

Principal Component Analysis untuk Reduksi Dimensi pada Clustering Ulasan Buku di Goodreads

Iffatu Wardani¹, Ailsa Vania Prabawati², Ane Dwi Untari³, Lutfiana Azizah⁴, Adelina Fitri Maharani⁵

^{1,2,3,4,5}Ilmu Komputer, Fakultas Sains dan Teknologi, Institut Teknologi dan Bisnis Trenggalek
¹iffatu.wardani@gmail.com, ²ailsavpailsa@gmail.com, ³anndwiy31@gmail.com, ⁴ltfna.07@gmail.com,
⁵adelinaftrmaharani@gmail.com

Abstrak

Penelitian ini bertujuan menganalisis dan memetakan pola tanggapan pembaca Indonesia terhadap buku *The Da Vinci Code* melalui pengelompokan (*clustering*) ulasan berbahasa Indonesia di *Goodreads*. *Dataset* dikumpulkan menggunakan alat bantu *Instant Data Scraper*. Data teks yang diperoleh kemudian diproses melewati beberapa tahapan *preprocessing* meliputi *case folding*, penghapusan *stopword*, penghapusan karakter khusus, serta *stemming* dengan pustaka *Sastrawi* untuk mengurangi *noise* dan menyatukan variasi kata. Selanjutnya, representasi numerik dibentuk melalui metode *word embedding FastText* model *skip-gram* dengan dimensi vektor 100. Metode ini dinilai efektif untuk *word embedding* teks berbahasa Indonesia karena mampu menangkap informasi sub-kata dan dapat menangani variasi imbuhan. Eksperimen pada langkah *clustering* dilakukan dengan dua skema. Pada skema pertama *clustering* langsung menggunakan algoritma *K-Means* pada vektor *FastText*. Sedangkan pada skema kedua, terlebih dahulu dilakukan reduksi dimensi menggunakan *Principal Component Analysis* (PCA) sebelum *K-Means*, dengan penurunan dimensi dari 100 menjadi 50 fitur. Penentuan jumlah *cluster* optimal dilakukan melalui evaluasi *silhouette score* pada nilai k sama dengan 2, 3, 4, dan 5. Hasil menunjukkan nilai terbaik pada $k = 2$ untuk kedua skema, dengan peningkatan kualitas yang tipis setelah penerapan PCA (*silhouette* 0,5899 tanpa PCA dan 0,5901 dengan PCA). Temuan ini mengindikasikan bahwa PCA dapat memperbaiki struktur *cluster*, namun diperlukan pengujian lanjutan untuk memastikan signifikansi perbedaannya.

Kata kunci: PCA, *K-Means*, *data mining*, NLP

1. Pendahuluan

Buku fiksi *The Da Vinci Code* karya penulis Dan Brown merupakan suatu fenomena budaya global (60 Minutes Australia, 2019). Meskipun buku ini sudah terbit tahun 2003, namun buku ini masih dibicarakan dan naskah serta terjemahannya masih terus dicetak ulang hingga tahun 2026, termasuk oleh penerbit di Indonesia (Gramedia, 2026). Hal ini disebabkan masih menariknya karya ini untuk dijadikan bahan perdebatan mengingat penulis mengklaim semua dokumen dan latar belakang sejarah dalam bukunya adalah fakta.

Jika dilihat ulasannya di google playbook, *The Da Vinci Code* mendapatkan *rating* 4,1 dari 7,06 ribu ulasan. Sedangkan di *website amazon* buku ini memperoleh *rating* cukup tinggi yaitu 4,6 dari 25.145 ulasan. Sementara itu di forum *goodreads* buku ini mendapat *rating* 3,9 dari 61.271 ulasan.

Analisis terhadap ulasan *The Da Vinci Code* ini dapat membuka jalan untuk memahami bagaimana suatu budaya populer global dikonsumsi, diterima, atau ditolak. Selain itu analisis juga dapat membantu memetakan bagaimana narasi fiksi dapat membentuk opini publik terhadap permasalahan yang di-

angkat (Fazri dan Voutama, 2025). Oleh karena itu penelitian ini akan menganalisis, khususnya pengelompokan (*clustering*) ulasan buku *The Da Vinci Code* yang ada di *goodreads*. Dari berbagai macam bahasa yang mengulas, dipilih ulasan yang berbahasa Indonesia untuk melihat tanggapan pembaca dari Indonesia serta untuk memvisualkan polarisasi pendapatnya.

