JASIE "Jurnal Aplikasi Sistem Informasi Dan Elektronika"
Vol. 2, No. 1, Juli 2022, Hal: 43-51
p-ISSN: 2714-612X, e-ISSN: Dalam Proses

Implementasi *Text Summarization* Pada *Reading Comprehension* Menggunakan *Library Python*

Rosita Yanuarti[1], Habibatul Azizah Alfaruq[2]
Teknik Informatika, Fakultas Teknik, Universitas Muhammadiyah Jember,
Email: rosita.yanuarti@unmuhjember.ac.id

ABSTRAK

Document (text) summarization atau peringkasan dokumen adalah aplikasi *information retrieval* (IR) lain yang terdiri dari pembuatan teks dengan versi yang dipersingkat untuk mengurangi kelebihan informasi dengan mempertahankan gagasan utama dari teks tersebut. *Text summarization* digunakan secara luas, misalnya untuk membuat ringkasan pada sekumpulan dokumen berita atau artikel guna mendapatkan topik berita terkait. Ringkasan otomatis juga diimplementasikan pada bidang bisnis seperti mendapatkan inti ringkasan pada *customer feedback* atau kepuasan pelanggan terhadap suatu produk atau layanan. Pada artikel ini, *text summarization* diimplementasikan pada dokumen *reading comprehension*, yang bertujuan sebagai media pembelajaran bagi instruktur dan murid dalam memahami teks *reading comprehension*.

Pada artikel ini, teknik *extractive summarization* dilakukan dengan menggunakan library python yaitu Gensim dan NLTK. *Text summarization* terdiri dari beberapa tahapan, simulai dengan input data, selanjutnya dilakukan proses *text preprocessing*, meliputi proses pembersihan pada teks, penghilangan *stopword*, dan proses tokenisasi. Selanjutnya pembobotan dan seleksi kalimat untuk setiap kata yang ada di dalam paragraf, dilanjutkan dengan proses f*iltering* dan penggabungan kalimat. Pada proses ini, setiap kalimat akan diberi peringkat sesuai dengan bobot kalimatnya. Oleh karena itu, kalimat-kalimat yang dipilih untuk dimasukkan ke dalam ringkasan.

Berdasarkan hasil pengujian, proses evaluasi dilakukan dengan membandingkan hasil *text summarization* sistem dengan menggunakan library Gensim dan NLTK dengan hasil summarization secara manual (human), maka didapatkan akurasi sebesar 72,6%.

Kata Kunci : *Text Summarization*, seleksi kalimat, *library python*

ABSTRACT

Document (text) summarization is another IR application that consists in creating a shortened version of a text in order to reduce the information overload while keeping the main idea of it is necessary. Text summarization is widely used, for example to make a summary of a set of news documents or articles in order to get related news topics. Automatic summarization is also implemented in business areas such as getting a summary of customer feedback or customer satisfaction about their a product or service. In this article, text summarization is implemented in reading comprehension document, which aims as a learning media for instructors and students in understanding the reading comprehension.

In this article, the extractive summarization technique is carried out using the python library, namely Gensim and NLTK. Text summarization consists of several steps, starting with data input, followed by text preprocessing process, which is including the cleaning process for the text, removing stop words, and tokenization processes. The weight assignment and sentence selection is carried out for each word in the paragraph, followed by the filtering process and assembling. In this process, each sentence will be ranked according to the weight of the sentence. Therefore, sentences that were selected to be included in the summary.

Based on the experimental results, the process evaluation by comparing summary that was generated by system using library Gensim and NLTK to human summaries, with the accuracy %.

