APLIKASI CLUSTERING FILM BERDASARKAN SINOPSISNYA

LAPORAN

Disusun untuk Memenuhi Tugas Kelompok
pada Mata Kuliah Pengolahan Bahasa Alami Semester VIII
yang diampu oleh Ibu Sukmawati Nur Endah, S.Si., M.Kom.

Disusun oleh:
Linggar Maretva Cendani (24060117120031)
Fetty Krisnaeni (24060117120032)

DEPARTEMEN ILMU KOMPUTER/INFORMATIKA
FAKULTAS SAINS DAN MATEMATIKA
UNIVERSITAS DIPONEGORO
SEMARANG
2021

ABSTRAK

Dunia perfilman memiliki kisah perjalanan yang cukup panjang, mulai dari film bisu yang tidak berwarna hingga saat ini telah menjadi film yang kaya akan efek dan dapat dengan mudah ditemukan di dunia hiburan. Hingga saat ini sudah banyak film-film berkualitas dengan berbagai macam genre yang dapat disaksikan oleh masyarakat. Genre merupakan salah satu preferensi penonton terhadap film berdasarkan atribut filmnya. Sebagian besar orang akan menonton film apabila mereka suka dengan genre yang diminati oleh mereka. Oleh karena itu pengelompokkan film sangat diperlukan bagi orang-orang yang ingin mendapatkan informasi mengenai film-film yang berada dalam lingkup genre yang sama. Pada penelitian ini akan dibangun Aplikasi Clustering Film Berdasarkan Sinopsisnya yang merupakan aplikasi text clustering untuk mengelompokkan film berdasarkan sinopsisnya. Data judul film dan sinopsis yang akan digunakan pada penelitian ini diambil dari website https://www.themoviedb.org/ yaitu berupa judul film dan sinopsisnya. Aplikasi ini dibuat dalam bentuk web dengan menggunakan bahasa pemrograman Python dan HTML.

Kata kunci: Text Clustering, Clustering Film, Python, KMeans.

KATA PENGANTAR

Dengan menyebut nama Allah SWT yang Maha Pengasih lagi Maha Penyayang, atas karunia dan rahmat-Nya penulis dapat menyelesaikan pembuatan Aplikasi Teks Clustering dan menyusun laporan yang berjudul "Aplikasi Clustering Film Berdasarkan Sinopsisnya" dengan baik. Laporan ini disusun untuk memenuhi tugas kelompok pada mata kuliah Pengolahan Bahasa Alami di Departemen Ilmu Komputer, Universitas Diponegoro.

Ucapan terima kasih penulis sampaikan kepada Ibu Sukmawati Nur Endah, S.Si., M.Kom. selaku dosen pengampu mata kuliah Pengolahan Bahasa Alami yang senantiasa membimbing penulis. Terima kasih juga penulis sampaikan kepada segenap pihak yang telah memberikan dukungan dan bantuan sehingga penulis dapat menyelesaikan laporan ini tepat pada waktunya.

Penulis menyadari bahwa dalam laporan ini masih banyak kekurangan baik dari segi materi ataupun dalam penyajiannya. Kritik dan saran sangat penulis harapkan untuk perbaikan pada penulisan ilmiah yang akan datang. Penulis berharap laporan ini dapat bermanfaat bagi pembaca pada umumnya dan penulis sendiri pada khususnya.

Semarang, 10 Juni 2021

Penulis

DAFTAR ISI

DAFTAR GAMBAR

BAB I
PENDAHULUAN

Pada bab ini akan membahas mengenai latar belakang, rumusan masalah, dan sistematika penulisan yang dibuat dalam proses pembuatan Aplikasi Clustering Film Berdasarkan Sinopsisnya.

1.1. Latar Belakang

Dunia perfilman memiliki kisah perjalanan yang cukup panjang, mulai dari film bisu yang tidak berwarna hingga saat ini telah menjadi film yang kaya akan efek dan dapat dengan mudah ditemukan di dunia hiburan. Seiring berjalannya waktu hingga sekitar tahun 1980/1990-an film Indonesia semakin meningkat. Namun peningkatan tersebut juga diikuti dengan masuknya film luar baik Hollywood ataupun Bollywood yang akhirnya mendominasi perfilman negeri. Masyarakat Indonesia cenderung menyukai film luar karena memiliki karakteristik tersendiri, alur cerita yang menarik, dan mindset bahwa menonton film luar lebih membuat mereka keren. Selain itu, kreatifitas dalam pembuatan film luar jauh lebih bagus, berbeda dengan film Indonesia yang bermain aman dengan hanya memproduksi film dengan genre tertentu yang akan banyak ditonton masyarakat.

Hingga saat ini sudah banyak film-film berkualitas dengan berbagai macam genre yang dapat disaksikan oleh masyarakat. Genre merupakan salah satu preferensi penonton terhadap film berdasarkan atribut filmnya. Sebagian besar orang akan menonton film apabila mereka suka dengan genre yang diminati oleh mereka. Genre film dapat kita ketahui dari judul maupun alur cerita yang ada pada film. Kita juga dapat mengetahui genre suatu film berdasarkan sinopsis film yang kita baca.

Pengelompokkan film sangat diperlukan bagi orang-orang yang ingin mendapatkan informasi mengenai film-film yang berada pada satu jenis genre. Selain itu pengelompokkan film akan bermanfaat bagi orang yang hanya menyukai suatu genre film tertentu sehingga mereka dapat menonton film dengan konteks cerita yang disuakinya. Contohnya seperti seseorang yang hanya menyukai film dengan genre horror akan terbantu, karena mereka akan mendapatkan informasi kumpulan film bergenre horror.

Dalam hal ini penulis melakukan penelitian mengenai text clustering yang akan mengelompokkan film berdasarkan sinopsisnya. Data yang akan digunakan pada penelitian ini diambil dari website https://www.themoviedb.org/ yaitu berupa judul film dan

sinopsisnya. Hasil penelitian berupa aplikasi text clustering untuk menentukan genre film berdasarkan sinopsisnya yang dibuat dalam bentuk web.

1.2. Rumusan Masalah

Berdasarkan uraian permasalahan pada latar belakang di atas, maka penulis merumuskan masalah sebagai berikut:

1. Merancang aplikasi text clustering untuk menentukan genre film berdasarkan sinopsisnya.
2. Merancang desain antar muka aplikasi text clustering untuk menentukan genre film berdasarkan sinopsisnya.

1.3. Sistematika Penulisan

Untuk memberikan gambaran yang urut dan jelas mengenai pembahasan penyusunan Aplikasi Clustering Film Berdasarkan Sinopsisnya maka dibuatlah sistematika penulisan sebagai berikut:

BAB I PENDAHULUAN

Bab ini membahas mengenai latar belakang, rumusan masalah, dan sistematika penulisan yang dibuat dalam proses pembuatan Aplikasi Clustering Berdasarkan Sinopsisnya.

BAB II DASAR TEORI

Bab ini menjelaskan dasar-dasar teori yang mendukung dalam pembuatan Aplikasi Clustering Film Berdasarkan Sinopsisnya.

BAB III METODOLOGI

Bab ini akan membahas metodologi penelitian yang digunakan dalam pembuatan Aplikasi Clustering Film Berdasarkan Sinopsisnya.

BAB IV HASIL DAN PEMBAHASAN

Bab ini menjabarkan tampilan antar muka, hasil penelitian, dan analisa hasil dalam pembuatan Aplikasi Clustering Film Berdasarkan Sinopsisnya sesuai dengan metodologi yang digunakan.

BAB V PENUTUP

Bab ini merupakan penutup yang berisi kesimpulan dari bab-bab yang dibahas sebelumnya.

BAB II
DASAR TEORI

Bab ini menjelaskan dasar-dasar teori yang mendukung dalam pembuatan Aplikasi Clustering Film Berdasarkan Sinopsisnya. Dasar teori yang digunakan dalam penelitian ini adalah sebagai berikut:

2.1. Clustering

Clustering merupakan suatu proses mengelompokkan suatu objek ke dalam kelompok-kelompok objek yang sejenis. Bentuk data yang paling umum digunakan dalam clustering yaitu data unsupervised learning. Dalam proses pengelompokannya algoritma clustering akan membagi suatu objek menjadi subset objek, di mana setiap subset berisi objek-objek yang dianggap sejenis.

Clustering dapat digunakan untuk mengorganisasikan dokumen yang diperoleh. Beberapa alasan perlu adanya pegelompokkan dokumen adalah sebagai berikut:

a. Untuk analisa keseluruhan korpus.
b. Untuk visualisasi koleksi dokumen dan topiknya.
c. Untuk memperbaiki recall pada hasil pelacakan.
d. Untuk navigasi yang lebih baik dari hasil pelacakan.

Dalam proses clustering terdapat beberapa istilah yang perlu dipahami, yaitu cluster, objek, dan properti. Cluster merupakan kelompok yang dihasilkan dari proses clustering. Objek merupakan sesuatu yang ditempatkan pada kelompok/cluster. Sedangkan properti merupakan cara yang digunakan untuk merepresentasikan hasil cluster tersebut.

Menurut Nadia Nejdah, et all (2009), pada dasarnya dalam setiap algoritma Feature Selection pasti memiliki tiga produk atau hasil akhir yang menyangkut tentang text clustering, yaitu:

1. Kemampuan memitigasi permasalahan yang disebabkan oleh tingginya dimensionalitas dan sparitas terkait penilaian kesamaan dokumen.
2. Kemampuan mengurangi biaya komputasi dari algoritma pengelompokan mengenai waktu pemrosesan dan ruang memori.
3. Kemampuan menyediakan seperangkat istilah yang ringkas yang dapat mencirikan dan membedakan klaster yang terbentuk.

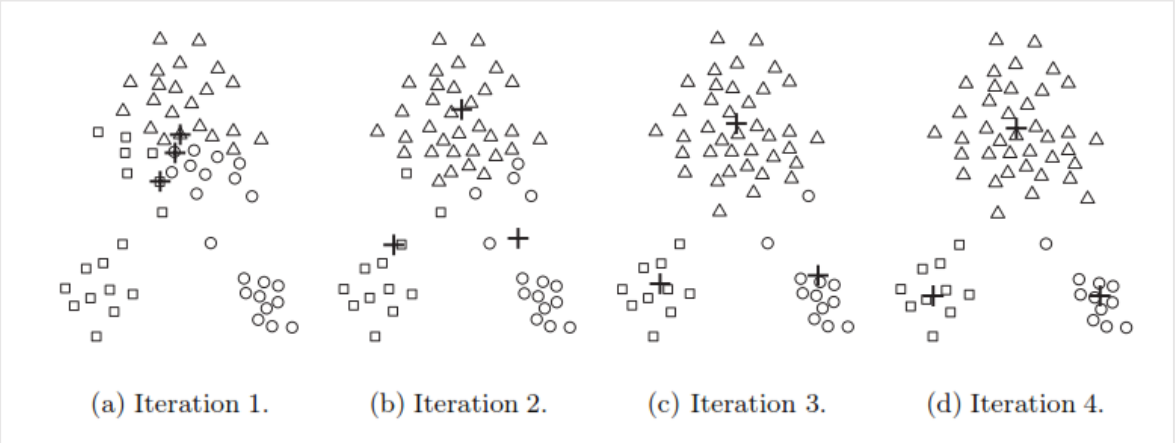
2.2. K-Means Clustering

Pada konsepnya proses K-Means Clustering akan mengelompokkan n objek ke dalam K cluster berdasarkan nilai atribut yang dimiliki oleh objek tersebut. K merupakan jumlah cluster yang berupa bilangan integer positif. K-Means Clustering termasuk jenis hard clustering, di mana satu objek hanya dapat menjadi anggota dari satu cluster secara eksklusif.

Berikut merupakan algoritma dasar K-Means Clustering:

1: Pilih suatu set objek sebagai inisial centroid dari cluster.
2: **Repeat**
3: Tempatkan setiap objek pada cluster terdekat.
4: Hitung kembali centroid sebagai center of mass (average) dari anggota-anggotanya.
5: **Until** Titik centroid tidak berubah lagi.

