



Peringkasan Teks Otomatis Berita Online Komisi Pemilihan Umum Menggunakan Algoritma K-Means Clustering

Ezra Matthew Warouw Runturamby*¹, Vivi Peggie Rantung², Kristofel Santa³
^{1,2,3}Program Studi Teknik Informatika Universitas Negeri Manado
19210021@unima.ac.id

Abstract

This research aims to develop an automatic text summarization system capable of summarizing online news about the General Election Commission (KPU) using the K-Means Clustering algorithm. In the current digital era, online news has become a primary source of information for the public, but the overwhelming amount of available information often makes it difficult for readers to filter and comprehend news efficiently. The low reading interest of the public further exacerbates this issue. Therefore, the automatic text summarization system is expected to provide a solution by helping readers quickly and effectively grasp the essence of the news. The K-Means Clustering algorithm will group sentences in the news into several clusters, which will then be used to create a representative summary. This research also identifies challenges such as the accuracy of the summary and the diversity of language in the news. The implementation of this system is expected to improve readers' time efficiency, provide better access to information, and support increased public participation in the democratic process.

Keywords: Automatic Text Summarization, K-Means Clustering, Online News, General Election Commission (KPU), Information Filtering, Reading Interest, News Summarization.

Abstrak

Penelitian ini bertujuan untuk mengembangkan sistem peringkasan teks otomatis yang mampu meringkas berita online tentang Komisi Pemilihan Umum (KPU) menggunakan algoritma K-Means Clustering. Di era digital saat ini, berita online telah menjadi sumber informasi utama bagi masyarakat, namun banyaknya informasi yang tersedia sering kali membuat pembaca kesulitan untuk menyaring dan memahami isi berita secara efisien. Minat baca masyarakat yang rendah juga memperburuk situasi ini. Oleh karena itu, sistem peringkasan teks otomatis diharapkan dapat memberikan solusi dengan membantu pembaca mendapatkan intisari berita dengan lebih cepat dan efektif. Algoritma K-Means Clustering akan mengelompokkan kalimat dalam berita menjadi beberapa kluster, yang nantinya digunakan untuk menyusun ringkasan yang representatif. Penelitian ini juga mengidentifikasi tantangan seperti akurasi ringkasan dan keragaman bahasa dalam berita. Penerapan sistem ini diharapkan dapat meningkatkan efisiensi waktu pembaca, memberikan akses informasi yang lebih baik, serta mendukung peningkatan partisipasi masyarakat dalam proses demokrasi.

Kata kunci: Peringkasan Teks Otomatis, K-Means Clustering, Berita Online, Komisi Pemilihan Umum (KPU), Penyaringan Informasi, Minat Baca, Peringkasan Berita.

1. Pendahuluan

Di era digital saat ini, perkembangan teknologi informasi dan komunikasi telah memberikan dampak yang signifikan dalam berbagai aspek kehidupan, termasuk cara masyarakat mengakses dan mengonsumsi informasi [1]. Berita online telah menjadi salah satu sumber informasi utama bagi banyak orang karena kemudahan akses dan kecepatan penyebarannya. Namun, kemudahan ini juga membawa tantangan baru, salah satunya adalah banyaknya informasi yang tersedia membuat pembaca kesulitan dalam menyaring dan mencerna berita secara efisien [2].

Di sinilah pentingnya pengembangan sistem peringkasan teks otomatis. Sistem ini bertujuan untuk membantu pembaca mendapatkan intisari atau ringkasan dari berita-berita yang panjang, sehingga informasi penting

dapat diserap dengan lebih cepat dan efisien. Peringkasan teks otomatis menggunakan algoritma seperti K-Means Clustering dapat menjadi solusi efektif dalam menyederhanakan proses ini. Algoritma K-Means Clustering mampu mengelompokkan kalimat-kalimat dalam berita ke dalam beberapa cluster, yang kemudian dapat digunakan untuk menyusun ringkasan yang representatif.

Penerapan algoritma K-Means Clustering pada peringkasan teks otomatis berita online tentang KPU akan memberikan beberapa manfaat penting. Pertama, membantu masyarakat untuk tetap mendapatkan informasi penting tentang kegiatan dan kebijakan KPU tanpa harus membaca seluruh berita secara mendetail. Kedua, meningkatkan efisiensi waktu bagi pembaca yang memiliki keterbatasan waktu untuk membaca berita secara lengkap. Ketiga, mendukung upaya

meningkatkan partisipasi masyarakat dalam proses demokrasi dengan menyediakan informasi yang lebih mudah diakses dan dipahami [3].

Salah satu area penting yang sering menjadi sorotan dalam berita online adalah informasi mengenai Komisi Pemilihan Umum (KPU). KPU memainkan peran krusial dalam menyelenggarakan pemilihan umum yang jujur dan adil, sehingga informasi mengenai kegiatan, kebijakan, dan hasil kerja KPU sangat penting bagi publik. Namun, dengan jumlah berita yang begitu banyak, pembaca sering kali mengalami kesulitan untuk mengikuti seluruh informasi yang relevan. Hal ini diperparah dengan keterbatasan minat baca masyarakat yang cenderung menurun terhadap berita - berita panjang dan kompleks [4].

Format berita online yang tergolong kompleks dan biasanya tidak dibaca karena kurangnya minat baca di Indonesia. Berdasarkan penelitian, minat baca di Indonesia masih rendah, dipengaruhi oleh beberapa faktor seperti akses informasi yang terbatas dan kurangnya kesadaran akan pentingnya literasi. Selain itu, masyarakat cenderung lebih memilih hiburan visual dibandingkan dengan konten berbasis teks. Kondisi ini diperparah oleh ketidakmampuan sebagian besar masyarakat untuk menyaring informasi secara efektif di tengah banyaknya sumber berita yang tersedia [5].

Namun, tantangan yang dihadapi dalam implementasi sistem ini juga tidak sedikit. Beberapa di antaranya termasuk memastikan akurasi dan relevansi ringkasan yang dihasilkan oleh algoritma, serta mengatasi keragaman bahasa dan gaya penulisan dalam berita-berita online. Oleh karena itu, penelitian lebih lanjut diperlukan untuk mengoptimalkan algoritma K-Means Clustering dan mengevaluasi kinerja sistem peringkasan otomatis ini dalam konteks berita online tentang KPU.

Dengan demikian, pengembangan peringkasan teks otomatis menggunakan algoritma K-Means Clustering ini diharapkan dapat menjadi solusi yang efektif untuk mengatasi keterbatasan minat baca terhadap berita online mengenai KPU, serta mendukung masyarakat dalam memperoleh informasi yang akurat dan relevan dengan cara yang lebih efisien [6].

2. Metode Penelitian

Berikut adalah penjelasan yang lebih rinci terkait proses dan alasan di balik setiap tahapannya Algoritma K-Means Clustering:

Frekuensi Kata: Mengidentifikasi kata-kata yang paling sering muncul dalam teks. Lakukan tokenization (proses memecah teks menjadi kata-kata individual). Hitung kemunculan setiap kata (frekuensi). Dalam contoh teks di atas, kita menghitung berapa kali setiap kata muncul. Misalnya, "Pemilihan" muncul 3 kali, "Umum" muncul 3 kali, "Komisi" muncul 2 kali, dan seterusnya. Frekuensi kata memberikan indikasi kata-kata yang paling penting atau sering digunakan dalam teks. Ini bisa

digunakan sebagai dasar untuk analisis lebih lanjut, seperti clustering atau pembuatan model.

Pemilihan Kata untuk Dataset: Memilih kata-kata yang akan dijadikan data dalam dataset. Pilih kata-kata berdasarkan frekuensi tertinggi atau yang memiliki nilai informasi paling penting.

Misalnya, dalam contoh di atas, kita memilih 10 kata teratas berdasarkan frekuensi kemunculannya. Memilih kata-kata yang paling sering muncul adalah salah satu pendekatan untuk membatasi ukuran dataset tanpa kehilangan informasi penting dari teks. Ini juga memudahkan dalam analisis clustering karena data menjadi lebih manageable.

Pemberian Koordinat (Fitur x dan y): Memberikan nilai numerik yang representatif untuk setiap kata dalam dataset. Untuk membuat dataset numerik, kita memberikan nilai koordinat acak untuk setiap kata. Nilai koordinat ini (x dan y) bisa disimulasikan sebagai karakteristik dari kata-kata tersebut, misalnya panjang kata, urutan kemunculan dalam teks, atau posisi dalam kalimat. Dalam contoh, kita hanya menggunakan angka acak dalam rentang tertentu, misalnya $x = (-12, 12)$ dan $y = (-10, 11)$. Dalam analisis clustering, kita perlu menempatkan setiap data (kata) dalam ruang multidimensi (diwakili oleh fitur x dan y). Meskipun di sini kita menggunakan angka acak untuk kesederhanaan, dalam aplikasi nyata, koordinat ini bisa didasarkan pada karakteristik yang lebih spesifik dan bermakna.

