Analisis Komparatif Model Abstractive Summarization dalam Pembentukan Struktur Hierarkis Dokumen pada Arsitektur RAPTOR

Wahyu Syaifullah Jauharis Saputra[1], Andri Fauzan Adziima[2], Mukhamad Khafid Maassobirin[3]
[1,2,3]*Sains Data Universitas Pembangunan Nasional Veteran Jawa Timur*
[1]wahyu.s.j.saputra.if@upnjatim.ac.id
[2]andri.fauzan.fasilkom@upnjatim.ac.id
[3]22083010072@student.upnjatim.ac.id
Corresponding author email: wahyu.s.j.saputra.if@upnjatim.ac.id

ABSTRAK

Penelitian ini menganalisis performa tiga model *abstractive summarization* PEGASUS-X Large, Qwen2.5-7B-Instruct-AWQ, dan Gemini 2.5 Flash-Lite dalam pembentukan struktur hierarkis dokumen pada arsitektur RAPTOR. RAPTOR membangun pohon ringkasan multilevel melalui proses *embedding, clustering,* dan *summarization* secara berulang sampai dengan ditemukannya *root node*, sehingga kualitas model ringkasan sangat berpengaruh terhadap koherensi dan stabilitas struktur. Dataset QASPER digunakan sebagai sumber dokumen panjang berbasis bukti ilmiah, dengan batas ringkasan maksimum 150 token untuk menjaga konsistensi antar model. Evaluasi dilakukan pada beberapa aspek, seperti karakteristik struktural, keselarasan informasi *parent–child*, serta efisiensi kompresi. Hasil penelitian menunjukkan bahwa ketiga model mampu membentuk struktur stabil dengan kedalaman 2-3 level dan pola distribusi *node* yang serupa. Namun, kualitas ringkasan berbeda secara signifikan. PEGASUS-X Large memberikan hasil terbaik dengan skor ROUGE dan BERTScore tertinggi serta *compression ratio* paling rendah, menunjukkan keseimbangan optimal antara kompresi dan retensi informasi. Qwen2.5-7B memberikan performa menengah, sedangkan Gemini 2.5 Flash-Lite cenderung menghasilkan ringkasan yang lebih abstraktif. Secara keseluruhan, PEGASUS-X Large merupakan model paling efektif untuk digunakan sebagai komponen *summarization* pada arsitektur RAPTOR.

Keywords: *Abstractive Summarization,* Struktur Hierarkis RAPTOR, QASPER

I. PENDAHULUAN

Perkembangan teknologi digital yang semakin pesat pada era modern mendorong munculnya berbagai pendekatan baru dalam pengolahan informasi berskala besar. Lonjakan volume data yang besar terutama data yang tidak terstruktur menjadi tantangan tersendiri dalam pengolahannya untuk mendapatkan informasi yang relevan secara efisien. *Information Retrieval* hadir menjadi pilar penting yang berperan untuk membantu pengguna dalam menemukan informasi yang umumnya dari data tidak terstruktur seperti teks pada koleksi data berskala besar [1]. *Information Retrieval* secara luas mencakup mulai dari produksi, organisasi, penyimpanan, pengambilan, penyebaran, dan penggunaan informasi yang ada [2]. Beberapa teknik terdahulu dari *Information Retrieval s*eperti TF-IDF [3]dan BM25 [4] yang berbasis pencocokan leksikal telah berevolusi menuju pendekatan *dense retrieval* yang memanfaatkan representasi semantik dari *pre-trained language models* (PLM) sehingga mampu

memahami teks secara konseptual [5]. PLM sendiri menjadi fondasi utama dalam lahirnya model dengan kapabilitas yang lebih besar yaitu *Large Language Models* (LLM) yang mampu memahami konteks panjang serta menghasilkan teks yang menyerupai manusia sehingga berpotensi menjadi menjadi pengetahuan universal. Namun LLM sendiri memiliki beberapa kelemahan, seperti masih terbatas oleh bias data pralatih, kecenderungan menghasilkan halusinasi, kurangnya transparansi, hingga kesulitan dalam menjaga pengetahuan tetap mutakhir [6]. Untuk mengatasi kelemahan tersebut, muncul arsitektur *Retrieval-Augmented Generation* (RAG) yang memadukan model generatif dengan mekanisme *retrieval* dari sumber pengetahuan eksternal, sehingga dapat melengkapi memori parametrik statis LLM dan menghasilkan jawaban yang lebih faktual dan kontekstual [7]. Namun, RAG standar yang hanya memanfaatkan *flat retrieval* relatif kurang efektif ketika dihadapkan dengan dokumen yang kompleks.

