

Text Summarization terhadap Berita Bahasa Indonesia menggunakan Dual Encoding

Made Dwiki Budi Laksana^{a1}, AAIN Eka Karyawati^{a2}, Luh Arida Ayu Rahning Putri^{a3}, I Wayan Santiyasa^{a4}, Ngurah Agus Sanjaya ER^{a5}, I Gusti Agung Gede Arya Kadnyanan^{a6}

^aProgram Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Udayana
Badung, Bali, Indonesia

¹budilaksana.dwiki@gmail.com

²eka.karyawati@unud.ac.id

³rahningputri@unud.ac.id

⁴santiyasa@unud.ac.id

⁵agus_sanjaya@unud.ac.id

⁶gungde@unud.ac.id

Abstract

Text summarization or automatic text summarization can make readers receive information quickly without having to read the entire news text, so readers can get more time in reading other news texts. Making text summarization can use two techniques, namely, extractive and abstractive techniques. Abstractive techniques have the aim of producing summary sentences with concepts as humans take the essence of a document that is read. In this study, the author builds an abstractive summarization model using the Dual Encoding method consisting of GRU. The evaluation was carried out using K-Fold Cross Validation, the number of folds used was 5. By using K-Fold Cross Validation, the ROUGE-1, ROUGE-2, and ROUGE-L values were 0.2127749, 0.119851, dan 0.1880595, respectively. For testing when using new data ROUGE-1, ROUGE-2, and ROUGE-L values were 0.3387776, 0.2395176, dan 0.3077376, respectively.

Keywords: *Recurrent Neural Network, Text Summarization, Indonesia News, Dual Encoding, Gated Recurrent Unit*

1. Pendahuluan

Perkembangan informasi pada internet yang tumbuh dengan pesat, dapat menimbulkan pelimpahan informasi di dalam internet. Informasi – informasi yang terus bertambah banyak seiring dengan berjalannya waktu dapat membuat penikmat informasi semakin bingung dalam memilih informasi yang diinginkan untuk dibaca. Salah satu informasi yang terus bertambah adalah teks berita. Teks berita ini bukan ditulis secara singkat tetapi ditulis secara panjang, sehingga membuat pembaca malas dalam membaca. Untuk mempermudah dalam hal membaca, maka diperlukannya sebuah alat. Alat tersebut adalah *text summarization*. *Text summarization* atau peringkasan teks otomatis dapat membuat pembaca menerima informasi dengan cepat tanpa harus membaca keseluruhan teks berita, sehingga pembaca dapat mendapat waktu lebih dalam membaca teks berita lainnya.

Ada dua metode yang tersedia untuk meringkas teks: teknik ekstraksi dan teknik abstrak. Teknik ekstraksi menggunakan setiap kata dalam teks asli untuk meringkas, mengelompokkan dan mengatur kata-kata dalam teks asli tidak berubah. Pernyataan yang dihasilkan biasanya memiliki struktur yang sama dengan pernyataan pada dokumen aslinya [1]. Metode abstrak merangkum abstrak dengan konsep bagaimana menangkap esensi dari teks yang dibaca orang [1].

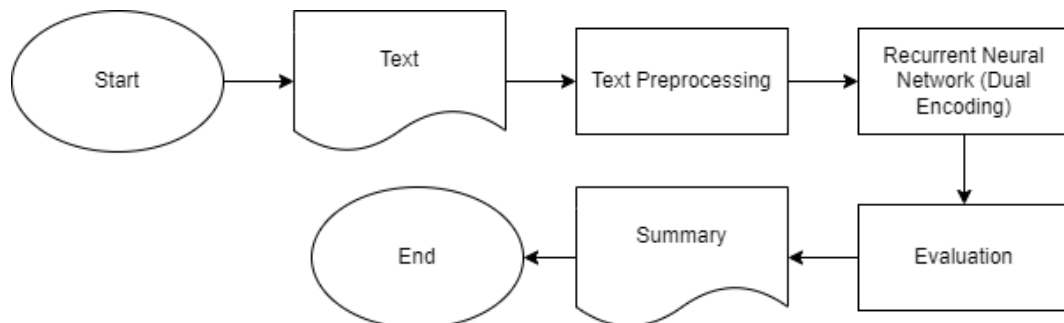
Dalam penerapan *text summarization* menggunakan abstraktif dibutuhkan pendekatan *Deep Learning*. Dengan *deep learning*, masukan akan dihitung melalui *layer* yang lebih banyak dibandingkan dengan Jaringan Syaraf Tiruan sederhana [2]. *Deep learning* yang digunakan ialah *Recurrent Neural Network* (RNN). Penelitian mengenai *text summarization* menggunakan RNN pernah dilakukan oleh beberapa

