BAB 2
LANDASAN TEORI

2.1 Segmentasi Pasar

Segmentasi pasar merupakan salah satu konsep fundamental dalam bidang pemasaran yang bertujuan untuk memahami karakteristik konsumen secara lebih spesifik. Secara umum, segmentasi pasar dapat diartikan sebagai proses pembagian suatu pasar yang bersifat heterogen menjadi beberapa kelompok yang lebih homogen berdasarkan karakteristik tertentu, seperti kebutuhan, perilaku, maupun preferensi konsumen. Dengan adanya segmentasi pasar, suatu organisasi atau instansi dapat menentukan strategi yang lebih tepat dalam menyampaikan produk atau layanan kepada target pengguna.

Dalam penerapannya, segmentasi pasar membantu perusahaan dalam mengenali perbedaan karakteristik pelanggan sekaligus menyusun strategi pemasaran yang lebih terarah. Melalui pengelompokan tersebut, perusahaan dapat menyesuaikan pendekatan pemasaran serta memanfaatkan sumber daya yang dimiliki secara lebih efektif. Perdana *et. al* [8] menyatakan bahwa segmentasi pelanggan merupakan strategi yang dilakukan dengan cara mengelompokkan pelanggan berdasarkan perbedaan karakteristik, perilaku, maupun kebutuhan. Penerapan segmentasi ini penting karena dapat digunakan untuk mengetahui tingkat loyalitas pelanggan serta menjadi dasar dalam menentukan strategi pemasaran yang efektif dan efisien.

Sejalan dengan hal tersebut, Ahsina *et al.*[9] menyatakan bahwa segmentasi merupakan cara untuk meningkatkan efektivitas komunikasi dengan pelanggan. Tujuan utamanya adalah menyesuaikan produk, layanan, serta pesan pemasaran agar lebih relevan dengan masing-masing segmen. Dengan demikian, setiap kelompok pelanggan dapat dilayani secara lebih personal sesuai dengan karakteristiknya.

Dalam konteks data pengeluaran per kapita, segmentasi pasar menjadi semakin relevan karena adanya perbedaan pola konsumsi masyarakat di berbagai wilayah di Indonesia. Perbedaan daya beli di setiap daerah tentu memengaruhi kebutuhan serta karakteristik segmen yang terbentuk. Oleh karena itu, diperlukan pendekatan berbasis data agar proses segmentasi yang dilakukan lebih objektif dan dapat dipertanggungjawabkan secara ilmiah. Salah satu metode yang sering

digunakan dalam proses ini adalah teknik *clustering*, khususnya *K-Means clustering* [12].

2.2 *K-Means Clustering*

Clustering atau pengelompokan merupakan salah satu teknik dalam *data mining* yang bertujuan untuk mengelompokkan sekumpulan objek atau data ke dalam beberapa kelompok (*cluster*) berdasarkan kemiripan atau kedekatan antar data. Objek-objek yang berada dalam satu *cluster* memiliki tingkat kesamaan yang tinggi satu sama lain, sedangkan objek yang berada di *cluster* yang berbeda memiliki tingkat perbedaan yang cukup besar. Karakteristik inilah yang membedakan *clustering* dari metode klasifikasi biasa, di mana pada *clustering* tidak diperlukan label kelas terlebih dahulu. Sebagai metode *unsupervised learning*, *clustering* tidak memerlukan data latih yang sudah berlabel. Algoritma ini bekerja sepenuhnya berdasarkan struktur yang ada di dalam data itu sendiri, mencari pola tersembunyi tanpa panduan dari variabel target yang sudah ditentukan sebelumnya. Hal ini menjadikan *clustering* sangat fleksibel dan dapat diterapkan pada berbagai jenis data, termasuk data sosial-ekonomi seperti data pengeluaran per kapita masyarakat [16].