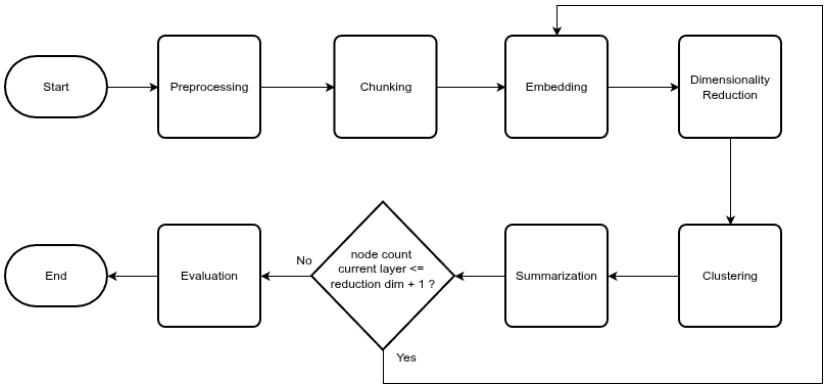
Salah satu pendekatan yang menawarkan solusi terhadap keterbatasan RAG dalam menangani dokumen kompleks adalah RAPTOR (*Recursive Abstractive Processing for Tree-Organized Retrieval*) yang dengan kombinasi RAPTOR dan GPT-4 dilaporkan meningkatkan akurasi absolut 20 poin pada dataset QuALITY dibandingkan *state of the art* sebelumnya [8]. Berbeda dari metode retrieval konvensional yang hanya mengambil potongan konteks pendek, RAPTOR membangun representasi dokumen dalam bentuk pohon ringkasan multilevel melalui proses *embedding*, *clustering*, dan *summarization* secara rekursif [8]. Struktur hierarkis ini memungkinkan sistem menangkap hubungan antarbagian teks global, sehingga informasi penting tetap terjaga. Pada mekanisme RAPTOR, setiap node dalam pohon merupakan hasil peringkasan abstraktif dari sekelompok *chunk*, sehingga kualitas ringkasan sangat menentukan koherensi, kedalaman, dan stabilitas struktur yang dihasilkan. Dengan demikian, pemilihan model *abstractive summarization* memegang peran kunci dalam pembentukan hierarki tersebut. Model peringkasan yang tidak stabil dapat menghasilkan ringkasan yang terlalu dangkal, redundan, atau kehilangan informasi penting, yang pada akhirnya mengurangi efektivitas *retrieval.* Oleh karena itu, pemahaman terhadap karakteristik dan performa berbagai model summarization menjadi krusial untuk mengoptimalkan representasi hierarkis dalam arsitektur RAPTOR.

RAPTOR sendiri memakai GPT-3.5-Turbo sebagai model *summarization*, namun memiliki sejumlah kelemahan yang membuatnya kurang sesuai untuk tugas *abstractive summarization*. Berdasarkan hasil analisis Ye et al. [9] model ini menunjukkan robustness rendah, terbukti dari penurunan F1 dan EM yang sangat tajam pada SQuAD ketika input sedikit diubah. Selain itu, GPT-3.5 Turbo bersifat *chat-oriented* sehingga cenderung menghasilkan keluaran panjang dan *verbose*, bertentangan dengan kebutuhan ringkasan padat pada RAPTOR. Performanya juga tidak konsisten antartugas, misalnya menjadi yang terlemah pada POS tagging, NER, dan beberapa skenario *zero-shot*. Model ini juga sensitif terhadap panjang *prompt*, menyebabkan ringkasan yang dihasilkan tidak stabil antarlevel. Oleh karena itu, penelitian ini akan melakukan analisis komparatif model *abstractive summarization* untuk menguji beberapa model yang berpotensi dapat menghasilkan ringkasan terbaik pada arsitektur RAPTOR. Model pertama yang digunakan adalah PEGASUS-X Large yang dipilih karena merupakan model yang memang dirancang secara khusus untuk *abstractive summarization.* PEGASUS menggunakan mekanisme *Gap-Sentences Generation* (GSG) sebagai *objective* pra-latihannya, yang terbukti memberikan performa unggul pada *long-document summarization* dan berbagai *benchmark* seperti CNN/DailyMail, XSum, dan arXiv [10]. Varian PEGASUS-X dikembangkan dengan dukungan konteks panjang dan pemrosesan paralel yang lebih efisien, sehingga lebih sesuai untuk pembentukan pohon ringkasan multilevel pada RAPTOR. Selain itu, sifatnya sebagai model *encoder–decoder* membuat ringkasan yang dihasilkan lebih padat dan tidak

