202110370311169
Fathul Agit Darmawan
Prodi Informatika

BAB II
TINJAUAN PUSTAKA

2.1. Peringkasan Teks (*Text Summarization*)

Peringkasan teks merupakan salah satu bidang kajian penting dalam *Natural Language Processing* (NLP) yang bertujuan untuk menyajikan representasi singkat dari suatu teks dengan tetap mempertahankan informasi utama di dalamnya. Dalam praktiknya, peringkasan sangat dibutuhkan ketika seseorang harus memahami dokumen panjang dalam waktu yang singkat. Hal ini semakin relevan dengan meningkatnya ketersediaan data teks dari berbagai sumber, termasuk artikel berita, laporan, dan transkripsi video.

Secara umum, peringkasan teks dapat dibagi menjadi dua pendekatan utama, yaitu ekstraktif dan abstraktif [10]. Peringkasan ekstraktif bekerja dengan cara memilih kalimat atau frasa penting yang terdapat pada dokumen asli sehingga ringkasan yang dihasilkan tetap berisi kalimat asli tanpa perubahan struktur [11], [12]. Sementara itu, peringkasan abstraktif mencoba membangun kalimat baru berdasarkan pemahaman konteks, mirip dengan cara manusia membuat ringkasan [8].

2.2. Transkripsi Video

Transkripsi video adalah proses konversi audio menjadi teks melalui teknologi *Automatic Speech Recognition* (ASR). Salah satu teknologi yang cukup populer adalah Whisper, model yang dikembangkan oleh OpenAI, yang mampu menghasilkan transkripsi dengan tingkat akurasi tinggi meskipun terdapat perbedaan aksen, variasi bahasa, maupun kondisi kebisingan pada audio [13], [14], [15]. Keunggulan utama Whisper adalah kemampuannya untuk bekerja pada berbagai bahasa dan domain, sehingga cocok digunakan pada video publik seperti YouTube.

Walaupun demikian, hasil transkripsi dari video sering kali masih berupa teks mentah yang panjang, penuh pengulangan, jeda, serta perubahan topik yang mendadak. Karakteristik tersebut membuat transkripsi video berbeda dari dokumen tertulis seperti

artikel atau laporan. Ketidakrapian ini mengakibatkan kesulitan dalam memahami informasi inti secara cepat, sehingga peringkasan menjadi langkah penting agar transkripsi dapat dimanfaatkan secara lebih efektif.

Peringkasan transkripsi video diperlukan agar pengguna dapat memahami konten inti tanpa harus membaca keseluruhan teks. Terlebih dalam konteks video yang berdurasi panjang, ringkasan memberikan kemudahan bagi audiens yang memiliki keterbatasan waktu atau akses. Dengan demikian, integrasi antara teknologi transkripsi otomatis dan metode peringkasan teks menjadi solusi yang strategis dalam memproses informasi berbasis video.

2.3. Model BERT untuk Representasi Kalimat

Bidirectional Encoder Representations from Transformers (BERT) merupakan model berbasis transformer yang diperkenalkan oleh Google pada tahun 2018. Keunggulan utama BERT terletak pada mekanisme bidirectional training, yang memungkinkan model memahami konteks kata dari arah kiri maupun kanan secara bersamaan. Hal ini berbeda dengan model sebelumnya yang hanya membaca konteks dari satu arah, sehingga pemahaman semantik pada BERT menjadi lebih mendalam [16].

Dalam konteks peringkasan ekstraktif, BERT digunakan untuk menghasilkan representasi vektor dari setiap kalimat. Representasi ini memungkinkan pengukuran tingkat kesamaan semantik antar kalimat [17]. Misalnya, dua kalimat dengan makna yang mirip akan memiliki vektor representasi yang dekat dalam ruang vektor. Hal ini membantu dalam menentukan kalimat mana yang lebih penting untuk dijadikan bagian ringkasan.

Penelitian sebelumnya menunjukkan bahwa BERT memberikan hasil yang lebih baik dalam peringkasan ekstraktif dibandingkan dengan pendekatan tradisional seperti TF-IDF atau LSA [18]. Hal ini karena BERT tidak hanya memperhatikan frekuensi kata, tetapi juga hubungan semantik antar kata dalam sebuah kalimat. Dengan

demikian, penggunaan BERT menjadi sangat relevan untuk menganalisis transkripsi video yang panjang dan kompleks.

2.4. Algoritma K-Means Clustering

K-Means clustering adalah salah satu algoritma pengelompokan data (*unsupervised learning*) yang paling banyak digunakan. Algoritma ini bekerja dengan membagi sekumpulan data ke dalam sejumlah cluster berdasarkan jarak terdekat ke pusat cluster (*centroid*). Proses ini dilakukan secara iteratif sampai posisi centroid konvergen atau perubahan jarak minimum tercapai [19]. Rumus umum untuk menghitung fungsi objektif k-means ditunjukkan seperti **Eq. 2.1** berikut

$$J = \sum_{I=1}^{k} \sum_{x_j \in S_i} \left\| x_j - \mu_i \right\|^2 \qquad \textit{Eq. 2.1}$$

Dengan k adalah jumlah cluster, x_j adalah data ke-j yang termasuk dalam cluster S_i, dan μ_i adalah pusat cluster ke-i. Tujuan algoritma ini adalah meminimalkan fungsi objektif J, yaitu jumlah kuadrat jarak antara titik data dan pusat cluster.