Di antara berbagai algoritma *clustering* yang tersedia, *K-Means* merupakan salah satu yang paling banyak digunakan baik di kalangan akademisi maupun praktisi. Algoritma ini tergolong populer karena konsepnya yang sederhana, mudah diimplementasikan, serta memiliki performa komputasi yang baik bahkan pada *dataset* berukuran besar. Rianti *et al.* [17] menjelaskan bahwa *clustering* merupakan algoritma yang mengelompokkan data berdasarkan jarak ke *centroid* terdekat, sehingga menghasilkan kelompok-kelompok yang homogen secara internal. Konsep ini sejalan dengan pendapat Ashari *at al.* [16], yang mendefinisikan *clustering* sebagai algoritma yang digunakan untuk mengklasifikasikan objek data berdasarkan kemiripannya, dan algoritma ini termasuk dalam kategori *unsupervised learning* yang dapat digunakan untuk mengklasifikasikan data tanpa label.

Ahsina *et al.*[9] mendefinisikan *K-Means* sebagai metode *clustering* yang sering digunakan di berbagai bidang karena penggunaannya sederhana, mudah diimplementasikan, dan mampu mengelompokkan data yang besar. Secara konseptual, algoritma ini bekerja dengan cara mengelompokkan data ke dalam sejumlah *cluster* berdasarkan jarak terdekat ke titik pusat *cluster* yang disebut

centroid. Proses ini dilakukan secara iteratif hingga posisi *centroid* tidak lagi berubah secara signifikan. Siagian *et al.* [12] menambahkan bahwa *K-Means clustering* adalah salah satu metode *data clustering* non-hierarki yang mengelompokkan data dalam bentuk satu atau lebih *cluster*. Data-data yang memiliki karakteristik yang sama dikelompokkan dalam satu *cluster*, sedangkan data yang memiliki karakteristik berbeda dikelompokkan dalam *cluster* yang berbeda, sehingga data yang berada dalam satu *cluster* memiliki tingkat variasi yang kecil.

Secara algoritmis, langkah-langkah dalam *K-Means Clustering* dapat diuraikan sebagai berikut:

1. Tentukan jumlah *cluster K* yang diinginkan.
2. Inisialisasi *K centroid* secara acak dari kumpulan data yang ada.
3. Hitung jarak setiap titik data ke masing-masing *centroid* menggunakan *Euclidean Distance*.
4. Masukkan setiap titik data ke *cluster* dengan *centroid* terdekat.
5. Perbarui posisi *centroid* dengan menghitung rata-rata (*mean*) dari seluruh titik data dalam *cluster* tersebut.
6. Ulangi langkah 3–5 hingga *centroid* tidak berubah atau memenuhi kriteria konvergensi.

Salah satu kelemahan mendasar dari inisialisasi *centroid* secara acak adalah kemungkinan terpilihnya titik-titik awal yang saling berdekatan atau tidak mewakili struktur data secara keseluruhan, sehingga proses iterasi berpotensi terjebak pada solusi optimum lokal (*local optimum*) yang kurang baik. Untuk mengatasi kelemahan ini, dikembangkan metode inisialisasi yang dikenal dengan *K-Means++*. Berbeda dengan pendekatan acak biasa yang memilih seluruh titik awal secara serentak tanpa pertimbangan, *K-Means++* menentukan *centroid* pertama secara acak dari data yang ada, kemudian *centroid* berikutnya dipilih berdasarkan probabilitas yang berbanding lurus dengan kuadrat jarak titik data tersebut terhadap *centroid* terdekat yang sudah terpilih sebelumnya. Dengan mekanisme ini, titik data yang berada jauh dari *centroid-centroid* yang sudah ada memiliki peluang lebih besar untuk terpilih sebagai *centroid* selanjutnya, sehingga posisi *centroid* awal yang dihasilkan cenderung lebih tersebar dan

merepresentasikan struktur data secara lebih menyeluruh. Penggunaan *K-Means++* pada umumnya mampu mempercepat proses konvergensi algoritma serta meningkatkan kualitas hasil pengelompokan dibandingkan inisialisasi acak konvensional, sehingga metode ini banyak diterapkan sebagai metode inisialisasi *default* pada berbagai pustaka pemrograman, termasuk *scikit-learn*.