Untuk dapat dianalisis menggunakan algoritma *machine learning*, ulasan yang masih berupa data teks dengan tipe *string* atau *character* perlu diubah menjadi data numerik. Metode untuk mengubah data teks menjadi data numerik ada beberapa macam, salah satunya adalah *fasttext* (Nurdin et al, 2020). Hasil dari penelitian terdahulu menunjukkan *fast-text* memiliki performa lebih baik dan juga mampu memahami sub kata dan efektif untuk bisa menangani kosakata langka ataupun data dalam jumlah besar (Harris et al, 2026). Sementara itu *K-Means* sebagai salah satu metode *clustering* dapat digunakan untuk analisis ulasan buku pada penelitian ini. *K-Means* yang merupakan metode *clustering* non-hierarki digunakan untuk mengelompokkan data berdasarkan kemiripan karakteristik. Data yang memiliki karak-

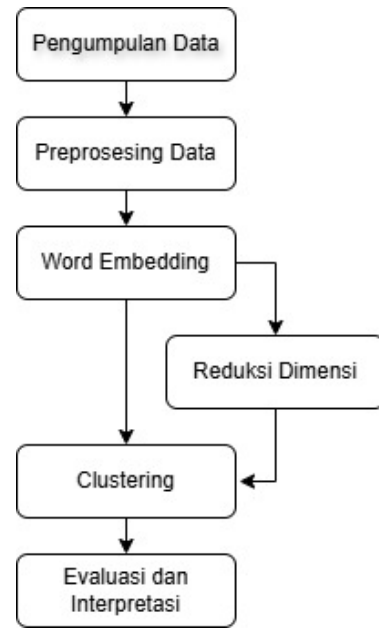
teristik serupa akan ditempatkan pada *cluster* yang sama, sedangkan data yang berbeda akan ditempatkan pada *cluster* yang berbeda (Haryono et al, 2025; Wicaksono et al, 2025; Sari et al, 2023). Jika dibandingkan dengan algoritma *clustering* yang lain, *K-Means* memiliki kelebihan yaitu lebih efisien menangani *dataset* berjumlah besar (Yolandari, 2025).

Kinerja algoritma *K-Means* dapat dibantu dengan menggunakan metode reduksi dimensi. *Principal Component Analysis* (PCA) sebagai salah satu algoritma untuk reduksi dimensi, digunakan untuk menyederhanakan data dengan mengidentifikasi komponen utama yang memiliki variansi terbesar. Dengan demikian, data dapat direpresentasikan dalam dimensi yang lebih rendah sambil mempertahankan sebagian besar informasi penting yang terkandung di dalamnya (Rianti et al, 2024). Evaluasi analisis pada penelitian ini, digunakan *silhouette score* untuk melihat jumlah *cluster* yang optimal. Penggunaan metode ini mampu menghasilkan struktur *cluster* yang lebih stabil, bisa meningkatkan kualitas proses *clustering* data, dan juga memberikan hasil analisis yang akurat (Saputra dan Purnama, 2024; Hasan 2024).

Penelitian terdahulu yang pernah dilakukan dalam topik ini sudah banyak. Diantaranya adalah penerapan algoritma *K-Means* untuk *clustering* ulasan pengguna aplikasi *cookpad* (Khairuna et al, 2025), situs *tripadvisor* (Nurjanah dan Arifin, 2021), media sosial UMKM (Saputro et al, 2025), serta *e-commerce* Lazada (Pamungkas dan Feberuariyanti, 2022). Selain itu algoritma ini juga pernah digunakan untuk *clustering* ulasan Amanzi *Waterpark* Palembang (Daely et al, 2026). Sementara itu penggunaan PCA untuk reduksi dimensi juga pernah dilakukan dalam proses *clustering* maupun klasifikasi data (Ritonga dan Muhadhis, 2021; Putra et al, 2025; Hediyaati dan Suartana, 2021).

2. Metode

Tahapan-tahapan yang dilaksanakan dalam penelitian ini diilustrasikan dalam Gambar 1. Setiap tahapan dilakukan secara berurutan mulai dari pengumpulan data dan dilanjutkan *preprocessing* data. Setelah data dibersihkan pada tahap *preprocessing* selanjutnya dilakukan *word embedding*. Setelah tahap ini data diproses dengan dua langkah berbeda. Langkah pertama adalah data dikelompokkan menggunakan algoritma *K-Means clustering* untuk selanjutnya dievaluasi. Sedangkan langkah kedua, data akan diproses terlebih dahulu menggunakan *principal component analysis* untuk selanjutnya dikelompokkan menggunakan algoritma *clustering*. Dan pada langkah terakhir, dibandingkan dan diinterpretasi hasil dari kedua langkah tersebut.



Gambar 1. Alur Penelitian

	data_mentah
0	buku ini bagus sekali tata bahasanya dan sulit...
1	THE DA VINCI CODE karya Dan Brown 588 halaman...
2	gue suka bgt njir
3	- Elemen misteri dan teka-tekinya luar biasa! ...
4	tidak pernah kecewa membaca buku dari film yan...
5	300 halaman yang lambat banget, setelah masih ...
6	pusing dikit tapi ok lah.
7	"Kita takut terhadap sesuatu yang tidak kita p...

Gambar 2. Data Hasil *Scraping*

2.1 Pengumpulan Data

Data yang dikumpulkan adalah ulasan buku *The Da Vinci Code* di *Goodreads*. Ulasan tersebut kemudian difilter yang berbahasa Indonesia dan data yang diambil adalah hasil filter tersebut. Dari hasil filter diperoleh 496 data. Untuk *tools* yang digunakan pada proses pengumpulan data adalah *Instant Data Scraper*.