Dalam konteks peringkasan teks, k-means digunakan untuk mengelompokkan kalimat-kalimat ke dalam beberapa cluster berdasarkan representasi vektornya. Dari setiap cluster, dipilih satu kalimat yang paling representative. Pendekatan ini bermanfaat untuk mengurangi redundansi informasi, menjaga keragaman topik, serta memastikan bahwa ringkasan mencakup keseluruhan isi dokumen secara proporsional.

2.5. Model BART untuk Peringkasan Abstraktif

BART (*Bidirectional and Auto-Regressive Transformers*) adalah model *encoder–decoder* yang dirancang untuk menghasilkan ringkasan baru secara lebih alami dan mudah dipahami [20]. Encoder membaca seluruh teks secara dua arah untuk memahami konteks keseluruhan, sedangkan decoder menghasilkan kalimat ringkasan secara bertahap. Secara umum, encoder mengubah setiap token menjadi representasi kontekstual.

Secara arsitektural, BART menggabungkan prinsip *denoising autoencoder*, yaitu melatih model untuk memulihkan teks asli dari versi yang "dirusak" melalui berbagai teknik seperti *token masking*, *token deletion*, hingga *sentence shuffling* [21]. Proses ini membuat BART lebih kuat dalam memahami struktur bahasa dan lebih fleksibel dalam menghasilkan ringkasan yang bersifat abstraktif, karena model belajar memahami konteks global sekaligus memprediksi teks secara autoregresif melalui decodernya.

Dalam proses pelatihan, BART mengoptimalkan fungsi *loss* berbasis *Negative Log-Likelihood (NLL)* antara keluaran decoder dan token referensi. Rumusnya dapat dituliskan seperti **Eq. 2.2** berikut.

$$\mathcal{L}_{NLL} = -\sum_{t=1}^{T} \log P\left(y_t \mid y_{<t}, x\right) \qquad \textit{Eq. 2.2}$$

Di mana x adalah input teks, y_t adalah token ringkasan yang benar pada langkah ke- t, dan $y_{<t}$ adalah token-token sebelumnya yang sudah dihasilkan [22]. Rumus ini memastikan bahwa model belajar menghasilkan token yang paling mungkin sesuai konteks sebelumnya dan makna keseluruhan teks.

Selain itu, perhatian (*self-attention* dan *cross-attention*) berperan penting dalam arsitektur BART. Mekanisme ini memungkinkan encoder dan decoder memusatkan perhatian pada bagian-bagian penting dalam teks sumber. Mekanisme *scaled dot-product attention* yang digunakan ditulis sebagai berikut.

$$Attention\left(Q, K, V\right) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \qquad \textit{Eq. 2.3}$$

Eq. 2.3 menggambarkan Q, K, dan V masing-masing adalah *query*, *key*, dan *value*, sedangkan d_k merupakan dimensi *key* [22]. Mekanisme ini membantu BART mengidentifikasi informasi penting yang akan dipertahankan dalam ringkasan.

Dengan kombinasi encoder bidirectional, decoder autoregresif, dan mekanisme *attention*, BART terbukti efektif dalam berbagai tugas peringkasan abstraktif dengan menghasilkan ringkasan yang lebih koheren, ringkas, dan mendekati gaya bahasa manusia [23], [24].

2.6. Metrik Evaluasi ROUGE

Evaluasi ringkasan otomatis umumnya dilakukan dengan menggunakan metrik ROUGE (*Recall-Oriented Understudy for Gisting Evaluation*). Metrik ini membandingkan ringkasan yang dihasilkan sistem dengan ringkasan acuan (*reference summary*) berdasarkan tingkat kesamaan n-gram, urutan kata, dan tumpang tindih frasa [25], [26].

Beberapa varian ROUGE yang paling sering digunakan adalah ROUGE-1, ROUGE-2, dan ROUGE-L. ROUGE-1 menghitung kesamaan unigram (kata tunggal), ROUGE-2 menghitung kesamaan bigram (pasangan kata), sedangkan ROUGE-L menghitung kesamaan berdasarkan *Longest Common Subsequence* (LCS). Secara umum, skor ROUGE dapat dihitung menggunakan rumus **Eq. 2.4** dibawah ini.

$$ROUGE - N = \frac{\sum_{S \in (Reference\ Summaries)} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in (Reference\ Summaries)} \sum_{gram_n \in S} Count(gram_n)} \quad \textit{Eq. 2.4}$$

Dengan n menunjukkan panjang n-gram, $Count_{match}(gram_n)$ adalah jumlah n-gram yang sama antara ringkasan sistem dengan ringkasan acuan, dan $Count(gram_n)$ adalah jumlah total n-gram dalam ringkasan acuan.

Metrik ROUGE banyak digunakan dalam penelitian peringkasan teks karena mampu memberikan evaluasi kuantitatif yang objektif terhadap kualitas ringkasan. Dalam penelitian ini, ROUGE digunakan untuk mengukur kesesuaian ringkasan otomatis hasil metode ekstraktif dengan ringkasan manual.