BAB II TINJAUAN PUSTAKA

2.1 Sistem Rekomendasi

Sistem rekomendasi merupakan fitur yang banyak digunakan pada perangkat lunak zaman sekarang. Sistem rekomendasi merupakan salah satu fitur software yang sangat berguna bagi pengguna. Beberapa pengertian sistem rekomendasi diuraikan berikut ini :

1. Sistem rekomendasi adalah sistem program yang mencoba untuk merekomendasikan produk atau jasa yang paling cocok untuk pengguna tertentu.
2. Sistem rekomendasi bisa juga diartikan sebagai mesin keputusan otomatis yang mengevaluasi kesamaan antara orang (yaitu “pengguna”) dan atau item untuk membuat rekomendasi tentang item apa yang cocok bersama.
3. Sistem rekomendasi juga dapat ditafsirkan sebagai sistem yang memprediksi informasi yang mungkin diminati oleh pengguna dan mendukung pengambilan keputusan pengguna.
4. Pengertian sistem rekomendasi yang lainnya yaitu algoritma untuk menyaring dan menyortir item dan informasi. Hal ini menggunakan opini komunitas pengguna untuk membantu individu dalam komunitas tersebut menemukan konten yang menarik dan relevan dari serangkaian pilihan yang berpotensi berlebihan.
5. Sistem pemberi rekomendasi adalah sistem yang mencoba merekomendasikan item (produk atau layanan) yang paling sesuai kepada pengguna tertentu (individu atau perusahaan) dengan menentukan minat pengguna terhadap item tersebut berdasarkan item, pengguna, dan informasi yang relevan tentang item tersebut. sebagai program yang memprediksi interaksi antara elemen dan pengguna.

Sistem Rekomendasi dilakukan secara individu atau bisnis dengan memprediksi minat pengguna terhadap produk [9]. Pengembangan sistem rekomendasi bertujuan untuk mengurangi informasi yang berlebihan dengan

mengambil informasi yang paling relevan dari sejumlah besar data, sehingga mampu memberikan layanan dengan baik [20]. Selain itu sistem rekomendasi juga memberikan manfaat bagi kedua belah pihak yaitu perusahaan sebagai penyedia layanan dan pengguna yang menggunakan layanan tersebut. Sistem rekomendasi juga memberikan kemudahan komunikasi dan transaksi antara perusahaan dan pengguna. Selain itu sistem rekomendasi terbukti mampu meningkatkan kualitas pengambilan keputusan. Misalnya sistem rekomendasi e-commerce yang mampu meningkatkan pendapatan, karena merupakan cara yang efektif untuk menjual banyak produk dengan jangkauan pasar yang lebih luas [21].

Pengembangan sistem rekomendasi berawal dari observasi yang sederhana, bahwa hampir setiap individu bergantung pada rekomendasi yang diberikan oleh orang lain dalam menentukan pilihan yang dijumpai pada aktifitas sehari-hari. Contohnya untuk menentukan film yang hendak ditonton, buku yang hendak dibaca, tempat makan yang layak untuk dikunjungi, dan hal-hal lainnya [4].

Gagasan memanfaatkan komputer untuk merekomendasikan item terbaik bagi pengguna telah ada sejak awal komputasi. Implementasi pertama dari konsep sistem rekomendasi muncul pada tahun 1979, dalam sebuah sistem bernama Grundy, seorang pustakawan berbasis komputer yang memberikan saran kepada pengguna tentang buku apa yang harus dibaca. Sistem ini cukup primitif karena pengelompokan data pengguna berdasarkan hasil dari wawancara dan *hard-code* dari sistem untuk mendapatkan daftar rekomendasi buku. Hal tersebut merupakan awal yang baik bagi ruang sistem rekomendasi. Ini adalah cikal bakal terbentuknya sistem rekomendasi otomatis.

Selanjutnya pada awal 1990-an terjadi peluncuran Tapestry, sistem rekomendasi komersial pertama. Implementasi sistem rekomendasi lainnya untuk membantu orang menemukan artikel pilihan mereka diluncurkan pada awal 1990-an oleh GroupLens, sebuah laboratorium penelitian di University of

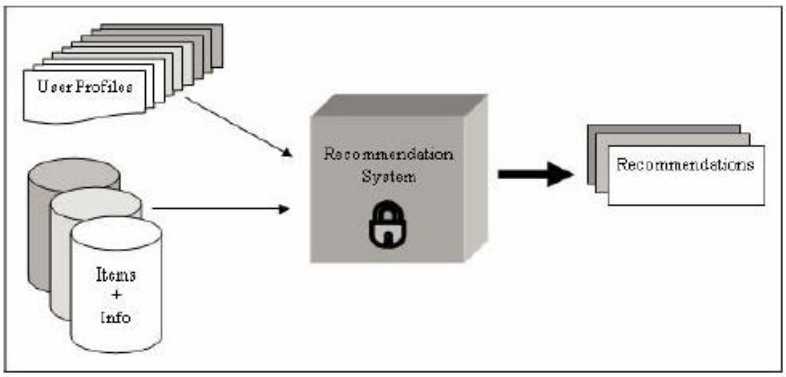
Minnesota, AS. Mereka menamai sistem itu dengan nama *GroupLens Recommender System*. Sistem ini mengklaim memiliki semangat yang mirip dengan Tapestry, Ringo, BellCore dan Jester. Pengembangan sistem rekomendasi lebih lanjut pada akhir 1990-an adalah implementasi Amazon Collaborative Filtering, salah satu teknologi sistem rekomendasi yang paling banyak dikenal.