Algoritma dasar K-Means Clustering dapat lebih jelas dan mudah dipahami seperti yang disajikan pada gambar berikut:



Gambar 2. 1 Visualisasi Algoritma Dasar K-Means Clustering

Untuk menetapkan titik ke centroid terdekat, kita memerlukan ukuran kedekatan yang mengkuantifikasi gagasan "terdekat" untuk data spesifik yang sedang dipertimbangkan. Beberapa cara yang digunakan untuk menetapkan titik ke centroid terdekat yaitu dapat menggunakan Euclidean Distance, Minkowski Distance, dan Manhattan Distance. Penjelasannya sebagai berikut:

1. Euclidean Distance

Euclidean distance merupakan ukuran jarak yang paling sering digunakan untuk data numerik. Jarak Euclidean antara 2 titik atau tuple, $\boldsymbol{x}$ = (x_1,x_2,…,x_d) dan $\boldsymbol{y}$ = (y_1,y_2,…,y_n) pada ruang dimensi d adalah

$$d_{euc}(\boldsymbol{x},\boldsymbol{y}) = \sqrt{\sum_{i=1}^{d}(x_i - y_i)^2}$$

2. Manhattan Distance

Manhattan distance dihitung berdasarkan jumlah jarak dari semua atribut. Jarak Manhattan antara 2 titik atau tuple, $\boldsymbol{x}$ = (x_1,x_2,…,x_d) dan $\boldsymbol{y}$ = (y_1,y_2,…,y_n) pada ruang dimensi d adalah

$$d_{man}(\boldsymbol{x},\boldsymbol{y}) = \sum_{i=1}^{d}|x_i - y_i|$$

3. Minkowski Distance

Minkowski distance merupakan generalisasi dari jarak Euclidean dan Manhattan. Jarak Manhattan antara 2 titik atau tuple, $\boldsymbol{x}$ = (x_1,x_2,…,x_d) dan $\boldsymbol{y}$ = (y_1,y_2,…,y_n) pada ruang dimensi d adalah

$$d_{min}(\boldsymbol{x},\boldsymbol{y}) = \left(\sum_{i=1}^{d}|x_i - y_i|^p\right)^{\frac{1}{p}} \quad , p \geq 1$$

Jika p=1 maka Manhattan distance

Jika p=2 maka Euclidean distance

BAB III
METODOLOGI

Bab ini akan membahas metodologi penelitian yang digunakan dalam pembuatan Aplikasi Clustering Film Berdasarkan Sinopsisnya. Penjelasan mengenai metodologi penelitian yang digunakan pada penelitian ini adalah sebagai berikut:

3.1. Pengumpulan Data

Data yang digunakan dalam penelitian ini merupakan data sinopsis film yang didapatkan dari website https://www.themoviedb.org/. Pada website tersebut terdapat film-film dengan berbagai macam genre yang disajikan bersama dengan sinopsis masing-masing film tersebut. Sinopsis film pada website tersebut disajikan dalam Bahasa Inggris.

Pada bagian pengumpulan data ini dilakukan proses pengambilan data genre film serta melakukan pengambilan data judul dan sinopsis film pada website https://www.themoviedb.org/. Genre film yang ada pada website tersebut berjumlah sebanyak 19 genre, diantaranya adalah Drama, Crime, Comedy, Action, Thriller, Documentary, Adventure, Sience Fiction, Animation, Family, Romance, Mistery, Horror, Fantasy, War, Music, History, Western, dan TV Movie. Sedangkan data judul film dan sinopsis yang diambil dari website https://www.themoviedb.org/ berjumlah 8.457 film. Berikut merupakan tampilan 5 data teratas judul film dan sinopsis yang didapatkan dari website https://www.themoviedb.org/:

	title	synopsis
0	Four Rooms	It's Ted the Bellhop's first night on the job....
1	Judgment Night	While racing to a boxing match, Frank, Mike, J...
2	Life in Loops (A Megacities RMX)	Timo Novotny labels his new project an experim...
3	Star Wars	Princess Leia is captured and held hostage by ...
4	Finding Nemo	Nemo, an adventurous young clownfish, is unexp...

Gambar 3. 1 Data Judul Film dan Sinopsisnya dari website https://www.themoviedb.org/

3.2. Preprocessing Data

Prepocessing data merupakan proses yang dilakukan untuk mendapatkan data yang bersih agar dapat digunakan secara maksimal. Pada proses preprocessing data ini dilakukan

dengan menghapus semua missing value yang ada pada data judul film dan sinopsis, kemudian dilakukan case folding, dan dilanjutkan dengan melakukan TF-IDF training.

Menghapus missing value pada data film dan sinopsis dilakukan karena data tersebut tidak memenuhi kualifikasi atau mungkin datanya null sehingga hasilnya tidak valid dan pastinya tidak akan digunakan dalam penelitian. Seperti pada penelitian ini data awal yang didapatkan dari website https://www.themoviedb.org/ berjumlah 8.457 film, namun karena terdapat beberapa data yang perlu dihapus maka dilakukan proses delete missing value sehingga data yang digunakan untuk melakukan penelitian ini sebanyak 8.214 film.

Case folding dilakukan untuk mengubah semua huruf dalam sinopsis film tersebut menjadi lowercase, kemudian dilanjutkan dengan melakukan TF-IDF training. Berikut merupakan tampilan data judul film dan sinopsis setelah dilakukan proses case folding:

	title	synopsis
0	Four Rooms	it's ted the bellhop's first night on the job....
1	Judgment Night	while racing to a boxing match, frank, mike, j...
2	Life in Loops (A Megacities RMX)	timo novotny labels his new project an experim...
3	Star Wars	princess leia is captured and held hostage by ...
4	Finding Nemo	nemo, an adventurous young clownfish, is unexp...

Gambar 3. 2 Data Judul Film dan Sinopsis Setelah dilakukan Case Folding

3.3. K-Means Model Training

Setelah melewati proses preprocessing data, kemudian data film dan sinopsis yang berjumlah 8.214 akan dilakukan training model K-Means. Proses training model K-Means ini menggunakan jumlah cluster sebanyak 19 cluster dan akan dilakukan iterasi sebanyak 500 kali iterasi. Pada proses ini data 8.214 judul film dan sinopsis akan dikelompokkan ke dalam 19 cluster berdasarkan genre yang sesuai.

3.4. Data Labelling

Proses data labelling dilakukan untuk memberikan label ke semua data film dan sinopsis yang berjumlah 8.214 data. Label yang dimaksud yaitu berupa angka 1 sampai 19 yang menunjukkan cluster ke-1 sampai cluster ke-19. Berikut merupakan tampilan data judul film dan sinopsis setelah dilakukan proses data labelling:

	title	synopsis	label
0	Four Rooms	it's ted the bellhop's first night on the job....	3
1	Judgment Night	while racing to a boxing match, frank, mike, j...	3
2	Life in Loops (A Megacities RMX)	timo novotny labels his new project an experim...	9
3	Star Wars	princess leia is captured and held hostage by ...	3
4	Finding Nemo	nemo, an adventurous young clownfish, is unexp...	13

Gambar 3. 3 Data Judul Film dan Sinopsis Setelah dilakukan Proses Data Labelling

3.5. Data Preview for each Cluster

Proses ini dilakukan untuk menampilkan seluruh data-data judul film beserta sinopsisnya sesuai dengan hasil pengelompokkan clusternya. Setiap cluster akan memberikan informasi film apa saja yang termasuk dalam cluster tersebut diikuti dengan sinopsis filmnya.

3.6. Save All Data from each Cluster

Proses ini dilakukan untuk menyimpan semua data dari masing-masing cluster. Data judul film dan sinopsisnya akan disimpan dalam format file csv.

3.7. Feature Names of each Cluster

Pada proses ini ditampilkan nama-nama fitur untuk setiap cluster. Fitur tersebut merupakan kata-kata yang paling relevan untuk setiap cluster yang menunjukkan karakteristik cluster. Kata-kata untuk nama pada tiap cluster tersebut berasal dari sinopsis film yang termasuk pada cluster tersebut.

3.8. Predict Sectences

Proses predict sentence ini dilakukan untuk memprediksi genre film berdasarkan kalimat atau kata-kata yang diinputkan oleh pengguna pada aplikasi. Kalimat dan kata-kata yang diinputkan dapat berupa sinopsis maupun judul film, kemudian sistem akan menampilkan tersmasuk ke dalam genre apakah film tersebut. Selain itu, aplikasi juga akan menampilkan judul film yang relevan dengan film tersebut berdasarkan genre yang sama.

3.9. Evaluation

Proses evaluasi cluster dilakukan dengan menggunakan Elbow Method (SSE) dan Silhoute Score. Elbow method pada dasarnya digunakan untuk menentukan berapa banyak jumlah cluster yang paling optimal untuk suatu data clustering. Hal ini ditentukan

berdasarkan perubahan nilai SSE yang paling signifikan (perubahan yang paling banyak) jika dilihat dari grafiknya.

Sedangkan evaluasi menggunakan Silhoutte Score dilakukan untuk mengetahui kualitas hasil clustering. Kualitas yang dimaksud yaitu apabila titik-titik di suatu cluster memiliki kemiripan yang tinggi dan jarak/perbedaan antar cluster semakin tinggi maka hasil clustering tersebut akan semakin bagus. Nilai hasil evaluasi silhoutte berada diantara -1 dan 1. Jika nilainya mendekati -1 maka hasil evaluasi silhoutte ini semakin jelek dan clustering yang dihasilkan juga semakin jelek. Tetapi sebaliknya apabila nilainya mendekati 1 maka hasil evaluasi silhoutte ini semakin bagus dan hasil akhir clustering yang dihasilkan akan semakin bagus juga.

3.10. Data Visualization

Data visualization untuk hasil clustering judul film dan sinopsisnya dilakukan dalam bentuk visualisasi 2 dimensi dan 3 dimensi. Masing-masing bentuk visualisasi akan dilakukan dengan PCA Dimensionality Reduction dan t-SNE Dimensionality Reduction, baik untuk visualisasi 2 dimensi maupun 3 dimensi.

BAB IV
HASIL DAN PEMBAHASAN

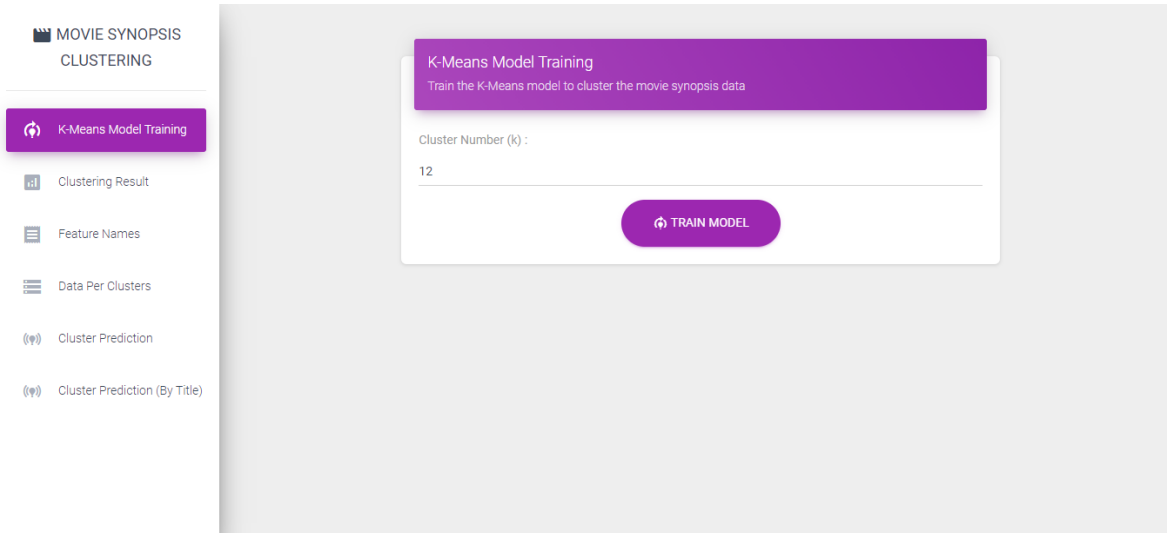
Bab ini menjabarkan tampilan antar muka, hasil penelitian, dan analisa hasil dalam pembuatan Aplikasi Clustering Film Berdasarkan Sinopsisnya sesuai dengan metodologi yang digunakan.

4.1. Tampilan Antar Muka

Berikut merupakan tampilan antar muka untuk setiap halaman pada Aplikasi Clustering Film Berdasarkan Sinopsisnya:

4.1.1. Tampilan Antar Muka Halaman K-Means Model Training

Berikut merupakan tampilan antar muka halaman awal pada Aplikasi Clustering Film Berdasarkan Sinopsisnya. Pada halaman ini pengguna dapat melakukan training model K-Means dengan memasukkan jumlah cluster yang diinginkan. Pada tampilan berikut sebagai contoh pengguna memasukkan jumlah cluster yaitu 12.