verbose, sebuah karakteristik penting untuk meningkatkan efektivitas *traversal*. Kemudian Qwen2.5-7B-Instruct-AWQ dipilih karena memiliki kemampuan pemrosesan konteks panjang dan kualitas bahasa yang kuat berdasarkan evaluasi ilmiah terbaru. Qwen2.5 *Technical Report* menunjukkan bahwa model ini mendukung konteks hingga 32K token dan dilatih pada 18T token, sehingga cocok untuk dokumen berskala panjang yang digunakan dalam RAPTOR. Selain itu, Qwen telah terbukti kompetitif pada *long-context summarization* melalui evaluasi pada benchmark CNNSum yang mencakup teks hingga 128K token [11]. Dukungan kuantisasi AWQ menjadikan model ini efisien dan stabil untuk penggunaan dalam *pipeline* hierarkis. Selanjutnya Gemini 2.5 Flash-Lite sebagai kandidat model *abstractive summarization* yang pada *Technical report* Gemini 2.5 menunjukkan bahwa keluarga model ini dirancang untuk *reasoning* dengan dukungan konteks sangat panjang serta varian Flash-Lite dibuat dengan tujuan menciptakan model yang memang dirancang untuk performa tinggi namun tetap ekonomis [12] Dengan kemampuan Gemini yang sudah terbukti maka Gemini 2.5 Flash-Lite layak dievaluasi lebih lanjut sebagai model *summarization* yang efisien dan skalabel dalam arsitektur RAPTOR.

Oleh karena itu, penelitian ini bertujuan untuk melakukan analisis komparatif model *abstractive summarization* dalam pembentukan struktur hierarkis pada arsitektur RAPTOR. Secara khusus, penelitian ini mengevaluasi tiga model ringkasan yaitu Pegasus-X Large, Qwen2.5-7B-Instruct-AWQ, dan Gemini 2.5 Flash-Lite untuk menilai stabilitas, konsistensi, dan kualitas ringkasan yang dihasilkan pada proses pembangunan *node* baru. Evaluasi dilakukan menggunakan dataset QASPER, salah satu dataset utama yang digunakan dalam penelitian RAPTOR dan mewakili skenario *long-context question answering* berbasis bukti ilmiah. Melalui analisis ini, penelitian ini diharapkan dapat mengidentifikasi model *summarization* yang paling optimal sebagai komponen pembentuk struktur hierarkis dokumen, serta memberikan kontribusi pada pengembangan sistem RAG yang lebih adaptif, akurat, dan efisien dalam memproses dokumen kompleks.

II. METODOLOGI PENELITIAN



Gambar 1. Diagram Alir Penelitian

Proses pembentukan struktur hierarkis dari dokumen dimulai dari tahap *preprocessing* hingga evaluasi akhir terhadap kualitas struktur yang dihasilkan. Penelitian ini menggunakan 50 sampel dokumen dari dataset QASPER yang dipilih secara acak sebagai representasi dokumen panjang berbasis bukti ilmiah. Tahap pertama yang dilakukan adalah p*reprocessing*, yaitu proses pembersihan dan penyiapan teks sebelum dilakukan pemrosesan lanjutan. *Preprocessing* mencakup normalisasi dan

penghapusan bagian teks yang tidak diperlukan serta penyesuaian struktur agar berada dalam format yang stabil dan konsisten untuk diproses pada tahap berikutnya. Selanjutnya adalah proses *chunking* dilakukan dengan membagi dokumen menjadi potongan teks kecil berukuran 100 token per *chunk.* Apabila *chunk* terpotong di tengah kalimat, maka kalimat tersebut dimasukkan ke dalam *chunk* berikutnya.