***Keyword** : Text Summarization, sentence selection, library python*

1. PENDAHULUAN

Terdapat sejumlah besar data yang muncul secara digital, oleh Karena itu pentingnya mengembangkan prosedur untuk mempersingkat teks panjang dengan mempertahankan gagasan utama dari teks tersebut. Peringkasan membantu mempersingkat waktu yang dibutuhkan untuk membaca, mempercepat pencarian

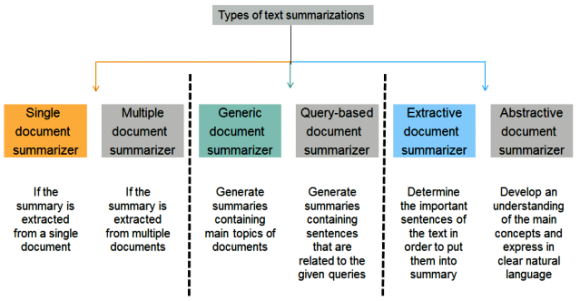
informasi dan membantu mendapatkan informasi sebanyak-banyaknya tentang satu topik (Romadhony, dkk, 2011).

Implementasi text summarization sangat luas, misalnya untuk membuat ringkasan pada sekumpulan dokumen berita atau artikel guna mendapatkan topik berita terkait. Ringkasan otomatis juga data diimplementasikan pada bidang bisnis seperti mendapatkan inti ringkasan pada *customer feedback* atau kepuasan pelanggan terhadap suatu produk atau layanan (Zhai dan Massung, 2016) Penelitian pada (Allahyari, dkk, 2017) mengimplementasikan *text summarization* pada dokumen berita berbahasa Indonesia. Penelitian pada (Rifano, dkk, 2020) juga mengimplementasikan *text summarization* pada dokumen berita bola.

Pada artikel ini, *text summarization* diimplementasikan pada dokumen *reading comprehension*, yang bertujuan sebagai media pembelajaran bagi instruktur dan murid pada kelas Bahasa Inggris TOEFL dalam memahami teks *reading comprehension*.

A. Document Summarization

Mesin pencari adalah aplikasi *information retrieval* (IR) yang paling penting dan tersebar luas, tetapi teknik IR juga mendasar untuk sejumlah tugas lainnya. Peringkasan dokumen adalah aplikasi IR lain yang terdiri dari pembuatan teks dengan versi yang dipersingkat untuk mengurangi kelebihan informasi. Peringkasan umumnya bersifat ekstraktif; yaitu, memilih kalimat yang paling relevan dari sebuah dokumen dan mengumpulkannya untuk membentuk versi pendek dari dokumen itu sendiri (Ceri, dkk, 2013).

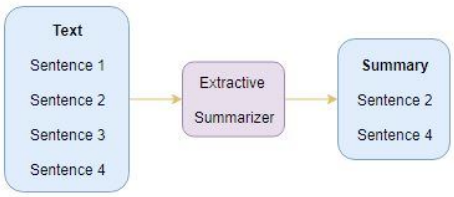


Gambar 1. Tipe-tipe peringkasan teks

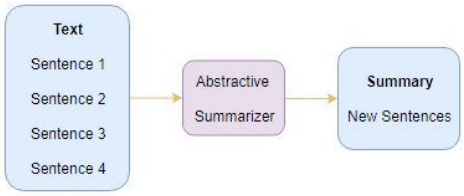
Ada dua metode utama dalam peringkasan teks. Yang pertama adalah peringkasan berbasis seleksi *(selection-based)* atau ekstraktif. Dengan metode ini, ringkasan terdiri dari urutan kalimat yang dipilih dari dokumen asli. Tidak ada kalimat baru yang ditulis, selanjutnya ringkasan dapat diekstraksi. Gambar 2 (a) menunjukkan model dari metode *extractive summarization*. Metode kedua adalah peringkasan abstraksi atau *generation-based summarization*. Pada model ini, ringkasan mungkin berisi kalimat baru yang tidak ada dalam dokumen asli mana pun (Zhai dan Massung, 2016). Model abstractive summarization ini ditunjukkan pada Gambar 2 (b).