Menentukan Cluster: Kelompokkan kata-kata ke dalam beberapa cluster berdasarkan fitur x dan y yang telah diberikan. Setiap kata diberikan label cluster, di sini kita memilih secara acak (misalnya cluster = 0, 1, atau 2). Dalam aplikasi nyata, algoritma clustering seperti K-Means digunakan untuk secara otomatis membagi kata-kata ke dalam beberapa cluster berdasarkan fitur x dan y. Proses clustering bertujuan untuk mengelompokkan data (kata-kata) yang memiliki karakteristik serupa ke dalam kelompok-kelompok yang sama (cluster). Dalam teks, cluster ini mungkin bisa mewakili kelompok kata-kata yang terkait secara tematis atau digunakan dalam konteks yang mirip.

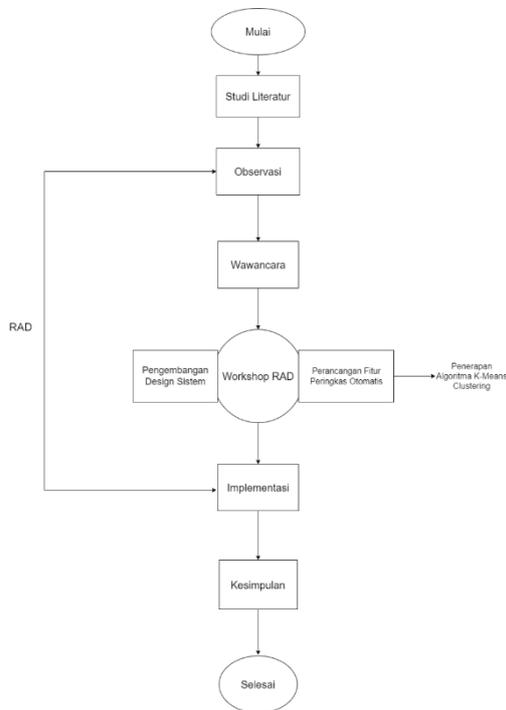
Evaluasi Hasil Clustering: Setelah menentukan cluster, kita bisa mengevaluasi hasilnya dengan menggunakan berbagai metrik, seperti sum of squared errors (SSE) atau adjusted rand score. SSE memberikan informasi tentang seberapa baik cluster dibentuk dengan menghitung jarak antara titik-titik (kata-kata) dalam cluster terhadap pusat cluster (centroid). Rand score mengukur kesesuaian antara cluster yang dihasilkan oleh algoritma dengan kategori asli (jika tersedia).

Visualisasi Hasil Clustering: Untuk memahami hasil clustering, kita bisa memvisualisasikannya dalam grafik 2D. Dengan menggunakan fitur x dan y, kita dapat membuat plot scatter (sebaran titik) di mana setiap titik mewakili sebuah kata, dengan warnanya menunjukkan

cluster yang sesuai. Ini membantu dalam memahami pola cluster dan distribusi kata-kata dalam dataset.

Kesimpulan: Dataset ini bisa digunakan untuk analisis lebih lanjut, clustering kata-kata berdasarkan fitur x dan y dapat dimodifikasi atau diperbaiki agar lebih representatif terhadap karakteristik kata-kata (misalnya, panjang kata, frekuensi, posisi, atau makna semantik). Peringkasan otomatis setelah clustering, kita dapat memilih kalimat atau kata yang paling representatif dari setiap cluster untuk membuat ringkasan teks [7].

Pada tahap penelitian. Penelitian ini menggunakan metode Rapid Application Development sebagai metode penelitian dan Algoritma K-Means untuk perancangan fitur peringkasan otomatis. Pada tahapan penelitian ini menggunakan teknik pengumpulan data di atas bisa dilihat pada Gambar 1.



Gambar 1 Tahapan Penelitian

Rapid Application Development (RAD) adalah suatu metode proses pengembangan perangkat lunak yang bersifat inkremental, dimana saya menggunakan metode ini terutama untuk merancang waktu pengerjaan yang singkat. Model RAD merupakan modifikasi dari model air terjun untuk mengembangkan setiap komponen perangkat lunak [8].

Berikut Penjelasan mengenai tahapan penelitian metode RAD :

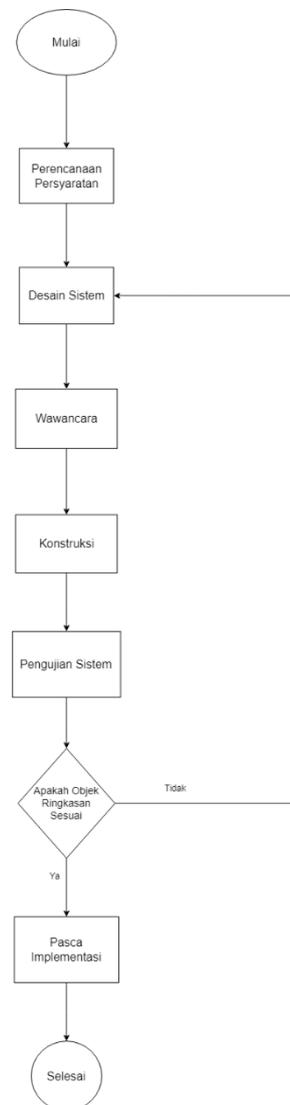
Requirements Planning (Perencanaan Persyaratan): Tahap ini melibatkan identifikasi kebutuhan dan persyaratan yang harus dipenuhi pada pengembangan sistem. Pada metode RAD melakukan identifikasi apa saja yang menjadi pokok bahasan dan apa saja yang perlu di tingkatkan pada desain sistem peringkasan otomatis sehingga pada tahap pengembangan fitur

peringkasan otomatis dapat meringkas apa saja yang menjadi pokok bahasan [9].

User Design (Desain Pengguna): Tahap ini adalah proses pembuatan desain sistem untuk fitur peringkasan otomatis yang sesuai dengan kebutuhan yang sudah di lakukan pada tahap pertama dan juga tahap ini adalah proses pengembangan apa – apa saja yang perlu di kembangkan dari hasil perencanaan persyaratan.

Construction (Konstruksi): Pada tahap ini pembangunan perangkat lunak sesuai desain telah disepakati. Dan setelah menggunakan metode RAD untuk mempercepat pengembangan dan melakukan evaluasi dari peringkasan otomatis apa saja komponen yang dapat digunakan Kembali.

Gambar 2 adalah gambar flowchart tahapan metode RAD :



Gambar 2 Flowchart Tahapan Metode RAD

Berikut Penjelasan mengenai tahapan penelitian Algoritma K-Means :

Pada penelitian ini menggunakan Algoritma K-means Clustering dengan beberapa tahapan seperti Text Pre-processing, Text Tokenization, Stopword Removal, Stemming, Pembobotan TF-IDF, dan Klasterisasi Kalimat. Di bawah ini adalah beberapa penjelasan mengenai tahapan algoritma K-means :

Text Pre-Processing: Tahap ini adalah untuk mempersiapkan dan membersihkan teks mentah sebelum diproses lebih lanjut dalam analisis teks atau pemrosesan bahasa alami. Tujuan utama dari text pre-processing adalah untuk mengurangi kekacauan dan menghasilkan representasi teks yang lebih terstruktur. meliputi text tokenizing, stemming dan stopword removal [10].

Text Tokenization: Proses memecah teks menjadi unit-unit yang lebih kecil yang disebut "token". Setiap token dapat berupa kata, frasa, kalimat, atau simbol tertentu. Tujuan dari text tokenization adalah untuk memecah teks menjadi unit-unit yang terpisah sehingga dapat dianalisis lebih lanjut dalam pemrosesan bahasa alami [11].

Stopword Removal: Proses menghapus kata-kata umum yang tidak memiliki makna khusus dalam analisis teks. Kata-kata tersebut disebut "stopword" karena mereka sering muncul dalam teks namun jarang memberikan kontribusi signifikan terhadap pemahaman atau analisis teks yang lebih dalam. Tujuan utama dari stopword removal adalah untuk meningkatkan efisiensi dan akurasi analisis teks dengan menghilangkan kata-kata yang tidak membawa informasi penting. Dengan menghapus stopword, kita dapat memfokuskan perhatian pada kata-kata kunci yang lebih berarti dan berkontribusi pada pemahaman teks secara keseluruhan.

Stemming: proses untuk mengubah kata-kata menjadi bentuk dasarnya atau kata dasar dengan cara menghapus akhiran kata. Tujuan utama dari stemming adalah untuk mengurangi variasi morfologi kata dalam teks sehingga kata-kata yang memiliki akar atau makna yang sama dapat dianggap sebagai satu entitas. Misalnya, dalam stemming, kata-kata seperti "berlari", "berlarian", dan "berlari-lari" akan diubah menjadi bentuk dasarnya yaitu "lari". Dengan melakukan stemming, kita dapat menggabungkan kata-kata yang terkait dan mendapatkan representasi yang lebih sederhana dan konsisten dari teks [12].