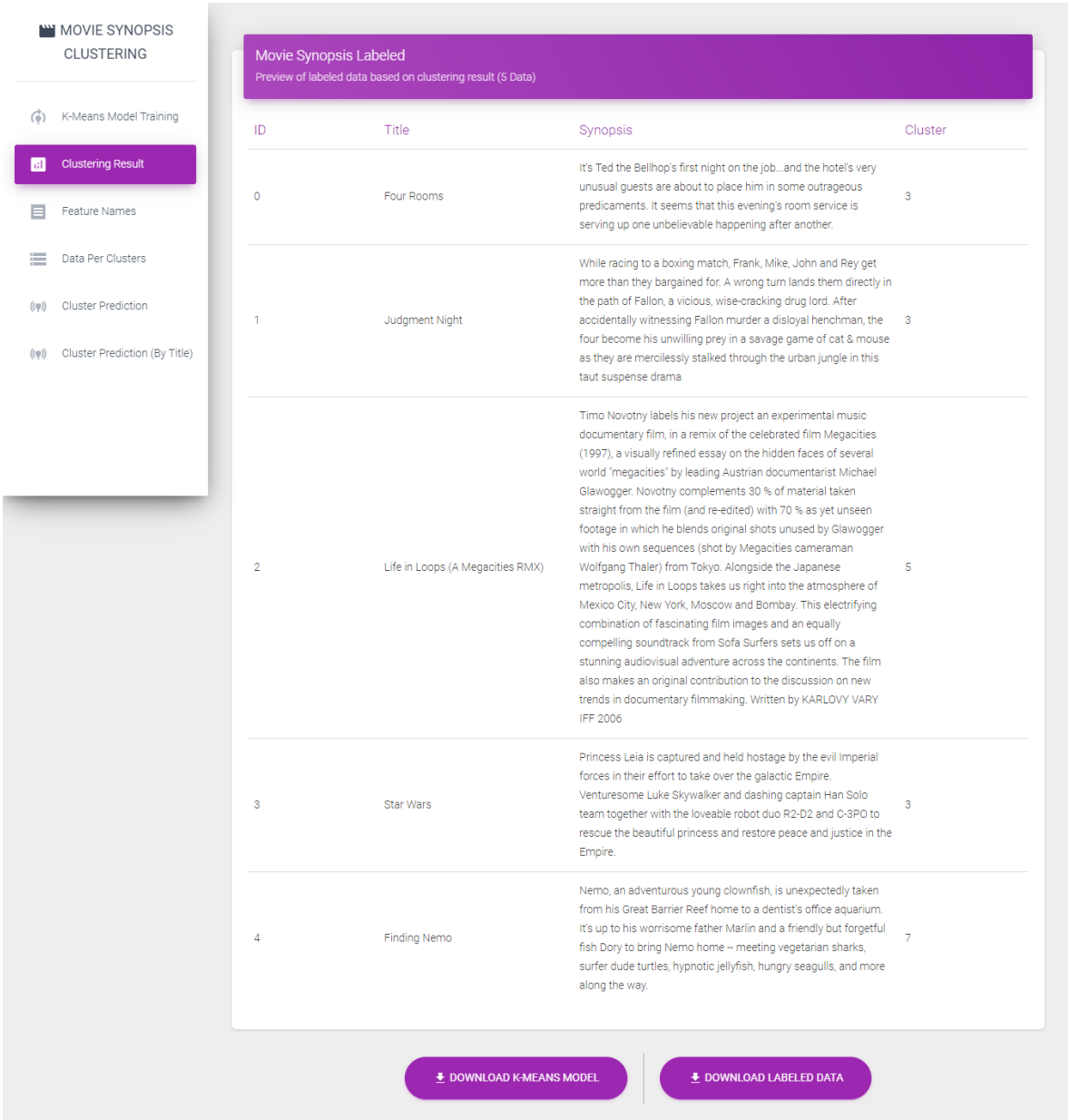


Ganbar 4. 1 Antar Muka Halaman K-Means Model Training

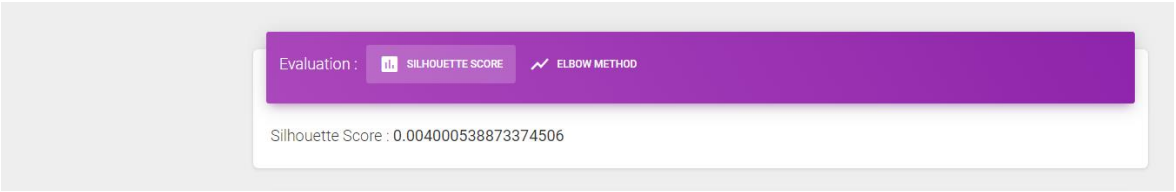
4.1.2. Tampilan Antar Muka Halaman Clustering Result

Pada halaman clustering result akan menampilkan lima data hasil clustering yang sudah dilabeli dengan nomor clusternya. Data hasil clustering yang sudah dilabeli dapat diunduh dengan format file .csv melalui tombol untuk mendownload yang sudah disediakan. Pada halaman ini juga disediakan tombol untuk mendownload model K-Means yang telah ditraining dengan format file .sav.

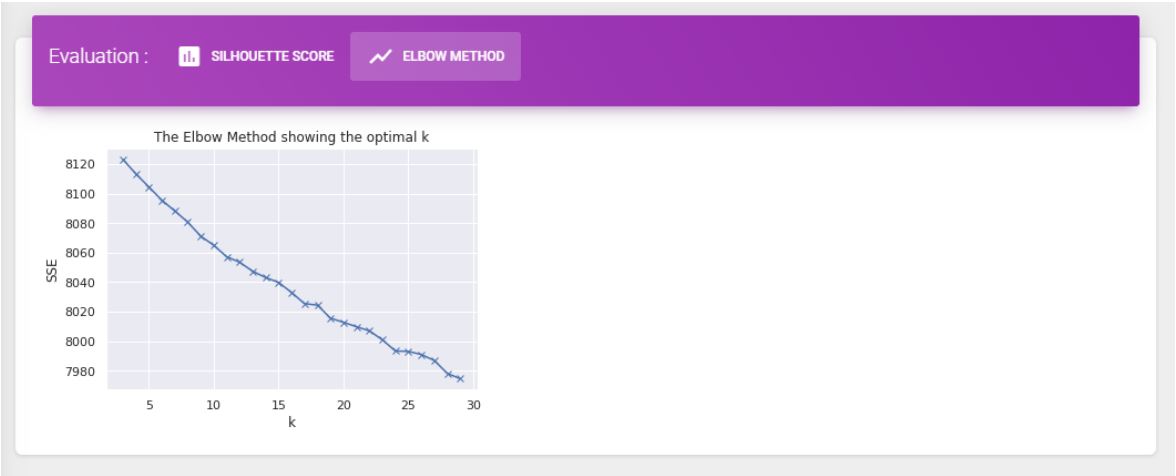
Halaman clustering result juga menampilkan evaluasi clustering. Evaluasi yang ditampilkan yaitu Silhouette Score dan Elbow Method. Kemudian visualisasi data yang digunakan akan ditampilkan menggunakan reduksi fitur dengan PCA dalam bentuk visualisasi 2 dimensi, dan 3 fitur untuk visualisasi 3 dimensi. Berikut merupakan tampilan antar muka halaman clustering result:



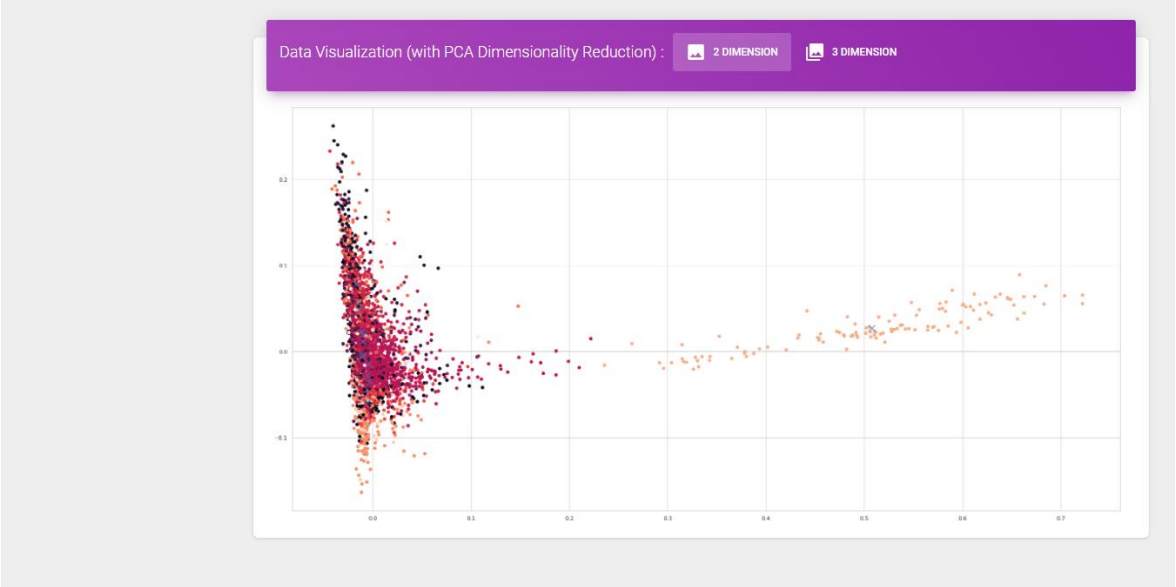
Ganbar 4. 2 Antar Muka Halaman Clustering Result: Movie Synopsis Labelled



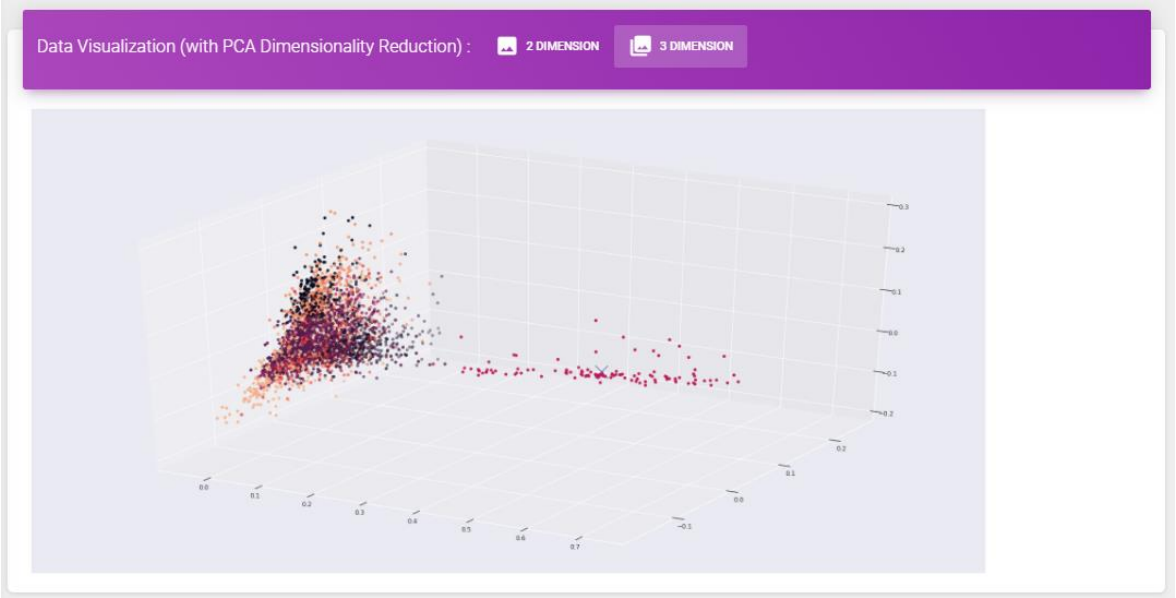
Ganbar 4. 3 Antar Muka Halaman Clustering Result: Evaluasi Silhouette



Ganbar 4. 4 Antar Muka Halaman Clustering Result: Evaluasi Elbow Method



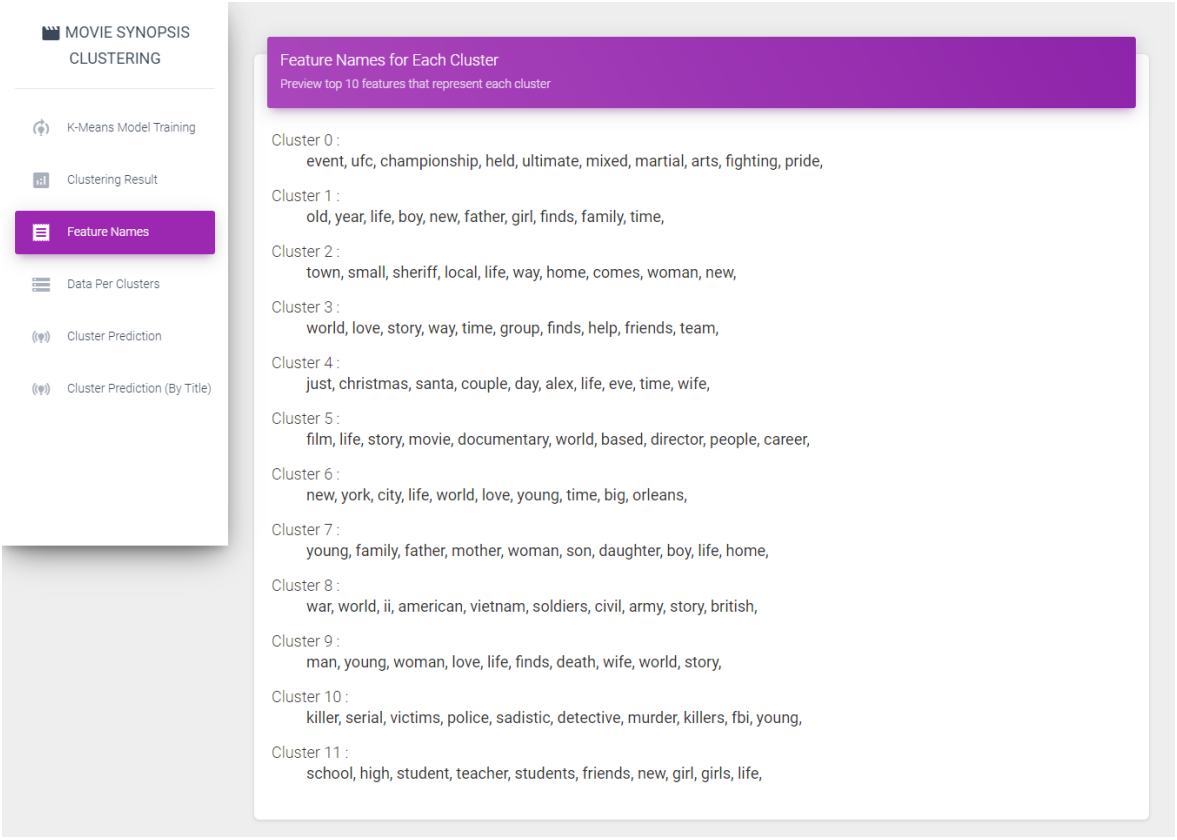
Ganbar 4. 5 Antar Muka Halaman Clustering Result: Visualisasi 2 Dimensi