Sistem rekomendasi merupakan sumber pendapatan penting bagi banyak bisnis dan digunakan di berbagai industri seperti ritel, berita, dan media. Istilah lainnya adalah *Million Dollar Investment*. Tujuan dari implementasi sistem rekomendasi ini dalam industri tersebut adalah menampilkan list item / produk yang relevan, terbarukan / *novelty*, kebetulan / *serendipity* dan beragam / *diversity*. Dengan pertumbuhan data yang cepat dan masif di Internet, peran sistem pemberi rekomendasi menjadi semakin penting. Dengan *proliferasi* data ini, memfilter informasi yang berguna / bermanfaat secara pribadi menjadi bagian penting. Peran sistem rekomendasi di sini adalah menyaring rekomendasi yang sesuai.

Keuntungan menggunakan sistem rekomendasi untuk penyedia layanan adalah rekomendasi biasanya mempercepat penelusuran, memudahkan pengguna mengakses konten yang menarik, dan mengejutkan mereka dengan penawaran yang tidak akan pernah mereka cari. Selain itu, perusahaan dapat menarik dan mempertahankan pelanggan dengan mengirimkan email berisi tautan ke penawaran baru yang sesuai dengan minat penerima, atau saran untuk film dan acara TV yang sesuai dengan profil mereka. Pengguna merasa dikenal dan dipahami serta membeli lebih banyak produk atau mengonsumsi lebih banyak konten. Dengan mengetahui apa yang diinginkan pengguna, perusahaan mendapatkan keunggulan kompetitif dan mengurangi risiko kehilangan pelanggan ke pesaing. Menambahkan nilai kepada pengguna dengan memasukkan rekomendasi ke dalam sistem dan produk sangat menarik. Selain itu, ini memungkinkan perusahaan untuk tetap berada di depan pesaing mereka dan pada akhirnya meningkatkan penjualan mereka.

Beberapa perusahaan e-commerce dunia / internasional yang menggunakan sistem rekomendasi untuk menunjang bisnis mereka di antaranya adalah Netflix, Amazon, YouTube, Facebook, Google, MovieLens, Last.fm, Alibaba, eBay, dan lain lain. Beberapa perusahaan nasional yang menggunakan sistem rekomendasi yaitu : gojek, telkomsel, traveloka dan ovo (https://bigbox.co.id/blog/7-perusahaan-ini-sukses-menerapkan-solusi-big-data-untuk-bisnisnya-2/ diakses 1 November 2022).

Namun, sistem rekomendasi tidak hanya digunakan untuk menjual barang dan membeli. Tetapi juga memiliki beberapa hal, seperti rekomendasi pertemanan, rekomendasi sewa hotel, dan rekomendasi objek wisata. Sistem rekomendasi akan mengevaluasi informasi yang menarik bagi pengguna dan membantu mereka dalam membuat pilihan. Ilustrasi dari sistem rekomendasi dapat dilihat pada gambar 2.1 berikut ini :

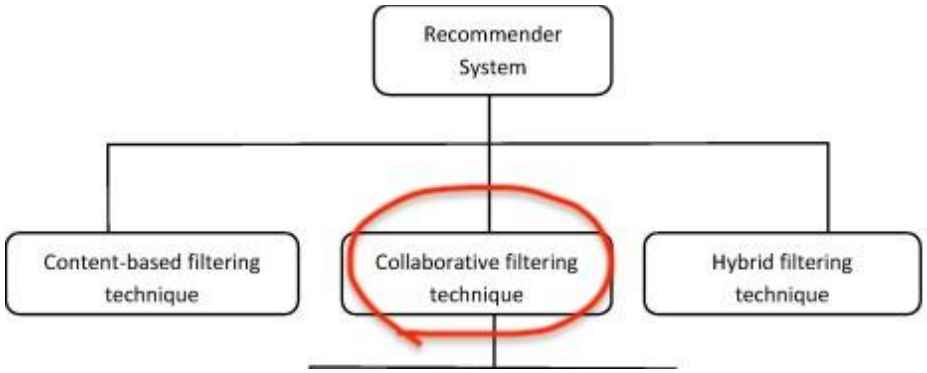


Gambar 2.1 Proses rekomendasi

Sumber : https://mti.binus.ac.id/2020/11/17/sistem-rekomendasi-content-based/ (diakses, 18 September 2022)

Untuk sistem rekomendasi audio atau video, informasi ini dapat berupa database berdimensi besar. Pada Gambar 2.1 dapat kita lihat bahwa produk akhir dari sistem akan berupa kumpulan rekomendasi kepada pengguna. Tampilan akhir dari rekomendasi ini tergantung pada sistem itu sendiri, tetapi dapat mencakup daftar item yang dipesan, tangkapan item, atau seluruh item. teknik rekomendasi berbasis konten adalah kasus khusus dari teknik berbasis informasi. Berdasarkan cara tersebut, pengguna berusaha membuat prediksi

berdasarkan analisis item atau *metadata* yang saling terkait. Teknik ini memungkinkan pengguna mengetahui semua item yang digunakan oleh pengguna di masa lampau dan dapat dijadikan sebagai landasan untuk menentukan seberapa mirip preferensi mereka dengan item yang ada saat ini. Untuk item yang memiliki kemiripan pola peringkat lama dapat digunakan untuk menghitung peringkat terhadap item yang baru. Ilustrasi dari sistem rekomendasi dan turunannya dapat dilihat pada gambar 2.2.