Potongan teks hasil *chunking* kemudian memasuki proses konstruksi *tree* yang berlangsung secara berulang pada setiap level atau *layer*. Dimulai dengan proses *embedding*, yaitu proses mengubah setiap teks menjadi representasi vektor menggunakan model *multilingual-e5-large-instruct* yang merupakan model *embedding* multibahasa yang dilatih dengan skema *contrastive pre-training* dan *instruction tuning* sehingga mampu menangkap kesetaraan semantik lintas bahasa serta menunjukkan kinerja yang kompetitif pada berbagai tugas *retrieval* dan s*emantic similarity* [13]. Representasi vektor yang dihasilkan kemudian direduksi dimensinya menggunakan algoritma *Uniform Manifold Approximation and Projection* (UMAP) yang merupakan teknik reduksi dimensi *non linier* berbasis teori *manifold* yang dirancang untuk mempertahankan struktur topologi lokal dan global sekaligus tetap relevan untuk data berukuran besar [14]. Penggunaan UMAP bertujuan mengurangi kompleksitas komputasi pada tahap selanjutnya namun tetap menjaga makna semantik dalam ruang berdimensi lebih rendah.

Selanjutnya, hasil reduksi dimensi diproses melalui *clustering* menggunakan *Gaussian Mixture Model* (GMM) yang memandang distribusi data sebagai campuran beberapa distribusi *Gaussian* sehingga mampu melakukan *soft clustering*, yaitu memberikan probabilitas keanggotaan suatu titik data terhadap masing-masing klaster [15]. Pendekatan ini lebih sesuai dengan karakteristik teks, di mana satu potongan teks dapat mengandung lebih dari satu topik. *Node-node* dalam *cluster* yang sama memiliki kedekatan semantik yang kuat dan topik yang serupa. Kemudian kumpulan *node* ini digabungkan dan diringkas menjadi sebuah *node* baru melalui tahap *summarization*. Model *summarization* menghasilkan ringkasan yang mewakili keseluruhan isi cluster tersebut, dan ringkasan inilah yang menjadi node pada level berikutnya di dalam struktur hierarkis. Terdapat 3 model *summarization* yang digunakan dalam penelitian ini, yaitu Pegasus-X Large, Qwen2.5-7B-Instruct-AWQ, dan Gemini 2.5 Flash-Lite dan akan dilakukan analisis komparatif untuk menilai model mana yang memiliki hasil terbaik. Pada tahap *summarization,* setiap model dibatasi untuk menghasilkan ringkasan dengan panjang maksimum 150 token agar proses konstruksi *tree* tetap konsisten pada seluruh model. Batas ini diterapkan secara seragam dan menjadi salah satu faktor yang dapat mempengaruhi kepadatan serta kesetiaan ringkasan yang dihasilkan, terutama pada model yang secara alami menghasilkan keluaran lebih panjang.

Proses dari *embedding* sampai dengan *clustering* akan dilakukan secara berulang hingga kondisi *node count current layer* ≤ *reduction dimension* + 1. Jika jumlah node masih lebih besar dari ambang batas tersebut, proses dilanjutkan ke level berikutnya. Sebaliknya, jika jumlah *node* sudah berada di bawah atau sama dengan batas minimal, proses konstruksi *tree* dihentikan dan struktur hierarkis dianggap telah selesai terbentuk. Tahap terakhir adalah *evaluation,* yaitu penilaian kualitas struktur hierarkis yang dibentuk oleh setiap model *summarization*. Evaluasi dalam penelitian ini difokuskan pada beberapa aspek utama. *Parent children consistency,* yaitu menilai koherensi vertikal antara ringkasan induk dan gabungan teks anak menggunakan ROUGE-1, ROUGE-L, dan BERTScore. Selanjutnya aspek *compression* & *retention*, yang mengukur efisiensi *compression ratio* serta kemampuan mempertahankan informasi *children* menggunakan *retention* ROUGE dan *retention* BERTScore. Selain itu, karakteristik struktural seperti *depth, node count, nodes* per level, distribusi *node*, dan total token per *tree* juga dianalisis sebagai informasi pendukung. Hasil evaluasi ini kemudian dibandingkan antar model untuk mengidentifikasi model *summarization*

yang menghasilkan struktur paling informatif, koheren, efisien, dan representatif.