Setelah data mentah terkumpul kemudian disimpan di google drive untuk di-mount ke google colab. Bahasa pemrograman yang digunakan dalam analisis ini adalah Python. Data yang sudah termuat kemudian mulai dianalisis dengan menjalankan langkah pertama yaitu *preprocessing*. Tampilan untuk 8 baris pertama data mentah yang diperoleh tersaji dalam Gambar 2.

2.2 Preprocessing Data

Data mentah yang dibersihkan dalam tahap *preprocessing* melewati 4 langkah yaitu *case folding*, menghilangkan *stopword*, menghapus karakter khusus, dan *stemming*. Pada tahap *case folding*, data mentah diseragamkan dengan mengganti huruf kapi-

hasil_case_folding	
0	buku ini bagus sekali tata bahasanya dan sulit...
1	the da vinci codekarya dan brown 588 halaman...
2	gue suka bgt njir
3	- elemen misteri dan teka-tekinya luar biasa! ...
4	tidak pernah kecewa membaca buku dari film yan...
5	300 halaman yang lambat banget, setelah masih ...
6	pusing dikit tapi ok lah.
7	"kita takut terhadap sesuatu yang tidak kita p...

Gambar 3. Data Hasil Case Folding

remove_stopword	
0	buku bagus tata bahasanya sulit dipahami
1	the da vinci codekarya brown 588 halaman ba...
2	gue suka bgt njir
3	- elemen misteri teka-tekinya biasa! disuguhka...
4	kecewa membaca buku film tonton.... buang kepe...
5	300 halaman lambat banget, halaman 400-selesai...
6	pusing dikit ok lah.
7	"kita takut pahami." - 55"simbol membawa arti ...

Gambar 4. Data Hasil Menghapus Stopword

tal menjadi huruf kecil semua. Pada tahap menghilangkan *stopword*, data yang memiliki kata sambung dihapus karena kata-kata tersebut dianggap sebagai *noise* dan tidak memiliki arti yang signifikan terhadap analisis (Prasetyo et al, 2023). Contoh yang termasuk kata sambung adalah, 'di', 'ke', 'dan', dll. Langkah *preprocessing* selanjutnya adalah menghapus karakter khusus seperti tanda baca, *emoticon*, dan angka. Dan pada langkah terakhir, dilakukan *stemming* untuk mengembalikan kata menjadi bentuk dasarnya (Sinaga dan Nainggolan, 2023). Pada tahap ini imbuhan pada suatu kata dihapus untuk memperoleh kata dasarnya.

a. Case Folding

Tahap ini dilakukan agar kata-kata yang ditulis menggunakan penulisan berbeda tetap bisa selaras dan dianggap sebagai satu kata yang sama. Pada penelitian ini, contoh katanya adalah 'DA VINCI CODE' yang diubah menjadi 'da vinci code', serta 'Sauniere' menjadi 'sauniere'. Hasil *preprocessing* tahap *case folding* tersaji dalam Gambar 3.

b. Menghapus Stopword

Langkah menghapus *stopword* dilakukan untuk menghilangkan kata-kata yang tidak memiliki arti. Dalam ulasan berbahasa Indonesia dari buku Da Vinci Code ini, yang termasuk *stopword* dan dihapus dalam *preprocessing* adalah 'tapi', 'yang', 'dan', dll. Data yang sudah dihilangkan *stopword*-nya dapat dilihat dalam Gambar 4.

hasil_normalisasi	
0	buku bagus tata bahasanya sulit dipahami
1	the da vinci codekarya brown halaman bahasa in...
2	gue suka bgt njir
3	elemen misteri tekatekinya biasa disuguhkan pr...
4	kecewa membaca buku film tonton buang kepepin...
5	halaman lambat banget halaman selesai serunya ...
6	pusing dikit ok lah
7	kita takut pahami simbol membawa arti berbeda ...

Gambar 5. Data Hasil Menghapus Karakter Khusus

hasil_stemming	
buku bagus tata bahasa sulit paham	
the da vinci codekarya brown halaman bahasa in...	
gue suka bgt njir	
elemen misteri tekatekinya biasa sugu premis ...	
kecewa baca buku film tonton buang penting dur...	
halaman lambat banget halaman selesai seru hal...	
pusing dikit ok lah	
kita takut paham simbol bawa arti beda latar b...	

Gambar 6. Data Hasil Stemming

c. Menghapus Karakter Khusus

Menghapus karakter khusus dalam *preprocessing* ditujukan untuk menghilangkan tanda baca, emoticon, dan angka. Dengan demikian data yang bersih akan terdiri dari kata-kata bermakna saja. Pada penelitian ini, karakter khusus yang dihapus adalah tanda seru '!', tanda pisah '-', angka '588', '300', '400', dll. Hasil dari penghapusan karakter khusus dapat dilihat dalam Gambar 5.

d. Stemming

Pada langkah *stemming*, kata-kata yang mempunyai awalan, akhiran, dan imbuhan diubah menjadi bentuk kata dasarnya atau akar katanya. Tujuan dari langkah ini adalah agar komputer mengenali kata-kata yang berasal dari akar sama sebagai kata yang serupa. Pada penelitian ini, beberapa kata yang di-*stemming* contohnya adalah 'bahasanya' menjadi 'bahasa', 'membawa' berubah menjadi 'bawa', dan 'ternama' menjadi 'nama'. Proses *stemming* dilakukan dengan memanfaatkan *library* Sastrawi Python. Untuk gambaran hasil proses *stemming*, dapat dilihat dalam Gambar 6.