Pada awal perkembangannya, *text summarization* atau peringkasan teks hanya memproses menggunakan dokumen tunggal di mana sistem menghasilkan

ringkasan dari satu dokumen, apakah sebuah berita, artikel ilmiah, acara siaran, atau kuliah. Seiring dengan kemajuan penelitian, jenis peringkasan teks yang baru muncul yaitu peringkasan dengan multi-dokumen. Peringkasan multi-dokumen dimotivasi oleh banyaknya penggunaan *source* di web. Mengingat banyaknya redundansi di web, peringkasan seringkali lebih berguna jika dapat memberikan ringkasan singkat dari banyak dokumen tentang topik yang sama atau peristiwa yang sama.



Gambar 2 (a) Model Extractive Summarization (Akhter dan Mehra, 2022)



Gambar 2 (b). Model Abstractive Summarization (Akhter dan Mehra, 2022)

Generic summarization merupakan ringkasan yang bersifat umum baik genre atau domain materi yang perlu diringkas. Dalam pengaturan ini, pentingnya informasi ditentukan melalui isi input dokumennya. Selanjutnya diasumsikan bahwa ringkasan akan membantu pembaca dengan cepat menentukan topik dalam dokumen tersebut. Sebaliknya, dalam peringkasan berbasis kueri (*query-based summarization*), tujuannya adalah untuk meringkas informasi dalam dokumen input hanya yang relevan dengan kueri tertentu dari pengguna. Sebagai contoh, dalam konteks pencarian informasi, dengan adanya query yang dimasukkan oleh pengguna dan sekumpulan dokumen relevan yang diambil oleh mesin pencari, ringkasan dari setiap dokumen dapat memudahkan pengguna untuk menentukan dokumen mana yang relevan. Untuk menghasilkan rangkuman yang berguna dalam konteks ini, peringkas otomatis perlu mempertimbangkan kueri dan juga dokumen. Peringkas mencoba menemukan informasi di dalam dokumen yang relevan dengan kueri atau dalam beberapa kasus, dapat menunjukkan berapa banyak informasi dalam dokumen yang terkait dengan kueri (Nenkova and McKeown, 2011).

Berdasarkan Bharti, dkk, (2017), pendekatan peringkasan teks dapat diklasifikasikan menjadi lima jenis, yaitu *statistical based, machine learning based, coherent based, graph based, algebraic based*. Pendekatan Statistical Based Approach sangat sederhana dan sering digunakan untuk ekstraksi kata kunci dari dokumen. Tidak ada kumpulan data standar yang diperlukan untuk pendekatan ini. Untuk mengekstrak kata kunci dari dokumen digunakan beberapa fitur statistik dokumen seperti, *term* atau *term frequency* (TF), *Term Frequency-inverse document frequency* (TF-IDF), *position of keyword* (POK). *Machine learning* adalah pendekatan yang bergantung pada fitur, dimana memerlukan kumpulan data label untuk melatih model. Ada beberapa pendekatan *machine learning* yang populer yaitu, *Nave Bayes, decision trees, Hidden Markov Model,*

Maximum Entropy (ME)*, Neural Network, Support Vector Machine* (SVM). Pendekatan berbasis koheren pada dasarnya berkaitan dengan hubungan kohesi antara kata-kata. Hubungan kohesi antar elemen dalam sebuah teks meliputi referensi, elipsis, substitusi, konjungsi, dan kohesi leksikal. Dua pendekatan *graph-based* yang populer digunakan untuk peringkasan teks yaitu, *Hyperlinked Induced Topic Search* dan *Google's PageRank* (GPR). Pendekatan aljabar, salah satunya menggunakan teori aljabar yaitu, matriks, transpos matriks, vektor Eigen, dan lain-lain. Ada banyak algoritma yang digunakan untuk peringkasan teks menggunakan pendekatan aljabar salah satunya adalah Latent Semantic Analysis (LSA) (Bharti, dkk, 2017).

Menurut Allahyari, dkk (2017), cara paling sederhana untuk mengevaluasi ringkasan adalah dengan meminta manusia (*expert*) untuk menilai kualitasnya. Faktor-faktor yang harus dipertimbangkan oleh *expert* ketika memberikan skor untuk setiap ringkasan kandidat adalah tata bahasa, non redundansi, integrasi potongan informasi yang paling penting, struktur dan koherensi (Allahyari, dkk, 2017). ROUGE adalah metrik yang paling banyak digunakan untuk evaluasi otomatis. ROUGE Lin memperkenalkan seperangkat metrik yang disebut Recall-Oriented Understudy for Gisting Evaluation (ROUGE) yang secara otomatis menentukan kualitas ringkasan dengan membandingkannya dengan ringkasan manusia (referensi).