III. HASIL DAN ANALISIS

Bagian ini akan menyajikan hasil dan analisis dari penelitian ini mengenai pembentukan struktur hierarkis menggunakan tiga model peringkasan, yaitu Pegasus X-Large, Qwen2.5–7B Instruct AWQ, dan Gemini 2.5 Flash-Lite, serta analisis terhadap beberapa aspek utama evaluasi, antara lain, evaluasi struktural, kualitas hubungan *parent–child*, dan evaluasi *compression–retention*.

III.1 Struktural Hierarkis

Evaluasi struktural bertujuan memahami karakteristik hierarki yang dihasilkan oleh ketiga model peringkasan. Parameter yang dianalisis meliputi *depth*, jumlah *node*, distribusi *node* per level, dan jumlah token per level. *Depth* dihitung berdasarkan index level tertinggi pada *layer_to_nodes*. Apabila kedalaman maksimum yang terdeteksi adalah d, maka jumlah level sebenarnya adalah d+1. Nilai depth untuk masing-masing model dirangkum pada Tabel 1.

Tabel 1. Rata-rata *depth*, *node count*, dan total token

Model	Avg Depth	Avg Node Count	Avg Total Token
Pegasus-X-Large	1.26	70.52	7174.80
Qwen2.5-7B-Instruct-AWQ	1.34	71.12	7356.60
Gemini 2.5 Flash-Lite	1.30	71.10	7306.14

Tabel 1 menunjukkan karakteristik umum struktur pohon ringkasan yang dihasilkan oleh ketiga model peringkasan. Nilai average depth pada kisaran 1.26–1.34 menunjukkan bahwa sebagian besar dokumen memiliki struktur dengan 2 hingga 3 level hierarki. Depth yang relatif dangkal ini mencerminkan bahwa ketiga model mampu melakukan kompresi informasi secara efektif tanpa menghasilkan rantai peringkasan terlalu panjang, yang sesuai dengan desain hierarchical summarization pada RAPTOR.

Jumlah node yang relatif seragam (70–72 *node* per dokumen) mengindikasikan bahwa variasi antamodel tidak terlalu besar dalam menentukan jumlah unit ringkasan yang dibentuk. Hal ini menunjukkan bahwa ketiga model mengekstraksi tingkat granularitas ringkasan yang hampir sama. Dari sisi total token, nilai rata-rata sekitar 7.1K–7.3K token per dokumen menunjukkan bahwa keseluruhan informasi yang terdapat pada struktur hierarki masih berada pada level detail yang memadai. Secara keseluruhan, Pegasus-X-Large sedikit lebih efisien daripada dua model lain karena menghasilkan total token paling rendah.

Tabel 2. Distribusi *Node* per *layer*

Model	L0	L1	L2
Pegasus-X-Large	59.4	10.14	0.98
Qwen2.5-7B-Instruct-AWQ	59.4	10.46	1.26
Gemini 2.5 Flash-Lite	59.4	10.60	1.10

Tabel 2 memperlihatkan bagaimana jumlah *node* tersebar pada setiap level hierarki. Pada L0, seluruh model menghasilkan sekitar 59 *node*, yang menggambarkan *chunk* awal dokumen. Ini menunjukkan konsistensi proses *chunking* pada dataset. Pada level 1 (L1), jumlah *node* turun secara drastis menjadi sekitar 10–11 *node*, menandakan bahwa model melakukan penggabungan dan peringkasan informasi secara agresif

namun tetap stabil. Pada level 2 (L2), hanya tersisa sekitar 1 *node*, yang merupakan *root summary*. Perbedaan kecil di antara model menunjukkan bahwa beberapa model, terutama Qwen, memilih mempertahankan lebih banyak representasi informasi pada level tertinggi dibandingkan Pegasus dan Gemini.