2.3 Word Embedding

Tahapan selanjutnya setelah *preprocessing* adalah *word embedding*. Pada tahap ini data yang berupa kata dalam ulasan diubah menjadi bentuk angka/representasi vektor. Metode *word embedding*

First 5 rows of FastText features (sample):

	0	1	2	3	4	5	6	7	8	9	...	90	91	92	93	94	95	96	97	98	99
0	-0.048431	0.324325	0.145980	0.379332	-0.137111	0.159306	0.338038	0.294890	-0.267174	0.242829	...	-0.086316	-0.485126	0.078417	0.169701	0.144136	-0.269392	-0.173942	0.079040	-0.403683	-0.095076
1	-0.049256	0.329273	0.147795	0.385263	-0.140821	0.161537	0.343551	0.297862	-0.273271	0.247198	...	-0.088773	-0.493184	0.078999	0.172083	0.145703	-0.272936	-0.176189	0.079389	-0.409449	-0.096470
2	-0.043250	0.282678	0.126402	0.329892	-0.119254	0.138092	0.294030	0.256689	-0.234162	0.210928	...	-0.076107	-0.424057	0.067749	0.147334	0.125096	-0.234408	-0.151872	0.068471	-0.351163	-0.083074
3	-0.046637	0.318951	0.139812	0.363183	-0.132128	0.152591	0.323923	0.281174	-0.257122	0.232859	...	-0.083348	-0.465311	0.074416	0.162198	0.137327	-0.257801	-0.166503	0.075338	-0.386037	-0.091210
4	-0.048793	0.325246	0.146396	0.380313	-0.138344	0.159238	0.339276	0.294550	-0.268627	0.243560	...	-0.087023	-0.487475	0.078012	0.169672	0.143709	-0.269964	-0.174177	0.078978	-0.404018	-0.095631

[5 rows x 100 columns]

Gambar 7. Vektor Hasil Word Embedding

yang digunakan dalam penelitian ini adalah *Fasttext*. *Fasttext* mengubah kata menjadi angka dengan cara memecah kata menjadi potongan karakter lebih kecil yang disebut sebagai n-gram, kemudian mengubahnya menjadi angka-angka dalam bilangan riil yang disebut vektor (Sepriadi et al, 2023). Penelitian ini menggunakan *fasttext* model *fasttext skipgram*. Model ini memberikan representasi vektor yang lebih akurat jika dibandingkan dengan TF-IDF (Afuan dan Hidayat, 2024). *Fasttext skipgram* ini sangat efektif digunakan untuk teks bahasa Indonesia yang memiliki berbagai macam imbuhan (Amalia et al, 2021). Kata-kata berakar sama yang mendapat berbagai macam imbuhan akan tetap bisa dipahami maksudnya dengan model ini.

Lima kata teratas sebagai perwakilan vektor hasil *word embedding* dapat dilihat dalam Gambar 7. Pada saat menjalankan *fasttext*, dimensi diatur memiliki nilai 100 (dalam Gambar 7 ditunjukkan dengan kolom 0 sampai kolom 99). Hal ini berarti bahwa setiap kata diatur memiliki 100 angka unik yang menangkap makna semantik, konteks, dan hubungan antar kata. Vektor inilah yang digunakan pada proses analisis selanjutnya.

Nilai-nilai pada setiap vektor bervariasi dari -1 sampai 1. Angka -1 atau yang mendekatinya menunjukkan bahwa kata tersebut memiliki arah yang berlawanan di dalam ruang vektor. Angka 0 menunjukkan bahwa kata tersebut netral dan tidak memiliki hubungan sama sekali. Demikian juga dengan nilai vektor yang mendekati 0, memiliki arti bahwa semakin kecil pula hubungan kata tersebut. Sedangkan angka 1 atau yang mendekatinya menunjukkan bahwa kata tersebut sangat mirip atau identik.

Perhitungan kemiripan kata tersebut dihitung menggunakan *cosine similarity* yaitu mengukur sudut antara dua vektor dalam ruang multi dimensi. Pencarian *cosine similarity* dapat dijabarkan pada persamaan (1).

$$\text{Similarity}(A, B) = \cos \theta = \frac{A \cdot B}{\|A\| \|B\|} \quad (1)$$

Jika persamaan (1) dijabarkan maka diperoleh komponen $A \cdot B$ yang merupakan dot product dari vektor A dan vektor B . Dan komponen lain yang berupa perkalian magnitude/panjang/norma dari kedua vek-

tor. Perolehan komponen-komponen tersebut dijelaskan dalam persamaan (2), (3), dan (4).

$$A \cdot B = \sum_{i=1}^n A_i B_i = (A_1 \times B_1) \times \dots \times (A_n \times B_n) \quad (2)$$

$$\|A\| = \sqrt{\sum_{i=1}^n A_i^2} = \sqrt{A_1^2 + A_2^2 + \dots + A_n^2} \quad (3)$$

$$\|B\| = \sqrt{\sum_{i=1}^n B_i^2} = \sqrt{B_1^2 + B_2^2 + \dots + B_n^2} \quad (4)$$

3. Hasil dan Pembahasan

Untuk langkah selanjutnya, vektor pada Gambar 7 dianalisis menggunakan 2 cara. Cara yang pertama yaitu dianalisis menggunakan *K-Means clustering*, dan cara kedua direduksi terlebih dahulu dimensinya menggunakan PCA kemudian dianalisis menggunakan *K-Means clustering*.