peneliti sebelumnya seperti penelitian yang dilakukan oleh Yoko, et al. (2018) menghasilkan model terbaik terhadap berita bahasa Indonesia diperoleh dari 6457 dataset dengan jumlah *word2vec* sebanyak 200 dan jumlah *hidden state* sebanyak 256 dengan *loss* sebesar 0.0006654 dan akurasi 99.8810% untuk penulisan sistem dan akurasi sebesar 46.65% untuk penulisan *Q&A Evaluation*. Kemudian penelitian yang dilakukan oleh Rajput dan Mandre (2020) menghasilkan nilai *ROUGE-1* yaitu 41.75 untuk CNN/DailyMail dan 35.12 untuk DUC 2004 terhadap berita bahasa Inggris. Penelitian yang dilakukan oleh Adelia, et al. (2019) menghasilkan model terbaik dengan menggunakan *BiGRU* dengan nilai *ROUGE-1* dan *ROUGE-2* sebesar 0.11975 dan 0.01199.

Berdasarkan paparan penelitian sebelumnya, pada penelitian ini akan melakukan *text summarization* dari berita bahasa Indonesia dengan menggunakan *recurrent neural network* dengan menggunakan *dual encoding model*. Penggunaan *dual encoding model* dapat digunakan dengan baik pada *text summarization* terhadap berita bahasa Indonesia.

2. Metode Penelitian

Penelitian yang dilakukan penulis terdiri dari beberapa tahapan, yaitu data berupa teks berita berbahasa Indonesia. Setelah data atau teks berita berbahasa Indonesia telah terkumpul, maka akan melalui tahap *text preprocessing*. Setelah data telah bersih maka akan masuk ke dalam arsitektur *recurrent neural network* dengan *dual encoding*. Dalam *dual encoding* ini akan menghasilkan *abstractive summarization*. Setelah menghasilkan *abstractive summarization* maka akan di evaluasi hasil tersebut. Secara umum, alur penelitian dapat dilihat pada gambar 1.



Gambar 1. Alur Umum Penelitian

2.1 Data Penelitian

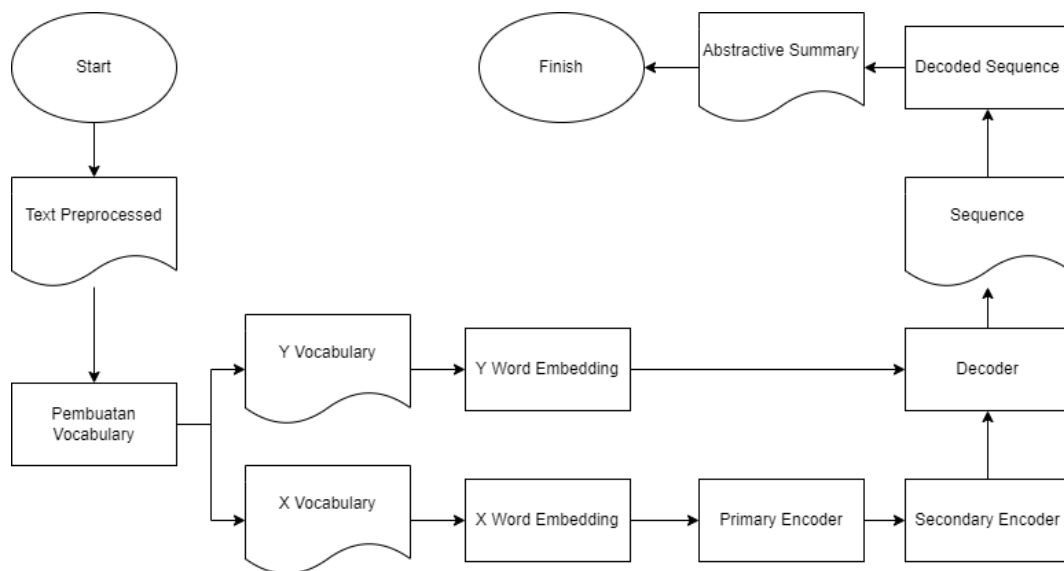
Data yang digunakan adalah data dalam bentuk dataset. Dataset yang digunakan adalah dataset *INDOSUM*. Dataset yang dikembangkan oleh Kurniawan dan Louvan dari Jurnal *INDOSUM: A New Benchmark Dataset for Indonesian Text Summarization* tahun 2018. Dataset ini terdiri dari 93.870 artikel berita. Setiap artikel mempunyai judul, kategori, sumber, URL dari berita asli, dan *abstractive summary*. Dataset ini mempunyai 6 kategori yaitu Inspirasi, Entertainment, Gosip, berita utama, Olahraga, dan Teknologi.