Gambar 2.2 Sistem rekomendasi dan turunannya
Sumber : https://blog.ariflaksito.net/2021/07/memahami-collaborative-filtering-di.html (diakses 13 Oktober 2022)

Metode ini merupakan salah satu teknik yang digunakan untuk menganalisis elemen dan melakukan ekstraksi ciri yang dapat menjelaskannya. Setelah fitur diekstraksi, sistem secara otomatis mencari item yang mirip dengan preferensi pengguna. Karena teknik ini bergantung pada analisis isi, kualitas data yang tersedia merupakan faktor kunci dalam kualitas hasil yang dihasilkan oleh sistem. Konsep penting dalam diskusi di atas adalah kesamaan antara fitur yang diekstraksi dan artikel. Teknik rekomendasi berbasis konten biasanya didasarkan pada gagasan bahwa sistem harus merekomendasikan item baru sesuai dengan kesamaan atau ketidakmiripannya dengan preferensi pribadi pengguna. Seperti yang Anda lihat, konsep kesamaan adalah salah satu bagian mendasar dari algoritme yang diperlukan untuk mengatur dan merekomendasikan musik. Untuk langganan musik digital, kesamaan konten musik dapat menjadi pendekatan yang layak untuk menyajikan musik kepada

pengguna. Contoh konten musik meliputi genre musik, artis, penulis, dan tahun pembuatan.

Dengan begitu banyak data yang tersedia tentang aktivitas pelanggan, *machine learning* dapat digunakan untuk membuat rekomendasi yang sangat relevan. Dengan banyaknya data tersebut dibutuhkan sistem yang mampu mengekstraksi esensi dari semua informasi yang tersedia dan membuat ringkasan untuk membantu pengambilan keputusan yang lebih baik.

Kemampuan sistem rekomendasi yang handal tersebut tentunya didukung oleh berbagai metode, seperti *content-based filtering* [5][6], *collaborative filtering* [7][9], dan *hybrid filtering* [10][11][12]. *Collaborative Filtering* memiliki beberapa kelebihan di antaranya mudah diimplementasikan dan dapat menyaring segala jenis informasi atau barang tanpa harus menganalisis komentar-komentar dari pengguna. Selain itu *Collaborative Filtering* menghasilkan rekomendasi kualitas tinggi daripada sistem rekomendasi berdasarkan konten [13][14][15][16].

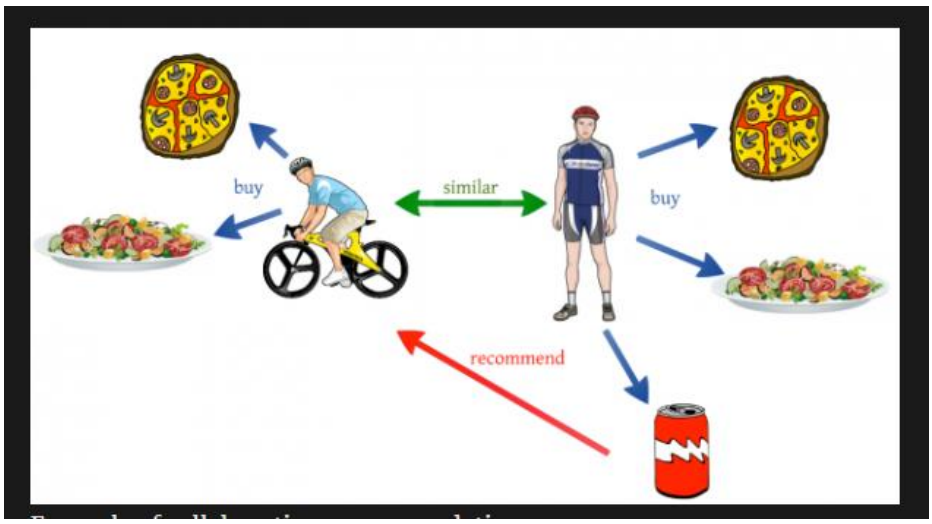
2.2 *Collaborative Filtering*

Metode *Collaborative Filtering* merupakan salah satu metode rekomendasi yang menggunakan data *rating* dari pengguna lain untuk menghasilkan rekomendasi. Metode tersebut menganggap bahwa selera pengguna terhadap suatu produk akan cenderung sama dari waktu ke waktu, begitu pula dengan pengguna lain yang memiliki selera sama. Lebih sederhananya metode *collaborative filtering* bekerja dengan cara memprediksi apa yang akan disukai pengguna, berdasarkan pada kemiripan dengan pengguna lain. Kemiripan tersebut diperoleh dari menganalisis sekumpulan besar informasi tentang perilaku, atau preferensi pengguna [18].

Dalam hal preferensi pengguna, biasanya dinyatakan dalam dua kategori yaitu peringkat eksplisit dan peringkat implisit. Peringkat eksplisit, adalah tarif yang diberikan oleh pengguna untuk item dalam skala geser, seperti 5 bintang untuk

Titanic. Ini adalah umpan balik paling langsung dari pengguna untuk menunjukkan betapa mereka menyukai suatu item. Implisit Rating / Peringkat implisit, menyarankan preferensi pengguna secara tidak langsung, seperti tampilan halaman, klik, catatan pembelian, mendengarkan trek musik atau tidak, dan sebagainya. Pada *Collaborative Filtering*, attribut yang digunakan bukan konten tetapi *user behaviour*. Contohnya kita merekomendasikan suatu item berdasarkan dari riwayat *rating* dari *user* tersebut maupun *user* lain.

Ide *Collaborative Filtering* adalah menemukan pengguna di komunitas yang berbagi apresiasi. Jika dua pengguna memiliki item dengan nilai yang sama atau hampir sama, maka mereka memiliki selera yang sama. Pengguna tersebut membangun grup atau yang disebut lingkungan. Pengguna mendapatkan rekomendasi untuk item yang belum pernah dinilai pengguna sebelumnya tetapi dinilai positif oleh pengguna di lingkungannya. Ilustrasi *Collaborative Filtering* dapat dilihat pada gambar 2.3.