Jarak antar titik data dan *centroid* dihitung menggunakan rumus *Euclidean Distance*. Menurut Rahmawati *et al.*[18], formula *Euclidean Distance* yang digunakan adalah sebagai berikut:

$$d(i,j) = \sqrt{\sum_{l=1}^{p}(x_{il} - x_{jl})^2} \tag{2.1}$$

Keterangan:

a. $d(i,j)$ = jarak antara objek i dan objek j

b. x_{il} = nilai objek i pada variabel ke-l

c. x_{jl} = nilai objek j pada variabel ke-l

d. p = jumlah variabel

Selain *Euclidean Distance*, inti dari *K-Means* juga dapat dijelaskan melalui fungsi objektifnya. Fungsi objektif *K-Means* adalah meminimalkan jumlah kuadrat jarak (*Sum of Squared Errors*/SSE) antara setiap titik data dengan *centroid cluster*-nya. Ahsina *et al.* [9] menyatakan bahwa kualitas dari suatu kelompok dapat diukur dengan variasi *within-cluster* yang merupakan jumlah kuadrat kesalahan antara semua objek dalam suatu kelompok dengan *centroid*-nya. Secara matematis, fungsi objektif tersebut dituliskan sebagai berikut:

$$J = \sum_{k=1}^{K}\sum_{x \in C_k} \|x - \mu_k\|^2 \tag{2.2}$$

Keterangan:

a. J = nilai fungsi objektif (total SSE)

b. K = jumlah *cluster*

c. C_k = himpunan titik data dalam *cluster* ke-k

d. x = titik data

e. μ_k = *centroid* dari *cluster* ke-*k*

Nilai *J* yang semakin kecil menandakan bahwa anggota-anggota dalam setiap *cluster* semakin kompak dan berdekatan dengan *centroid*-nya, yang berarti kualitas *clustering* semakin baik. Proses minimisasi fungsi objektif ini yang menjadi dasar iterasi dalam algoritma *K-Means* hingga tercapai kondisi konvergensi [8].

2.3 *Agglomerative Clustering*

Agglomerative Clustering merupakan salah satu jenis algoritma *hierarchical clustering* yang bekerja dengan pendekatan *bottom-up*, yaitu memulai proses pengelompokan dengan menganggap setiap titik data sebagai satu *cluster* tersendiri, kemudian secara bertahap menggabungkan *cluster-cluster* yang paling mirip satu sama lain hingga seluruh data tergabung dalam satu *cluster* tunggal atau hingga mencapai jumlah *cluster* yang telah ditentukan [15]. Pendekatan ini berbeda secara fundamental dengan *K-Means* yang bekerja melalui optimasi posisi *centroid* secara iteratif, karena *Agglomerative Clustering* tidak memerlukan inisialisasi *centroid* maupun penentuan jumlah *cluster* di awal proses, melainkan membentuk struktur hierarki yang dapat dipotong pada tingkat kemiripan tertentu untuk menghasilkan jumlah *cluster* yang diinginkan.

Secara algoritmis, proses *Agglomerative Clustering* diawali dengan memperlakukan setiap titik data sebagai *cluster* tunggal, sehingga apabila terdapat *N* data maka pada tahap awal terbentuk *N cluster*. Selanjutnya, dihitung jarak atau tingkat ketidakmiripan antar seluruh pasangan *cluster* yang ada. Dua *cluster* dengan tingkat kemiripan tertinggi atau jarak terkecil kemudian digabungkan menjadi satu *cluster* baru, dan matriks jarak diperbarui untuk merefleksikan penggabungan tersebut. Langkah ini diulang hingga seluruh data tergabung dalam satu *cluster* besar, atau hingga jumlah *cluster* yang tersisa sesuai dengan nilai *K* yang telah ditentukan. Hasil dari keseluruhan proses penggabungan ini dapat divisualisasikan dalam bentuk *dendrogram*, yaitu diagram pohon yang menggambarkan urutan dan jarak penggabungan setiap *cluster*.