Efektivitas keseluruhan dari sebuah ringkasan dapat diuji jika pengguna membaca ringkasan dan kemudian menjawab pertanyaan tentang teks asli. Apakah ringkasan dapat menangkap informasi penting yang dibutuhkan oleh evaluator? Jika teks asli adalah seluruh bab buku teks, dapatkah pengguna membaca ringkasan tiga paragraf dan memperoleh informasi yang cukup untuk menjawab pertanyaan yang disediakan? Ini adalah satu-satunya metrik yang dapat digunakan untuk ukuran ekstraktif dan abstraksi. Menggunakan model bahasa untuk menilai ringkasan ekstraktif dan abstraksi kemungkinan akan menjadi bias terhadap yang ekstraktif karena metode ini berisi frasa langsung dari teks asli, yang memberikan kemungkinan relevansi yang sangat tinggi (Zhai dan Massung, 2016).

B. Libraries Python

Summarization adalah alat yang berguna untuk berbagai aplikasi tekstual yang bertujuan untuk menyoroti informasi penting dalam korpus besar. Dengan ledakan informasi di Internet, Python menyediakan beberapa *tool* atau *library* untuk membantu meringkas teks.

1. Gensim

Gensim adalah *library* Python untuk *topic modelling, document indexing* dan *similarity retrieval* dengan kumpulan data yang sangat besar. Target audiens adalah komunitas natural language processing (NLP) dan information retrieval (IR). *Library* ini dirancang secara otomatis untuk mengekstrak topik semantik dari dokumen. Implementasi gensim didasarkan pada algoritma TextRank yang populer. Library ini adalah toolkit yang bersifat open-source untuk *vector space*

modelling dan *topic modelling,* dan diimplementasikan dalam bahasa pemrograman Python, menggunakan NumPy, SciPy dan opsional Cython untuk kinerjanya. Cara kerja summarization ini didasarkan pada barisan (ranking) kalimat teks menggunakan variasi dari algoritma TextRank. TextRank adalah *general-purpose*, algoritma peringkat berbasis grafik untuk NLP. TextRank adalah teknik ringkasan otomatis. Algoritma peringkat berbasis graf adalah teknik untuk menentukan pentingnya sebuah simpul dalam sebuah graf, berdasarkan informasi global yang diambil secara rekursif dari seluruh graf (Sareen, 2018).

2. Sumy

Library sederhana dan *command line utility* untuk mengekstrak ringkasan dari halaman HTML atau teks biasa. Paket ini juga berisi kerangka evaluasi sederhana untuk ringkasan teks. Sumy menawarkan beberapa algoritma dan metode untuk meringkas seperti :

a. **Luhn** : Algoritma Luhn adalah pendekatan yang berdasarkan TF-IDF, dan salah satu pendekatan paling awal untuk peringkasan teks. Luhn mengusulkan bahwa pentingnya setiap kata dalam sebuah dokumen menandakan betapa pentingnya itu. Idenya adalah bahwa setiap kalimat dengan kemunculan maksimum dari kata-kata frekuensi tertinggi dan kemunculan paling sedikit menandakan tidak memiliki makna terhadap dokumen.
b. **Latent Semantic Analysis :** Latent Semantic Analysis adalah teknik untuk membuat representasi vektor dari sebuah dokumen. Representasi vektor dari suatu dokumen digunakan untuk membandingkan dokumen dan kesamaannya dengan menghitung jarak antara vektor.
c. **LexRank :** LexRank adalah pendekatan berbasis grafik yang bersifat *unsupervised*. Pemberian skor pada kalimat dilakukan dengan menggunakan metode grafik. LexRank digunakan untuk menghitung penting atau tidaknya suatu kalimat berdasarkan konsep sentralitas vektor eigen dalam representasi grafik kalimat.
d. **TextRank :** TextRank menggunakan pendekatan ekstraktif dan berbasis grafik yang bersifat unsupervised dan berbasis PageRank untuk peringkasan teks. Pada TextRank, simpul grafik adalah kalimat, dan bobot tepi antar kalimat menunjukkan kesamaan antar kalimat.