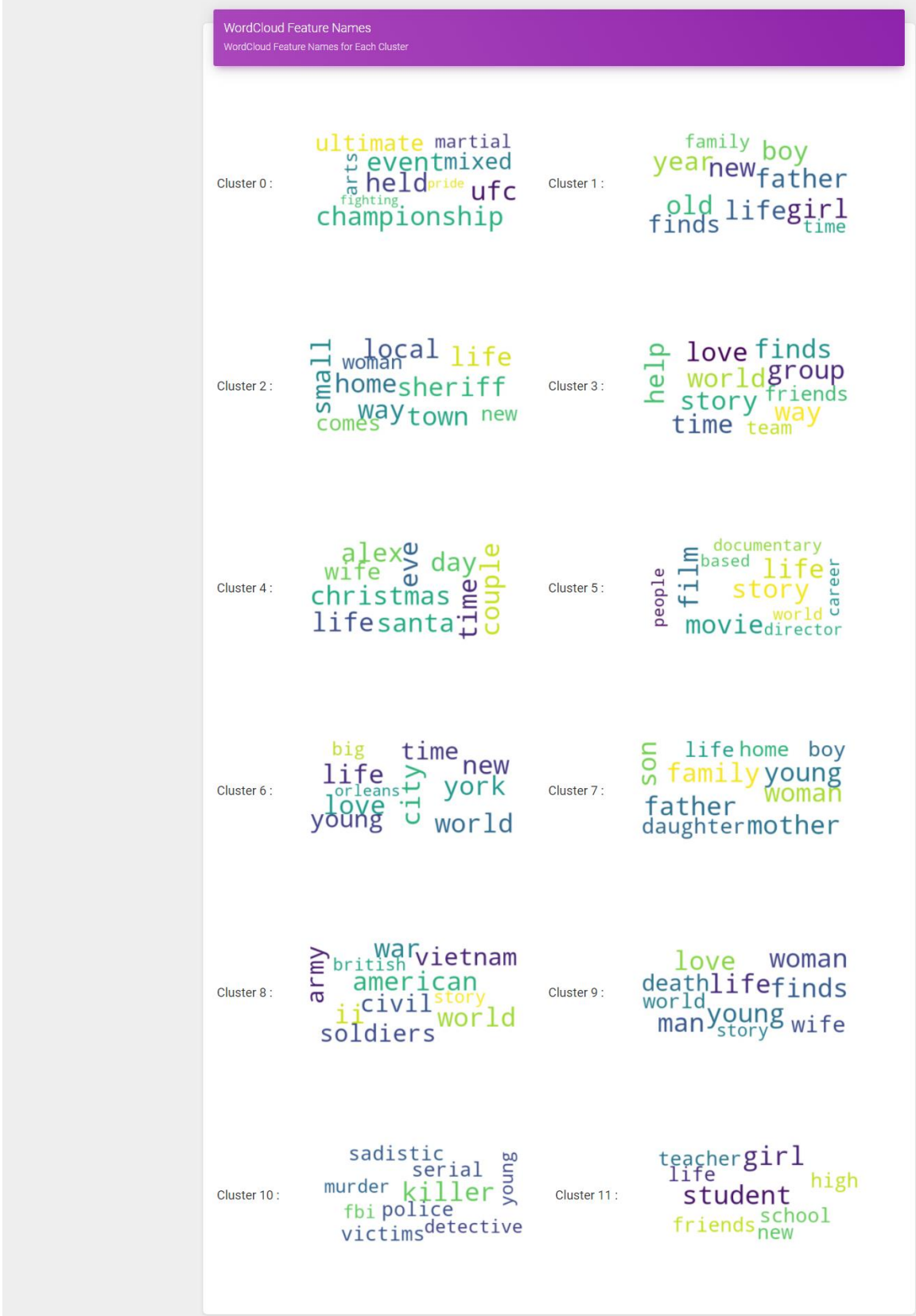
Ganbar 4. 6 Antar Muka Halaman Clustering Result: Visualisasi 3 Dimensi

4.1.3. Tampilan Antar Muka Halaman Feature Names

Halaman feature names merupakan halaman yang menampilkan 10 fitur kata paling relevan untuk tiap cluster. Pada halaman ini akan ditampilkan juga WordCloud dari fitur-fiitur pada tiap cluster. Berikut merupakan tampilan antar muka halaman feature names:



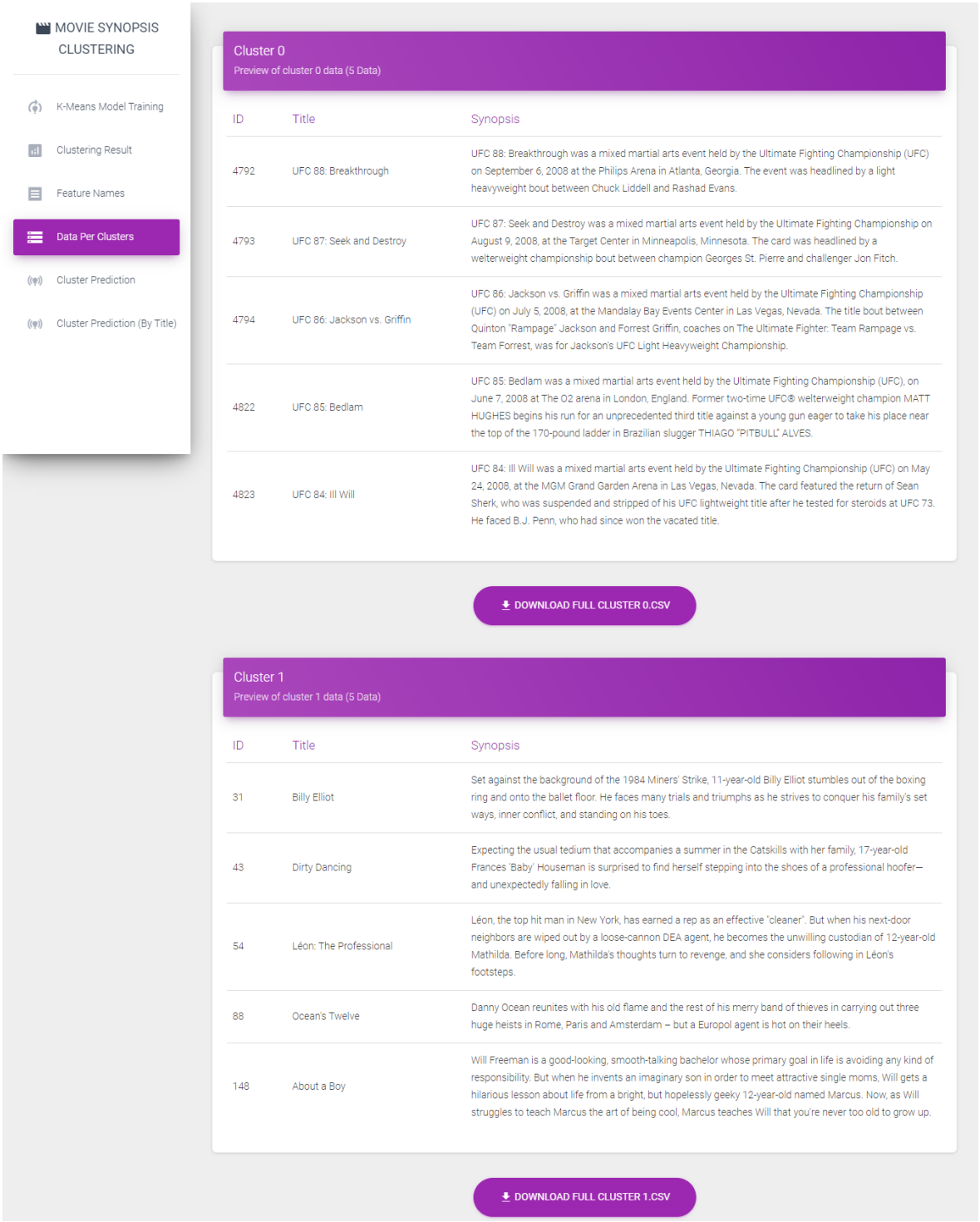
Ganbar 4. 7 Antar Muka Halaman Featue Names Tiap Cluster



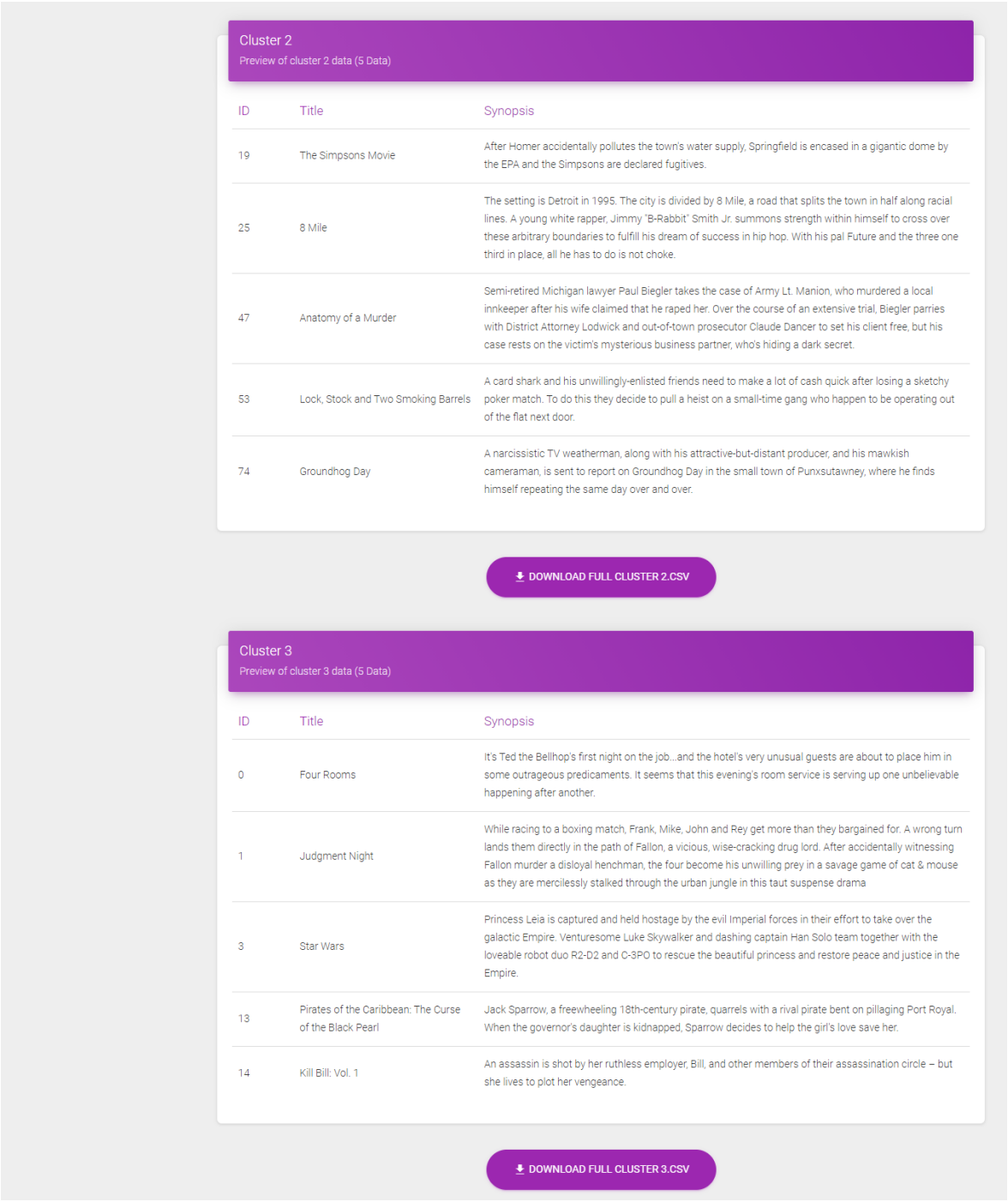
Ganbar 4. 8 Antar Muka Halaman Featue Names: WordCloud Featue Names