Salah satu komponen penting dalam *Agglomerative Clustering* adalah metode *linkage*, yaitu kriteria yang digunakan untuk menentukan jarak atau tingkat kemiripan antar *cluster* pada setiap tahap penggabungan. Beberapa metode *linkage* yang umum digunakan antara lain *single linkage* yang menghitung jarak minimum antar anggota dua *cluster*, *complete linkage* yang menghitung

jarak maksimum, *average linkage* yang menghitung rata-rata jarak antar seluruh pasangan anggota, serta *Ward linkage* [15]. Secara matematis, ketiga kriteria tersebut dapat dirumuskan sebagai berikut :

$$d_{single}(A,B) = \min_{a\in A,\, b\in B} d(a,b) \tag{2.3}$$

$$d_{complete}(A,B) = \max_{a\in A,\, b\in B} d(a,b) \tag{2.4}$$

$$d_{average}(A,B) = \frac{1}{|A||B|}\sum_{a\in A}\sum_{b\in B} d(a,b) \tag{2.5}$$

Keterangan:

a. A dan B = dua *cluster* yang dibandingkan.
b. a dan b = anggota data pada *cluster* A dan B.
c. $d(a,b)$ = jarak Euclidean antara titik data a dan b.
d. $|A|$ dan $|B|$ = jumlah anggota pada *cluster* A dan B.

Berbeda dengan ketiga kriteria tersebut, *Ward linkage* bekerja dengan prinsip meminimalkan total varians dalam *cluster* (*within-cluster variance*) pada setiap tahap penggabungan, sehingga menghasilkan *cluster* dengan bentuk dan ukuran yang relatif lebih seimbang dibandingkan metode *linkage* lainnya. Kriteria penggabungan pada *Ward linkage* dirumuskan berdasarkan besarnya kenaikan *SSE* yang terjadi apabila dua cluster digabungkan, sebagai berikut:

$$\Delta(A,B) = \frac{|A||B|}{|A|+|B|}\|\mu_A - \mu_B\|^2 \tag{2.6}$$

Keterangan:

a. $\Delta(A,B)$ = kenaikan SSE akibat penggabungan *cluster* A dan B

b. $|A|, |B|$ = jumlah anggota pada *cluster* A dan B

c. μ_A, μ_B = *centroid* dari *cluster* A dan B

Cluster yang dipilih untuk digabungkan pada setiap tahap adalah pasangan dengan nilai $\Delta(A,B)$ terkecil, sehingga peningkatan varians internal akibat penggabungan tetap minimal. Pendekatan *Ward* juga memiliki kemiripan konseptual dengan fungsi objektif *K-Means* yang sama-sama berupaya meminimalkan *Sum of Squared Errors* (SSE).

Hasil penelitian Hartono *et al.* [15] menunjukkan bahwa perbandingan antara *K-Means* dan *Agglomerative Hierarchical Clustering* (AHC) menghasilkan *cluster* yang identik pada data berjumlah kecil (10 data), namun mulai menunjukkan perbedaan ketika volume data bertambah (30 dan 60 data). Perbedaan tersebut disebabkan oleh perbedaan mekanisme kerja kedua algoritma, *K-Means* memungkinkan perpindahan anggota antar *cluster* secara iteratif, sedangkan AHC membangun hierarki yang bersifat tetap dan tidak dapat direvisi setelah penggabungan dilakukan.

2.4 Praproses Data

2.4.1 *Encoding* Variabel Kategorikal

Encoding merupakan proses transformasi data yang semula berbentuk kategorikal atau teks menjadi representasi numerik agar dapat diproses oleh algoritma yang hanya menerima input berupa angka. Pemilihan teknik *encoding* yang tepat menjadi hal yang penting karena setiap variabel kategorikal memiliki karakteristik yang berbeda-beda, sehingga teknik yang diterapkan harus disesuaikan agar makna asli dari kategori tersebut tidak hilang atau disalahartikan oleh algoritma. Terdapat beberapa pendekatan *encoding* yang umum digunakan, di antaranya *Label Encoding*, *Ordinal Encoding*, dan *One-Hot Encoding*.