3. NLTK

The Natural Language Toolkit, atau yang disebut dengan NLTK, merupakan rangkaian perpustakaan dan program untuk pemrosesan bahasa alami simbolis dan statistik (NLP) untuk bahasa Inggris yang ditulis dalam bahasa pemrograman Python. Library ini dikembangkan oleh Steven Bird dan Edward Loper di Departemen Ilmu Komputer dan Informasi di University of Pennsylvania. NLTK mencakup demonstrasi grafis dan data sampel yang disertai dengan buku yang menjelaskan konsep dasar dalam pemrosesan bahasa (Yao, 2019).

NLTK bertujuan untuk mendukung penelitian dan pengajaran di NLP atau bidang yang terkait erat dengan linguistik empiris, ilmu kognitif, kecerdasan

buatan, pencarian informasi, dan pembelajaran mesin. NLTK telah berhasil digunakan sebagai alat pengajaran, sebagai alat belajar individu, dan sebagai platform untuk membuat prototipe dan membangun sistem penelitian. NLTK mendukung fungsi klasifikasi, tokenization, stemming, tagging, parsing, dan penalaran semantik (Yao, 2019).

2. METODE PENELITIAN

A. Pengumpulan Data

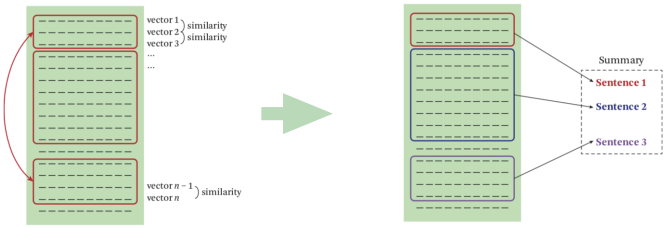
Pada penelitian ini, dataset yang digunakan adalah contoh soal *reading comprehension* sesi reading, yang didapatkan dari salah satu tutor atau instruktur Bahasa Inggris. Dataset terdiri dari 10 soal *reading comprehension*.

B. Pengolahan Data

Pengolahan data yang dilakukan pada artikel ini menggunakan teknik *extractive summarization* dengan menggunakan library python yaitu Gensim dan NLTK. *Text summarization* terdiri dari beberapa tahapan, sebagai berikut (Romadhony, dkk, 2011) :

1. *Preprocessing*, meliputi proses pembersihan pada teks, penghilangan *stopword,* dan proses tokenisasi.
2. Pembobotan dan ekstraksi kalimat. Proses pembobotan terlebih dahulu dilakukan untuk setiap kata yang ada di dalam paragraf. Kemudian pemboboton dilakukan untuk setiap kalimat.
3. *Filtering* dan Penggabungan kalimat. Langkah ini dilakukan dengan memilih kalimat yang memiliki ranking terbesar. Kalimat di-ranking berdasarkan nilai bobot dari kalimat tersebut. Bobot yang memiliki nilai terbesar akan di-ranking di posisi teratas. Artinya kemungkinan kalimat tersebut menjadi bagian dari ringkasan (*summary*) semakin besar.

Gambar 3 menunjukkan proses pemecahan text menjadi kalimat-kalimat, selanjutnya setiap kalimat akan diberikan nilai/bobot pada proses pembobotan. Selanjutnya masing-masing kalimat akan diukur similaritasnya, kemudian dilakukan proses pemilihan kalimat berdasarkan rangking nilai bobot dari yang paling besar. Proses yang terakhir adalah menggabungkan kalimat-kalimat yang telah dipilih menjadi bagian dari ringkasan (*summary*).