Pembobotan TF-IDF: digunakan dalam pengambilan informasi dan pemrosesan bahasa alami untuk menilai pentingnya suatu kata dalam dokumen atau korpus teks. Metode ini mengkombinasikan dua konsep kunci, yaitu frekuensi kata dalam dokumen (TF) dan kebalikan frekuensi kata dalam seluruh dokumen (IDF). Rumus 1 dan 2 berikut adalah persamaan untuk mendapatkan nilai pembobotan menggunakan TF-IDF :

$$IDF = 1 + \log D/df \quad (2)$$

$$TFIDF = TF * (1 + \log D/df) \quad (3)$$

Penghitungan Tf (Term Frequency): Tf mengukur seberapa sering sebuah kata muncul dalam sebuah dokumen. Ini dihitung dengan cara membagi jumlah kemunculan sebuah kata dalam dokumen dengan jumlah total kata dalam dokumen tersebut. Tujuan dari Tf adalah untuk memberikan bobot lebih kepada kata-kata yang muncul lebih sering dalam sebuah dokumen karena kemungkinan besar kata-kata tersebut lebih penting dalam mendeskripsikan isi dokumen tersebut.

Penghitungan Idf (Inverse Document Frequency): Idf mengukur seberapa penting suatu kata dalam seluruh kumpulan dokumen. Kata-kata yang muncul dalam banyak dokumen dianggap kurang penting dibandingkan kata-kata yang muncul dalam jumlah dokumen yang lebih sedikit. Idf dihitung dengan logaritma dari pembagian jumlah total dokumen dengan jumlah dokumen yang mengandung kata tersebut.

Perkalian Tf dengan Idf (Tf-idf Weighting): Setelah mendapatkan nilai Tf dan Idf untuk setiap kata dalam dokumen, langkah selanjutnya adalah mengalikan nilai Tf dengan nilai Idf untuk setiap kata. Hal ini menghasilkan bobot yang lebih tinggi untuk kata-kata yang sering muncul dalam sebuah dokumen tetapi jarang muncul dalam seluruh kumpulan dokumen [13].

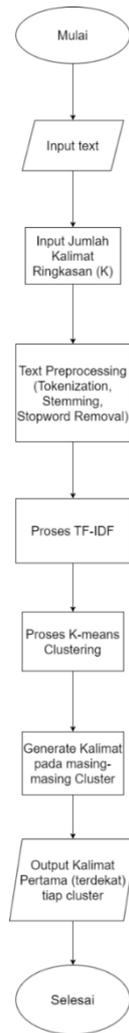
Klasterisasi Kalimat:

Klasterisasi kalimat, juga dikenal sebagai pengelompokan kalimat, adalah proses mengelompokkan kalimat-kalimat berdasarkan kesamaan topik atau makna mereka. Tujuan dari klasterisasi kalimat adalah mengidentifikasi pola dan struktur yang ada dalam teks yang memungkinkan pengelompokan kalimat-kalimat yang serupa berada dalam satu kelompok.

Gambar 3 adalah Flowchart Tahapan Algoritma K-Means.: Cutover (Beralih)

Tahap ini adalah tahap pengujian, evaluasi, dan persiapan untuk penggunaan sistem yang sudah dikembangkan dari tahap ketiga [14].

Post-Implementation (Pasca-implementasi): Setelah perangkat lunak selesai dikembangkan dan akan dijalankan kembali pada divisi hukum komisi pemilihan umum kabupaten minahasa, akan dilakukan pemantauan apakah sistem bekerja dengan baik dan perbaikan jika diperlukan, serta juga mengumpulkan umpan balik atau saran yang diberikan oleh instansi untuk pengembangan dan perbaikan lanjutan. Rapid Application Development (RAD) dalam pengembangan sistem informasi berbasis web. Dengan menggunakan algoritma K-means Clustering, sistem dapat mengelompokkan data secara efisien untuk meringkas informasi penting dari teks yang Panjang [15].



Gambar 3 Flowchart Tahapan Algoritma K-Means Clustering

3. Hasil dan Pembahasan

Penentuan koordinat acak dalam analisis kluster, khususnya untuk metode seperti K-means, sering digunakan sebagai langkah awal dalam proses clustering atau untuk membuat data sintesis untuk pengujian algoritma. Metode ini dikenal sebagai "random initialization" dan memiliki peran penting dalam kinerja algoritma clustering.

Dalam konteks ini, koordinat acak biasanya dihasilkan menggunakan distribusi uniform. Distribusi uniform memastikan bahwa setiap nilai dalam rentang yang ditentukan memiliki probabilitas yang sama untuk dipilih. Rumus 4 untuk menghasilkan nilai acak dari distribusi uniform adalah:

$$x = a + (b - a) * \text{random()} \quad (4)$$

x adalah nilai acak yang dihasilkan; a adalah batas bawah rentang; b adalah batas atas rentang; random() adalah fungsi yang menghasilkan angka acak antara 0 dan 1,

Dalam kasus kita, untuk koordinat x:

$$x = -12 + (12 - (-12)) * \text{random()} = -12 + 24 * \text{random}()$$

Dan untuk koordinat y:

$$y = -10 + (11 - (-10)) * \text{random()} = -10 + 21 * \text{random}()$$

Meskipun metode ini sederhana dan mudah diimplementasikan, perlu dicatat bahwa inisialisasi acak dapat mempengaruhi hasil akhir clustering, terutama untuk algoritma seperti K-means yang sensitif terhadap inisialisasi awal. Oleh karena itu, dalam praktik nyata, sering digunakan metode inisialisasi yang lebih canggih seperti K-means++ untuk meningkatkan kinerja dan konsistensi hasil clustering.

Frekuensi Kata: Teks yang diambil dari link: <https://liputankawanua.com/2024/05/kpu-minahasa-sosialisasikan-tahapan-pilkada-2024/>

"Komisi Pemilihan Umum menggelar sosialisasi Pemilihan Umum di Kabupaten Minahasa. Sosialisasi ini bertujuan untuk meningkatkan partisipasi masyarakat dalam Pemilihan Umum. Komisi Pemilihan Umum berharap masyarakat akan aktif berpartisipasi."

Penghitungan frekuensi kata: Rumus: Frekuensi kata = Jumlah kemunculan kata dalam teks

Perhitungan manual:

Komisi: 2; Pemilihan: 3; Umum: 3; menggelar: 1; sosialisasi: 2; di: 1; Kabupaten: 1; Minahasa: 1; ini: 1; bertujuan: 1; untuk: 1; meningkatkan: 1; partisipasi: 2; masyarakat: 2; dalam: 1; berharap: 1; akan: 1; aktif: 1; berpartisipasi: 1

Membuat tabel dengan koordinat acak: Kita akan mengambil 10 kata teratas berdasarkan frekuensi untuk membuat tabel.

Rumus 5 adalah untuk untuk koordinat acak:

$$x = \text{random}(-12, 12) \quad y = \text{random}(-10, 11) \quad \text{cluster} = \text{random}(0, 2)$$

Contoh perhitungan manual (menggunakan angka acak yang kita pilih):

"Pemilihan" (frekuensi: 3) $x = 5.123456789$ $y = -3.987654321$ cluster = 1

"Umum" (frekuensi: 3) $x = -8.234567890$ $y = 7.876543210$ cluster = 0

"Komisi" (frekuensi: 2) $x = 0.345678901$ $y = 9.765432109$ cluster = 2

"sosialisasi" (frekuensi: 2) $x = -3.456789012$ $y = -5.678901234$ cluster = 1

"partisipasi" (frekuensi: 2) $x = 7.890123456$ $y = 2.345678901$ cluster = 0

"masyarakat" (frekuensi: 2) $x = -1.234567890$ $y = 10.987654321$ cluster = 2

"menggelar" (frekuensi: 1) $x = 11.111111111$ $y = -9.999999999$ cluster = 1

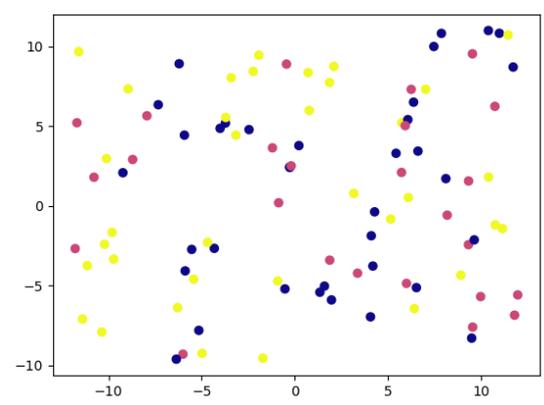
- 8) "di" (frekuensi: 1) x = -10.987654321 y = 10) "Minahasa" (frekuensi: 1) x = -4.321098765 y = 0.123456789 cluster = 0
 = -7.654321098 cluster = 1
- 9) "Kabupaten" (frekuensi: 1) x = 6.543210987 y = 8.765432109 cluster = 2