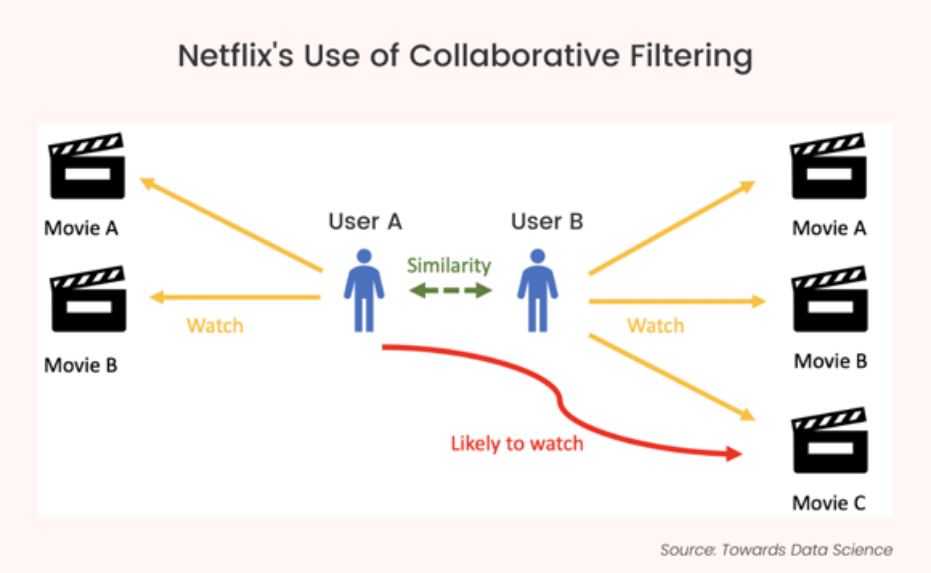
Gambar 2.3 Ilustrasi *Collaborative Filtering*
Sumber : https://d4datascience.com/2016/07/22/recommender-systems-101/
(diakses 13 Oktober 2022)

Dua orang yang memiliki hobi sama yaitu bersepeda, membeli 2 item yang sama yaitu pizza dan salad. Selain itu, orang sebelah kanan juga membeli minuman ringan. Karena hobi keduanya sama yaitu bersepeda, maka sistem

merekomendasikan minuman ringan kepada orang sebelah kiri agar membelinya juga.

Salah satu perusahaan dunia yang menerapkan *Collaborative Filtering* adalah perusahaan Movielens. MovieLens merupakan salah satu gudang data yang menyediakan data *movie*, *users* dan *ratings* dalam jumlah besar yang sering digunakan oleh banyak peneliti untuk pengujian performa atau membentuk model baru dalam sistem rekomendasi. MovieLens menggunakan *Collaborative Filtering* peringkat film dan ulasan film anggota. MovieLens berisi sekitar 11 juta peringkat untuk sekitar 8500 film. MovieLens dibuat pada tahun 1997 oleh GroupLens Research , sebuah lab penelitian di Departemen Ilmu dan Teknik Komputer di University of Minnesota, untuk mengumpulkan data penelitian tentang rekomendasi yang dipersonalisasi.

Selain Movielens, salah satu perusahaan internasional yang menggunakan sistem rekomendasi berbasis *Collaborative Filtering* adalah Netflix. Sistem rekomendasi Netflix adalah salah satu yang terbaik. Sulit dipercaya, tetapi sebelum menjadi besar, Netflix adalah perusahaan persewaan DVD yang dimulai pada tahun 1997. Bisnis mereka mulai merugi pada awal tahun 2000-an karena menghadapi tantangan besar yaitu orang-orang selalu meminta lebih. Film dan blockbuster baru tersedia untuk disewa, yang menyebabkan pelanggan tidak puas ketika mereka tidak menerimanya. Gambar 2.4. adalah contoh bagaimana Netflix menerapkan *Collaborative Filtering*.



Gambar 2.4 Ilustrasi *Collaborative Filtering* yang digunakan Netflix
Sumber : https://blog.clerk.io/ (diakses 19 November 2022)

Pada gambar 2.4 terdapat 2 orang user / pengguna yaitu *user* A dan *user* B sebagai penonton di website Netflix. Pengguna A dan Pengguna B menonton 2 film yang sama yaitu film A dan film B. Selain itu, Pengguna B juga menonton film C. Karena itu, Netflix memberikan rekomendasi kepada pengguna A agar menonton film C.

Metode *Collaborative Filtering* dengan segala kelebihannya, namun masih menghadapi masalah utama yaitu *cold start*, *sparsity* dan *scalability* [19]. *Cold start* adalah suatu kondisi pengguna baru yang belum pernah memberikan *rating* terhadap suatu produk, sehingga informasi yang didapatkan untuk arah peminatan pengguna sulit diketahui. Jika arah peminatan tidak diketahui, maka sulit untuk memberikan rekomendasi [19]. *Sparsity* merupakan kondisi suatu *dataset* yang belum terisi penuh atau datanya tidak lengkap. Salah satu penyebabnya adalah adanya produk baru yang kurang diinginkan oleh pengguna atau karena minimnya informasi sehingga pengguna belum bisa memberikan *rating* terharap produk tersebut. Selain itu bisa disebabkan kerena adanya pengguna baru yang tidak menyukai produk tersebut atau membiarkan produk tanpa memberikan *rating*. Jika data *rating* kondisinya jarang maka nilai

kesamaan antara pengguna menjadi rendah dan dengan demikian akan mempengaruhi kualitas rekomendasi [19]. *Scalability* merupakan kondisi sistem rekomendasi yang perlu meningkatkan kekuatan komputasi mereka untuk menawarkan rekomendasi yang akurat dan tepat waktu dengan kondisi data dalam skala besar [19].