2.2 Text Preprocessing

Dalam penelitian ini, data yang diperoleh diolah sebelum masuk ke model. Data yang tidak terkait dengan data tidak terstruktur merupakan kelemahan utama dalam kumpulan data dan dapat meningkatkan waktu pelatihan. Fase ini terdiri dari beberapa langkah, termasuk *case folding*, *Remove Punctuation*, dan *tokenization*. *Case Folding* akan mengubah semua karakter menjadi huruf kecil. *Remove Punctuation* adalah teknik untuk menghilangkan tanda baca yang digunakan dalam teks untuk membedakan antara kalimat dan komponennya dan untuk memperjelas maknanya. *tokenization* untuk memecah kalimat menjadi token yang dalam hal ini berupa kata, dan *case conversion* untuk mengkonversi bentuk huruf dalam teks menjadi seragam yaitu menjadi huruf kecil [3].

2.3 Dual Encoding

Setelah tahap *text preprocessing* dan dihasilkan data *text preprocessed* maka data akan diolah di dalam model. Model yang digunakan yaitu *dual encoding model*. Data yang dihasilkan melalui *dual encoding model* ini adalah *abstractive summary*.



Gambar 2. Arsitektur *Dual Encoding*

Sebelum masuk ke *primary encoder* maka akan masuk kedalam proses pembuatan *vocabulary*. Pembuatan *vocabulary* digunakan untuk menghasilkan *x vocabulary* (*vocabulary* untuk teks) sedangkan *y vocabulary* (*vocabulary* untuk ringkasan). Setelah pembuatan *vocabulary* akan diproses di dalam *word embedding layer*. Untuk *x word embedding* akan diproses ke dalam *primary encoder* sedangkan *y word embedding* akan diproses ke dalam *decoder*. Dalam *primary encoder* dilakukannya perhitungan *hidden state*. Setelah mendapatkan *hidden state* di *primary encoder*, mulai menghitung *content representation* untuk *x*. *Primary encoder* menghasilkan *semantic vector*. *Semantic vector* berisi *hidden state* yang telah dihitung dan *content representation*. Di *primary encoder*, *hidden state representation* dibuat hanya sekali dalam model ini. Setelah itu, *semantic vector* akan diolah di *decoder* dan *secondary encoder*.

Dalam *decoder*, akan dihitung *hidden state*. Setelah menghitung *hidden state* lalu akan dihitung *content representation of partial generated sequence* dan dikirim ke *secondary encoder*. Lalu *secondary encoder* akan menghasilkan *importance weight*. Di dalam *secondary encoder*, akan dilakukan penghitungan *hidden state*. *Importance weight* akan dikirim ke *decoder*. Setelah itu akan dilakukan penghitungan *hidden state* dari *decoder* berdasarkan *hidden state* yang terakhir dan *final state* yang baru. Setelah itu akan menghasilkan *sequence* dan akan masuk ke dalam proses *decoded sequence* untuk menghasilkan *abstractive summary*.

2.4 Primary Encoder

Blok *primary encoder* mengandung *bidirectional Gated Recurrent Unit (Bi-GRU)* sebagai *recurrent unit*. *Primary encoder* membaca panjang variable dari *input sequence*, menghasilkan *hidden state representation* menggunakan *bidirectional GRU*. Untuk setiap posisi kata, *forward GRU* secara berurut menghitung *hidden state representation* dan *backward GRU* menghitung *hidden representation* dalam *reversed sequence*. *Content representation* untuk keseluruhan *input sequence* dibuat oleh penggabungan *hidden state representation* dan menggunakan konten untuk menunjukan setiap kata dari *input sequence*. *Primary encoder* membuat *feature representation* disebut *semantic vector* untuk setiap kata dalam *input sequence*. *Semantic vector* berisi *hidden state representation* dan *content representation* [4]. GRU bisa secara adaptif menangkap ketergantungan dari skala waktu berbeda, yang terdapat dalam persamaan (1):

$$\begin{cases} u_t = \sigma(W_u[x_t, h_{t-1}]), \\ r_t = \sigma(W_r[x_t, h_{t-1}]), \\ h'_t = \tanh(W_h[x_t, r_t \odot h_{t-1}]), \\ h_t = (1 - u_t) \odot h_{t-1} + u_t \odot h'_t, \end{cases} \quad (1)$$