Tabel 1 Hasil Koordinat Acak

No	x	y	cluster	word	count
0	5.123456789	-3.987654321	1	Pemilihan	3
1	8.23456789	7.87654321	0	Umum	3
2	0.345678901	9.765432109	2	Komisi	2
3	-3.456789012	-5.678901234	1	sosialisasi	2
4	7.890123456	2.345678901	0	partisipasi	2
5	-1.23456789	10.98765432	2	masyarakat	2
6	11.11111111	-9.999999999	1	menggelar	1
7	-10.98765432	0.123456789	0	di	1
8	6.543210987	8.765432109	2	Kabupaten	1
9	-4.321098765	-7.654321098	1	Minahasa	1

Tabel 2 Bagian Cluster dari jumlah kata dari dalam link

cluster	count
0	36
2	35
1	25



Gambar 4 Tampilan dari Bagian Cluster

Pada Gambar 4 terdapat titik-titik data dalam tiga warna (kuning, merah muda, dan biru tua) tersebar di seluruh bidang. StandardScaler digunakan untuk menstandarisasi data sehingga memiliki mean = 0 dan standard deviation = 1. Rumus 5 untuk menghitung nilai yang sudah distandardisasi adalah:

$$z = (x - \mu) / \sigma \quad (2)$$

x adalah Nilai asli dari fitur; μ adalah Rata-rata dari fitur, σ adalah Standar deviasi dari fitur; z adalah Nilai fitur setelah standarisasi.

```
array([[ -0.24668132, -0.95193476],
       [ 1.44446741, -0.39288189],
       [ 0.92810779,  1.54878689],
       [ 0.72192671, -0.97762242],
       [-0.7245193 , -0.60382502]])
```

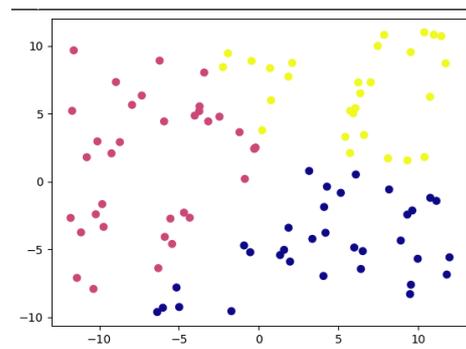
Sebelum menerapkan algoritma clustering, data diproses menggunakan StandardScaler. Langkah ini

penting karena dapat menormalisasi fitur-fitur dengan menghilangkan rata-rata dan menskalakan ke variansi unit dan memastikan semua fitur berkontribusi secara setara dalam proses clustering, mencegah fitur dengan skala lebih besar mendominasi analisis.

Model K-Means menentukan jumlah cluster yang akan digunakan dengan tujuan memilih centroid secara random dan tidak terjebak di lokal minimum sehingga menghasilkan label dibawah.

```
array([[0, 0, 2, 0, 1, 0, 0, 1, 1, 1, 2, 2, 1, 0, 1, 1, 1, 0, 2, 0, 2, 2,
        2, 1, 0, 0, 1, 1, 2, 0, 2, 1, 0, 1, 1, 0, 0, 0, 2, 1, 0, 0, 2, 1,
        2, 0, 1, 0, 1, 0, 0, 1, 0, 2, 0, 1, 2, 1, 2, 1, 1, 0, 2, 0, 1, 1,
        1, 2, 2, 2, 1, 1, 1, 1, 0, 2, 2, 0, 2, 0, 2, 1, 1, 1, 0, 0, 0, 1,
        1, 0, 2, 2, 2, 0, 2, 2], dtype=int32)
```

Setelah pra-pemrosesan, algoritma clustering K-Means diterapkan K-Means bertujuan untuk membagi n observasi menjadi k cluster, di mana setiap observasi termasuk dalam cluster dengan rata-rata (centroid) terdekat. Algoritma ini kemungkinan diatur untuk membuat 3 cluster, berdasarkan warna-warna dalam visualisasi. K-Means bekerja dengan cara: Menginisialisasi centroid secara acak; Menugaskan setiap titik ke centroid terdekat; Menghitung ulang centroid berdasarkan titik-titik yang ditugaskan; Mengulangi langkah 2-3 hingga konvergen seperti pada Gambar 5.

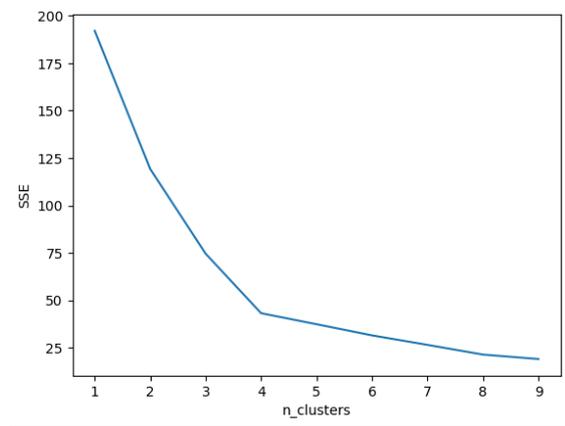


Gambar 5 Tampilan Setelah Menerapkan Model K-Means Clustering

Evaluasi SSE (Sum of Squared Errors): Digunakan ketika tidak punya label atau cluster dari awal dan mengevaluasi menggunakan Evaluasi SSE.

1. 192.000000000000006
2. 119.20560395936744
3. 74.5450062468814
4. 43.199362656537254
5. 37.38625365700291
6. 31.493218484029637
7. 26.5029217795395
8. 21.400841070999167
9. 19.054628474740397

Selama proses evaluasi menghasilkan 10 iterasi nilai SSE atau jarak dari setiap data terhadap centroid dari awal data besar dan semakin lama mengecil dan setelah cluster ke 3 tidak terjadi secara signifikan berkurang.



Gambar 6 Visualisasi Evaluasi Sum of Squared Errors

Grafik evaluasi algoritma K-Means menggunakan Sum of Squared Errors (SSE) menunjukkan penurunan yang signifikan ketika jumlah cluster ($n_clusters$) bertambah. Berikut adalah penjelasan dari hasil tersebut:

Penurunan Drastis di Awal ($n_clusters = 1$ hingga 3):

Alasan: Pada jumlah cluster yang sangat kecil (misalnya 1 atau 2), semua titik data dikelompokkan bersama dalam satu atau dua cluster, sehingga menghasilkan SSE yang sangat tinggi. Ini karena jarak antara titik-titik data dengan pusat cluster sangat besar. Ketika jumlah cluster bertambah, SSE menurun drastis karena data dapat dikelompokkan lebih baik, yang mengurangi jarak antara titik data dengan centroid cluster.

Penurunan Stabil Setelah $n_clusters > 3$:

Setelah mencapai sekitar 3 atau 4 cluster, penurunan SSE menjadi lebih lambat. Ini menunjukkan bahwa penambahan cluster tambahan masih memberikan peningkatan dalam pemisahan data, tetapi tidak sebanyak sebelumnya. Pada titik ini, K-Means mulai

menangkap struktur yang lebih baik dari data, tetapi penambahan cluster selanjutnya hanya sedikit memperbaiki hasil.

Titik Kestabilan atau Elbow Point ($n_clusters = 4$ hingga 5):

Pada jumlah cluster tertentu, penurunan SSE mulai melambat secara signifikan, yang biasanya dikenal sebagai elbow point. Pada grafik Anda, titik elbow kemungkinan berada di sekitar 4 atau 5 cluster. Ini adalah indikasi bahwa jumlah cluster ini adalah optimal, di mana penambahan cluster selanjutnya hanya memberikan sedikit penurunan SSE, tetapi kompleksitas model meningkat.

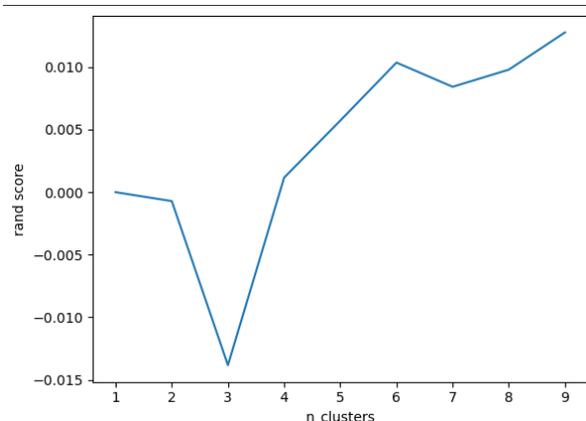
Penurunan yang Sangat Lambat Setelah $n_clusters > 5$:

Setelah 5 cluster, SSE terus menurun tetapi dengan laju yang sangat lambat. Ini menunjukkan bahwa penambahan cluster memberikan sedikit manfaat tambahan. Pada titik ini, cluster tambahan mungkin mulai menyebabkan overfitting, di mana cluster terlalu memecah data secara berlebihan.

Secara keseluruhan, SSE akan terus menurun ketika jumlah cluster bertambah, namun penurunan signifikan hanya terjadi di tahap awal. Titik elbow dalam grafik ini menunjukkan jumlah cluster optimal yang dapat digunakan dalam pemisahan data.