4.1.4. Tampilan Antar Muka Halaman Data Per Cluster

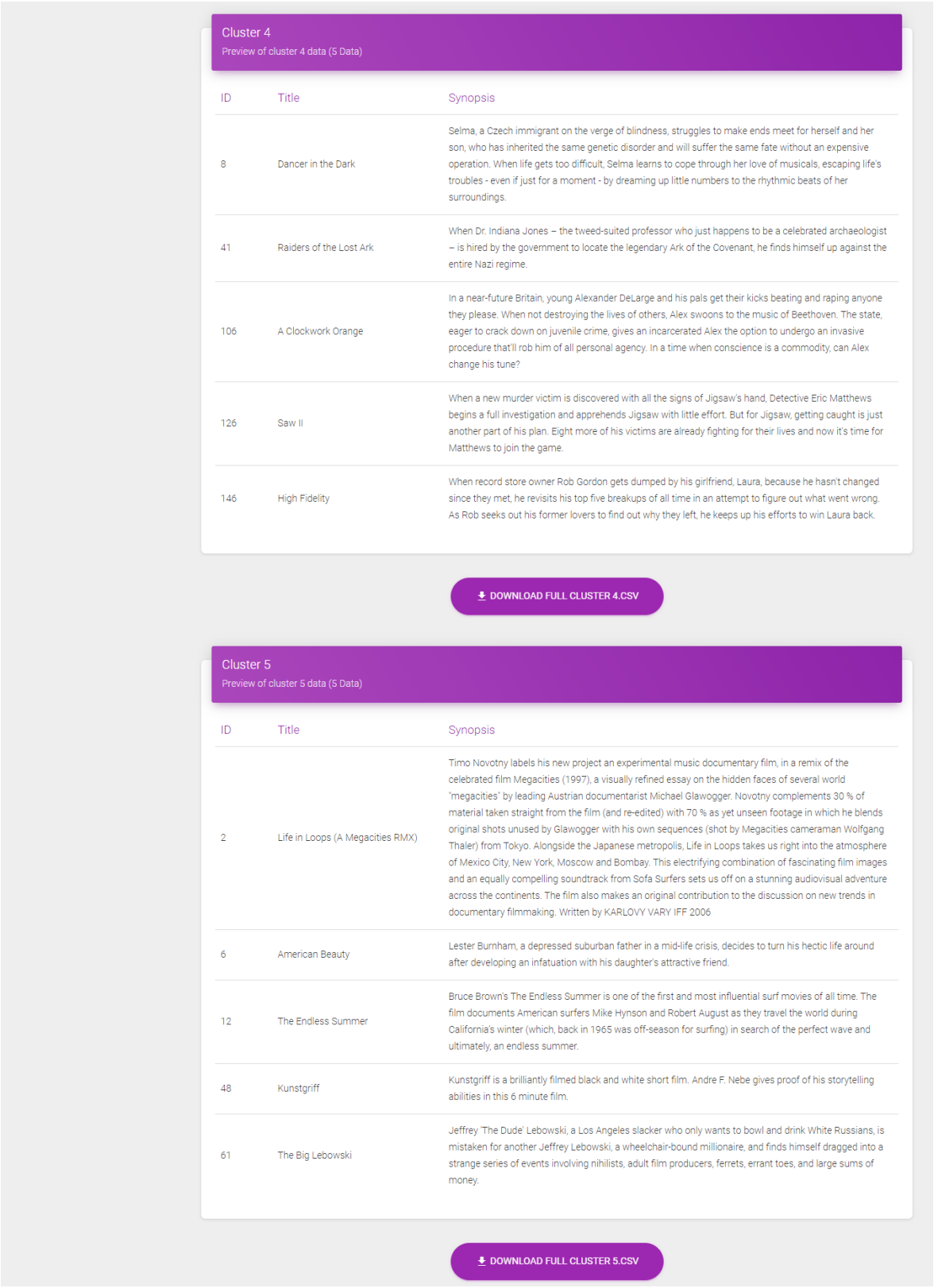
Pada halaman ini akan ditampilkan tabel preview 5 data untuk tiap cluster. Data yang ditampilkan pada masing-masing cluster berupa data id film, judul film, serta sinopsisnya. Pada halaman ini juga disediakan tombol untuk download full data per cluster dengan format .csv. Berikut merupakan tampilan antar muka halaman data per cluster:



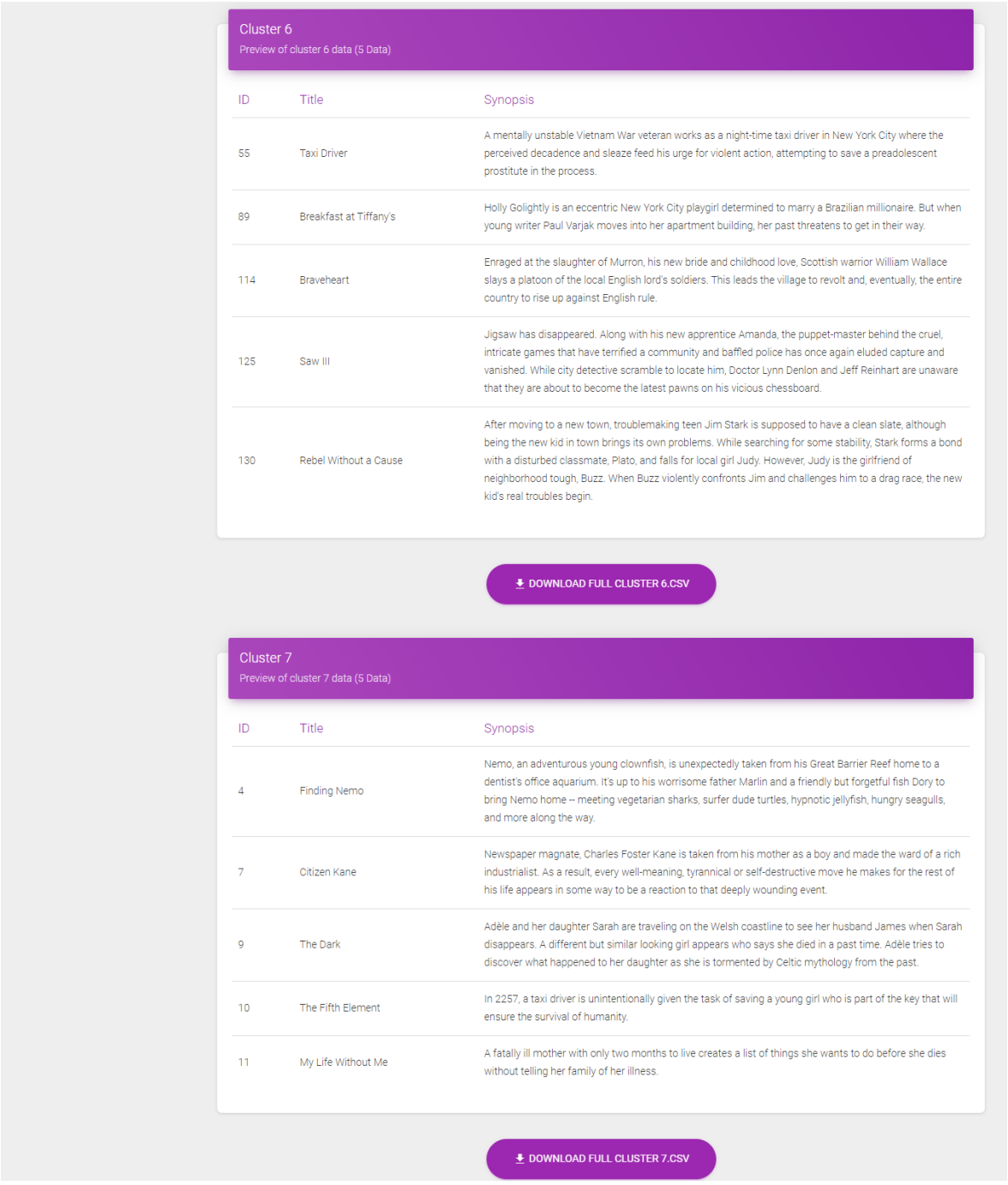
Ganbar 4. 9 Antar Muka Halaman Data Cluster: 0 dan 1



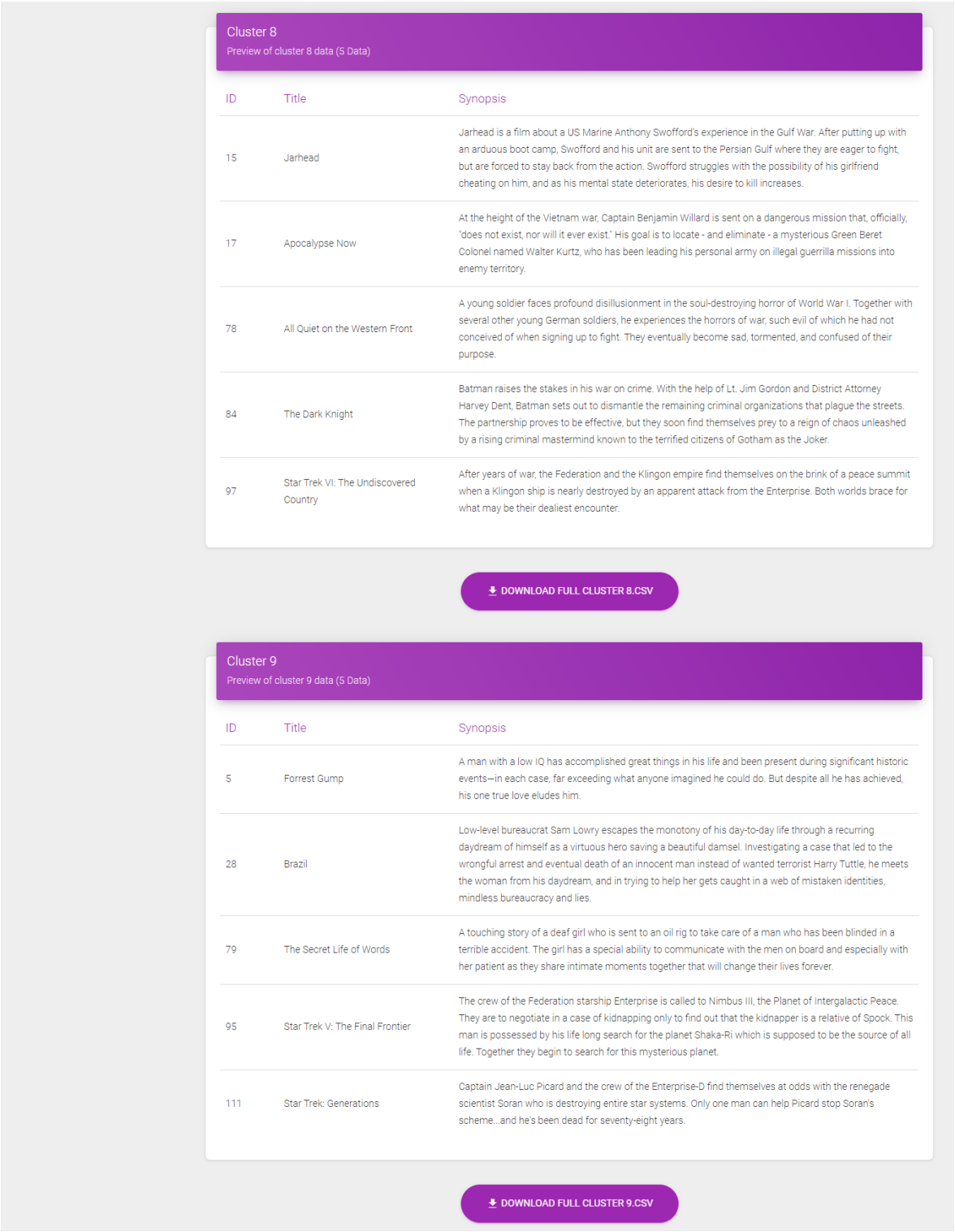
Ganbar 4. 10 Antar Muka Halaman Data Cluster: 2 dan 3