Keterangan:

Wu,Wr,Wh = matrik parameter

x_t = *input embedding vector* yang sesuai

h_t = *hidden state vector* saat waktu ke-t

\odot = operator perkalian

Forward hidden state dan *backward hidden state* didefinisikan dalam persamaan (2).

$$\begin{cases} \vec{h}_t^p = GRU^p(x_t, \vec{h}_{t-1}^p), \\ \overleftarrow{h}_t^p = GRU^p(x_t, \overleftarrow{h}_{t-1}^p), \\ h_t^p = [\vec{h}_t^p, \overleftarrow{h}_t^p], \end{cases} \quad (2)$$

Keterangan:

\vec{h}_t^p = *hidden state representation* yang dihasilkan menggunakan *forward GRU* untuk setiap posisi kata secara sekuensial berdasarkan *current word embedding* dan *hidden state* sebelumnya

\overleftarrow{h}_t^p = *hidden state representation* yang dihasilkan menggunakan *backward GRU* untuk setiap kata dari *input sentence* dalam *reversed sequence*

h_t^p = gabungan *hidden state* dari *forward GRU* dan *backward GRU*

h_t^p digunakan untuk menunjukkan setiap kata dalam *input sequence*. Content representation dihitung menggunakan persamaan (3).

$$C^p = \tanh(W_p \frac{1}{N} \sum_{t=1}^N h_t^p + b_p) \quad (3)$$

Keterangan:

C^p = *content representation*

W_p dan b_p = parameter

N = panjang dari *input sequence*

2.5 Secondary Encoder

Secondary encoder terdiri dari *unidirectional GRU* sebagai *Recurrent Neural Network*. *Hidden representation*, *content representation*, dan *decoded content representation* digunakan oleh *secondary encoder* untuk menghitung *importance weight*. Menggunakan *importance weight*, *context vector* yang sesuai dihitung ulang dan diumpankan ke *decoder* untuk menghasilkan *output* yang diinginkan. *Secondary encoder* melakukan encoding lebih baik di setiap *decoding stage* [4]. *Secondary encoder* membaca *input sequence* dan menghitung *importance weight* untuk setiap langkah *decoding* berdasarkan *decoded information* dari setiap tahap. Untuk setiap kata dalam *input sequence*, *importance weight* dihitung berdasarkan informasi itu sendiri. Informasi tersebut ialah *saliency* dan *redundancy*. *Importance weight* dihitung menggunakan persamaan (4).

$$\alpha_t = \sigma \left(W_2 (\tanh(W_1 [h_t^p, C^p, C^d] + b_1)) + h_t^{pT} W_s C^p + h_t^{pT} W_s C^d - C^{pT} W_r C^d + b_2 \right) \quad (4)$$

Keterangan:

α_t = *importance weight*

$W_1, W_2, W_s, W_r, b_1, b_2$ = *learning parameter*

C^p = *content representation* dari seluruh *input sequence*

C^d = *content representation* dari seluruh *output sequence* yang dihasilkan oleh *decoder*

$h_t^{pT} W_s C^p$ = *saliency* antara setiap kata

$h_t^{pT} W_s C^d$ = seluruh konten dari teks sumber

$C^{pT} W_r C^d$ = *redundancy* antara konten teks sumber dan *decoded content* dari tahap sekarang

Hidden state di *encoded* oleh *secondary encoder* yang mengambil mayoritas dari informasi secara langsung dari *previous hidden* dan mengabaikan efek dari *current word* menggunakan (5).

$$h_t^S = (1 - \alpha) \odot h_{t-1}^S + \alpha_t \odot GRU^S(x_t, h_{t-1}^S) \quad (5)$$

Keterangan:

h_t^S = *hidden state*

h_{t-1}^S = *previous hidden*

2.6 Decoder

Decoder dilengkapi dengan *attention mechanism* yang menunjukkan *decoding operation* secara bertahap dan menghasilkan *partial fixed length output* pada setiap tahap. *Partial fixed length output* berisi *decoded content representation*. Decoder menghasilkan *output sequence* lebih akurat dengan menggunakan *semantic context vector* yang baru yang dihasilkan oleh *secondary encoder* [4]. *context vector* dihitung sebagai *weighted sum* dari *hidden state* menggunakan persamaan (6).

$$C_i = \sum_{j=1}^n a_{ij} h_j^p \quad (2.6)$$

Keterangan:

C_i = *context vector*

a_{ij} dari persamaan (6) adalah *weight* yang dihitung untuk setiap *hidden state* menggunakan persamaan (7).

