = (themed[:max_sents] if themed else [])
    if len(out) < max_sents:
        for s in sents:
            if s not in out:
                out.append(s)
                if len(out) >= max_sents: break
    return out

def _shape_two_paragraphs(summary_text: str, source_text: str) -> str:
    pool = " ".join([
        _normalize_indonesian(summary_text or ""),
        _normalize_indonesian(source_text or "")
    ])
    pool = _collapse_dupes(_CLEAN_SPACES.sub(" ", pool))
    risk_sents = _choose_best_sents(pool, max_sents=4, theme_pat=_THEMES_RISK)
    use_sents = _choose_best_sents(pool, max_sents=3, theme_pat=_THEMES_USE)
    p1_default = ("Menghirup oksigen murni dalam jangka waktu lama dapat berbahaya karena
memicu "
        "produksi radikal bebas berlebihan dan merusak sel-sel tubuh.")
    p2_default = ("Meski berisiko bila berlebihan, oksigen murni bermanfaat dalam kondisi
terkontrol "
        "seperti terapi medis atau prosedur pra-peluncuran bagi astronaut.")
    def _tidy_lines(lines, cap=3):
        text = " ".join(lines[:cap])
        text = _resolve_contradictions(text)
        text = _style_refine_id(text)
        text = _collapse_dupes(text)
        text = re.sub(r"\s+([,;:!?])", r"\1", text)
        if text and not re.search(r"[!?!?...]$", text): text += "."
        return text
    para1 = _tidy_lines(risk_sents, cap=4) if risk_sents else _style_refine_id(p1_default)
    para2 = _tidy_lines(use_sents, cap=3) if use_sents else _style_refine_id(p2_default)
    if not para1.strip(): para1 = _style_refine_id(p1_default)
    if not para2.strip(): para2 = _style_refine_id(p2_default)
    final = f"{para1}\n\n{para2}"
    final = _CLEAN_SPACES.sub(" ", final).strip()
    return final

def summarize_text(text: str) -> str:
    src = (text or "").strip()

```

```

if not src:
    return ""
chunks = _chunk_by_tokens_fast(src)
partial, global_oracle_pool = [], []
with torch.inference_mode():
    for i in range(0, len(chunks), BATCH_SIZE):
        batch = chunks[i:i+BATCH_SIZE]
        if not batch: break
        guided_texts = []
        for ch in batch:
            guidance, _ oracle = _make_guidance_prefix(ch)
            guided_texts.append((guidance + ch).strip())
            global_oracle_pool.extend(oracle[:2])
        outs = _summarize_chunk_batch(guided_texts)
        partial.extend(outs)
narrative = _fuse_summaries(partial, global_oracle_pool[:8]).strip()
narrative = _strip_heading_artifacts(narrative)
narrative = _final_sanitize(narrative)
narrative = _trim_words(narrative, max_words=110)
pretty = _shape_two_paragraphs(narrative, src)
pretty = _trim_words(pretty, max_words=200)
return pretty

# ===== JOB WORKER =====
def _process_job(job_id: str, url: str):
    JOBS[job_id]["status"] = "running"
    t0 = time.time()
    try:
        vid = extract_video_id(url)
        with concurrent.futures.ThreadPoolExecutor(max_workers=2) as ex:
            fut_meta = ex.submit(fetch_video_meta, vid)
            fut_trans = ex.submit(get_transcript_text, vid)
            meta = fut_meta.result()
            transcript = fut_trans.result()

        if not transcript or len(transcript.split()) < 15:
            meta = meta or {}
            # Masih kosong? gabungkan judul + deskripsi agar UI tak kosong
            try:
                with yt_dlp.YouTubeDL(**YTDLP_OPTS_COMMON) as ydl:
                    info = ydl.extract_info(f"https://www.YouTube.com/watch?v={vid}", download=False)
                    desc = (info.get("description") or "")
            except Exception:
                desc = ""
            transcript = _clean_text("{} {}".format(meta.get("title", ""), desc)).strip()

        summary = summarize_text(transcript) if transcript else ""

```

```

if not summary and transcript:
    ext = _tfidf_mmr_sentences(transcript, max_sents=5, diversity=0.7)
    summary = _final_sanitise(" ".join(ext))
    summary = _trim_words(summary, max_words=180)

metrics = compute_Rouge(transcript, summary) if (transcript and summary) else None
meets = meets_Rouge_standard(metrics) if metrics else False
metrics_pct = _percentify_metrics(metrics)

elapsed = time.time() - t0
result = {
    "model_name": _model_name,
    "device": str(_device),
    "batch_size": BATCH_SIZE,
    "url": url,
    "title": (meta or {}).get("title", "-"),
    "published": (meta or {}).get("published", "-"),
    "duration": (meta or {}).get("duration", "-"),
    "transcript": transcript,
    "summary": summary,
    "metrics": metrics,
    "metrics_display": metrics_pct,
    "processing_seconds": round(elapsed, 2),
    "processing_hms": fmt_hms(elapsed),
    "r1_min": ROUGE_MIN_R1F,
    "r2_min": ROUGE_MIN_R2F,
    "rl_min": ROUGE_MIN_RLF,
    "meets_standard": meets
}