3.1 K-Means Clustering

Proses *clustering* menggunakan algoritma *K-Means* dilakukan dengan menentukan k terlebih dahulu. K adalah jumlah *cluster* dalam algoritma. Kemudian dilakukan langkah berulang untuk menentukan suatu data ikut *cluster* mana. Iterasi berhenti ketika diperoleh nilai SSE (*sum of square error*) yang minimum. SSE dapat dicari menggunakan persamaan (5).

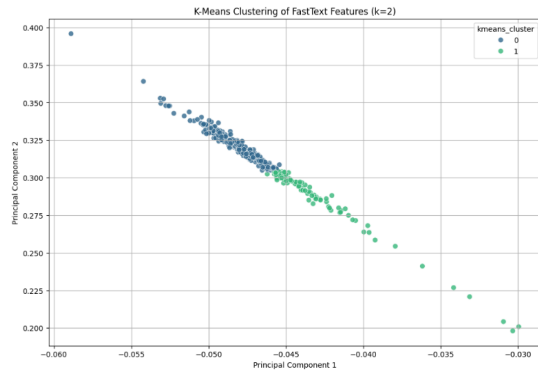
$$J = \sum_{j=1}^k \sum_{i=1}^n \|x_i^{(j)} - c_j\|^2 \quad (5)$$

dengan keterangan:

- J : Total nilai error yang ingin diminimalkan
- k : Jumlah *cluster* yang ditentukan
- n : Jumlah data dalam *cluster* ke- j
- $x_i^{(j)}$: Titik data ke- i yang berada di dalam *cluster* ke- j
- c_j : Titik pusat (*centroid*) dari *cluster* ke- j
- $\|x_i^{(j)} - c_j\|^2$: Jarak kuadrat antara data dan *centroid*

Untuk mengukur seberapa dekat setiap titik data dengan semua *centroid* yang ada, digunakan rumus jarak *euclidean distance* pada persamaan (6). Sedangkan posisi *centroid* yang terus diperbarui dihitung dengan mencari rata-rata posisi seluruh data yang masuk dalam *cluster* tersebut. Rumus mencari *centroid* ini dapat dilihat pada persamaan (7).

$$d(x, c) = \sqrt{\sum_{m=1}^D (x_m - c_m)^2} \quad (6)$$

Gambar 8. Hasil *Clustering K-Means* dengan $k = 2$

dengan keterangan: D = jumlah fitur/dimensi data; x_m = nilai fitur ke- m pada data; c_m = nilai fitur ke- m pada *centroid*.

$$c_j = \frac{1}{n_j} \sum_{x \in S_j} x \quad (7)$$

dengan keterangan: c_j = koordinat *centroid* baru untuk *cluster* ke- j ; n_j = jumlah total data yang menjadi anggota *cluster* ke- j ; $\sum x$ = jumlah total nilai koordinat semua data anggota *cluster* ke- j .

Dengan melakukan uji coba beberapa nilai k berbeda, diperoleh k terbaik adalah $k = 2$. Untuk evaluasi algoritma *K-Means* menggunakan *silhouette score* dijelaskan pada akhir pembahasan. Sedangkan visualisasi hasil pengelompokan atau *clustering* ulasan buku *The Da Vinci Code*, dapat dilihat dalam Gambar 8. Dari Gambar 8 dapat dilihat bahwa untuk $k = 2$, *cluster* 0 ditunjukkan oleh titik-titik berwarna biru sedangkan *cluster* 1 ditunjukkan oleh titik-titik berwarna hijau.

3.2 PCA kemudian *K-Means Clustering*

Untuk langkah ini, vektor hasil *word embedding* pada Gambar 7 tidak langsung dikelompokkan menjadi *cluster*. Vektor yang awalnya memiliki 100 dimensi tersebut direduksi terlebih dahulu agar memiliki dimensi yang lebih rendah baru kemudian dilakukan *clustering*. Pada penelitian ini proses reduksi dimensi vektor dilakukan menggunakan PCA untuk meningkatkan kinerja algoritma *clustering* dan efisiensi komputasi (Rosyada dan Utari, 2024).

Untuk melakukan PCA, ada beberapa langkah yang harus dilakukan yaitu mengubah data menjadi bentuk standar, menghitung matriks kovarian, kemudian memproyeksi data ke ruang baru. Langkah pertama standarisasi data dilakukan menggunakan persamaan (8).

$$z = \frac{x - \mu}{\sigma} \quad (8)$$

dengan keterangan: z = nilai yang distandarisasi; x = nilai asli; μ = nilai rata-rata fitur; σ = standar deviasi fitur.