$$\begin{cases} a_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^n \exp(e_{ik})} \\ e_{ij} = v_a^T \tanh(W_a h_{i-1}^d + U_a h_j^p) \\ h_i^d = GRU^d(y_i, h_{i-1}^d) \end{cases} \quad (2.7)$$

Keterangan:

e_{ij} = nilai yang menunjukkan seberapa baik *input* sekitar posisi j yang cocok dengan *output* di posisi i

h_i^d = *hidden state*

h_{i-1}^d = *hidden state* yang terakhir

y_i = *target word*

Partial fixed-length sequence secara bertahap di *decode* menggunakan persamaan (8).

$$C^d = \tanh(W_d \frac{1}{L} \sum_{i=1}^L h_i^p + b_d) \quad (8)$$

Keterangan:

W_d dan b_d = parameter

L = panjang dari *current decode sequence*

Final state yang baru dihasilkan setelah setiap *fixed length decoding* dan *decoder* yang ditulis ulang dengan menggunakan persamaan sebagai berikut:

$$h_i^d = \begin{cases} GRU^d(y_i, [h_{i-1}^d, h_m^s]), & \text{if } L \% K == 0 \\ GRU^d(y_i, h_{i-1}^d), & \text{otherwise} \end{cases} \quad (9)$$

Keterangan:

h_m^s = final state

Current context vector dihasilkan melalui *primary encoder* dan *decoder hidden state* digabungkan dan diumpkan ke *decoder* melalui satu *linear layer* untuk menghasilkan *vocabulary distribution* menggunakan persamaan (10)

$$P_v = P(y_i | y_1, \dots, y_{i-1}; x) = \text{softmax}(W_v [h_i^d, C_i] + b_v) \quad (10)$$

Keterangan:

W_v, b_v = learning parameter

P_v = conditional probability distributions untuk target word atas semua kata dalam vocabulary saat time-step

2.7 ROUGE

ROUGE atau *Recall-Oriented Understudy for Gisting Evaluation* adalah pengukuran evaluasi otomatis yang diturunkan dari *BLEU* (*BiLingual Evaluation Understudy*). *ROUGE* menggunakan *n-gram* yang overlap antara *system-generated summary* dan *handwritten gold standard summaries*. Nilai dalam *DUC corpora* digunakan untuk memvalidasi sistem *ROUGE*. *ROUGE* biasanya digunakan untuk peringkasan teks karena *recall-oriented* tidak seperti *BLEU* yang fokus dalam precision. *ROUGE* adalah metrik yang umum digunakan untuk mengukur coverage dalam peringkasan. Namun, *ROUGE* mempunyai batasan yang besar. *ROUGE* tidak memperhitungkan struktur kalimat atau ketepatan tata bahasa. Ringkasan yang diberi peringkat tinggi menggunakan beberapa kombinasi parameter *ROUGE* akan tetap memiliki peringkat tinggi jika semua kata – katanya bercampur aduk yang menghasilkan kata – kata tidak berarti. *ROUGE* juga tidak memperhitungkan *lexical variations* yang memiliki kemiripan secara semantik. Namun, karena kemudahan penggunaan dan tidak adanya yang bisa diandalkan dalam teknik evaluasi otomatis sehingga *ROUGE* cukup populer [5].

ROUGE-N pada dasarnya adalah perhitungan penarikan berdasarkan perbandingan *N-gram* dari ringkasan standar emas dan teks ringkasan mesin. Jumlah *n-gram* yang digunakan bervariasi. Namun, *n-gram* dengan angka $n = 1$ (*ROUGE-1*) dan $n = 2$ (*ROUGE-2*) paling sering digunakan [6]. Maka *ROUGE-N* dapat dihitung dengan persamaan (11).

$$ROUGE - N = \frac{p}{q} \quad (11)$$

Keterangan:

p = jumlah *n-gram* yang sama dengan *gold standard summary* dan teks hasil peringkasan mesin

q = jumlah *n-gram* pada *gold standard summary*

ROUGE-L mengevaluasi ringkasan teks dengan membandingkan *common subsequence length* (LCS) atau hasil ringkasan teks mesin dengan urutan kata terpanjang yang sama antara ringkasan standar emas [6]. *ROUGE-L* dapat dihitung menggunakan persamaan (12)

$$ROUGE - L = \frac{LCS}{m} \quad (12)$$

Keterangan:

LCS = longest common subsequence

m = jumlah kata pada *gold standard summary*

3. Hasil dan Pembahasan

3.1. Evaluasi *Dual Encoding* terhadap *Abstractive Summarization*

Pada penelitian ini, untuk mengetahui performa *abstractive summarization* pada teks berbahasa Indonesia menggunakan *Dual Encoding*, maka akan dilakukan pengujian atau evaluasi untuk