# Excel
xls = None
if transcript:
    row_data = {
        "Judul": result["title"],
        "Publikasi": result["published"],
        "Durasi": result["duration"],
        "URL": url,
        "Transkrip": transcript,
        "Ringkasan": summary,
        "Processing Seconds": result["processing_seconds"],
        "Processing HH:MM:SS": result["processing_hms"],
    }
if metrics_pct:
    row_data.update({
        "ROUGE-1 Precision": metrics_pct['Rouge1']['precision'],
        "ROUGE-1 Recall": metrics_pct['Rouge1']['recall'],
    })

```

```

        "ROUGE-1 F1":    metrics_pct['Rouge1']['f1'],
        "ROUGE-2 Precision": metrics_pct['Rouge2']['precision'],
        "ROUGE-2 Recall":  metrics_pct['Rouge2']['recall'],
        "ROUGE-2 F1":    metrics_pct['Rouge2']['f1'],
        "ROUGE-L Precision": metrics_pct['RougeL']['precision'],
        "ROUGE-L Recall":  metrics_pct['RougeL']['recall'],
        "ROUGE-L F1":    metrics_pct['RougeL']['f1']
    })
else:
    row_data.update({
        "ROUGE-1 Precision": "", "ROUGE-1 Recall": "", "ROUGE-1 F1": "",
        "ROUGE-2 Precision": "", "ROUGE-2 Recall": "", "ROUGE-2 F1": "",
        "ROUGE-L Precision": "", "ROUGE-L Recall": "", "ROUGE-L F1": ""
    })

df = pd.DataFrame([row_data])
buf = io.BytesIO()
with pd.ExcelWriter(buf, engine="openpyxl") as w:
    df.to_excel(w, index=False, sheet_name="Data")
buf.seek(0)
xls = buf
JOBS[job_id]["result"] = result
JOBS[job_id]["xls"] = xls
JOBS[job_id]["status"] = "done"
except Exception as e:
    JOBS[job_id]["status"] = "error"
    JOBS[job_id]["error"] = str(e)
    print("[JOB ERROR]", e)
    print(traceback.format_exc(), file=sys.stderr)

# ===== HTML Templates (UI TIDAK DIUBAH) =====
TEMPLATE_FORM = """
<!doctype html><html lang="id"><head><meta charset="utf-8">
<title>Ringkasan Ekstraktif</title>
<style>
    body{font-family:system-ui,-apple-system,BlinkMacSystemFont,'Segoe
    UI',Roboto,Oxygen,Ubuntu,Cantarell,'Open Sans','Helvetica Neue',sans-
    serif;background:#f8f9fa;color:#212529;padding:1rem}
    .container{max-width:900px;margin:auto;background:white;padding:2rem;border-
    radius:8px;box-shadow:0 4px 6px rgba(0,0,0,.1)}
    h1{text-align:center;color:#343a40}
    input[type=text]{width:100%;padding:12px;font-size:16px;box-sizing:border-box;border:1px solid
    #ced4da;border-radius:4px}
    .btn{margin-top:1rem;padding:12px 20px;font-
    size:16px;background:#007bff;color:white;border:none;border-
    radius:4px;cursor:pointer;width:100%}
    .btn:hover{background:#0056b3}

```