Langkah selanjutnya adalah menghitung matriks kovarian yang merupakan matriks persegi dan dapat menunjukkan bagaimana setiap variabel dalam

First 5 rows of PCA-reduced FastText features (sample):

	0	1	2	3	4	5	6
0	0.079147	-0.001887	0.000111	-0.000754	-0.000873	0.000870	-0.000255
1	0.116753	0.004141	-0.001042	0.000250	0.000560	-0.000471	0.000335
2	-0.229465	0.000145	0.000734	-0.001086	0.001165	0.000445	-0.000712
3	-0.021401	0.001171	-0.000512	-0.000211	-0.000128	0.000588	-0.000531
4	0.085512	-0.000228	-0.000062	-0.000344	-0.000381	-0.000112	-0.000107

	7	8	9	...	40	41	42
0	-0.000595	-0.000234	0.000832	...	0.000195	0.000308	2.854767e-04
1	-0.000445	0.000087	-0.000124	...	0.000223	-0.000122	-1.083141e-07
2	0.000113	-0.000305	-0.001555	...	0.000535	-0.000156	-1.540149e-04
3	-0.000017	-0.000222	-0.000030	...	-0.000173	-0.000314	-2.184453e-04
4	-0.000165	-0.000168	0.000597	...	0.000085	-0.000086	-1.035593e-04

	43	44	45	...	46	47	48	49
0	-0.000300	-0.000360	-0.000524	...	0.000512	-0.000058	-0.000054	-0.000023
1	-0.000098	0.000138	-0.000119	...	0.000039	0.000067	0.000092	-0.000013
2	-0.000048	-0.000289	0.000557	...	-0.000561	0.000023	-0.000140	0.000043
3	0.000071	0.000221	0.000168	...	0.000063	-0.000192	-0.000026	-0.000066
4	-0.000120	0.000089	0.000181	...	0.000058	-0.000098	0.000046	0.000090

[5 rows x 50 columns]

Gambar 9. Hasil Reduksi Dimensi Menggunakan PCA

kumpulan data saling berhubungan atau berubah bersamaan. Dari matriks kovarian selanjutnya diekstrak nilai eigen untuk mengukur varians dan vektor eigen untuk menentukan arah sumbu baru atau komponen utama. Secara matematis, hubungan antara matriks, vektor eigen, dan nilai eigen dapat dilihat pada persamaan (9). Pada saat memilih komponen utama, pertimbangan yang digunakan adalah nilai eigen terbesar yang dipilih. Kemudian proyeksikan data tersebut ke ruang dimensi yang baru.

$$Ax = \lambda x \quad (9)$$

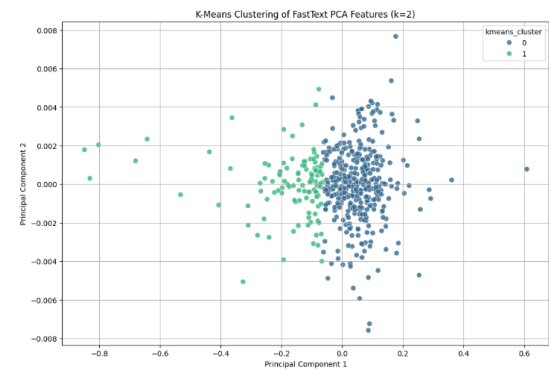
dengan keterangan: A = matriks persegi; x = vektor eigen (bukan vektor nol); λ = nilai eigen yang bernilai skalar.

Data hasil proyeksi merupakan hasil dari reduksi dimensi pada PCA. Fitur atau dimensi yang sebelumnya ada 100, pada penelitian ini direduksi menjadi 50. Untuk hasil reduksi dimensi menggunakan python dapat dilihat dalam Gambar 9. Kolom 0 sampai 49 menunjukkan fitur yang baru.

Langkah analisis dilanjutkan dengan melakukan *clustering* pada data hasil reduksi dimensi. Langkah ini sama dengan yang sudah dilakukan pada subbab 3.1. Ketika dilakukan proses *clustering* menggunakan algoritma *K-Means*, diperoleh bahwa k terbaik adalah $k = 2$. Evaluasi kembali dilakukan dengan menggunakan *silhouette score*.

Visualisasi untuk langkah 3.2 dapat dilihat dalam Gambar 10. Titik-titik berwarna biru adalah titik dari *cluster* 0 sedangkan titik berwarna hijau adalah titik dari *cluster* 1. Terdapat banyak titik terkumpul yang memiliki koordinat berdekatan. Hal ini menunjukkan banyak ulasan menggunakan kata-kata yang sama atau mirip. Namun ada juga beberapa titik yang posisinya cukup jauh dari titik yang lain. Ini menunjukkan bahwa titik tersebut merupakan data unik yang menggunakan kata kunci berbeda dari data ulasan yang lain.