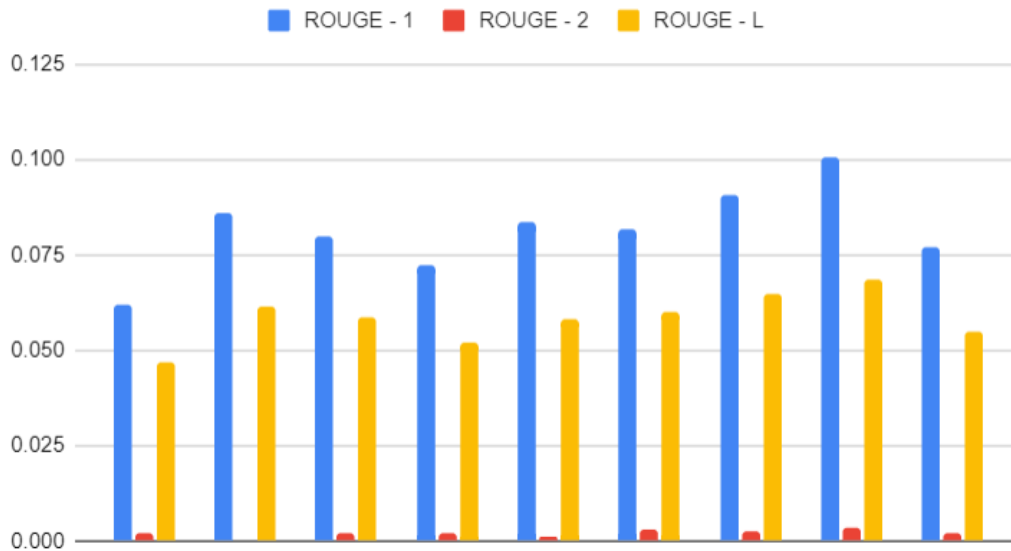
mendapatkan nilai dari *ROUGE-1*, *ROUGE-2*, dan *ROUGE-L*. Pada pengujian ini, menggunakan data artikel berita bahasa Indonesia dataset yang akan dibagi dengan perbandingan 90% untuk data training dan validasi, dan untuk 10% termasuk data testing. Selanjutnya validasi model dilakukan dengan menggunakan metode *K-Fold Cross Validation*, dengan jumlah fold sebesar 5 dan berdasarkan konfigurasi *hyperparameter* (*Learning Rate*, *Batch Size*, dan *Epoch*)

Dalam melakukan *K-Fold Cross Validation*. Jumlah unit pada *GRU* digunakan sebesar 100 dan untuk dimensi pada *word embedding* sebesar 300. *Hyperparameter* yang digunakan ialah *learning rate*, *batch size*, dan *epoch*. *Learning rate* yang diuji sebesar 0.01 dan 0.001. *Batch size* yang diuji sebesar 8, 16, dan 32. *Epoch* yang diuji sebesar 5, 10, dan 20. Parameter – parameter tersebut akan dikombinasikan dan menghasilkan 18 nilai setiap rata – rata *ROUGE-1*, rata - rata *ROUGE-2*, dan rata - rata *ROUGE-L*. Hasil dari validasi model *Dual Encoding* dapat dilihat pada Tabel 1.