Gambar 3. Proses pembobotan dan penggabungan kalimat

Pada artikel ini, tipe *text summarization* yang dilakukan adalah extractive summarization dengan menggunakan library python antara lain Gensim, Sumy, dan NLTK. Algoritma pada library Gensim meringkas text secara otomatis, dengan mengekstrak satu atau lebih kalimat-kalimat penting dari teks awal.

Algoritma-algoritma pada library ini bersifat unsupervised, yang berarti tidak memerlukan input dari manusia dan hanya memerlukan kumpulan dokumen *plain text* (Kumar, dkk, 2021).

3. HASIL DAN PEMBAHASAN

A. Hasil dan Pembahasan

Tahapan dalam melakukan text summarization dimulai dengan membaca dataset atau *original text*. Proses ini merupakan proses load dataset yaitu membaca data *reading comprehension*. Tahap selanjutnya adalah *text preprocessing*. Tokenisasi dan penghilangan *stopword* dilakukan dalam tahap ini. Tokenisasi yang dilakukan bertujuan untuk memecah *original text* menjadi kalimat-kalimat. Selanjutnya, membuat *matrix similarity* antar kalimat. Pada tahap ini menggunakan cosine similarity untuk mencari similaritas antar kalimat. Kemudian, berdasarkan *score* atau nilai similaritas tersebut, dan membuat ranking setiap kalimat dalam matrix similarity, sehingga diambil sebanyak n teratas kalimat-kalimat dengan score yang paling tinggi.

Perbandingan hasil text summarization yang dihasilkan dengan menggunakan library Gensim dan NLTK, dengan hasil summary yang dibuat oleh *expert.*

```
The first flying vertebrates were true reptiles in which one of the fingers of the front of limbs became very elongat
ed, providing support for a flap of stretched skin that served as a wing. These were the pterosaurs, literallythe "wi
ng lizards." The earliest pterosaurs arose near the end of the Triassic period of the Mesozoic Era, some 70 million y
ears before the first known fossils of true birds occur, and they presumably dominates the skies until they were even
tually displaced by birds. Like the dinosaurs, some of pterosaurs became gigantic; the largest fossil discovered is o
f an individual that had a wingspan of 50 feet or more, larger than many airplanes. These flying reptiles had large,
tooth-filled jaws, but their bodies were small and probably without the necessary powerful muscles for sustained wing
movement. They must have been expert gliders, not skillful fliers, relying on wind power for their locomotion.
Birds, despite sharing common reptilian ancestors with pterosaurs, evolved quite separately and have been much more s
uccessful in their dominance of the air. They are an example of a common theme in evolution, the more or less paralle
l development of different types of body structure and function for the same reason-in this case, for flight. Althoug
h the fossil record, as always, is not complete to determine definitively the evolutionary lineage of the birds or in
as much detail as one would like, it is better in this case than for many other animal groups. That is because of the
unusual preservation in a limestone quarry in southern Germany of Archaeopteryx, a fossil that many have called the l
ink between dinosaurs and birds. Indeed, had it not been for the superb preservation of these fossils, they might wel
l have been classified as dinosaurs. They have the skull and teeth of a reptileas well as a bony tail, but in the lin
e-grained limestone in which these fossils occur there are delicate impressions of feathers and fine details of bone
structure that make it clear that Archaeopteryx was a bird. All birds living today, from the great condors of the And
es to the tiniest wrens, race their origin back to the Mesozoic dinosaurs.
```

Gambar 4. *Original text*

```
In [62]:                                                          Slide Type -
         from gensim.summarization import summarize
         summarize(data, split=True)

Out[62]: ['These flying reptiles had large, tooth-filled jaws, but their bodies were small and probably without the necessary
          powerful muscles for sustained wing movement.',
           'That is because of the unusual preservation in a limestone quarry in southern Germany of Archaeopteryx, a fossil th
          at many have called the link between dinosaurs and birds.']
```