```

.warn{background:#fff3cd;color:#856404;padding:1rem;margin:1rem 0;border:1px solid
#ffeeba;border-radius:4px}
.muted{color:#6c757d;font-size:0.9rem;text-align:center}
</style></head><body>
<div class="container">
<h1>Ringkasan Video YouTube (Ekstraktif)</h1>
<p class="muted">Model: {{ model_name }} | Device: {{ device }} | Batch Size: {{ batch_size }}</p>
{% with messages = get_flashed_messages() %}
  {% if messages %}{% for m in messages %}<div class="warn">{{ m }}</div>{% endfor %}{% endif
%}
{% endwith %}
<form method="post" action="/">
  <label for="YouTube_url"><b>URL Video YouTube</b></label>
  <input type="text" id="YouTube_url" name="YouTube_url"
placeholder="https://www.YouTube.com/watch?v=..." required>
  <button class="btn" type="submit">Buat Ringkasan</button>
</form>
</div></body></html>
""""
TEMPLATE_STATUS = """"
<!doctype html><html lang="id"><head><meta charset="utf-8">
<title>Memproses...</title>
<style>
  body{font-family:system-ui,sans-serif;background:#f8f9fa;padding:1rem;text-
align:center;color:#343a40}
  .container{max-width:600px;margin:auto;background:white;padding:2rem;border-
radius:4px;box-shadow:0 4px 6px rgba(0,0,0,.1)}
  .loader{border:8px solid #f3f3f3;border-top:8px solid #3498db;border-
radius:50%;width:60px;height:60px;animation:spin 1s linear infinite;margin:auto}
  @keyframes spin{0%{transform:rotate(0deg)}100%{transform:rotate(360deg)}}
</style>
<script>
  const jobId = "{{ job_id }}";
  async function poll(){
    try {
      const r = await fetch(`/status/${jobId}?_=${Date.now()}`);
      if (!r.ok) { throw new Error('Network error'); }
      const js = await r.json();
      if(js.status === "done") window.location.href = `/result/${jobId}`;
      else if(js.status === "error") document.getElementById("msg").innerText = "Terjadi kesalahan:
" + (js.error || "Unknown");
    } catch(e) { /* ignore */ }
    setTimeout(poll, 1800);
  }
  window.onload = poll;
</script>
</head><body>

```

```

<div class="container">
  <h1>Sedang Memproses Video Anda...</h1>
  <div class="loader"></div>
  <p id="msg">Mohon tunggu. Video panjang atau tanpa subtitle bisa sedikit lebih lama.</p>
  <p style="color:#6c757d;font-size:0.9rem;">Job ID: {{ job_id }}</p>
</div>
</body></html>
""""

```

```

TEMPLATE_RESULT = """"

```

```

<!doctype html><html lang="id"><head><meta charset="utf-8">
<title>Hasil Ringkasan</title>
<style>
body{font-family:system-ui,sans-serif;background:#f8f9fa;color:#212529;padding:1rem}
.container{max-width:900px;margin:auto;background:white;padding:2rem;border-
radius:8px;box-shadow:0 4px 6px rgba(0,0,0,.1)}
h1{text-align:center;color:#343a40}
h2{border-bottom:2px solid #007bff;padding-bottom:0.5rem;margin-top:2rem;color:#343a40}
.panel-info{background:#e9ecef;padding:1rem;border-radius:4px;margin-bottom:1.5rem}
.panel-info p{margin:0.5rem 0}
.metrics table{width:100%;border-collapse:collapse;margin-top:1rem}
.metrics th, .metrics td{border:1px solid #dee2e6;padding:8px;text-align:center}
.metrics th{background-color:#f2f2f2}
.btn-group{display:flex;gap:1rem;margin-top:1.5rem}
.btn{flex:1;text-align:center;padding:12px;font-size:16px;text-decoration:none;border-
radius:4px;border:1px solid #007bff;color:#007bff;background:white}
.btn-primary{background:#007bff;color:white}
.btn:hover{opacity:0.8}
.small{color:#6c757d;font-size:0.9rem}
.content-block{border:1px solid #ced4da;background:#f8f9fa;border-radius:4px;padding:16px}
.content-block p{margin:.6rem 0;line-height:1.65;text-align:justify}
</style></head><body>
<div class="container">
<h1>Hasil Proses</h1>
<div class="panel-info">
  <p><b>Judul:</b> {{ title or '-' }}</p>
  <p><b>Publikasi:</b> {{ published or '-' }} | <b>Durasi:</b> {{ duration or '-' }}</p>
  <p><b>URL:</b> <a href="{{ url }}" target="_blank">{{ url }}</a></p>
  <p><b>Waktu Proses:</b> {{ processing_hms }} <span class="small">(~{{ processing_seconds }}
detik)</span></p>
</div>

<h2>Transkrip Asli</h2>
<div class="content-block">{{ transcript_fmt|safe }}</div>

<h2>Ringkasan Abstraktif Alami</h2>
<div class="content-block">{{ summary_fmt|safe }}</div>

```