Tabel 1. Hasil dari *K-Fold Cross Validation*

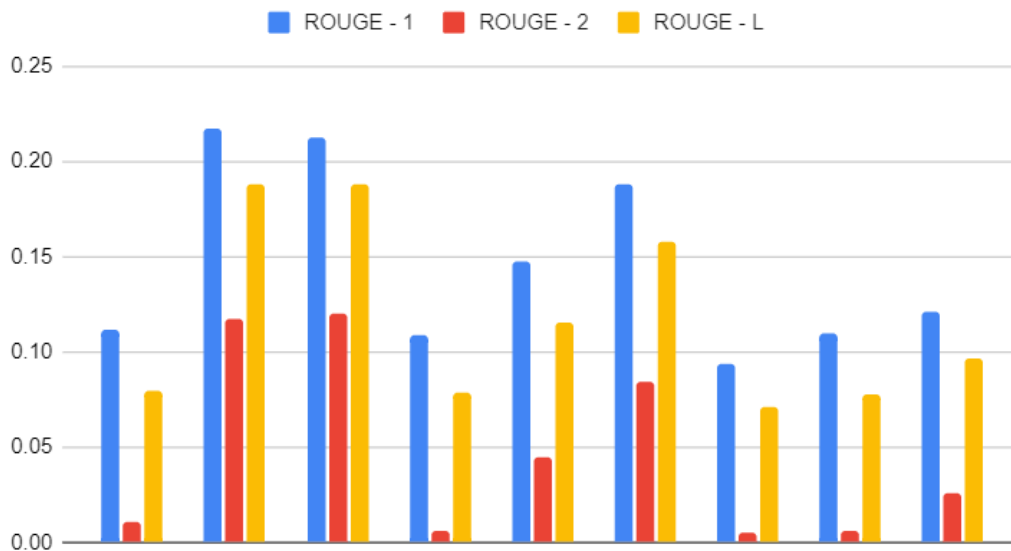
Learning Rate	Batch Size	Epoch	Avg ROUGE 1	Avg ROUGE 2	Avg ROUGE L
0.01	8	5	0.0620598	0.0020986	0.0467862
		10	0.0859643	0.0004356	0.0612732
		20	0.0799099	0.0018761	0.0585936
	16	5	0.0722505	0.001867	0.052169
		10	0.0837565	0.001233	0.0583148
		20	0.0817789	0.0028873	0.0600106
	32	5	0.090794	0.0024744	0.0647594
		10	0.100424	0.003337	0.0685092
		20	0.07696	0.002076	0.0548748
0.001	8	5	0.1113044	0.0106942	0.0794237
		10	0.2168853	0.117364	0.1879771
		20	0.2127749	0.1198551	0.1880595
	16	5	0.1091135	0.0065141	0.078212
		10	0.1473753	0.0446482	0.1151162
		20	0.1876084	0.0845609	0.1576544
	32	5	0.0937695	0.0048913	0.0712081
		10	0.1100292	0.0056799	0.0773099
		20	0.1209372	0.0261349	0.0966511

Pada Tabel 1 di atas dapat dilihat nilai *ROUGE-1*, *ROUGE-2*, dan *ROUGE-L* yang didapatkan melalui pengujian *Learning Rate*, *batch size* dan *epoch*. Selanjutnya dari hasil pengujian tersebut akan dianalisis untuk menemukan model dengan performa terbaik yang selanjutnya akan diuji kembali menggunakan data testing.



Gambar 3. Hasil K-Fold-Cross Validation dengan Learning Rate 0.01

Gambar 3 menunjukkan perbandingan hasil pengujian pada setiap *batch size* dan *epoch* terhadap *learning rate* 0.01. Dapat dilihat bahwa pengujian dengan *batch size* 8 dan *epoch* 5 memperoleh hasil terendah dengan nilai *Rouge-1*, *Rouge-2*, dan *Rouge-L* yaitu 0.0620598, 0.0020986, dan 0.0467862. Kemudian pengujian dengan *batch size* 32 memperoleh hasil tertinggi dengan *Rouge-1* 0.100424, *Rouge-2* 0.003337, dan *ROUGE-L* 0.0685092.



Gambar 4. Hasil K-Fold-Cross Validation dengan Learning Rate 0.001

Gambar 4 menunjukkan perbandingan hasil pengujian pada setiap *batch size* terhadap *learning rate* 0.001. Dapat dilihat bahwa pengujian dengan *batch size* 128 memperoleh hasil terendah dengan nilai *Rouge-1*, *Rouge-2*, dan *Rouge-L* yaitu 0.0937695, 0.0048913, dan 0.0712081. Kemudian pengujian dengan *batch size* 32 memperoleh hasil tertinggi dengan *Rouge-1* 0.2127749, *Rouge-2* 0.1198551, dan *ROUGE-L* 0.1880595.

Berdasarkan analisa diatas didapatkan bahwa setiap kenaikan *batch size* akan mengalami penurunan nilai *ROUGE*. Sehingga *batch size* yang paling baik digunakan sebesar 8 dan *epoch* 20. Untuk *learning rate* yang digunakan ialah 0.001 dapat dilihat pada Gambar 4 memiliki grafik yang paling tinggi dari pada Gambar 3.

3.2. Hasil Pengujian Model Terbaik dengan Data Baru

Berdasarkan analisis yang telah dilakukan hasil pengujian model dengan *K-Fold Cross-Validation*, didapatkan model *abstractive summarization* menggunakan *Dual Encoding* dengan performa terbaik pada *learning rate* 0.001 dan *batch size* 32. Kemudian model terbaik ini diuji kembali menggunakan data baru yang belum pernah melewati pelatihan dan validasi sebelumnya. Hasil pengujian menggunakan data baru dapat dilihat pada Tabel 2.