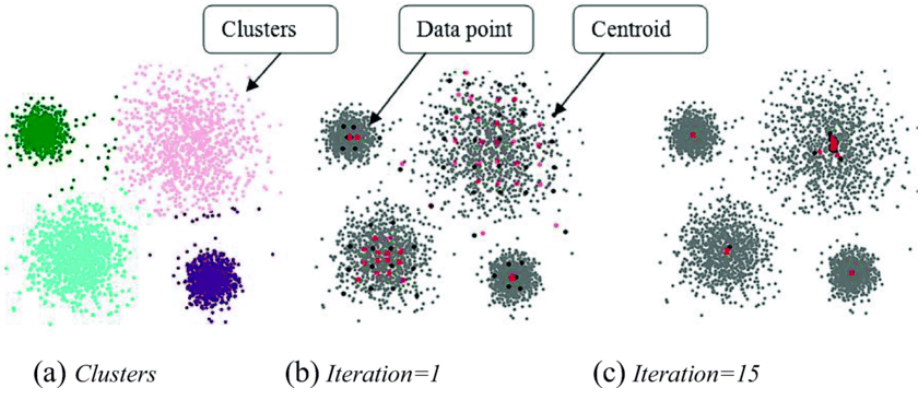
Beberapa penelitian telah dilakukan untuk menyelesaikan masalah tersebut, di antaranya :

1. Ardimansyah, MI, et. al yang menyelesaikan masalah *sparsity* (*sparse data rating*) dengan mengusulkan penggunaan matrik faktorisasi sebagai *pre-processing* untuk mengisi nilai *rating* yang kosong. *Sparsity* merupakan salah satu masalah yang dihadapi *Collaborative Filtering*, khusunya pada kategori berbasis memori karena memiliki kekurangan yaitu saat kumpulan data mengandung *sparsity* (data banyak yang kosong/jarang) maka akurasi menjadi kurang optimal. Hasil penelitian menunjukkan bahwa adanya *pre-prosessing* menggunakan matriks faktorisasi mampu meningkatkan akurasi [5].
2. Ji, R., Tian, Yi., dan Ma, Mengdi menyelesaikan masalah *sparsity* dan *cold start* menggunakan algoritma *Collaborative Filtering* berdasarkan karakteristik pengguna. Langkah yang dilakukan yaitu diawali dengan melakukan analisis perilaku pengguna, fungsi *bobot time-interest* digunakan untuk meningkatkan formula modifikasi *cosine similarity*. Kemudian tingkat preferensi pengguna dan tingkat kepercayaan pengguna digunakan untuk meningkatkan akurasi hasil rekomendasi. Eksperimen dilakukan menggunakan *dataset* hetrec 2011, dengan hasil adanya peningkatan yang signifikan terhadap nilai akurasi rekomendasi [22].

2.3 Algoritma K-Means

Klasterisasi berbasis *Centroid* adalah metode pengelompokan data ke dalam klaster non-hierarki. Klasterisasi jenis ini lebih efisien tetapi rentan terhadap *outlier*. Tipe ini juga merupakan algoritma iteratif untuk klasterisasi, di mana klaster dibentuk dari jarak minimum antara titik data ke pusat klaster

(*centroid*). secara umum metode ini memiliki fungsi tujuan yaitu meminimumkan jarak (*dissimilarity*) dari seluruh data ke pusat klaster masing-masing. Ilustrasi klasterisasi berbasis *centroid* (*Centroid-based clustering*) disajikan pada gambar 2.5.



Gambar 2.5 Klastering berbasis *centroid*

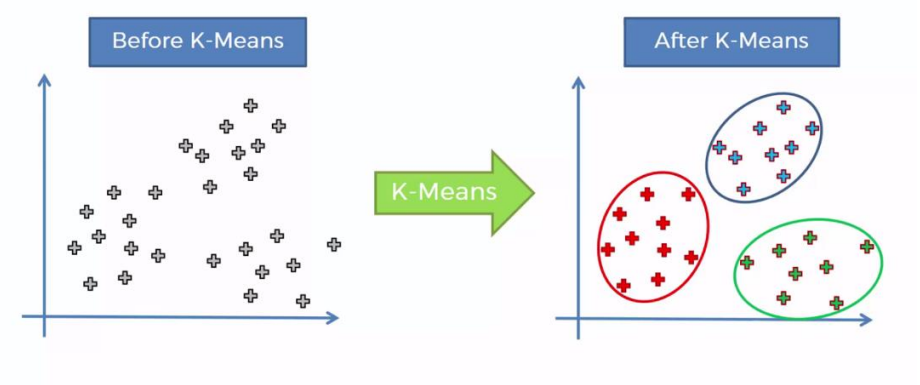
Sumber : www.researchgate.net/Centroid-based-klastering

K-Means adalah salah satu algoritma yang paling populer untuk klasterisasi berbasis *centroid*. Hal ini dikarenakan K-Means dapat mengelompokkan data dalam jumlah besar dengan waktu komputasi yang cepat dan efisien.

Secara umum metode K-Means menggunakan langkah-langkah berikut untuk melakukan proses pengelompokan.

* Tentukan jumlah klaster
* Tetapkan data ke klaster yang ada secara acak
* Hitung rata-rata setiap klaster dari data yang disertakan
* Alokasikan ulang semua data ke klaster berikutnya
* Ulangi proses nomor 3 sampai tidak ada perubahan lagi atau perubahan di bawah ambang batas.