Label Encoding merupakan teknik *encoding* yang mengonversi setiap kategori pada suatu variabel menjadi nilai numerik berurutan (misalnya 0, 1, 2, dan seterusnya) tanpa mempertimbangkan adanya hierarki atau tingkatan tertentu antar kategori. Teknik ini umumnya diterapkan pada variabel nominal, yaitu variabel yang kategorinya tidak memiliki urutan logis satu sama lain, seperti jenis kelamin atau status pekerjaan. Meskipun sederhana dan mudah diterapkan, *Label Encoding* memiliki risiko apabila digunakan pada variabel dengan jumlah kategori lebih dari dua, karena algoritma yang sensitif terhadap besaran numerik dapat secara tidak sengaja menafsirkan urutan angka hasil *encoding* sebagai suatu hierarki atau bobot tertentu, padahal kategori tersebut sejatinya tidak memiliki hubungan tingkatan apa pun.

Ordinal Encoding diterapkan pada variabel yang kategorinya memang memiliki urutan atau tingkatan yang bermakna secara substantif, seperti tingkat pendapatan, frekuensi pembelian, atau kisaran harga. Berbeda dengan *Label Encoding* yang pemetaan angkanya bersifat sembarang, pada *Ordinal Encoding*

urutan nilai numerik ditentukan secara eksplisit mengikuti tingkatan kategori yang sesungguhnya, sehingga selisih antar nilai dapat merepresentasikan perbedaan tingkatan secara lebih tepat. Penggunaan teknik ini memungkinkan algoritma untuk tetap memperhitungkan informasi urutan yang melekat pada variabel tersebut, informasi yang akan hilang apabila variabel ordinal diproses menggunakan teknik *encoding* yang tidak mempertimbangkan urutan.

One-Hot Encoding merupakan teknik yang mengonversi setiap kategori unik dalam suatu variabel menjadi kolom biner tersendiri, di mana nilai 1 menandakan keberadaan kategori tersebut pada suatu baris data dan nilai 0 menandakan sebaliknya [19]. Teknik ini umumnya digunakan pada variabel nominal yang memiliki jumlah kategori lebih dari dua, karena dapat menghindari munculnya asumsi keliru mengenai adanya hubungan ordinal antar kategori yang sebenarnya tidak saling berurutan. Konsekuensi dari penerapan *One-Hot Encoding* adalah bertambahnya jumlah kolom atau dimensi pada *dataset*, sebanding dengan jumlah kategori unik pada variabel yang dienkoding, sehingga teknik ini perlu diterapkan secara selektif pada variabel yang memang relevan agar tidak menimbulkan dimensi data yang berlebihan. Perlu diperhatikan pula bahwa atribut hasil konversi *One-Hot Encoding* cenderung didominasi oleh atribut bertipe kontinu selama proses *clustering*, dan secara umum tidak ada skema pembobotan yang dapat mengatasi hal tersebut [19].

2.4.2 Normalisasi Data

Normalisasi diperlukan karena variabel-variabel dalam suatu *dataset* pada umumnya memiliki skala dan rentang nilai yang berbeda-beda satu sama lain. Tanpa adanya normalisasi, variabel yang memiliki skala nilai lebih besar akan mendominasi perhitungan jarak *Euclidean* pada algoritma *K-Means*, sehingga kontribusi variabel lain yang skalanya lebih kecil menjadi tidak proporsional dan dapat menyebabkan hasil pengelompokan menjadi bias. Salah satu metode normalisasi yang umum digunakan adalah *StandardScaler*, yang bekerja dengan cara mentransformasi setiap nilai pada suatu variabel sehingga distribusi datanya memiliki rata-rata (*mean*) sama dengan nol dan standar deviasi sama dengan satu. Transformasi ini dikenal dengan istilah standarisasi atau *z-score*, yang dirumuskan sebagai berikut :

$$z = \frac{x - \mu}{\sigma} \tag{2.7}$$

Keterangan:

a. z = nilai hasil standarisasi

b. x = nilai asli data

c. μ = rata-rata data

d. σ = standar deviasi data

Berbeda dengan metode normalisasi lain seperti *MinMaxScaler* yang mentransformasikan data ke dalam rentang nilai tertentu (umumnya 0 hingga 1) berdasarkan nilai minimum dan maksimum, *StandardScaler* tidak terikat pada batas nilai tersebut sehingga relatif lebih tangguh (*robust*) terhadap keberadaan nilai ekstrem atau *outlier* dalam data, termasuk pada variabel hasil *One-Hot Encoding* yang bersifat biner. Setelah proses normalisasi dilakukan, seluruh variabel akan berada pada skala yang setara sehingga algoritma *K-Means* dapat menghitung jarak antar titik data secara lebih adil tanpa dipengaruhi oleh perbedaan satuan maupun rentang nilai asli masing-masing variabel.