Tabel 3. Total token per *layer*

Model	L0	L1	L2
Pegasus-X-Large	5200.32	1798.96	175.52
Qwen2.5-7B-Instruct-AWQ	5200.32	1923.00	233.28
Gemini 2.5 Flash-Lite	5200.32	1904.76	201.06

Tabel 3 menampilkan jumlah token yang dihasilkan oleh setiap model pada tiap level hierarki. Pada L0, seluruh model menghasilkan jumlah token yang identik (sekitar 5200 token), mencerminkan konsistensi ukuran chunk input. Pada L1, jumlah token turun secara signifikan menjadi sekitar 1800–1900 token, menandakan kompresi informasi sebesar ±65%. Penurunan ini menunjukkan bahwa model mampu merangkum informasi dari *node* dasar secara efisien. Pada L2, seluruh model menghasilkan ringkasan root dengan panjang 175–233 token, yang berarti tingkat kompresi total mencapai sekitar 96% dari L0. Perbedaan token antara model pada L2 menunjukkan variasi gaya peringkasan dengan Pegasus-X-Large adalah yang paling ringkas, Qwen2.5-7B-Instruct-AWQ merupakan yang paling panjang, dan Gemini 2.5 Flash-Lite yang berada di tengah-tengah. Hasil ini memperlihatkan bahwa ketiga model mampu membentuk ringkasan sangat padat pada level tertinggi tanpa kehilangan struktur informasi.

III.2 Hubungan *Parent-Child*

Tabel 4. Rata-rata ROUGE & BERTScore Parent–Child

Model	Avg ROUGE-1	Avg ROUGE-L	Avg BERTScore
Pegasus-X-Large	0.5440	0.5082	0.9096
Qwen2.5-7B-Instruct-AWQ	0.3939	0.2703	0.8593
Gemini 2.5 Flash-Lite	0.3624	0.2289	0.8453

Tabel 4 menunjukkan kualitas hubungan antara *node parent* dan *children* pada setiap level, diukur menggunakan ROUGE-1, ROUGE-L, dan BERTScore. Hasil ini mengevaluasi seberapa baik ringkasan pada level lebih tinggi tetap mempertahankan informasi dari *node* level sebelumnya. Model Pegasus-X-Large menunjukkan performa terbaik pada seluruh metrik, dengan ROUGE-1 sebesar 0.5440, ROUGE-L sebesar 0.5082, dan BERTScore sebesar 0.9096. Hal ini menunjukkan bahwa ringkasan yang dihasilkan Pegasus lebih *faithful*, yaitu lebih selaras dengan isi *children* baik secara leksikal maupun semantik. Model Qwen2.5–7B memiliki performa menengah, sementara Gemini 2.5 menghasilkan skor paling rendah. Skor Gemini yang lebih kecil menunjukkan bahwa model ini melakukan *summarization* yang lebih abstraktif, sehingga sering kali melepaskan sebagian detail leksikal *children*. Skor BERT yang tetap tinggi pada seluruh model (≥ 0.84) menandakan bahwa informasi semantik tetap dipertahankan secara umum meskipun gaya ringkasannya berbeda.

III.3 *Compression Ratio*

Tabel 5. Rata-rata *Compression Ratio*

Model	Avg Compression Ratio
Pegasus-X-Large	0.3266
Qwen2.5-7B-Instruct-AWQ	0.3385

Gemini 2.5 Flash-Lite	0.3412

Compression Ratio mengukur seberapa ringkas ringkasan parent dibandingkan total token *children*. Nilai yang lebih rendah berarti ringkasan lebih padat. Pada tabel terlihat bahwa Pegasus-X-Large memiliki nilai terendah (0.3266), menandakan model ini paling efisien dalam melakukan kompresi tanpa memperpanjang ringkasan. Qwen2.5-7B dan Gemini 2.5 menghasilkan ringkasan yang sedikit lebih panjang (sekitar 0.338-0.341), menandakan kecenderungan mempertahankan detail lebih banyak di *parent node*. *Compression Ratio* yang rendah pada ketiga model menunjukkan bahwa semua model menjalankan proses peringkasan secara efektif, dengan kompresi yang mencapai lebih dari 60% di setiap langkah peringkasan.

IV. KESIMPULAN

Penelitian ini menyimpulkan bahwa ketiga model peringkasan Pegasus-X Large, Qwen2.5-7B Instruct AWQ, dan Gemini 2.5 Flash-Lite mampu membentuk struktur hierarkis RAPTOR secara stabil dengan *depth* rata-rata 2-3 level dan pola distribusi *node* yang konsisten. Meskipun demikian, kualitas struktur yang terbentuk menunjukkan variasi penting antar model. Pegasus-X Large terbukti paling unggul karena menghasilkan ringkasan yang paling koheren dan paling selaras dengan isi *children*, ditunjukkan oleh skor ROUGE-1, ROUGE-L, dan BERTScore tertinggi. Selain itu, Pegasus-X Large juga menghasilkan kompresi yang paling efisien tanpa mengorbankan kesetiaan informasi, sehingga struktur pohon yang dibentuk lebih ringkas namun tetap informatif.