Ilustrasi K-Means dapat dilihat pada gambar 2.6 berikut ini :



Gambar 2.6 Ilustrasi K-Means

Sumber : https://miro.medium.com/

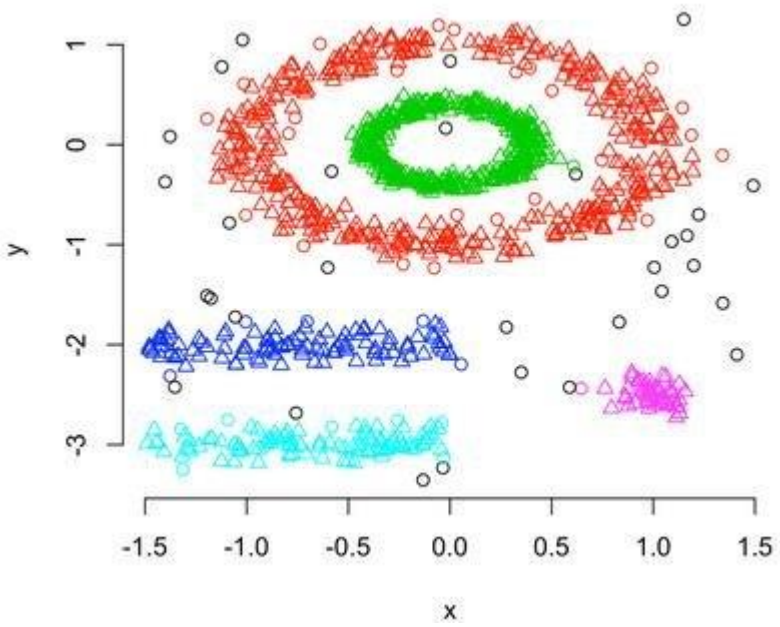
Evaluasi *clustering* dilakukan dengan tujuan untuk mengetahui seberapa baik kualitas dari hasil *clustering*. Pada penelitian ini, evaluasi hasil *clustering* yang digunakan adalah *Davies Bouldin Index (DBI)* untuk mengetahui jumlah *cluster* yang paling optimal. *Davies Bouldin Index (DBI)* diperkenalkan oleh David L. Davies dan Donald W. Bouldin pada tahun 1979. *Davies-Bouldin Index* merupakan salah satu metode yang digunakan untuk mengukur validitas atau jumlah *cluster* paling optimal pada metode K-Means. Muningsih, Elly., dkk [23] menggunakan DBI dalam menentukan jumlah kluster propinsi terbaik berdasarkan potensi desa yaitu 3 *cluster* dengan DBI 0,175. Selain itu, Az-zahra, A.A, dkk [24] menggunakan DBI dalam menentukan jumlah *cluster* terbaik untuk tingkat minat belanja online di Provinsi Yogyakarta yaitu 9 *cluster* dengan DBI 1,3427

2.4 Algoritma *DBSCAN* (*Density-Based Spatial Clustering of Application with Noise*)

Density-based Klastering adalah pengelompokan data berbasis kepadatan dengan menggabungkan wilayah dengan kepadatan yang sama ke dalam kelompok. Tipe ini memiliki kesulitan menangani data multi-densitas berdimensi tinggi. Metode ini membuat klaster berdasarkan kepadatan setiap titik data. Area yang padat karena banyaknya titik data di area tersebut

kemudian dianggap kelompok/klaster. Di sisi lain, area dengan titik data yang sangat sedikit dianggap sebagai *noise* atau *outlier*.

Pengelompokan berbasis kepadatan menghubungkan wilayah dengan kepadatan sampel tinggi ke dalam kelompok. Hal ini memungkinkan distribusi menjadi sewenang-wenang, selama daerah padat dapat dihubungkan. Algoritma ini memiliki masalah dengan data kepadatan yang berbeda dan dimensi yang tinggi. Selain itu, dengan desain, algoritma tidak menetapkan *outlier* ke klaster. Ilustrasi *density based clustering* disajikan pada gambar 2.7 berikut ini :



Gambar 2.7 *Density Based Clustering*

Sumber : www.researchgate.net/Density-based-Klastering

Algoritma yang merupakan *Density-Based Clustering* di antaranya adalah:

- *OPTICS (Ordering Points to Identify Klastering Structure)*
- *DBSCAN (Density-Based Spatial Klastering of Applications with Noise)*
- *HDBSCAN (Hierarchical Density-Based Spatial Klastering of Applications with Noise)*

Dewi Anjainah dan Siti Monalisa [25] menggunakan DBSCAN untuk melakukan analisis rekomendasi produk berdasarkan segmentasi pelanggan. Dalam segmentasi pelanggan, diperoleh 5 *cluster* dan 31 data *noise*. Selain itu, Teguh Iman Hermanto dan Yusuf Muhyidin [26] menggunakan DBSCAN untuk melakukan analisis data sebaran bandwidth menggunakan algoritma DBSCAN untuk menentukan tingkat kebutuhan bandwidth di Kabupaten Purwakarta. Jumlah *Cluster* yang terbentuk yaitu sebanyak 2 *cluster*.