Evaluasi Adjusted Rand

1. 0.0
2. -0.0007138309571122644
3. -0.01382678398483381
4. 0.0011605272770146245
5. 0.005719788613923186
6. 0.010360097847430763
7. 0.008423695578792464
8. 0.009792918966285917
9. 0.012770508548933282



Gambar 7 Tampilan Visualisasi Evaluasi Adjusted Rand

Grafik hasil evaluasi algoritma K-Means menggunakan Rand Score menunjukkan fluktuasi nilai dari jumlah cluster yang berbeda. Pergerakan naik turun pada grafik tersebut dapat dijelaskan dengan alasan-alasan berikut:

Penurunan Awal pada Rand Score ($n_clusters = 2$ hingga 3) sehingga pada saat jumlah cluster rendah (misalnya $2-3$), K-Means mungkin tidak mampu menangkap struktur alami dalam data. Hal ini disebabkan oleh underfitting, di mana jumlah cluster yang kecil tidak cukup untuk membagi data secara efektif sesuai dengan pola yang ada. Akibatnya, skor evaluasi, seperti Rand Score, bisa menurun karena cluster yang terbentuk kurang representatif.

Peningkatan Setelah $n_clusters > 3$:

Ketika jumlah cluster meningkat, K-Means mulai mampu mengelompokkan data secara lebih baik sesuai dengan distribusi aslinya. Cluster yang lebih banyak memberikan lebih banyak fleksibilitas bagi algoritma untuk memisahkan data secara lebih tepat. Ini menjelaskan mengapa Rand Score mulai naik setelah 3 cluster.

Titik optimal K-Means mendekati jumlah cluster optimal yang mencerminkan struktur sebenarnya dalam data. Ini biasanya terjadi pada rentang di mana Rand Score meningkat secara signifikan, seperti antara $n_clusters$ 4 hingga 9 dalam grafik Anda.

Fluktuasi di $n_clusters$ yang lebih tinggi sehingga pada jumlah cluster yang lebih tinggi (misalnya $7-9$), terdapat fluktuasi kecil yang mungkin disebabkan oleh overfitting. Jumlah cluster yang terlalu banyak dapat menyebabkan K-Means memecah data menjadi cluster yang terlalu kecil atau tidak relevan. Meskipun Rand Score masih meningkat, fluktuasi kecil dapat terjadi karena adanya kesulitan dalam membagi data secara efisien ketika cluster terlalu banyak.

Secara umum, pola naik-turun ini biasa terjadi pada evaluasi K-Means karena algoritma harus menemukan keseimbangan antara underfitting dan overfitting, yang tercermin dari evaluasi seperti Rand Score.

Proses tahapan pembuatan sistem ini menggunakan metode penelitian RAD (Rapid Application Development) dan berikut adalah penjelasan mengenai tahapan – tahapan yang sudah dilaksanakan.

Pada tahap Requirements Planning (Perencanaan Persyaratan) dalam penelitian ini, terdapat beberapa kebutuhan yang harus dipenuhi, yaitu seperti pada Tabel 3.

Tabel 3 Kebutuhan Penggunaan

No	Kebutuhan	Deskripsi Kode
1.	Sistem harus dapat mengekstrak konten teks dari halaman HTML yang diberikan	Fungsi <code>extract_content_from_html()</code> digunakan untuk memenuhi kebutuhan ini. Fungsi ini menggunakan library BeautifulSoup untuk

	melalui URL.	mengambil semua paragraf teks dari halaman HTML, dan menggabungkannya menjadi satu teks utuh, serta mengekstrak informasi tambahan seperti penulis dan tanggal publikasi berita.
2.	Sistem harus dapat meringkas teks berita menjadi beberapa kalimat inti yang mewakili isi berita.	Fungsi <code>summarize_text_with_kmeans(text, num_clusters=5)</code> digunakan untuk memenuhi kebutuhan ini. Fungsi ini menggunakan library scikit-learn untuk melakukan tokenisasi kalimat, perhitungan TF-IDF, dan pengelompokan kalimat menggunakan algoritma K-means Clustering. Hasilnya adalah kumpulan kalimat terpilih yang mewakili ringkasan berita.
3.	Sistem harus menyediakan antarmuka pengguna yang memungkinkan pengguna untuk memasukkan URL berita dan melihat ringkasan yang dihasilkan.	Fungsi <code>summarize_from_url(request)</code> menangani permintaan pengguna, mengambil konten dari URL yang diberikan, meringkas teks, dan menampilkan hasil ringkasan serta informasi tambahan seperti penulis dan tanggal publikasi.

Dengan memenuhi kebutuhan-kebutuhan di atas, sistem dapat mengekstrak konten berita, meringkas teks menggunakan algoritma K-means Clustering, dan menyajikan hasil ringkasan kepada pengguna. Tahap Requirements Planning ini menjadi dasar bagi tahap-tahap selanjutnya dalam mengembangkan sistem sesuai dengan metode Rapid Application Development.

Construction (Konstruksi): Pada tahap ini, pengembangan aplikasi dilakukan berdasarkan desain yang telah dibuat. Kode aplikasi diimplementasikan dalam framework Django dengan menggunakan pustaka BeautifulSoup untuk pengambilan konten dan NLTK serta Scikit-learn untuk pemrosesan teks dan algoritma K-means Clustering. Berikut adalah fungsi utama untuk meringkas teks dari URL yang diberikan.

Text Pre-Processing

Tujuan: Mempersiapkan teks berita agar dapat diproses lebih lanjut oleh algoritma K-means Clustering.

Implementasi: Dalam kode, tidak ada fungsi khusus untuk text pre-processing. Namun, asumsi yang digunakan adalah bahwa teks berita telah melalui tahapan pembersihan, seperti penghapusan karakter khusus, konversi teks menjadi huruf kecil, dan lain-lain.

Tahap Text Pre-Processing adalah langkah pertama dalam proses peringkasan teks yang bertujuan untuk membersihkan dan mempersiapkan teks mentah untuk

analisis lebih lanjut. Langkah ini mencakup beberapa sub-tahap penting: Mengambil Teks dari URL

Mengambil teks dari halaman web:

URL : <https://liputankawanua.com/2024/05/kpu-minahasa-sosialisasikan-tahapan-pilkada-2024/>.

Menggunakan pustaka web scraping seperti requests untuk mengunduh konten halaman web dan BeautifulSoup untuk mem-parsing konten HTML.

requests mengunduh halaman web, sedangkan BeautifulSoup mem-parsing HTML dan mengekstrak teks dari elemen-elemen yang relevan seperti paragraf (<p>), judul (<h1>, <h2>, dll.), dan lainnya.

Menghapus Tag HTML: Setelah teks diambil dari halaman web, teks tersebut masih mengandung tag HTML yang tidak diperlukan untuk analisis teks. Proses ini meliputi: Mem-parsing konten HTML dengan BeautifulSoup; Menghapus semua tag HTML, menyisakan hanya teks murni; Menghapus elemen-elemen seperti skrip (<script>), gaya (<style>), dan komentar HTML (<!-- -->).

Menghapus Elemen yang Tidak Relevan: Langkah ini bertujuan untuk lebih membersihkan teks dari elemen-elemen yang tidak relevan dan berpotensi mengganggu analisis selanjutnya. Ini meliputi: Menghapus whitespace yang berlebihan, garis baru (\n), dan karakter-karakter khusus lainnya yang tidak diperlukan; Menghapus teks-teks atau elemen-elemen non-teks yang mungkin masih tertinggal.

Tahap Text Pre-Processing memastikan bahwa teks yang akan dianalisis sudah bersih dan siap untuk diolah lebih lanjut dengan teknik-teknik seperti tokenisasi, penghapusan stopword, stemming, pembobotan TF-IDF, dan klasterisasi kalimat menggunakan algoritma K-means.

Text Tokenization: Tujuan tahapan ini yaitu untuk Memecah teks berita menjadi kumpulan kalimat. dengan Implementasi sebagai berikut:

Tahap Text Tokenization adalah proses memecah teks yang sudah dibersihkan menjadi unit-unit yang lebih kecil, seperti kalimat atau kata. Pada konteks ini, kita akan memecah teks menjadi kalimat-kalimat. Tujuan utama dari tahap ini adalah untuk memudahkan analisis dan pengolahan teks pada langkah-langkah berikutnya.

Langkah-langkah Text Tokenization

Pemecahan Teks Menjadi Kalimat: Menggunakan alat atau pustaka pemrosesan bahasa alami (Natural Language Processing, NLP) untuk mengenali batas kalimat; Dalam bahasa Python, pustaka seperti nltk atau spaCy sering digunakan untuk melakukan tugas ini.