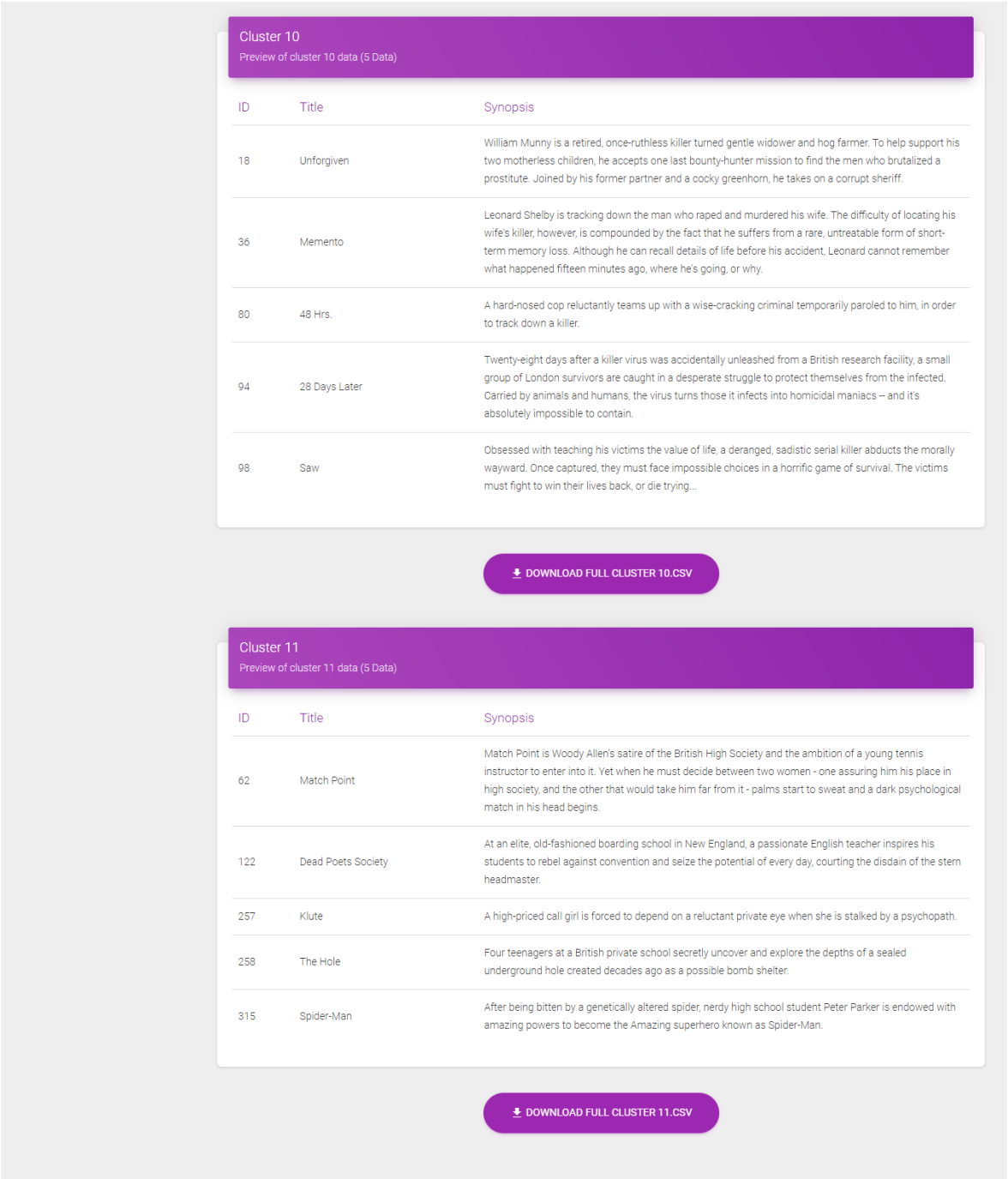
Ganbar 4. 11 Antar Muka Halaman Data Cluster: 4 dan 5



Ganbar 4. 12 Antar Muka Halaman Data Cluster: 6 dan 7



Ganbar 4. 13 Antar Muka Halaman Data Cluster: 8 dan 9

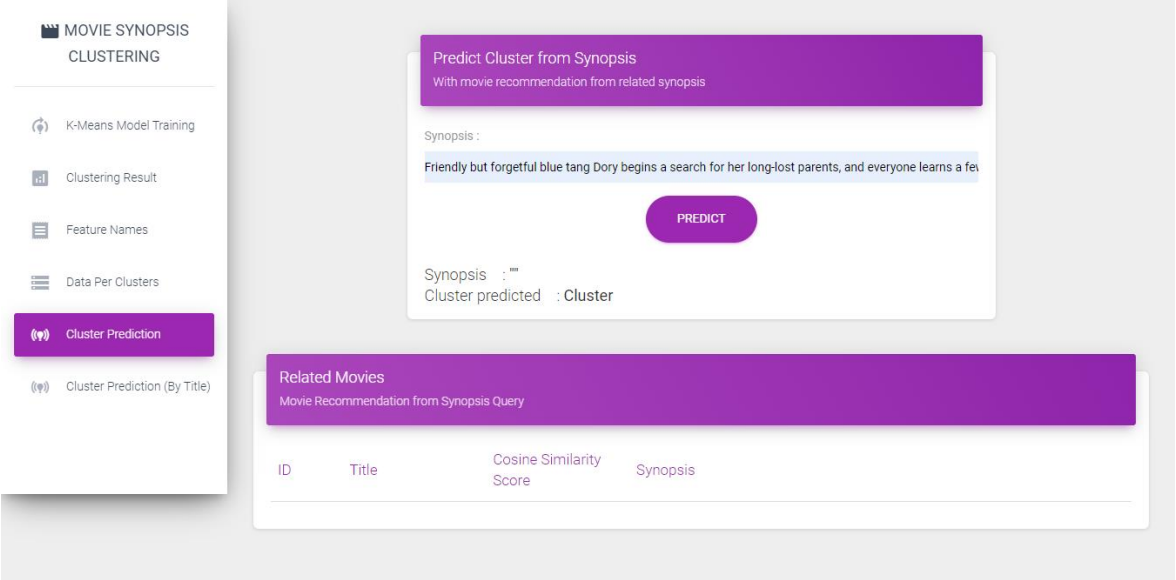


Ganbar 4. 14 Antar Muka Halaman Data Cluster: 10 dan 11

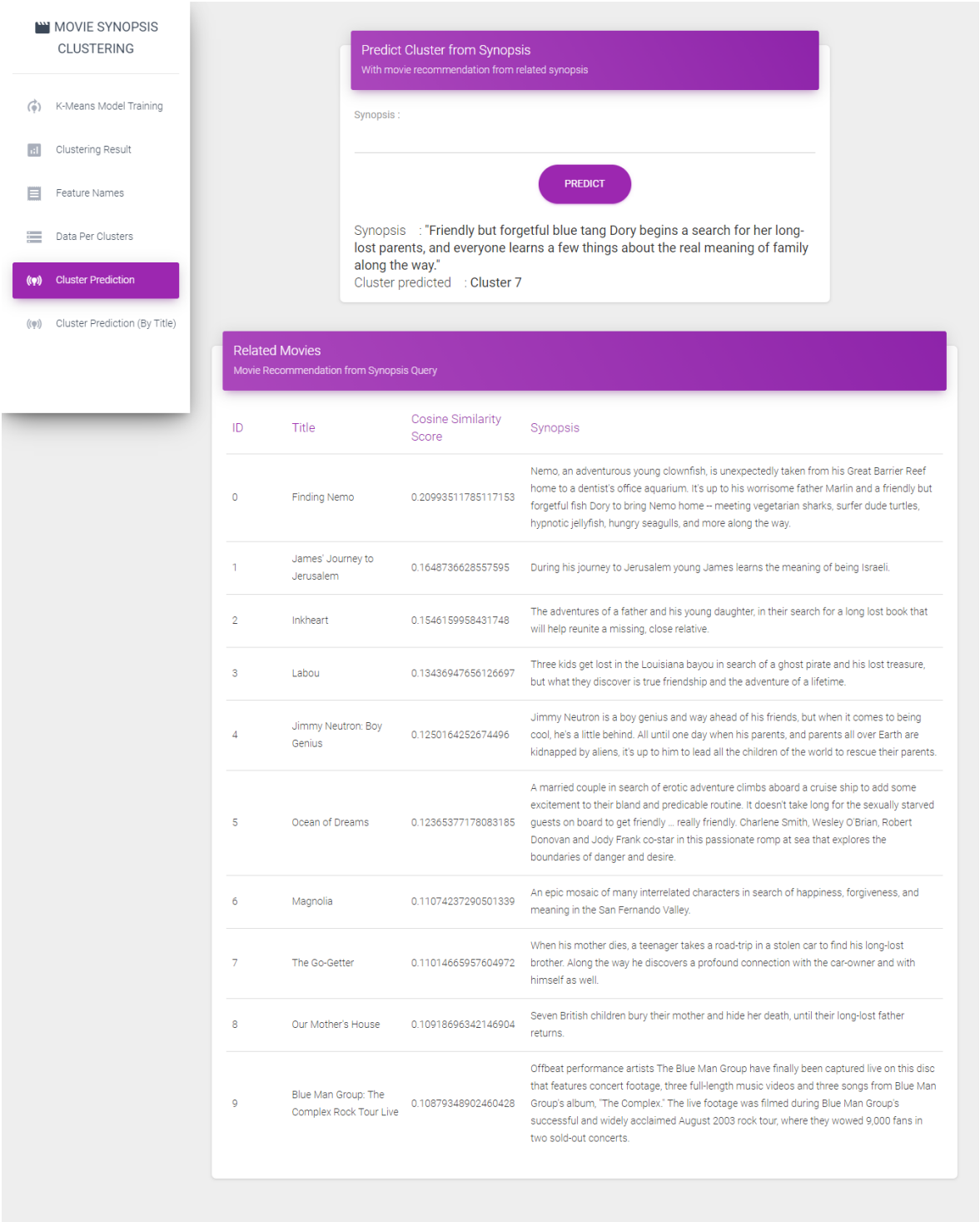
4.1.5. Tampilan Antar Muka Halaman Cluster Prediction

Pada halaman ini pengguna dapat melakukan prediksi cluster dengan memasukkan data sinopsis suatu film. Sebagai contoh sinopsis yang dimasukkan adalah sinposis film Finding Dory, yaitu "After his son is captured in the Great Barrier Reef and taken to Sydney, a timid clownfish sets out on a journey to bring him home". Setelah meginputkan sinopsis tersebut didapatkan prediksi cluster-nya adalah cluster 7.

Pada halaman ini juga menampilkan 10 film related movies, di mana 10 film tersebut merupakan film yang terkait dengan sinopsis yang dimasukkan (ditampilkan juga nilai cosine similarity berurutan dari yang terbesar ke terkecil, semakin besar nilai cosin similarity maka akan semakin mirip dengan sinopsis yang dimasukkan). Berikut merupakan tampilan antar muka halaman cluster prediction:



Ganbar 4. 15 Antar Muka Halaman Cluster Prediction



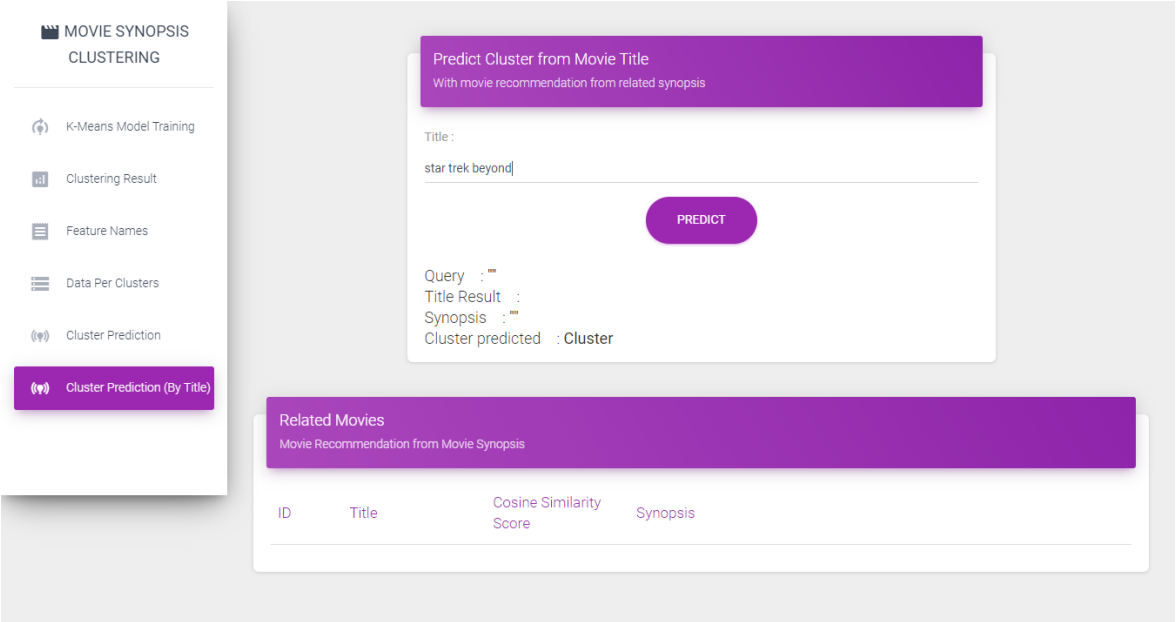
Ganbar 4. 16 Antar Muka Halaman Cluster Prediction: Related Movies