Penelitian selanjutnya dilakukan oleh Das, et al [3] yang mengkombinasikan teknik *clustering* dengan algoritme voting. Teknik *clustering* yang digunakan adalah *DBSCAN* dengan meng-klaster pengguna untuk mengidentifikasi kelompok pengguna yang memiliki karakteristik serupa. Kombinasi *DBSCAN* dan Borda akan menghasilkan rekomendasi berupa film yang paling disukai sesuai dengan genrenya. Selanjutnya penelitian Tang dan Tong [4], dengan mengusulkan BordaRank yang terdiri dari dua tahapan yaitu item *Collaborative Filtering* dan agreasi item dengan metode *Borda Count*. Pada tahapan item *Collaborative Filtering* untuk mengukur *user* dan item *similarity* menggunakan *Euclidean Distance Based Similarity* dengan memanfaatkan informasi demografi. Tujuannya adalah untuk mengisi matrik yang jarang (*sparsity*) dengan memprediksi *rating*. Setelah itu dilanjutkan dengan proses agregasi sehingga menghasilkan *ranking* produk untuk direkomendasikan ke *user* target.

DBSCAN memerlukan dua input parameter sebelum melakukan proses *clustering* yaitu *epsilon* (eps) dan *minimum points* (minPts). *Epsilon* merupakan jarak maksimum antara dua data dalam satu *cluster* yang diperbolehkan, dan titik minimum adalah banyaknya data minimal dalam jarak *epsilon* agar terbentuk suatu *cluster*. Metode jarak yang digunakan dalam DBSCAN adalah jarak *Euclidian*.

2.5 *Weight Point Rank (*WP-Rank*)*

Beberapa penelitian dengan pendekatan berbasis *ranking* telah dilakukan P. A. Riyaz and S. M. Varghese. Keduanya mengusulkan Algoritma *Collaborative Filtering* (CF) diimplementasikan di Apache Hadoop dengan memanfaatkan M*apR*educe untuk *Bigdata* (*scalability*) [1]. Wu, et al. [2] menggunakan metode user-based collaborative filtering (CF), graph-based method dan social-based CF, yang selanjutnya diintegrasikan dengan menggunakan metode agregasi yaitu CombSum, CombMED dan Borda untuk membentuk rekomendasi hybrid. Pendekatan tersebut mampu meningkatkan kinerja rekomendasi dan mengatasi masalah *scalability*. Namun karena penentuan genre dilakukan secara random maka terjadi penurunan kualitas rekomendasi, selain itu akurasi pada metode Borda masih rendah.

Lestari, S., et al [7] mengusulkan pendekatan baru yaitu metode agregasi WP-Rank yang memaksimalkan penggunaan data peringkat untuk menghasilkan bobot produk. Hasil eksperimen menunjukkan bahwa metode WP-Rank lebih unggul daripada metode Borda.

Metode *Weight Point Rank* (WP-Rank) berfungsi untuk meningkatkan kualitas rekomendasi produk untuk pengguna. Metode ini bekerja dengan memanfaatkan data *rating* sebagai faktor tambahan dalam penentuan poin dan bobot. Proses WP-Rank terdiri dari 4 tahap, yaitu:

1) Menghitung jumlah rating yang sama.

$$S_{u_g,p_h} = \sum_{k=1}^{l} SR(R_{u_g,p_h}, R_{u_k,p_h}) \quad (2.1)$$

$$SR\left(R_{u_g,p_h}, R_{u_k,p_h}\right) = \begin{cases} 1, if\ R_{u_g,p_h} = R_{u_k,p_h} \\ 0\ if\ R_{u_g,p_h} \neq R_{u_k,p_h} \end{cases} \quad (2.2)$$

dengan $R_{u_g,p_h}, R_{u_k,p_h} \neq 0$

Pengguna dilambangkan sebagai U = {u1, u2, u3, …, ul} dan produk dilambangkan sebagai P = {p_1, p_2, p_3, …, p_m}, rating dari pengguna ke-g untuk produk ke-h diwakili sebagai Ru_g,p_h dan Ru_k,p_h dengan k = {1, 2, 3, … l}.

Persamaan 2.1 digunakan untuk menghitung jumlah *rating* yang sama dengan notasi Su_g,p_h. *Rating* yang sama akan bernilai 1, dan jika tidak sama maka akan bernilai 0, dengan notasi $SR(Ru_g,p_h,\ Ru_k,p_h)$ seperti pada Persamaan (2.2).

2) Menentukan poin produk

$$P_{u_g,p_h} = 1 + \sum_{k=1}^{m} PR_{u_g,p_h,p_k} \tag{2.3}$$

$$PR_{u_g,p_h,p_k} = \begin{cases} 1, if\ R_{u_g,p_h} > R_{u_k,p_h} \\ 0\ if\ R_{u_g,p_h} < R_{u_k,p_h} \end{cases} \tag{2.4}$$

$$PR_{u_g,p_h,p_k} = \begin{cases} 1, if\ R_{u_g,p_h} = R_{u_k,p_k}, S_{u_g,p_h} > S_{u_g,p_k} \\ 0\ if\ R_{u_g,p_h} = R_{u_k,p_k}, S_{u_g,p_h} < S_{u_g,p_k} \end{cases} \tag{2.5}$$

$$PR_{u_g,p_h,p_k} = \begin{cases} 1, if\ R_{u_g,p_h} = R_{u_k,p_k}, S_{u_g,p_h} > S_{u_g,p_k}, h < k \\ 0\ if\ R_{u_g,p_h} = R_{u_k,p_k}, S_{u_g,p_h} < S_{u_g,p_k}, h > k \end{cases} \tag{2.6}$$

dengan h ≠ k

Persamaan 2.3 merupakan proses menentukan poin produk dengan notasi Pug,ph. Poin produk diperoleh dengan cara menambahkan nilai 1 dengan hasil penjumlahan poin yang memenuhi syarat seperti pada Persamaan 2.4 – 2.6. Poin produk akan bernilai 1 jika:

a. *Rating* dari pengguna ke-g pada produk ke-h (R_{u_g,p_h}) lebih besar dari *rating* dari pengguna ke-g pada produk ke-k (R_{u_g,p_k}).
b. *Rating* dari pengguna ke-g pada produk ke-h (R_{u_g,p_h}) sama dengan *rating* dari pengguna ke-g pada produk ke-k (R), dan jumlah *rating* yang sama u_g,p_k pada produk ke-h (S_{u_g,p_h}) lebih besar dari jumlah *rating* yang sama pada produk ke-k (S_{u_g,p_k}).