Hasil evaluasi langkah-langkah pada subbab 3.1 dan 3.2 disajikan pada Tabel 1. Dari tabel terlihat bahwa *clustering* dengan reduksi dimensi memperoleh *silhouette score* yang lebih mendekati angka 1. Hal ini menandakan bahwa metode tersebut lebih bagus (Sudrajat et al, 2025). Meskipun demikian, un-



Gambar 10. Hasil Analisis *K-Means* dengan PCA, untuk $k = 2$

Tabel 1. Evaluasi Menggunakan *Silhouette Score*

Jumlah k	Tanpa PCA	Dengan PCA
2	0.5899	0.5901
3	0.5539	0.5541
4	0.5262	0.5266
5	0.5424	0.5428

tuk klaim bahwa analisis studi kasus yang menggunakan PCA ini berbeda secara signifikan dengan analisis tanpa PCA, diperlukan pengujian lebih lanjut.

4. Kesimpulan

Berdasarkan hasil penelitian, penerapan algoritma *K-Means* pada ulasan berbahasa Indonesia buku *The Da Vinci Code* di *Goodreads* mampu mengelompokkan data ulasan dengan baik, dengan jumlah *cluster* optimal sebanyak dua kelompok yaitu *cluster* 0 dan *cluster* 1. Hasil evaluasi menunjukkan bahwa penggunaan PCA sebelum proses *clustering* menghasilkan nilai *silhouette score* yang sedikit lebih tinggi dibandingkan tanpa PCA. Hal ini menunjukkan bahwa reduksi dimensi dapat membantu meningkatkan kualitas pengelompokan data meskipun peningkatannya tidak terlalu besar. Dengan demikian, kombinasi PCA dan *K-Means* dapat menjadi pendekatan yang efektif dalam analisis dan pengelompokan ulasan teks berbahasa Indonesia.

Sedangkan saran untuk penelitian selanjutnya, reduksi dimensi pada *clustering* bisa dilakukan dengan mengubah jumlah fitur yang direduksi (tidak berjumlah 50 fitur). Modifikasi juga bisa dilakukan dengan mengubah metode reduksi dimensinya sebagai contoh menggunakan metode UMAP.

Daftar Pustaka

Afuan, L., & Hidayat, N. (2024): *Sentiment Analysis of the Kampus Merdeka Program on Twitter Using Support Vector Machine and a Feature Extraction Comparison: TF-IDF vs. FastText*, Journal of Applied Data Sciences, 5(4), 1738–1753. <https://doi.org/10.47738/jads.v5i4.436>

Amalia, A., Sitompul, O. S., Mantoro, T., & Naba-

ban, E. B. (2021): *Morpheme Embedding for Bahasa Indonesia Using Modified Byte Pair Encoding*, IEEE Access, 9, 155699–155710. <https://doi.org/10.1109/ACCESS.2021.3128439>

Fazri, M., & Voutama, A. (2025): *Analisis Sentimen Publik Terhadap Danantara di Media Sosial X Menggunakan NLP dan Pembelajaran Mesin*, JOISIE (Journal of Information Systems and Informatics Engineering), 9(1), 197. <https://doi.org/10.35145/joisie.v9i1.4924>

Gramedia. (2026): *The Da Vinci Code (Edisi Ke-3, Republish 2025)*, Gramedia.com. <https://www.gramedia.com/products/the-da-vinci-code-edisi-ke3-republish-2025>

Harris, A., Nugroho, A., Novianto, Y., Jasmir, J., & Fatma, D. (2026): *Fitur Word Embedding untuk Meningkatkan Kinerja Machine Learning pada Analisis Sentimen Game Honor of Kings (Word Embedding Features to Improve Machine Learning Performance in Sentiment Analysis of the Honor of Kings Game)*, Sistemasi: Jurnal Sistem Informasi. <http://sistemasi.ftik.unisi.ac.id>

Hasan, Y. (2024): *Pengukuran Silhouette Score dan Davies-Bouldin Index pada Hasil Cluster K-Means dan DbSCAN* (Vol. 06, Issue 01).

Hediyati, D., & Suartana, I. M. (2021): *Penerapan Principal Component Analysis (PCA) untuk Reduksi Dimensi pada Proses Clustering Data Produksi Pertanian di Kabupaten Bojonegoro*.

Khairuna, R., Nurdin, & Ar Razi. (2025): *Analisis Perbandingan Algoritma K-Means dan K-Medoids untuk Klusterisasi Teks Ulasan pada Aplikasi Cookpad*, Rabbit: Jurnal Teknologi Dan Sistem Informasi Univrab, 10(2), 445–458. <https://doi.org/10.36341/rabit.v10i2.6131>

60 Minutes Australia (2019): *The Da Vinci Code phenomenon | 60 Minutes Australia*, Youtube. <https://www.youtube.com/watch?v=qJY1ak5VQcs>

Nurdin, A., Anggo, B., Aji, S., Bustamin, A., & Abidin, Z. (2020): *Perbandingan Kinerja Word Embedding Word2vec, Glove, dan Fasttext pada Klasifikasi Teks*, Jurnal Teknokompak, 14(2), 74.