Selain proses transformasi tersebut, *StandardScaler* juga menyediakan mekanisme transformasi balik yang disebut *inverse transform*, yaitu proses mengembalikan nilai data yang telah dinormalisasi ke dalam skala aslinya. Mekanisme ini menjadi penting terutama pada tahap interpretasi hasil *clustering*, karena nilai *centroid* yang dihasilkan oleh algoritma *K-Means* berada dalam skala data yang telah dinormalisasi sehingga sulit dimaknai secara langsung. Melalui proses *inverse transform*, nilai *centroid* tersebut dapat dikembalikan ke satuan aslinya, sehingga karakteristik setiap *cluster* dapat diinterpretasikan secara lebih bermakna dan sesuai dengan konteks data sesungguhnya.

2.5 Penentuan Jumlah *Cluster* Optimal

2.5.1 *Elbow Method*

Elbow Method adalah salah satu pendekatan yang paling sering digunakan untuk menentukan jumlah *cluster* optimal dalam *K-Means*. Metode ini bekerja dengan menghitung nilai SSE (*Sum of Squared Errors*) untuk setiap nilai K yang dicoba, kemudian memvisualisasikannya dalam bentuk grafik. Titik di mana penurunan nilai SSE mulai melambat secara signifikan yang menyerupai bentuk siku-siku atau '*elbow*' yang dianggap sebagai jumlah *cluster* yang paling optimal.

Rahmawati *et al.*[18] menjelaskan bahwa keunggulan dari *Elbow Method* adalah kemampuannya dalam menghitung persentase perbandingan antara jumlah *cluster* yang membentuk titik siku untuk menentukan jumlah *cluster* terbaik. Hal ini sejalan dengan temuan Perdana *et al.* [8], yang dalam penelitiannya pada segmentasi pelanggan aplikasi Alfagift menggunakan *Elbow Method* dan berhasil menentukan K optimum dari perhitungan SSE (*Sum of Squared Errors*) sebesar tiga *cluster*. Ashari *et al* [16] dalam penelitiannya juga menggunakan *Elbow Method* sebagai salah satu metode evaluasi, dan menemukan bahwa nilai $K = 3$ merupakan titik *elbow* terbaik berdasarkan pengamatan pada grafik SSE yang dibandingkan dengan nilai K. Penurunan terbesar terjadi pada $K = 3$, setelah titik itu penurunan mulai melandai.

Meskipun *Elbow Method* banyak digunakan karena kesederhanaannya, penentuan posisi *elbow* secara visual bersifat subjektif dan pada banyak *dataset* tidak menghasilkan titik belok yang tegas. Untuk mengurangi subjektivitas tersebut, penelitian ini juga memanfaatkan algoritma *Kneedle* sebagai pelengkap inspeksi visual manual [20]. *Kneedle* bekerja dengan mendeteksi titik kelengkungan maksimum pada kurva SSE secara matematis, sehingga posisi *elbow* dapat ditentukan secara objektif dan dapat direproduksi, tidak lagi bergantung sepenuhnya pada penilaian mata peneliti. Meskipun demikian, titik *elbow* yang terdeteksi secara objektif tidak selalu berimpit dengan konfigurasi K yang paling stabil maupun paling mudah diinterpretasikan secara bisnis, sehingga *Elbow Method* dalam penelitian ini tetap diposisikan sebagai salah satu masukan awal, bukan penentu tunggal nilai K final, sebagaimana dijelaskan lebih lanjut pada sub-bab 2.5.2 dan 3.5.3.