4.1.6. Tampilan Antar Muka Halaman Cluster Prediction (By Title)

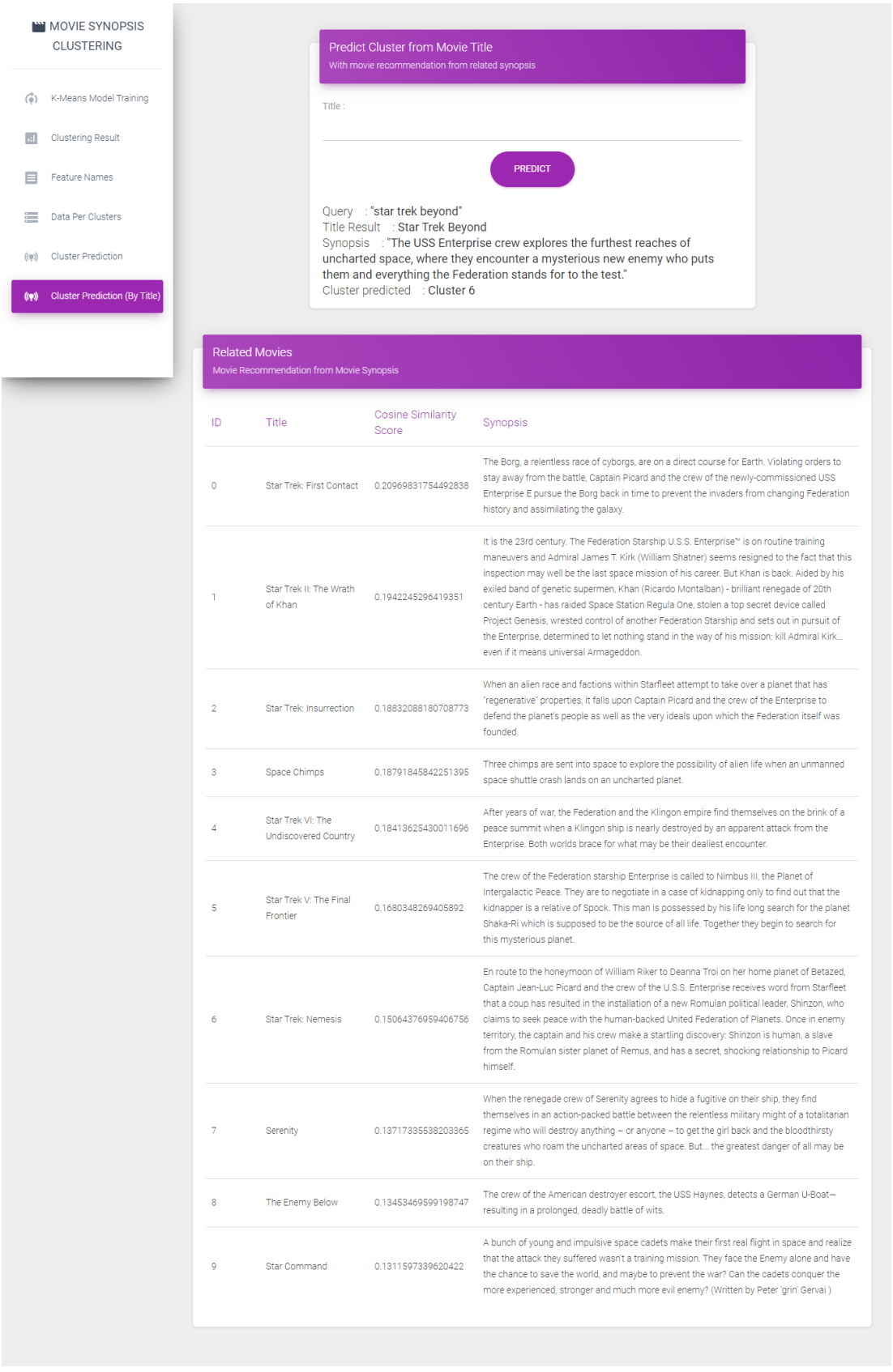
Halaman ini merupakan halaman untuk menentukan prediksi cluster. Namun berbeda dengan halaman sebelumnya yang menentuka prediksi cluster dengan memasukkan sinopsis film sebagai masukkannya, halaman prediksi cluster ini menggunakan judul film sebagai data masukkannya. Sebagai contoh, dimasukkan judul film "star trek beyond" kemudian

didapatkan hasil pencarian judul dan sinopsis dari database. Setelah itu data judul dan sinopsis yang telah ditemukan akan ditampilkan pada halaman aplikasi disertai dengan tampilan prediksi cluster yang didapatkan yaitu cluster 6.

Pada halaman ini juga menampilkan 10 film related movies, di mana 10 film tersebut merupakan film yang terkait dengan judul film yang dimasukkan (ditampilkan juga nilai cosine similarity berurutan dari yang terbesar ke terkecil, semakin besar nilai cosin similarity maka akan semakin mirip dengan sinopsis yang dimasukkan). Berikut merupakan tampilan antar muka halaman cluster prediction (by title):



Ganbar 4. 17 Antar Muka Halaman Cluster Prediction (By Title)



ID	Title	Cosine Similarity Score	Synopsis
0	Star Trek: First Contact	0.20969831754492838	The Borg, a relentless race of cyborgs, are on a direct course for Earth. Violating orders to stay away from the battle, Captain Picard and the crew of the newly-commissioned USS Enterprise E pursue the Borg back in time to prevent the invaders from changing Federation history and assimilating the galaxy.
1	Star Trek II: The Wrath of Khan	0.1942245296419351	It is the 23rd century. The Federation Starship U.S.S. Enterprise™ is on routine training maneuvers and Admiral James T. Kirk (William Shatner) seems resigned to the fact that this inspection may well be the last space mission of his career. But Khan is back. Aided by his exiled band of genetic supermen, Khan (Ricardo Montalban) - brilliant renegade of 20th century Earth - has raided Space Station Regula One, stolen a top secret device called Project Genesis, wrested control of another Federation Starship and sets out in pursuit of the Enterprise, determined to let nothing stand in the way of his mission: kill Admiral Kirk... even if it means universal Armageddon.
2	Star Trek: Insurrection	0.18832088180708773	When an alien race and factions within Starfleet attempt to take over a planet that has "regenerative" properties, it falls upon Captain Picard and the crew of the Enterprise to defend the planet's people as well as the very ideals upon which the Federation itself was founded.
3	Space Chimps	0.18791845842251395	Three chimps are sent into space to explore the possibility of alien life when an unmanned space shuttle crash lands on an uncharted planet.
4	Star Trek VI: The Undiscovered Country	0.18413625430011696	After years of war, the Federation and the Klingon empire find themselves on the brink of a peace summit when a Klingon ship is nearly destroyed by an apparent attack from the Enterprise. Both worlds brace for what may be their dealiest encounter.
5	Star Trek V: The Final Frontier	0.1680348269405892	The crew of the Federation starship Enterprise is called to Nimbus III, the Planet of Intergalactic Peace. They are to negotiate in a case of kidnapping only to find out that the kidnapper is a relative of Spock. This man is possessed by his life long search for the planet Shaka-Ri which is supposed to be the source of all life. Together they begin to search for this mysterious planet.
6	Star Trek: Nemesis	0.15064376959406756	En route to the honeymoon of William Riker to Deanna Troi on her home planet of Betazed, Captain Jean-Luc Picard and the crew of the U.S.S. Enterprise receives word from Starfleet that a coup has resulted in the installation of a new Romulan political leader, Shinzon, who claims to seek peace with the human-backed United Federation of Planets. Once in enemy territory, the captain and his crew make a startling discovery: Shinzon is human, a slave from the Romulan sister planet of Remus, and has a secret, shocking relationship to Picard himself.
7	Serenity	0.13717335538203365	When the renegade crew of Serenity agrees to hide a fugitive on their ship, they find themselves in an action-packed battle between the relentless military might of a totalitarian regime who will destroy anything – or anyone – to get the girl back and the bloodthirsty creatures who roam the uncharted areas of space. But... the greatest danger of all may be on their ship.
8	The Enemy Below	0.13453469599198747	The crew of the American destroyer escort, the USS Haynes, detects a German U-Boat—resulting in a prolonged, deadly battle of wits.
9	Star Command	0.1311597339620422	A bunch of young and impulsive space cadets make their first real flight in space and realize that the attack they suffered wasn't a training mission. They face the Enemy alone and have the chance to save the world, and maybe to prevent the war? Can the cadets conquer the more experienced, stronger and much more evil enemy? (Written by Peter 'grin' Gervai)

Ganbar 4. 18 Antar Muka Halaman Cluster Prediction (By Title): Related Movies

4.2. Hasil Penelitian

Hasil penelitian berupa aplikasi text clustering untuk menentukan genre film berdasarkan sinopsisnya. Aplikasi text clustering ini dibuat dalam bentuk website sehingga dapat mempermudah user untuk menggunakannya. Aplikasi dibuat menggunakan bahasa pemrograman Python dan untuk halaman antar muka dibuat menggunakan bahasa pemrograman HTML.

Sesuai dengan fungsinya aplikasi ini memiliki enam halaman utama sebagai halaman fungsionalnya. Keenam halaman tersebut yaitu:

1. Halaman K-Means Model Training
2. Halaman Clustering Result
3. Halaman Feature Names
4. Halaman Data Per Cluster
5. Halaman Cluster Prediction
6. Halalaman Cluster Prediction (By Title)

Aplikasi ini memberikan informasi mengenai clustering film berdasarkan sinopsisnya dan menampilkan hasil clustering tersebut. Aplikasi ini juga akan menampilkan evaluasi hasil clustering yang telah dilakukan serta dapat menampilkan visualisasi hasil clustering dalam bentuk visualisasi 2 dimensi dan 3 dimensi.

4.3. Analisa Hasil

Berdasarkan dari hasil penelitian yang telah dijelaskan pada bagian sebelumnya, Aplikasi Clustering Film Berdasrkan Sinopsisnya sudah dapat berjalan dengan baik sesuai dengan fungsionalitasnya. Clustering yang dihasilkan sudah sesuai dengan jumlah cluster yang dimasukkan oleh user pada halaman K-Means Model Training. Pada halaman clustering result Aplikasi Clustering Film Berdasrkan Sinopsisnya juga sudah menampilkan hasil cluster yang sesuai. Hasil evaluasi cluster yang ditampilkan pada halaman clustering result juga sudah sesuai, juga dengan tampilan visualisasi data hasil clustering yang ditampilkan sudah cukup baik.

Pada halaman feature names Aplikasi Clustering Film Berdasrkan Sinopsisnya, nama-nama fitur yang ditampilkan juga sudah sesuai dengan genre film yang terkait. Halaman data per cluster yang ditampilkan pada Aplikasi Clustering Film Berdasrkan Sinopsisnya juga sudah dapat menampilkan data judul film dan sinopsisnya sesuai dengan cluster masing-masing.

Fungsi yang ada di halaman cluster prediction pada Aplikasi Clustering Film Berdasrkan Sinopsisnya juga sudah berjalan dengan baik. Hal ini dibuktikan dengan hasil prediksi cluster yang dihasilkan pada Aplikasi Clustering Film Berdasrkan Sinopsisnya, baik dengan masukan berupa sinopsis film maupun dengan masukan judul film. Selain itu, 10 Film yang ditampilkan sebagai related movies juga sudah sesuai dengan film yang terkait.

BAB V
PENUTUP

Bab ini merupakan penutup yang berisi kesimpulan dari pembuatan Aplikasi Text Clustering untuk Menentukan Genre Film Berdasarkan Sinopsisnya.

5.1. Kesimpulan

Aplikasi Clustering Film Berdasarkan Sinopsisnya merupakan text clustering untuk menentukan genre film berdasarkan sinopsisnya. Aplikasi ini dibuat dalam bentuk website dengan menggunakan bahasa pemrograman Python dan untuk halaman antar muka dibuat menggunakan bahasa pemrograman HTML. Data yang digunakan dalam aplikasi ini menggunakan data judul film dan sinopsis dari website https://www.themoviedb.org/.

Berdasarkan fungsionalitasnya aplikasi ini memiliki enam halaman utama sebagai halaman fungsionalnya, yaitu halaman K-Means Model Training, halaman Clustering Result, halaman Feature Names, halaman Data Per Cluster, halaman Cluster Prediction, halalaman Cluster Prediction (By Title). Aplikasi Clustering Film Berdasarkan Sinopsisnya menampilkan hasil clustering memberikan informasi mengenai clustering film berdasarkan sinopsisnya sesuai dengan jumlah cluster yang diinputkan oleh user. Aplikasi ini juga akan menampilkan evaluasi hasil clustering yang telah dilakukan dengan menggunakan Evaluasi Silhouette dan Elbow Method, serta dapat menampilkan visualisasi hasil clustering dalam bentuk visualisasi 2 dimensi dan 3 dimensi.

Sejauh ini Aplikasi Clustering Film Berdasarkan Sinopsisnya sudah berjalan dengan baik. Semua fitur yang ada pada Aplikasi Clustering Film Berdasarkan Sinopsisnya juga sudah bekerja sesuai dengan fungsinya. Penulis berharap Aplikasi Clustering Film Berdasarkan Sinopsisnya ini dapat tetap berjalan dengan baik sehingga dapat menghasilkan cluster yang sesuai dengan inputan yang dimasukan user dan dapat memberikan informasi sesuai dengan apa yang diharapkan oleh user.

DAFTAR PUSTAKA

Adriani, Mirna.2016. *Pemrosesan Tek*s. Jakarta: Fasilkom, Universitas Indonesia.

Bishop, Christopher M. 2006. Pattern Recognition and Machine Learning. Springer.

Jain, Anil K. 1988. *Algorithms for Clustering Data*. Prentice Hall.

Nedjah, Nadia, et all. 2009. *Intelligent Text Categorization and Clustering*. Springer.