Gambar 5. Hasil summarization menggunakan library Gensim

```
Summarize Text:
 The first flying vertebrates were true reptiles in which one of the fingers of the front of limbs became very elonga
ted, providing support for a flap of stretched skin that served as a wing. These flying reptiles had large, tooth-fil
led jaws, but their bodies were small and probably without the necessary powerful muscles for sustained wing movement
```

Gambar 6. Hasil summarization menggunakan library NLTK

Evaluasi yang digunakan untuk mengukur performansi dari sistem yang dibangun adalah dengan membandingkan hasil summary yang dihasilkan sistem dengan hasil summary yang dibuat oleh manusia. Tabel berikut, adalah hasil text

summarization yang dihasilkan oleh program, dan hasil summarization yang dibuat oleh instruktur TOEFL.

B. Kesimpulan

Text summarization salah satu implementasi IR (*information retrieval*) yang membuat teks dengan versi yang lebih pendek. Text summarization dilakukan dengan melalui beberapa tahapan antara lain *preprocessing document*, selanjutnya pembobotan dan ekstraksi kalimat, kemudian *filtering* dan penggabungan kalimat. Proses *text summarization* ini dapat dilakukan dengan menggunakan library python Gensim dan NLTK. Dari hasil evaluasi perbandingan hasil text summarization yang dilakukan secara manual (human) dengan automated menggunakan library Gensim dan NLTK didapatkan akurasi sebesar 72,8%.

4. DAFTAR PUSTAKA

1. Abualigah, L., Bashabsheh, M.Q., Alabool, H., and Shehab, M. 2020. Text Summarization: A Brief Review.
2. Akhter, B., and Mehra, M. 2022. A Study of Implementation of Deep Learning Techniques for Text Summarization. *International Journal of Innovative Research in Engineering & Management (IJIREM)*.
3. Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E.D., Gutierrez, J.B., and Kochut, K. 2017. Text Summarization Techniques: A Brief Survey. *In Proceedings of arXiv*, USA, July 2017, 9 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn
4. Bharti, S.K.,, Babu, K.S., and Jena, S.K. 2017. Automatic Keyword Extraction for Text Summarization: A Survey.
5. Ceri, S., Bozzon, A., Brambilla, M., Valle, E.D., Fraternali, P., and Quarteroni, S. 2013. Web Information Retrieval. Springer.
6. Kumar, K.S., Priyanka, M., Rishitha, M., Teja, D.D., Madhuri, N. 2021. Text Summarization with Sentimental Analysis. *International Journal of Innovative Research in Computer Science & Technology (IJIRCST)*.
7. Nenkova A., McKeown K. 2011. Automatic Summarization. *Foundations and Trends in Information Retrieval*. Vol. 5, Nos. 2–3 (2011).
8. Rifano, E.J., Fauzan, A.C., Makhi, A., Nadya, E., Nasikin, Z., Putra, F.N. 2020. Text Summarization pada Berita Bola Menggunakan Library Natural Language Toolkit (NLTK) Berbasis Pemrograman Python. *ILKOMNIKA: Journal of Computer Science and Applied Informatics*. Vol. 2, No. 1, April 2020, Halaman 8-17.
9. Romadhony, A., Fariska Z.R, Yusliani, N., Abednego, L. 2011. Text Summarization untuk Dokumen Berita Berbahasa Indonesia.
10. Sareen, S. 2018. Text Summarisation with Gensim (TextRank Algorithm). https://medium.com/@shivangisareen/text-summarisation-with-gensim-textrank-46bbb3401289
11. Yao, J. 2019. Automated Sentiment Analysis of Text Data with NLTK.

Department of Communication, Beijing University of Posts and Telecommunications, China. *Journal of Physics*.

12. Zhai, C., Massung, S. 2016. *Text Data Management and Analysis A Practical Introduction to Information Retrieval and Text Mining*. ACM Books.
13. https://dockship.io/articles/600d07855c9276402bd79023/text-summarization-using-sumy-and-gensim
14. https://www.geeksforgeeks.org/python-extractive-text-summarization-using-gensim/