c. *Rating* dari pengguna ke-g pada produk ke-*h* (R_{u_g,p_h}) sama dengan *rating* dari pengguna ke-g pada produk ke-k (*R*), jumlah *rating* yang sama u_g,p_k pada produk ke-*h* (S_{u_g,p_h}) sama dengan jumlah *rating* yang sama pada produk ke-k (S_{u_g,p_k}), dan urutan kemunculan produk ke-*h* lebih kecil dari urutan kemunculan produk ke-k.

d. Selain dari ketiga kondisi tersebut yaitu saat $R_{u_g,p_h} = R_{u_g,p_k} < R_{u_g,p_k}$ atau

$R_{u_g,p_h}, S_{u_g,p_h} < S_{u_g,p_k}$ atau $R_{u_g,p_h} = R_{u_g,p_k}, S_{u_g,p_h} = S_{u_g,p_k}$, h > k

$Ru_g,p_h = Ru_g,p_k < Ru_g,p_k$ atau Ru_g,p_h, $Su_g,p_h < Su_g,p_k$ atau $Ru_g,p_h = Ru_g,p_k$, $Su_g,p_h = Su_g,p_k$, $h > k$ maka akan bernilai 0.

3) Menghitung *Weight Point*

$$WPu_g,p_h = (Su_g,p_h + Ru_g,p_h)\ Pu_g,p_h \quad (2.7)$$

Persamaan 2.7 merupakan proses perhitungan bobot poin (WPu_g,p_h), yang diperoleh dari proses perkalian poin produk (Pu_g,p_h) dengan hasil penjumlahan *rating* yang sama (Su_g,p) dengan *rating* (Ru_g,phh).

4) Menghitung *Weight Point Rank* (WP-Rank)

$$WPRank_{p_h} = \sum_{k=1}^{m} WP_{u_k,p_h} \quad (2.8)$$

Persamaan 2.8 menyatakan *Weight Point Rank* ($WPRank_{p_h}$) yang diperoleh dengan menjumlahkan bobot poin pada setiap produk (WP_{uk},p_h).

2.6 *Normalized Discounted Cumulative Gain* (NDCG)

Normalized Discounted Cumulative (NDCG) adalah ukuran kualitas peringkat. *Machine Learning* sering menggunakan NDCG untuk mengevaluasi performa mesin telusur, rekomendasi, atau sistem pengambilan informasi lainnya.

NDCG sering digunakan pada mesin pencari populer untuk perusahaan yang memiliki aplikasi yang berinteraksi langsung dengan pelanggan, seperti Amazon, Netflix, dan Spotify.

Nilai NDCG ditentukan dengan membandingkan relevansi item yang dikembalikan oleh mesin pencari dengan relevansi item yang akan dikembalikan oleh mesin pencari "ideal" hipotetis. Misalnya, jika anda menelusuri "Hero" di aplikasi streaming musik populer, anda mungkin mendapatkan 10+ hasil dengan kata "Hero" di lagu, artis, atau album.

Relevansi setiap lagu atau artis diwakili oleh skor (juga dikenal sebagai "nilai") yang ditetapkan ke kueri penelusuran. Skor dari rekomendasi ini kemudian didiskon berdasarkan posisinya di hasil pencarian – apakah mereka direkomendasikan pertama atau terakhir? Skor diskon kemudian diakumulasikan dan dibagi dengan skor diskon maksimum yang mungkin, yang merupakan skor diskon yang akan diperoleh jika mesin pencari mengembalikan dokumen sesuai urutan relevansinya yang sebenarnya.

Jika seorang pengguna menginginkan lagu "My Hero" dari Foo Fighters, misalnya, semakin dekat lagu tersebut ke atas untuk rekomendasi, semakin baik pencarian untuk pengguna tersebut. Pada akhirnya, urutan relatif dari hasil atau rekomendasi yang dikembalikan penting untuk kepuasan pelanggan.

2.7 Imputasi Data

Imputasi data adalah proses mengisi atau memperkirakan nilai yang hilang atau tidak lengkap dalam sebuah kumpulan data. Imputasi data juga dapat diartikan sebagai proses memperkirakan nilai yang hilang dari suatu pengamatan berdasarkan nilai-nilai valid dari variabel lain[27]. Ketika data yang diperoleh tidak lengkap atau terdapat nilai yang hilang, imputasi data menjadi metode yang penting untuk menghasilkan data yang lebih lengkap dan dapat digunakan dalam analisis lebih lanjut. Sebagai solusi yang tepat agar ukuran sampel tidak

berkurang, maka jika terdapat missing data adalah dengan melakukan imputasi [28]. Sumber utama untuk imputasi data adalah teknik statistik.

Teknik statistik adalah teknik yang melibatkan penggunaan pendekatan statistik yang didasarkan pada pola dan hubungan dalam data yang ada. Misalnya, metode imputasi statistik seperti *mean imputation* dapat digunakan untuk mengisi nilai yang hilang dengan rata-rata dari nilai yang tersedia. Dengan imputasi data yang tepat, kumpulan data yang awalnya tidak lengkap atau memiliki nilai yang hilang dapat digunakan untuk menghasilkan hasil analisis yang lebih bermakna dan akurat.