```

<div class="metrics">
  <h2>Metrik ROUGE</h2>
  <table>
    <tr><th>Metrik</th><th>Precision</th><th>Recall</th><th>F1-Score</th></tr>
    {% if metrics_display %}
      <tr><td>ROUGE-1</td><td>{{ metrics_display.Rouge1.precision }}</td><td>{{
metrics_display.Rouge1.recall }}</td><td>{{ metrics_display.Rouge1.f1 }}</td></tr>
      <tr><td>ROUGE-2</td><td>{{ metrics_display.Rouge2.precision }}</td><td>{{
metrics_display.Rouge2.recall }}</td><td>{{ metrics_display.Rouge2.f1 }}</td></tr>
      <tr><td>ROUGE-L</td><td>{{ metrics_display.RougeL.precision }}</td><td>{{
metrics_display.RougeL.recall }}</td><td>{{ metrics_display.RougeL.f1 }}</td></tr>
    {% else %}
      <tr><td colspan="4">Gagal menghitung metrik.</td></tr>
    {% endif %}
  </table>
</div>
<div class="btn-group">
  <a class="btn" href="/download-excel/{{ job_id }}">Download Excel</a>
  <a class="btn btn-primary" href="/">Proses Video Lain</a>
</div>
</div></body></html>
''''

# ===== Routes =====
from werkzeug.serving import make_server

def _collapse_dupes(txt: str):
    # placeholder (dipakai di route)
    return txt

@app.route("/health")
def health():
    return "OK", 200

@app.route("/", methods=["GET", "POST"])
def index():
    if request.method == "POST":
        url = request.form.get("YouTube_url", "").strip()
        if not url:
            flash("Masukkan URL video YouTube.")
            return render_template_string(TEMPLATE_FORM, model_name=_model_name,
device=str(_device), batch_size=BATCH_SIZE)

        job_id = uuid.uuid4().hex[:12]
        JOBS[job_id] = {"status": "queued", "error": None, "result": None, "xls": None}

```

```

th = threading.Thread(target=_process_job, args=(job_id, url), daemon=True)
th.start()

return render_template_string(TEMPLATE_STATUS, job_id=job_id)

return render_template_string(TEMPLATE_FORM, model_name=_model_name,
device=str(_device), batch_size=BATCH_SIZE)

@app.route("/status/<job_id>")
def status(job_id):
    info = JOBS.get(job_id)
    if not info:
        return jsonify({"status": "error", "error": "Job tidak ditemukan"}), 404
    return jsonify({"status": info["status"], "error": info["error"]})

@app.route("/result/<job_id>")
def result(job_id):
    info = JOBS.get(job_id)
    if not info:
        flash("Job tidak ditemukan.")
        return redirect(url_for("index"))
    if info["status"] != "done":
        return render_template_string(TEMPLATE_STATUS, job_id=job_id)
    ctx = info["result"].copy()

    transcript_display = _collapse_dupes(ctx["transcript"])
    transcript_fmt = paragraphize(transcript_display) or "<p><em>Tidak ada subtitle tersedia;
diambil dari deskripsi video agar tampilan tidak kosong.</em></p>"
    summary_fmt = paragraphize(ctx["summary"]) or "<p><em>Ringkasan belum
tersedia.</em></p>"

    return render_template_string(
        TEMPLATE_RESULT,
        job_id=job_id,
        title=ctx["title"], published=ctx["published"], duration=ctx["duration"],
        url=ctx["url"], transcript=ctx["transcript"], summary=ctx["summary"],
        metrics_display=ctx["metrics_display"],
        processing_seconds=ctx["processing_seconds"], processing_hms=ctx["processing_hms"],
        r1_min=ctx["r1_min"], r2_min=ctx["r2_min"], rl_min=ctx["rl_min"],
        meets_standard=ctx["meets_standard"],
        transcript_fmt=transcript_fmt, summary_fmt=summary_fmt
    )

@app.route("/download-excel/<job_id>")
def download_excel(job_id):
    info = JOBS.get(job_id)
    if not info or info["status"] != "done" or not info["xls"]:
```

```

        flash("Data belum siap untuk diunduh.")
        return redirect(url_for("result", job_id=job_id))
    return send_file(info["xls"], as_attachment=True,
download_name=f"transkrip_ringkasan_{job_id}.xlsx")