LAMPIRAN

1. Source Code Program

Import Dependencies

```
import json
import requests
import os, shutil
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import pickle
from google.colab import drive
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
from wordcloud import WordCloud
from    sklearn.metrics.pairwise    import    cosine_similarity,
euclidean_distances
from sklearn.decomposition import PCA
from sklearn.manifold import TSNE

drive.mount('/content/gdrive/')
path = "/content/gdrive/MyDrive/Upload/Akademik/Tugas Besar PBA"
path_api = "/content/gdrive/MyDrive/Upload/API/"
```

Data Collecting

Retrieve Genres

```
# Get All Genres

genres = []
for i in range(20000) :
  url = ("https://api.themoviedb.org/3/movie/%d?api_key=63fea4c7
09da1f1496b7a1ca7a3c6083" % i)
  r = requests.get(url)
  json_data = json.loads(r.text)
  try:
    if (json_data['genres'] != "") :
```

```
      # print(json_data['genres'])
      for j in json_data['genres'] :
        genre = j.get('name')
        if (genre not in genres) :
          genres.append(genre)
  except Exception:
    pass
print(genres)
len(genres)
```

Retrieve Titles & Synopsis

```
titles = []
synopsis = []

for i in range(20000) :
  url = ("https://api.themoviedb.org/3/movie/%d?api_key=63fea4c7
09da1f1496b7a1ca7a3c6083" % i)
  r = requests.get(url)
  json_data = json.loads(r.text)
  try:
    if (json_data['overview'] != "" and json_data['overview'] !=
 "No overview found." and json_data['original_language'] == 'en'
) :
      titles.append(json_data['title'])
      synopsis.append(json_data['overview'])
  except Exception:
    pass
import pandas as pd

df_movies = pd.DataFrame({'title': titles, 'synopsis': synopsis}
)
df_movies.head()
df_movies.info()
with open('/content/gdrive/MyDrive/Upload/movie_synopsis.csv', '
w') as f:
  df_movies.to_csv(f)
```

Data Preprocessing

Import Data

```
from google.colab import drive

drive.mount('/content/gdrive/')

import pandas as pd
```

```
df_movies = pd.read_csv('/content/gdrive/MyDrive/Upload/Akademik
/Tugas Besar PBA/movie_synopsis_8000_no overview.csv', linetermi
nator='\n')
df_movies.head()
df_movies.info()
```

Delete Missing Value

```
# Get names of indexes for which column synopsis has no overview
index_drop = df_movies[df_movies['synopsis'] == "No overview fou
nd."].index

# Delete these row indexes from dataFrame
df_movies.drop(index_drop , inplace=True)

# Reset the index
df_movies.reset_index(drop =True, inplace=True)


df_movies.info()
```

Case Folding

```
df_movies["synopsis"] = df_movies["synopsis"].str.lower()
df_movies.head()
```

TF-IDF Training

```
from sklearn.feature_extraction.text import TfidfVectorizer

vectorizer = TfidfVectorizer(stop_words='english')
features = vectorizer.fit_transform(df_movies['synopsis'].values
) # training vector TF-IDF pada tiap data

# vector TF-
IDF tiap kalimat (features), pasangan index kata dan nilainya
for i in features :
  print(i)
  print('-----------------------------')


import pickle

# Save Model
pickle.dump(vectorizer, open('/content/gdrive/MyDrive/Upload/Aka
demik/Tugas Besar PBA/vectorizer.sav', 'wb'))


import pickle
```

```
# Load Model
vectorizer = pickle.load(open('/content/gdrive/MyDrive/Upload/Ak
ademik/Tugas Besar PBA/vectorizer.sav', 'rb'))
```

KMeans Model

KMeans Model Training

```
from sklearn.cluster import KMeans

k = 19 # jumlah cluster
kmeans_model = KMeans(n_clusters = k, n_init = 3, max_iter = 500
)
synopsis_clusters = kmeans_model.fit(features)


import pickle

# Save Model
pickle.dump(kmeans_model, open('/content/gdrive/MyDrive/Upload/A
kademik/Tugas Besar PBA/kmeans_model.sav', 'wb'))


import pickle

# Load Model
kmeans_model = pickle.load(open('/content/gdrive/MyDrive/Upload/
Akademik/Tugas Besar PBA/kmeans_model.sav', 'rb'))
```

Data Labelling

```
df_movies['label'] = kmeans_model.labels_
df_movies.head()


with open('/content/gdrive/MyDrive/Upload/movie_synopsis_labeled
.csv', 'w') as f:
  df_movies.to_csv(f)
```

Data Distribution

```
movie_clusters = df_movies.groupby('label')


clusters_count = []
for cluster in movie_clusters :
    clusters_count.append(len(cluster[1]))
```

```
plt.figure(figsize=(14, 7))
sns.set_theme(style="whitegrid")
graph = sns.barplot(x = np.arange(k), y = clusters_count,
palette='hls')
```

Data from each Cluster

Data Preview for each Cluster

```
clusters = df_movies.groupby('label')

for cluster in clusters.groups :
  print("Cluster %d : " % cluster)
  data_cluster = clusters.get_group(cluster)[['title','synopsis'
]]
  for i in range(5) :
    data = data_cluster.iloc[i]
    print('   ', data['title'], ': ', data['synopsis'])
  print('\n')
```

Save All Data from each Cluster

```
clusters = df_movies.groupby('label')

for cluster in clusters.groups :
    f = open('cluster'+str(cluster)+ '.csv', 'w') # buat file cs
v untuk tiap cluster
    data = clusters.get_group(cluster)[['title','synopsis']] # j
udul dan sinposis tiap data pada tiap cluster
    f.write(data.to_csv(index_label='id')) # simpan ke csv
    f.close()
```

Feature Names of each Cluster

```
order_centroids = kmeans_model.cluster_centers_.argsort()[:, ::-
1] # diurutkan berdasarkan indeks -> lalu di-reversed
terms = vectorizer.get_feature_names()
n_terms = 10

for i in range(k) :
  print("Cluster %d :" % i)
  for j in order_centroids [i, :n_terms] :
    print('   %s' % terms[j])
  print('----------')


from wordcloud import WordCloud
import matplotlib.pyplot as plt
```

```
c = []
keywords = []
for i in range(k) :
  c.append(i)
  key = ""
  for j in order_centroids [i, :n_terms] :
    key = key+(" ")+(terms[j])
  keywords.append(key)

for i in range(k) :
  print('Cluster: %d' % c[i])
  text = keywords[i]
  wordcloud = WordCloud(max_font_size=50, background_color="whit
e").generate(text)
  plt.figure()
  plt.imshow(wordcloud, interpolation="bilinear")
  plt.axis("off")
  plt.show()
```

Cluster Prediction and Recomendation

```
def predict_cluster(sentence) :
  Y = vectorizer.transform([sentence])
  prediction = kmeans_model.predict(Y)
  cluster_prediction = prediction[0]
  print("Sentence           : ", sentence)
  print("Cluster predicted  : ", cluster_prediction)

def movie_recommendation(n, query, features, vectorizer, df_movi
es) :
  query = [query.lower()]
  query = vectorizer.transform(query)

  cosine_score = []
  index = 0
  for i in features :
    cosine_score.append(cosine_similarity(query, i))
    index +=1

  cosine_score_update = []
  for i in cosine_score :
    cosine_score_update.append(i[0][0])
  cosine_score_update = np.array(cosine_score_update)
  cosine_score_update
  indices = np.argsort(cosine_score_update)[::-1]

  for i in range(n) :
```

```
    index_now = indices[i]
    if (cosine_score_update[index_now] != 0) :
      print(i, " . ", df_movies['title'][index_now], " : ", cosi
ne_score_update[index_now], " : ", df_movies['synopsis'][index_n
ow])

def movie_recommendation_by_title(n, title, features, vectorizer
, df_movies) :
  index = 0
  movie_index = 0
  for i in df_movies['title'] :
    if (i == title) :
      movie_index = index
      break
    index +=1

  synopsis_ori = ""
  title_new = ""
  url = ("https://api.themoviedb.org/3/search/movie?api_key=63fe
a4c709da1f1496b7a1ca7a3c6083&language=en-
US&query=%s&page=1&include_adult=false" % title)
  r = requests.get(url)
  json_data = json.loads(r.text)
  try:
    if (json_data['results'][0]['overview'] != "") :
      title_new = json_data['results'][0]['title']
      synopsis_ori = json_data['results'][0]['overview']
  except Exception:
    pass

  synopsis = [synopsis_ori.lower()]
  synopsis = vectorizer.transform(synopsis)

  cosine_score = []
  index = 0
  for i in features :
    cosine_score.append(cosine_similarity(synopsis, i))
    index +=1

  cosine_score_update = []
  for i in cosine_score :
    cosine_score_update.append(i[0][0])
  cosine_score_update = np.array(cosine_score_update)
  cosine_score_update
  indices = np.argsort(cosine_score_update)[::-1]

  print("Title Query : ", title)
  print(title_new, " : ", synopsis_ori, "\n")
```

```
  for i in range(n) :
    index_now = indices[i]
    if (cosine_score_update[index_now] != 0) :
      print(i, " . ", df_movies['title'][index_now], " : ", cosi
ne_score_update[index_now], " : ", df_movies['synopsis'][index_n
ow])
```

Predict Sentences

```
sentence = "Britain declared war on Germany in september 1939 an
d begin world war 2"

predict_cluster(sentence)


sentence = "There are a lot of aliens at space"

predict_cluster(sentence)


sentence = "Johnstone's Family is very nice family"

predict_cluster(sentence)
```

Movie Recommendation (by Query/ Synopsis)

```
n = 10
query = "Johnstone's Family is very nice family"

print("Synopsis : ", query, "\n")
movie_recommendation(n, query, features, vectorizer, df_movies)
```

Movie Recommendation (by Title)

```
n = 10
title = "Star Trek Beyond"

movie_recommendation_by_title(n, title, features, vectorizer, df
_movies)
```

Evaluation

Elbow Method (SSE)

```
SSE = []
K = range(3,30)
for n_k in K:
  kmeans_model_iteration = KMeans(n_clusters = n_k).fit(features
)
  SSE.append(kmeans_model_iteration.inertia_)
```

```
import matplotlib.pyplot as plt

plt.plot(K, SSE, 'bx-')
plt.xlabel('k')
plt.ylabel('SSE')
plt.title('The Elbow Method showing the optimal k')
plt.show()
```

Silhouette Score

```
from sklearn.metrics import silhouette_score

# nilai silhouette score antara -
1 dan 1, semakin tinggil semakin bagus
print(f'Silhouette Score : {silhouette_score(features, labels =
kmeans_model.labels_)}')
```

Data Visualization

2 Dimensi

PCA Dimensionality Reduction

```
from sklearn.decomposition import PCA

pca = PCA(n_components = 2)
reduced_features = pca.fit_transform(features.toarray())
reduced_cluster_centers = pca.transform(synopsis_clusters.cluste
r_centers_)


import matplotlib.pyplot as plt

plt.figure(figsize=(30, 15), dpi=80)
plt.scatter(reduced_features[:, 0], reduced_features[:,1], c = k
means_model.predict(features))
plt.scatter(reduced_cluster_centers[:, 0], reduced_cluster_cente
rs[:,1], marker='x', s=150, c='b')
```

t-SNE Dimensionality Reduction

```
from sklearn.manifold import TSNE

tsne = TSNE(n_components=2)
reduced_features_tsne = tsne.fit_transform(features.toarray())


import pickle
```

```
# Save Model
pickle.dump(tsne, open('/content/gdrive/MyDrive/Upload/Akademik/
Tugas Besar PBA/tsne.sav', 'wb'))


import matplotlib.pyplot as plt

plt.figure(figsize=(30, 15), dpi=80)
plt.scatter(reduced_features_tsne[:, 0], reduced_features_tsne[:
,1], c = kmeans_model.predict(features))
```

3 Dimensi

PCA Dimensionality Reduction

```
from sklearn.decomposition import PCA

pca_3d = PCA(n_components = 3)
reduced_features_3d = pca_3d.fit_transform(features.toarray())
reduced_cluster_centers_3d = pca_3d.transform(synopsis_clusters.
cluster_centers_)


%matplotlib notebook
%matplotlib inline
from mpl_toolkits.mplot3d import Axes3D

sns.set(style = "darkgrid")

fig = plt.figure(figsize=(30, 15), dpi=80)
ax = fig.add_subplot(111, projection = '3d')

x = reduced_features_3d[:, 0]
y = reduced_features_3d[:, 1]
z = reduced_features_3d[:, 2]

ax.scatter(x, y, z, c = synopsis_clusters.labels_)
ax.scatter(reduced_cluster_centers_3d[:, 0], reduced_cluster_cen
ters_3d[:,1], reduced_cluster_centers_3d[:,2], marker='x', s=500
, c='b')

plt.show()
```

t-SNE Dimensionality Reduction

```
from sklearn.manifold import TSNE

tsne_3d = TSNE(n_components=3)
```

```
reduced_features_3d_tsne = tsne_3d.fit_transform(features.toarra
y())


import pickle

# Save Model
pickle.dump(tsne_3d, open('/content/gdrive/MyDrive/Upload/Akadem
ik/Tugas Besar PBA/tsne_3d.sav', 'wb'))


import matplotlib.pyplot as plt

from mpl_toolkits.mplot3d import Axes3D

sns.set(style = "darkgrid")

fig = plt.figure(figsize=(30, 20), dpi=80)
ax = fig.add_subplot(111, projection = '3d')

x = reduced_features_3d_tsne[:, 0]
y = reduced_features_3d_tsne[:, 1]
z = reduced_features_3d_tsne[:, 2]

ax.scatter(x, y, z, c = synopsis_clusters.labels_)

plt.show()
```