Mengidentifikasi Batas Kalimat: Algoritma NLP mendeteksi tanda baca seperti titik (.), tanda tanya (?), dan tanda seru (!) sebagai indikasi akhir kalimat; Algoritma ini juga mempertimbangkan konteks untuk

memastikan bahwa titik tidak disalahartikan sebagai akhir kalimat dalam kasus seperti singkatan (misalnya, "Mr.", "Dr.").

Menghasilkan Daftar Kalimat: Setelah identifikasi batas kalimat, teks dipecah sesuai dengan tanda baca yang telah dikenali; Setiap kalimat diperlakukan sebagai unit terpisah untuk dianalisis lebih lanjut.

Contoh Tokenisasi untuk URL : Teks Asli dari URL :

URL: <https://liputankawanua.com/2024/05/kpu-minahasa-sosialisasikan-tahapan-pilkada-2024/>

Teks yang Telah Dibersihkan:

"KPU Minahasa sosialisasikan tahapan Pilkada 2024. Tahapan Pilkada ini penting untuk keberhasilan demokrasi."

Hasil Tokenisasi: "KPU Minahasa sosialisasikan tahapan Pilkada 2024."; "Tahapan Pilkada ini penting untuk keberhasilan demokrasi."

Dalam contoh ini, teks asli yang sudah dibersihkan dipecah menjadi dua kalimat. Pemecahan ini didasarkan pada titik (.) sebagai penanda akhir kalimat. Hasilnya adalah dua kalimat terpisah yang akan dianalisis lebih lanjut pada tahap berikutnya seperti Stopword Removal, Stemming, Pembobotan TF-IDF, dan Klasterisasi Kalimat.

Tahap Text Tokenization penting karena setiap kalimat akan dianalisis untuk menentukan relevansinya dalam konteks dokumen dan untuk proses klasterisasi. Proses ini mempersiapkan teks untuk langkah-langkah pengolahan selanjutnya, memastikan setiap unit teks yang lebih kecil (kalimat) dapat dievaluasi dan diringkas dengan tepat.

Stopword Removal adalah tahap dalam pemrosesan teks di mana kata-kata umum yang sering muncul namun memiliki sedikit nilai informasi (dikenal sebagai stopwords) dihapus dari teks. Kata-kata ini termasuk berita, preposisi, dan kata-kata umum lainnya seperti "dan", "atau", "di", "yang", dll. Proses ini membantu fokus pada kata-kata yang lebih bermakna untuk analisis lebih lanjut.

Langkah-langkah Stopword Removal, Identifikasi Stopwords: Menggunakan daftar stopwords yang sudah ada (seperti yang disediakan oleh pustaka NLP seperti nltk atau spaCy); Daftar ini berisi kata-kata umum yang sering muncul dalam bahasa tertentu tetapi tidak menambahkan nilai signifikan dalam analisis.

Menghapus Stopwords: Setiap kalimat atau teks dipecah menjadi kata-kata individu (token); Setiap kata dibandingkan dengan daftar stopwords; Kata-kata yang cocok dengan daftar stopwords dihapus dari teks.

Kalimat Sebelum Stopword Removal: "KPU Minahasa sosialisasikan tahapan Pilkada 2024."; "Tahapan Pilkada ini penting untuk keberhasilan demokrasi."

Kalimat Setelah Stopword Removal: "KPU Minahasa sosialisasikan tahapan Pilkada 2024."; "Tahapan Pilkada penting keberhasilan demokrasi."

Dalam contoh ini, kata-kata seperti "ini" dan "untuk" telah dihapus karena dianggap sebagai stopwords yang tidak menambahkan banyak nilai informasi dalam analisis teks. Hasilnya adalah teks yang lebih fokus pada kata-kata yang lebih bermakna seperti "KPU", "Minahasa", "sosialisasikan", "tahapan", "Pilkada", "penting", "keberhasilan", dan "demokrasi".

Proses Stopword Removal membantu meningkatkan kualitas analisis teks dengan menghilangkan kata-kata yang tidak relevan, sehingga algoritma seperti TF-IDF dan K-means clustering dapat bekerja lebih efektif pada kata-kata yang lebih signifikan dalam teks.

Stemming adalah tahap dalam pemrosesan teks yang bertujuan untuk mengurangi kata-kata ke bentuk dasarnya (root form) atau bentuk kata dasar (stem). Proses ini membantu menyatukan variasi kata yang berbeda yang memiliki arti yang sama sehingga analisis teks menjadi lebih efektif.

Langkah-langkah Stemming, Identifikasi Bentuk Kata: Menggunakan algoritma stemming yang tersedia dalam pustaka NLP seperti nltk (Porter Stemmer, Snowball Stemmer) atau spaCy; Algoritma ini mengidentifikasi sufiks atau akhiran yang umum digunakan dalam bahasa tertentu.

Menghapus atau Mengubah Sufiks: Algoritma stemming menghapus atau mengubah sufiks dari kata-kata untuk mengurangi mereka ke bentuk dasarnya.; Proses ini dilakukan dengan aturan yang sudah ditentukan, misalnya mengubah "sosialisasikan" menjadi "sosialisasi".

Contoh Stemming untuk URL 1, Kalimat Sebelum Stemming: "KPU Minahasa sosialisasikan tahapan Pilkada 2024."; "Tahapan Pilkada ini penting untuk keberhasilan demokrasi."

Kalimat Setelah Stemming: "KPU Minahasa sosialisasi tahap Pilkada 2024."; "Tahap Pilkada penting berhasil demokrasi."

Dalam contoh ini: "sosialisasikan" dikurangi menjadi "sosialisasi"; "tahapan" dikurangi menjadi "tahap"; "keberhasilan" dikurangi menjadi "berhasil"

Proses stemming membantu menyatukan variasi morfologis dari kata-kata sehingga kata-kata seperti "sosialisasi", "sosialisasikan", dan "sosialisasi" diperlakukan sebagai satu kata dasar yang sama, yaitu "sosialisasi". Hal ini mempermudah analisis teks karena mengurangi redundansi dan meningkatkan fokus pada makna dasar dari teks.

Pembobotan TF-IDF: Menghitung bobot Term Frequency-Inverse Document Frequency (TF-IDF) untuk setiap kalimat.

Rumus 6 adalah rumus TF (Term Frequency).

$$TF(t, d) = \frac{f(t, d)}{\sum_{t' \in d} f(t', d)} \quad (6)$$

$f(t, d)$ adalah jumlah kemunculan term t dalam dokumen d .

Rumus 7 adalah rumus IDF (Inverse Document Frequency).

$$IDF(t) = \log \frac{N}{|\{d \in D; t \in d\}|} \quad (7)$$

N adalah total dokumen, $|\{d \in D; t \in d\}|$ adalah jumlah dokumen yang mengandung term t .

Rumus 8 adalah rumus TF-IDF (Term Frequency-Inverse Document Frequency)

$$TF - IDF(t, d) = TF(t, d) \times IDF(t) \quad (8)$$

Contoh Pembobotan TF-IDF, Misalnya ada 2 kalimat: "KPU Minahasa sosialisasi tahap Pilkada 2024."; "Tahap Pilkada penting berhasil demokrasi."

Tabel 4 Pembobotan TF-IDF

Term	TF (Kalimat 1)	TF (Kalimat 2)	IDF	TF-IDF (Kalimat 1)	TF-IDF (Kalimat 2)
KPU	1/6	0	0.301	0.050	0
Minahasa	1/6	0	0.301	0.050	0
sosialisasi	1/6	0	0.301	0.050	0
tahap	1/6	1/5	0.0	0.0	0.0
Pilkada	1/6	1/5	0.0	0.0	0.0
2024	1/6	0	0.301	0.050	0
penting	0	1/5	0.301	0	0.060
berhasil	0	1/5	0.301	0	0.060
demokrasi	0	1/5	0.301	0	0.060

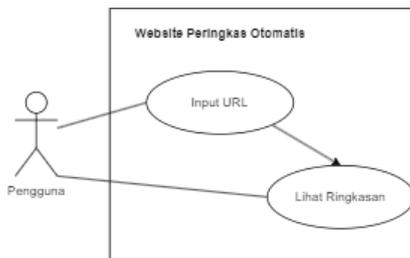
Klasterisasi Kalimat: Algoritma K-means digunakan untuk mengelompokkan kalimat-kalimat berdasarkan kesamaan konten. Kalimat terpanjang dari setiap kluster dipilih sebagai bagian dari ringkasan.

Pada tahap Desain Pengguna, diadakan workshop Rapid Application Development (RAD) untuk memperoleh masukan langsung dari pengguna akhir dan memastikan bahwa kebutuhan serta desain sistem sesuai dengan harapan mereka. Workshop RAD melibatkan sesi

kolaboratif di mana pengembang, pengguna akhir, dan pemangku kepentingan bekerja sama dalam iterasi desain dan pengembangan untuk mempercepat proses pembuatan prototipe dan validasi.