2.5.2 *Silhouette Method*

Silhouette Method adalah metode lain yang dapat digunakan untuk menentukan jumlah *cluster* optimal sekaligus mengevaluasi kualitas *clustering* yang dihasilkan. Berbeda dengan *Elbow Method* yang hanya melihat kompaksi dalam *cluster*, *Silhouette Method* mempertimbangkan dua aspek sekaligus: *cohesion* (seberapa kompak data dalam satu *cluster*) dan *separation* (seberapa jauh jarak antar *cluster* yang berbeda). Menurut Rahmawati *et al.* [18], *Silhouette Coefficient* merupakan metode yang efektif untuk menentukan jumlah *cluster* terbaik karena metode ini dapat memperkirakan seberapa dekat data di suatu kelompok dan selisih antar kelompok yang berlainan. Ripai *et al.*[14]

menambahkan bahwa *Silhouette Index* dipandang paling ideal karena menilai dua aspek sekaligus, yaitu *cohesion* dan *separation*.

Nilai *Silhouette* berkisar antara -1 hingga $+1$. Nilai yang mendekati $+1$ menunjukkan bahwa objek tersebut berada di *cluster* yang tepat, nilai mendekati 0 menunjukkan objek berada di perbatasan antara dua *cluster*, dan nilai negatif mengindikasikan bahwa objek mungkin lebih cocok masuk ke *cluster* yang lain. Secara umum, nilai rata-rata *Silhouette Score* yang direkomendasikan adalah di atas 0.5 untuk dianggap sebagai pengelompokan yang baik [21].

Meskipun *Elbow* dan *Silhouette Method* menjadi standar dalam menentukan jumlah *cluster* optimal, keduanya sering kali memberikan rekomendasi nilai *K* yang kontradiktif. *Elbow Method* berbasis nilai SSE secara matematis akan terus menurun seiring bertambahnya kelompok, sehingga cenderung mengarah pada pemilihan nilai *K* yang lebih besar demi meminimalkan *error*. Sebaliknya, *Silhouette Method* yang mengukur aspek *cohesion* sekaligus *separation* kerap menghasilkan nilai *K* yang lebih kecil, terutama pada *dataset* berdimensi tinggi dengan batas antar-kelompok yang melandai secara geometris. Ketika terjadi benturan rekomendasi, penentuan jumlah *cluster* tidak dapat diputuskan secara sepihak, melainkan butuh parameter tambahan seperti pengujian stabilitas konfigurasi *cluster* terhadap perubahan inisialisasi *centroid* menggunakan *Adjusted Rand Index* (ARI), sebagaimana diuraikan pada sub-bab 2.7. Jika suatu nilai *K* memiliki skor evaluasi internal yang tinggi namun strukturnya sangat sensitif terhadap inisialisasi acak, maka hasil pengelompokan tersebut menjadi tidak andal, sehingga analisis stabilitas ini dapat bertindak sebagai kriteria pemutus (*decisive criterion*) saat metode konvensional gagal mencapai konsensus yang jelas.

2.6 Evaluasi Kualitas *Cluster*

2.6.1 *Silhouette Score*

Silhouette Score merupakan metrik evaluasi yang mengukur seberapa baik penempatan setiap titik data dalam *cluster*-nya dibandingkan dengan *cluster* lain. Metrik ini menghasilkan nilai rata-rata dari seluruh nilai *Silhouette* individu dalam *dataset*. Hasan [22] menyatakan bahwa *Silhouette Score* memberikan ukuran seberapa baik setiap titik data ditempatkan dalam *cluster*-nya dibandingkan dengan *cluster* lain. Secara matematis, nilai *Silhouette* untuk setiap titik data i dihitung dengan rumus berikut:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{2.8}$$

Keterangan:

a. $s(i)$ = nilai silhouette titik data *i*

b. $a(i)$ = rata-rata jarak titik *i* terhadap anggota dalam *cluster* yang sama

c. $b(i)$ = rata-rata jarak minimum titik *i* terhadap *cluster* lain

Nilai *Silhouette Score* keseluruhan diperoleh dari rata-rata nilai s(i) seluruh titik datadalam dataset, dimana nilai yang mendekati 1 mengindikasikan bahwa struktur *cluster* yang terbentuk sangat baik. Aziz *et al.*[21] dalam penelitiannya mengenai pengelompokan ekspor kopi menemukan *Silhouette Coefficient* sebesar 0,73, yang termasuk dalam kategori *strong structure cluster* sehingga mengindikasikan kualitas pengelompokan yang sangat baik.