Sementara itu, Qwen2.5-7B Instruct AWQ menunjukkan kinerja menengah dengan retensi semantik yang baik meskipun ringkasannya relatif lebih panjang. Gemini 2.5 Flash-Lite menghasilkan ringkasan paling abstraktif dengan *compression ratio* tertinggi, namun kesetiaan leksikalnya lebih rendah dibanding dua model lainnya. Secara keseluruhan, Pegasus-X Large merupakan model yang paling optimal untuk digunakan sebagai komponen *summarization* dalam arsitektur RAPTOR pada dokumen panjang berbasis bukti ilmiah, karena menawarkan keseimbangan terbaik antara kompresi, koherensi, dan stabilitas struktur.

REFERENSI

[1] J. Guo *et al.*, “A Deep Look into neural ranking models for information retrieval,” *Inf. Process. Manag.*, vol. 57, no. 6, p. 102067, Nov. 2020, doi: 10.1016/j.ipm.2019.102067.

[2] W. Dietmar, “Applications Of Informetrics To Information Retrieval Research,” *Informing Sci. Int. J. Emerg. Transdiscipl.*, vol. 3, Jan. 2000, doi: 10.28945/581.

[3] G. Salton and C. Buckley, “Term-weighting approaches in automatic text retrieval,” *Inf. Process. Manag.*, vol. 24, no. 5, pp. 513–523, Jan. 1988, doi: 10.1016/0306-4573(88)90021-0.

[4] S. Robertson and H. Zaragoza, “The Probabilistic Relevance Framework: BM25 and Beyond,” *Found. Trends Inf. Retr.*, vol. 3, pp. 333–389, Jan. 2009, doi: 10.1561/1500000019.

[5] Z. Xu *et al.*, “A Survey of Model Architectures in Information Retrieval,” Aug. 26, 2025, *arXiv*: arXiv:2502.14822. doi: 10.48550/arXiv.2502.14822.

[6] R. Yang *et al.*, “Retrieval-augmented generation for generative artificial intelligence in health care,” *Npj Health Syst.*, vol. 2, no. 1, p. 2, Jan. 2025, doi: 10.1038/s44401-024-00004-1.

[7] P. Lewis *et al.*, “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks,” in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2020, pp. 9459–9474. Accessed: Nov. 30, 2025. [Online]. Available:

https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html

[8] P. Sarthi, S. Abdullah, A. Tuli, S. Khanna, A. Goldie, and C. D. Manning, "RAPTOR: RECURSIVE ABSTRACTIVE PROCESSING FOR TREE-ORGANIZED RETRIEVAL," 2024.

[9] J. Ye *et al.*, "A Comprehensive Capability Analysis of GPT-3 and GPT-3.5 Series Models," Dec. 23, 2023, *arXiv*: arXiv:2303.10420. doi: 10.48550/arXiv.2303.10420.

[10] J. Zhang, Y. Zhao, M. Saleh, and P. J. Liu, "PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization," July 10, 2020, *arXiv*: arXiv:1912.08777. doi: 10.48550/arXiv.1912.08777.

[11] L. Wei, H. Yan, X. Lu, J. Zhu, J. Wang, and W. Zhang, "CNNSum: Exploring Long-Context Summarization with Large Language Models in Chinese Novels," June 02, 2025, *arXiv*: arXiv:2412.02819. doi: 10.48550/arXiv.2412.02819.

[12] "gemini_v2_5_report.pdf." Accessed: Nov. 30, 2025. [Online]. Available: https://storage.googleapis.com/deepmind-media/gemini/gemini_v2_5_report.pdf

[13] L. Wang, N. Yang, X. Huang, L. Yang, R. Majumder, and F. Wei, "Multilingual E5 Text Embeddings: A Technical Report," Feb. 08, 2024, *arXiv*: arXiv:2402.05672. doi: 10.48550/arXiv.2402.05672.

[14] L. McInnes, J. Healy, N. Saul, and L. Großberger, "UMAP: Uniform Manifold Approximation and Projection," *J. Open Source Softw.*, vol. 3, no. 29, p. 861, Sept. 2018, doi: 10.21105/joss.00861.