Pemodelan sistem ini menggunakan UML (Unified Modeling Language). UML yang digunakan yaitu Use Case Diagrams, Activity Diagrams, dan Sequence Diagrams. Berikut adalah penjelasan mengenai tahapan-tahapan algoritma K-means Clustering yang diimplementasikan dalam sistem ini:

Use Case Diagram menggambarkan interaksi antara aktor (pengguna) dan sistem dalam mencapai tujuan tertentu. Dalam konteks sistem peringkas otomatis, terdapat beberapa use case utama seperti menginput URL, dan melihat ringkasan. Aktor utama dalam diagram ini adalah pengguna, yang berinteraksi dengan sistem untuk memasukkan URL berita, dan melihat ringkasan yang dihasilkan oleh sistem. Diagram ini menggambarkan bagaimana setiap interaksi yang terjadi serta bagaimana sistem merespons dari setiap permintaan pengguna.



Gambar 8 Use Case Diagram Peringkas Otomatis

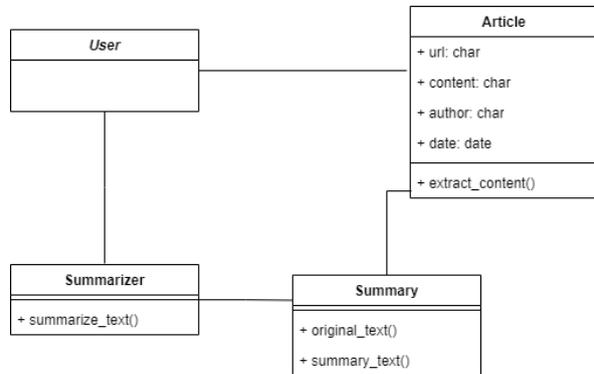
Pengguna: Aktor yang berinteraksi dengan sistem.;
Input URL: Pengguna memasukkan URL untuk meringkas teks berita.

Class Diagram menunjukkan struktur kelas dalam sistem, termasuk atribut dan metode dari setiap kelas, serta hubungan antar kelas. Dalam sistem peringkas otomatis ini, terdapat beberapa kelas utama seperti User, Article, Summarizer, dan Summary. Kelas User merepresentasikan pengguna dan memiliki atribut seperti username dan email, serta metode login dan logout. Kelas Artikel masing-masing merepresentasikan berita, dengan metode untuk mengekstrak konten. Kelas Summarizer mengelola proses peringkas, menggunakan metode untuk meringkas teks berita. Kelas Summary menyimpan hasil ringkasan, dengan atribut yang mencakup teks asli dan teks ringkasan. Diagram ini membantu dalam memahami struktur internal sistem dan bagaimana setiap kelas berinteraksi satu sama lain.

Pada tahap perancangan sistem peringkas otomatis yang diimplementasikan menggunakan algoritma K-means Clustering, Class Diagram menunjukkan struktur internal dari sistem dan hubungan antar kelas. Sistem ini tidak menggunakan fitur login dan logout karena dirancang untuk akses langsung tanpa autentikasi

pengguna. Berikut adalah penjelasan setiap kelas dalam sistem ini:

User: Meskipun kelas ini disebut User, dalam konteks sistem peringkas otomatis ini, kelas User tidak memiliki atribut atau metode untuk login dan logout. Sebaliknya, kelas ini hanya merepresentasikan pengguna umum yang mengakses sistem untuk menggunakan fitur peringkas otomatis.



Gambar 9 Class Diagram Peringkas Otomatis

Article: Kelas ini merepresentasikan berita yang diinput melalui URL. Atribut dari kelas ini termasuk URL, konten teks, penulis, dan tanggal publikasi. Metode dalam kelas ini meliputi `extract_content()` yang digunakan untuk mengambil dan memproses konten dari URL yang diberikan.

Summarizer: Kelas ini mengelola proses peringkas teks. Metode dalam kelas ini meliputi `summarize_text()` yang menggunakan algoritma K-means untuk meringkas teks yang diekstraksi dari berita.

Summary: Kelas ini menyimpan hasil ringkasan. Atribut dari kelas ini termasuk teks asli dan teks ringkasan. Tidak ada metode tambahan dalam kelas ini selain getter dan setter untuk mengakses dan mengubah atribut.

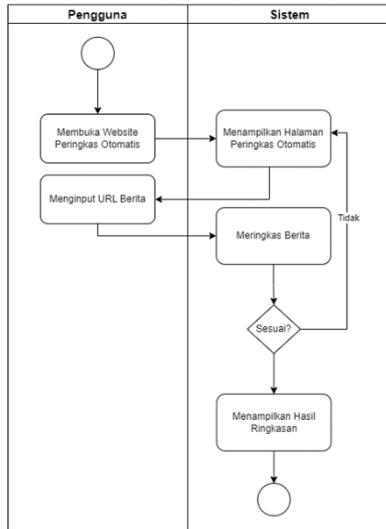
Activity Diagram: Activity Diagram menggambarkan aliran aktivitas dalam sistem, mulai dari awal hingga akhir proses. Proses dimulai saat pengguna memasukkan URL. Sistem kemudian mengekstrak konten dari URL yang diberikan. Setelah konten diekstraksi, sistem menggunakan algoritma K-means untuk meringkas teks. Hasil ringkasan kemudian ditampilkan kepada pengguna. Diagram ini membantu dalam memahami aliran kerja sistem dan bagaimana setiap langkah saling terkait untuk mencapai hasil akhir. Gambar 10 adalah gambar activity diagram.

Diagram aktivitas menggambarkan alur kerja sistem peringkas otomatis, yang diakses oleh pengguna untuk meringkas berita dari URL yang diberikan. Diagram ini terdiri dari dua swimlane yang menunjukkan tindakan dari pengguna dan sistem. Berikut adalah penjelasan rinci mengenai setiap aktivitas dalam diagram ini:

Pengguna Membuka Website Peringkas Otomatis.
Aktivitas: Pengguna memulai dengan membuka website yang menyediakan layanan peringkas otomatis.;

Penjelasan: Ini adalah langkah pertama di mana pengguna mengakses antarmuka web aplikasi peringkas otomatis. Aktivitas ini ditandai dengan simbol lingkaran di awal yang menunjukkan titik awal dari alur kerja.

Peringkas Otomatis



Gambar 10 Activity Diagram Peringkas Otomatis

Sistem Menampilkan Halaman Peringkas Otomatis. Aktivitas: Setelah pengguna membuka website, sistem menampilkan halaman utama dari peringkas otomatis. Penjelasan: Sistem merespon dengan menyediakan antarmuka di mana pengguna dapat memasukkan URL berita yang ingin diringkas.

Pengguna Menginput URL Berita. Aktivitas: Pengguna memasukkan URL berita yang ingin diringkas ke dalam form yang tersedia di halaman website.; Penjelasan: Ini adalah tindakan yang dilakukan oleh pengguna untuk menyediakan data yang akan diproses oleh sistem.

Sistem Meringkas Berita. Aktivitas: Setelah menerima URL dari pengguna, sistem memulai proses peringkas berita. Penjelasan: Sistem mengekstraksi konten teks dari URL yang diberikan, kemudian menggunakan algoritma K-means Clustering untuk meringkas teks menjadi beberapa kalimat inti.

Sistem Memeriksa Kesesuaian Ringkasan. Aktivitas: Sistem melakukan pengecekan apakah hasil ringkasan sesuai dengan kriteria tertentu atau tidak.; Penjelasan: Langkah ini melibatkan verifikasi internal untuk memastikan bahwa ringkasan yang dihasilkan memenuhi standar yang telah ditentukan.

Keputusan: Sesuai? Aktivitas: Sistem menentukan apakah ringkasan yang dihasilkan sesuai atau tidak.; Penjelasan: Ini adalah titik keputusan dalam alur kerja. Jika ringkasan sesuai, alur berlanjut ke langkah berikutnya. Jika tidak, sistem mungkin mengulang proses atau memberikan notifikasi kesalahan kepada pengguna.

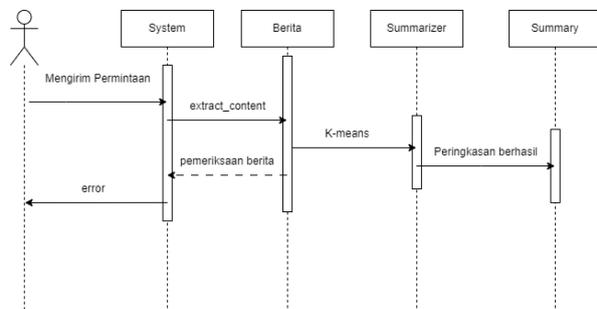
Sistem Menampilkan Hasil Ringkasan. Aktivitas: Jika ringkasan sesuai, sistem menampilkan hasil ringkasan kepada pengguna. Penjelasan: Hasil ringkasan yang sudah diverifikasi ditampilkan di antarmuka pengguna, memungkinkan pengguna untuk melihat ringkasan teks dari berita yang dimasukkan.

Diagram ini memberikan gambaran jelas tentang bagaimana interaksi antara pengguna dan sistem berlangsung dalam proses peringkas otomatis. Setiap aktivitas menunjukkan langkah-langkah spesifik yang diambil dari saat pengguna mengakses website hingga melihat hasil ringkasan berita.