Tabel 2. Hasil dari Pengujian Data Baru

Pengujian	ROUGE-1	ROUGE-2	ROUGE-L
Training Validasi	0.2127749	0.1198551	0.1880595
Testing	0.3387776	0.2395176	0.3077376

3.3. Hasil Perbandingan Ringkasan Mesin dengan Ringkasan Asli

Berikut merupakan hasil dari ringkasan mesin dengan ringkasan aslinya dapat dilihat pada tabel 3. Hasil ringkasan dari mesin sudah meringkas dengan membuat sebuah kalimat dan kata – kata unik dari teks. Terdapat kata – kata yang sama dari *original summary* dan *generate summary* seperti Gionee pabrikan ponsel asal tiengkok telah mengumumkan. Setiap kata – kata unik akan mempengaruhi nilai *ROUGE-1* dan *ROUGE-2*.

Tabel 3. Hasil dari Ringkasan Mesin dengan Ringkasan Aslinya

Teks	Original Summary	Generate Summary
harus diakui smartphone dengan bezel tipis dan rasio layar 189 atau yang disebut infinity display fullvision edge to edge dan fullview memiliki daya tarik yang luar biasa gionee pabrikan ponsel asal tiengkok ini pun telah mengumumkan 8 smartphone terbaru di mana 6 smartphone menggunakan layar penuh fullview display gionee membagi 8 smartphone barunya itu dalam beberapa seri seperti seri premium m7 yang dirancang untuk para profesional series s yang ...	gionee pabrikan ponsel asal tiengkok telah mengumumkan 8 smartphone terbaru di mana enam smartphone menggunakan layar penuh fullview display gionee membagi delapan smartphone barunya itu dalam beberapa seri seperti seri premium m7 yang dirancang untuk para profesional series s yang ditunjukkan untuk kalangan muda dan series f yang hadir dengan harga terjangkau	gionee pabrikan ponsel asal tiengkok telah mengumumkan waktu dua bulan disebutkan diklaim menjadi salah satu perangkat yang diminati oleh dua perangkat yang diluncurkan dalam menu e commerce atau mini vr pengumuman tersebut dikabarkan akan meluncur ke perangkat android menegaskan ai adalah fitur bernama timeline kehadiran google kemudian bisa menerima akun pooka video maupun google assistant

4. Kesimpulan

Berdasarkan penelitian yang telah dilakukan dan hasil yang diperoleh selama melakukan penelitian, dapat ditarik kesimpulan sebagai berikut

1. *Recurrent Neural Network* menggunakan model *Dual Encoding* dalam *text summarization* terhadap berita Bahasa Indonesia untuk membuat *abstractive summary* menghasilkan nilai *ROUGE-1* sebesar 0.2127749, *ROUGE-2* sebesar 0.119851, dan *ROUGE-L* sebesar 0.1880595 setelah melalui pengujian model menggunakan *K-Fold Cross Validation*. Dengan konfigurasi *learning rate* 0.001, *batch size* 8, dan *epoch* 20 menghasilkan model terbaik.
2. Model *recurrent neural network* dengan menggunakan *dual encoding* terbaik akan diuji dengan data yang baru untuk menghasilkan *abstractive summary*. Hasil pengujian tersebut menghasilkan nilai *ROUGE-1*, *ROUGE-2*, *ROUGE-L* sebesar 0.3387776, 0.2395176, dan 0.3077376.

Daftar Pustaka

- [1] K. Ivnedra and M. Mustikasari, *Implementasi Metode Recurrent Neural Network pada Text Summarization dengan Teknik Abstraktif*, Jurnal Teknologi Informasi dan Ilmu Komputer, 2019.
- [2] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, 2016.
- [3] A. E. Karyawati, P. A. Utomo, and I. G. A. Wibawa, "Comparison of SVM and LIWC for Sentiment Analysis of SARA," *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, vol. 16, no. 1, p. 45, 2022, doi: 10.22146/ijccs.69617
- [4] M. H. Rajput and B. R. Mandre, *Abstractive Summarization of Document using Dual Encoding Framework*, International Journal of Computer Applications, 2020, vol. 176, no. 39.
- [5] P. Mehta and P. Majumder, *From Extractive to Abstractive Summarization: A Journey*, Springer, 2019.
- [6] Yuliska and Syaliman, *Literatur Review Terhadap Metode, Aplikasi dan Dataset Peringkasan Dokumen Teks Otomatis untuk Teks Berbahasa Indonesia*, 2020, vol. 5, no. 1, pp. 19-31.