# ===== Start Flask (background) + Tunnel Robust =====
def start_background_flask():
    global app_server, server_thread
    try:
        if 'app_server' in globals() and app_server:
            print("⌂ Menutup server Flask lama...")
            app_server.shutdown()
            time.sleep(0.5)
    except Exception:
        pass
    app_server = make_server("0.0.0.0", PORT, app)
    server_thread = threading.Thread(target=app_server.serve_forever)
    server_thread.daemon = True
    server_thread.start()

def start_public_tunnel():
    """
    Robust tunnel:
    - Verifikasi /health lokal.
    - Coba ngrok (jika ada AUTHTOKEN) → verifikasi /health publik.
    - Jika ngrok gagal / tidak ada token, coba Cloudflare (Quick Tunnel) → verifikasi /health publik.
    """
    import urllib.request

    try:
        urllib.request.urlopen(f"http://127.0.0.1:{PORT}/health", timeout=5).read()
        print("✅ Health /health OK pada port", PORT)
    except Exception as e:
        print("❌ Flask /health gagal:", repr(e))
        print("→ PORT mungkin bentrok. Ganti PORT (mis. 5001), lalu jalankan ulang.")
        raise

def _verify_public(url: str, retries=3, delay=1.2):
    ok = False; last_err = None
    for _ in range(retries):
        try:
            with urllib.request.urlopen(f"{url}/health", timeout=8) as resp:
                body = resp.read().decode("utf-8", "ignore")
            if resp.status == 200 and "OK" in body:
                ok = True; break
        except Exception as e:
            last_err = e; time.sleep(delay)

```

```

return ok, last_err

def try_ngrok_once():
    if not NGROK_AUTHTOKEN:
        raise RuntimeError("NGROK_AUTHTOKEN kosong")
    conf.get_default().auth_token = NGROK_AUTHTOKEN
    try:
        conf.get_default().region = NGROK_REGION
    except Exception:
        pass
    try:
        for t in ngrok.get_tunnels():
            ngrok.disconnect(t.public_url)
        ngrok.kill()
    except Exception:
        pass
    options = {"bind_tls": True}
    if NGROK_RESERVED_DOMAIN:
        options["hostname"] = NGROK_RESERVED_DOMAIN
    try:
        tun = ngrok.connect(addr=f"http://127.0.0.1:{PORT}", **options)
        url = tun.public_url
        print("🔗 Ngrok URL:", url)
        ok, err = _verify_public(url)
        if not ok:
            raise RuntimeError(f"Verifikasi ngrok gagal: {err}")
        print("✅ NGROK aktif & tervalidasi:", url)
        return url
    except ngrok_process.PyngrokNgrokError as e:
        msg = str(e)
        print("❌ Ngrok gagal:", msg)
        if "ERR_NGROK_4018" in msg:
            print("➡ Authtoken salah/belum verifikasi. Cek dashboard ngrok.")
        if "ERR_NGROK_108" in msg:
            print("➡ Ada agent ngrok lain aktif. Terminate di Dashboard > Agents.")
        raise
    except Exception as e:
        print("❌ Ngrok error umum:", e)
        raise

def try_cloudflare_once():
    time.sleep(random.uniform(0.5, 1.2))
    url = try_cloudflare(port=PORT)
    print("🔗 Cloudflare URL:", url)
    ok, err = _verify_public(url)
    if not ok:
        raise RuntimeError(f"Verifikasi Cloudflare gagal: {err}")

```

```

    print(" ✅ Cloudflare aktif & tervalidasi:", url)
    return url
last_err = None
if NGROK_AUTHTOKEN:
    for _ in range(2):
        try:
            return type("Obj", (), {"public_url": try_ngrok_once()})
        except Exception as e:
            last_err = e; time.sleep(1.0)
    for _ in range(3):
        try:
            url = try_cloudflare_once()
            return type("Obj", (), {"public_url": url})
        except Exception as e:
            last_err = e
        try:
            import subprocess, sys as _sys
            subprocess.run([_sys.executable, "-m", "pip", "install", "-U", "pycloudflared"],
check=False)
        except Exception:
            pass
        time.sleep(1.2)

    raise RuntimeError(f"Tidak bisa membuat tunnel publik. Last error: {last_err}")

print("▶ Menyalakan Flask server baru...")
start_background_flask()

# Cek kesehatan raise
try:
    import urllib.request
    urllib.request.urlopen(f"http://127.0.0.1:{PORT}/health", timeout=5).read()
    print(" ✅ Server lokal siap di", f"http://127.0.0.1:{PORT}")
except Exception as e:
    print(" ⚠️ Server lokal belum merespons /health:", repr(e))
    print(" → Umumnya: PORT bentrok. Ganti PORT (mis. 5001) lalu Restart runtime.")
    raiseraise

print("▶ Membuka tunnel publik...")
try:
    public_tunnel = start_public_tunnel()
    print(" ✅ URL Publik:", public_tunnel.public_url)
except Exception as e:
    print(" ❌ Gagal membuka tunnel publik:", e)
    print(" → Lanjutkan akses via output Colab (preview HTML) atau set NGROK_AUTHTOKEN untuk
stabilitas.")

```