Sequence Diagram menggambarkan interaksi antar objek dalam sistem dalam urutan waktu tertentu. Dalam diagram ini, objek utama termasuk User, System, Berita, Summarizer, dan Summary. Proses dimulai ketika pengguna mengirim permintaan untuk meringkas berita. Sistem menerima permintaan ini dan memanggil metode `extract_content` untuk mengekstrak teks dari berita. Setelah konten diekstraksi, Summarizer melakukan proses peringkas menggunakan algoritma K-means. Hasil ringkasan kemudian disimpan dalam objek Summary dan dikembalikan kepada pengguna. Diagram ini memberikan gambaran yang jelas tentang urutan interaksi antara objek-objek dalam sistem selama proses peringkas [16].

Penjelasan: Sequence Diagram menggambarkan interaksi antar objek dalam sistem dalam urutan waktu tertentu.

Sequence Diagram:



Gambar 11 Sequence Diagram Peringkas Otomatis

User: Mengirim permintaan untuk meringkas berita;
System: Menerima permintaan dan memanggil metode `extract_content`.
Berita: Mengekstraksi konten teks.
Summarizer: Melakukan proses peringkas menggunakan K-means.;
Summary: Menyimpan hasil ringkasan dan mengembalikannya kepada pengguna.

Rencana pengujian sistem (Blackbox) pada Tabel 5 berfokus pada pengujian fungsionalitas sistem tanpa memeriksa kode internal. Pengujian dilakukan berdasarkan input dan output yang diharapkan.

Hasil pengujian sistem pada Tabel 6 berfokus pada pengujian fungsionalitas utama seperti memasukkan URL dan mendapatkan ringkasan tanpa memerlukan pengujian untuk fitur autentikasi

Tabel 5 Rencana Pengujian Black box

No	Fitur yang Diuji	Input	Expected Output	Jenis Pengujian
1	Input URL	URL berita	Ringkasan teks berita ditampilkan	Black box
2	Tampilkan Ringkasan	URL berita	Ringkasan isi teks sesuai dengan input URL	Black box
3	Validasi URL	URL yang tidak valid	Pesan error "URL tidak valid"	Black box
4	Hasil Ringkasan	Isi dari URL berita yang di input	Ringkasan berbeda setiap menekan tombol ringkas walaupun URL yang sama	Black Box

Tabel 6 Hasil Pengujian

No	Fitur yang Diuji	Input	Output yang Diharapkan	Status Pengujian
1	Memasukkan URL Berita	URL valid berita	Meringkas isi dari URL berita	Sesuai
2	Menampilkan Ringkasan	URL valid berita	Menampilkan hasil ringkasan dari URL berita	Sesuai
3	Memasukkan URL Tidak Valid	URL tidak valid	Menampilkan pesan error	Sesuai
4	Hasil Ringkasan	URL valid berita	Menampilkan ringkasan berbeda setiap menekan tombol ringkas walaupun URL yang sama	Sesuai

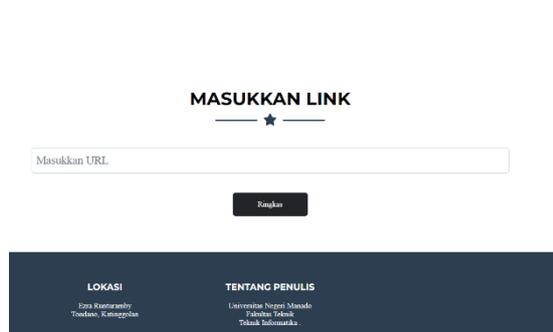
Tampilan Halaman Dashboard: Tampilan halaman dashboard menunjukkan tampilan halaman penginputan url berita yang ingin diringkas. Seperti pada Gambar 12.

Tampilan Halaman Summary (Hasil Ringkasan): Tampilan halaman summary ini berupa tampilan saat url berita yang dimasukkan ke dalam sistem itu valid dan dapat diringkas seperti pada Gambar 13.

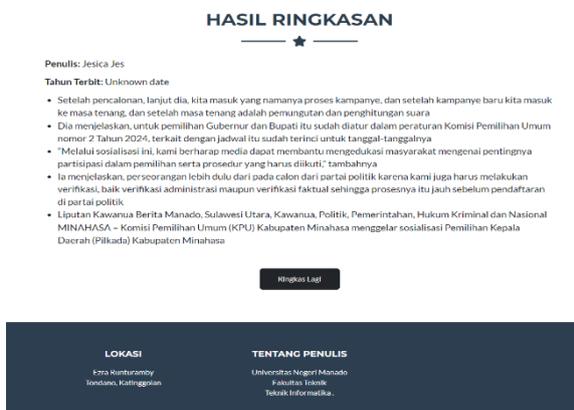
Cutover (Beralih): Tahap ini mencakup pengujian sistem secara menyeluruh, migrasi data jika diperlukan, dan peluncuran sistem. Pengujian dilakukan untuk memastikan bahwa aplikasi berfungsi sesuai dengan kebutuhan pengguna dan tidak terdapat bug yang signifikan.

Post-Implementation (Pasca-implementasi): Pada tahap ini, dilakukan pemeliharaan sistem, termasuk pembaruan fitur, perbaikan bug, dan penyesuaian berdasarkan umpan balik pengguna. Selain itu, dilakukan evaluasi terhadap kinerja sistem dan penggunaannya dalam memenuhi kebutuhan pengguna.

PERINGKAS OTOMATIS LINK BERITA



Gambar 12 Halaman Dashboard



Gambar 13 Tampilan Halaman Summary

4. Kesimpulan

Berdasarkan hasil penelitian dan pembahasan tentang Peringkat Teks Otomatis Berita Online Komisi Pemilihan Umum Menggunakan Algoritma K-Means Clustering, Peringkat teks otomatis ini dapat membantu para pembaca untuk mendapatkan intisari atau inti dari berita KPU secara cepat dan efisien tanpa harus membaca keseluruhan isi berita yang panjang. Algoritma K-Means Clustering digunakan untuk mengelompokkan kalimat-kalimat dalam berita berdasarkan kemiripan semantik, sehingga kalimat-kalimat yang paling representatif dari setiap kelompok dapat dipilih sebagai ringkasan berita.

Daftar Rujukan

- [1] Andre, D. (2023). Perkembangan Teknologi Informasi: Dampaknya Sampai Saat Ini. ToffeeDev. Retrieved from <https://toffeedev.com/perkembangan-teknologi-informasi/>
- [2] Madcoms. (2019). Karakteristik Masyarakat Era Digital dan Media Sosial. Jurnal Komunikasi Pembangunan, 17(2), 180-183..
- [3] Civita C.I.L, Vivi P. R, Gladly C. R., 2022. Data Mining Rekomendasi Sekolah Calon Siswa SMA di Kota Tomohon Menggunakan Metode K-Means Clustering

- [4] Komisi Pemilihan Umum Republik Indonesia. (2024). Electoral Governance: Jurnal Tata Kelola Pemilu Indonesia, Vol. 5 No. 2.
- [5] Komisi Pemilihan Umum Republik Indonesia. (2024). Riset Kepemiluan Indonesia.
- [6] Santoso, B. (2023). Pengembangan sistem peringkas teks otomatis menggunakan algoritma K-Means Clustering. Jurnal Teknologi Informasi, 15(2), 123-130.
- [7] Celebi, M. E., Kingravi, H. A., & Vela, P. A. (2013). A comparative study of efficient initialization methods for the k-means clustering algorithm. Expert systems with applications, 40(1), 200-210.
- [8] Rina N. M., Quido C. K., Ferdinan I. S., 2023. Sistem Pengendali Perangkat Elektronik Melalui Voice Assistant Dengan Metode Rapid Application Development (RAD)
- [9] Sutrisno, A. (2022). Rapid application development: A methodology for software development. Journal of Software Engineering, 10(1), 45-56.
- [10] Jain, A. K. (2020). A survey of K-means algorithm variants. International Journal of Computer Applications, 975(5), 6-10.
- [11] Xu, R., & Wunsch, D. (2018). K-means clustering: A comprehensive review. IEEE Transactions on Neural Networks and Learning Systems, 29(3), 786-797.
- [12] Sharmila, P., & Suresh, S. (2021). An enhanced K-means clustering algorithm for data mining. Journal of Data Science, 19(1), 45-60.
- [13] Gupta, M., & Singh, R. (2022). Performance evaluation of K-means clustering algorithm in big data. Journal of Big Data, 9(1), 1-15.
- [14] Chen, Y., & Kumar, R. (2021). The impact of RAD on software development time and quality. Software Engineering Journal, 12(4), 78-85.
- [15] Rahman, M., & Lee, S. (2023). Evaluating rapid application development in software projects. International Journal of Information Technology, 14(3), 201-210.
- [16] Ghadban, D., Awad, A., AlHajj, M., 2022. A Survey on Automatic Text Summarization