Nurjanah, M., & Arifin, T. (2021): *Penerapan Algoritma K-Means untuk Analisis Data Ulasan di Situs Tripadvisor*, Jurnal Responsif, 3(1), 75–82. <http://ejurnal.ars.ac.id/index.php/jti>

Pamungkas, M. D., & Februariyanti, H. (2022): *Penerapan Algoritma K-Means Clustering untuk Mengelompokan Data Review Barang pada E-Commerce Lazada*, SemanTIK, 8(2), 99. <https://doi.org/10.55679/semantik.v8i2.29058>

Prasetyo, S. D., Hilabi, S. S., & Nurapriani, F. (2023): *Analisis Sentimen Relokasi Ibukota Nusantara Menggunakan Algoritma Naïve Bayes dan KNN*, Jurnal Komtekinfo, 1–7. <https://doi.org/10.35134/komtekinfo.v10i1.330>

Putra, B. A., Mukhtar, H., Titasari Br Bangun, E., Gusnanda, A., Maisyarah, A., Kurniawan, M. I., Pradipa, R., & Ali, Z. M. (2025): *Optimisasi*

- Algoritma K-Means dengan Metode Reduksi Dimensi untuk Pengelompokan Big Data dalam Arsitektur Cloud Computing*, 5(1), 1–8.
- Ramang, E., Haryono, K., Lailiyah, S., & Sa'ad, M. I. (2025): *Implementation of Data Clustering for Informatics Engineering Study Program Students at STMIK Widya Cipta Dharma Using the K-Means Method Implementasi Clustering Data Mahasiswa Program Studi Teknik Informatika STMIK Widya Cipta Dharma Menggunakan Metode K-Means*, <https://doi.org/10.46984/sebatik.v29i1.0000>
- Rianti, R., Andarsyah, R., & Awangga, R. M. (2024): *Penerapan PCA dan Algoritma Clustering untuk Analisis Mutu Perguruan Tinggi di LLDIKTI Wilayah IV* (Vol. 18). <https://journal.fkom.uniku.ac.id/ilkom>
- Ritonga, A. S., & Muhandhis, I. (2021): *Teknik Data Mining untuk Mengklasifikasikan Data Ulasan Destinasi Wisata Menggunakan Reduksi Data Principal Component Analysis (PCA)*, In *Jurnal Ilmiah Edutic* (Vol. 7, Issue 2).
- Rosyada, I. A., & Utari, D. T. (2024): *Penerapan Principal Component Analysis untuk Reduksi Variabel pada Algoritma K-Means Clustering*, *Jambura Journal of Probability and Statistics*, 5(1), 6–13. <https://doi.org/10.34312/jjps.v5i1.18733>
- Saputra, R., & Purnama, I. (2024): *Optimizing K-Means Algorithm with Elbow and Silhouette Methods for National Exam Score Data Clustering*, In *Jurnal Ilmu Komputer Ruru* (Vol. 1).
- Saputro, S. W., Abadi, F., & Nugroho, R. A. (2025): *Analisis Sentimen Ulasan Media Sosial UMKM Kuliner dengan Pendekatan Lexicon-Based dan Kosakata Khusus*, *JIP (Jurnal Informatika Polinema)*.
- Sari, P., & Syahri, R. (2023): *Algoritma K-Means Clustering: Sebuah Studi Literatur*, *Jurnal Informatika (JURI)*. <https://doi.org/10.12345/juri>
- Sepriadi, N., Budianita, E., Fikry, M., Fakultas Sains dan Teknologi, & Institut Negeri Kota Riau. (2023): *Analisis Sentimen Review Aplikasi MyPertamina Menggunakan Word Embedding Fasttext dan Algoritma K-Nearest Neighbor*, *Informasi (Jurnal Informatika dan Sistem Informasi)*.
- Sinaga, A., & Nainggolan, S. P. (2023): *Analisis Perbandingan Akurasi dan Waktu Proses Algoritma Stemming Arifin-Setiono dan Nazief-Adriani pada Dokumen Teks Bahasa Indonesia*, *Sebatik*, 27(1), 63–69. <https://doi.org/10.46984/sebatik.v27i1.2072>
- Sudrajat, R., Hadiana, A. I., & Melina, M. (2025): *Evaluasi Kualitas Klaster Wilayah Rawan Bencana Menggunakan K-Means dengan Silhouette dan Elbow Method*, *Jurnal Algoritma*, 22(2), 127–139. <https://doi.org/10.33364/algoritma/v.22-2.2379>
- Wicaksono, S. A., Wijoyo, S. H., & Aryadita, H. (2025): *Penentuan Strategi Promosi Program Studi Pendidikan Teknologi Informasi Menggunakan Algoritma K-Means Clustering*, *JIP (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, 10(4), 4153–4162. <https://doi.org/10.29100/jipi.v10i4.9576>
- Yolandari, N. A., Butarbutar, L. E., Rajagukguk, G. C. H., Zulfi, M. F., Arnita, & Ramadhani, F. (2025): *Analisis Perbandingan K-Means dan DbSCAN dalam Pengelompokan Data Travel Review Ratings Menggunakan Evaluasi Silhouette Index dan Davies-Bouldin Index*, *Jurnal Informatika dan Teknik Elektro Terapan*, 13(3). <https://doi.org/10.23960/jitet.v13i3.6884>

Halaman ini sengaja dikosongkan