2.6.2 *Davies-Bouldin Index*

Davies-Bouldin Index (DBI) adalah metrik evaluasi *cluster* yang menilai kualitas pengelompokan berdasarkan rasio antara jarak *intra-cluster* (kompaksi) dan jarak *inter-cluster* (separasi). Semakin kecil nilai DBI, semakin baik kualitas *cluster* yang terbentuk karena *cluster* menjadi lebih kompak dan semakin jauh dari *cluster* lainnya. Shalsadilla *et al.*[23] mendefinisikan DBI sebagai suatu metode yang menentukan banyaknya *cluster* optimum berdasarkan kedekatan objek terhadap *centroid*-nya dalam satu *cluster* dan jarak antar *centroid cluster*. Secara matematis, DBI dirumuskan sebagai berikut:

$$DBI = \frac{1}{K} \sum_{i=1}^{K} \max_{j \neq i} \left(\frac{s_i + s_j}{d(c_i, c_j)} \right) \tag{2.9}$$

Keterangan:

a. DBI = nilai *Davies-Bouldin Index*

b. K = jumlah *cluster*

c. s_i, s_j = rata-rata jarak data terhadap centroid pada *cluster* i dan j

d. $d(c_i, c_j)$ = jarak antar centroid *cluster*

Naufan *et al.*[24] juga menegaskan bahwa tujuan optimasi dengan DBI adalah mencapai nilai DBI terendah yang mendekati angka 0, yang mengindikasikan bahwa *cluster* memiliki pola yang konsisten. Dalam penelitiannya, Muhammad *et al.* [11] berhasil mengoptimalkan nilai DBI menjadi 0,049 menggunakan kombinasi *K-Means* dan PCA, jauh lebih baik dibandingkan *K-Means* tanpa PCA yang menghasilkan DBI sebesar 0,466. Hasan [22] juga menemukan bahwa penggunaan DBI sebagai metrik evaluasi dapat memberikan panduan yang efektif dalam memilih algoritma *clustering* yang paling sesuai untuk suatu *dataset*.

2.6.3 *Calinski-Harabasz Index*

Calinski-Harabasz Index (CHI), yang juga dikenal sebagai *Variance Ratio Criterion*, adalah metrik evaluasi yang mengukur rasio antara dispersi antar-*cluster* (*between-cluster dispersion*) dan dispersi dalam *cluster* (*within-cluster dispersion*). Berbeda dengan DBI, pada *Calinski-Harabasz Index* nilai yang lebih tinggi justru mengindikasikan kualitas *cluster* yang lebih baik. Ashari *et al.*[16] menjelaskan bahwa *Calinski-Harabasz Index* menentukan *cluster* berdasarkan rata-rata tingkat dispersi *cluster*, di mana semakin tinggi nilai rata-ratanya, semakin baik jumlah *cluster* yang dipilih. Secara matematis, CHI dirumuskan sebagai berikut:

$$CHI = \frac{\mathrm{tr}(B_k)}{\mathrm{tr}(W_k)} \times \frac{N-K}{K-1} \tag{2.10}$$

Keterangan:

a. CHI = nilai *Calinski-Harabasz Index*

b. $\mathrm{tr}(B_k)$ = trace dispersi antar-*cluster*

c. $\mathrm{tr}(W_k)$ = trace dispersi dalam *cluster*

d. N = jumlah total data

e. K = jumlah *cluster*

Dalam penelitiannya, Rianti *et al.*[17] menemukan bahwa *K-Means* menghasilkan *Calinski-Harabasz Index* sebesar 757,06, yang menunjukkan hasil *clustering* yang cukup baik. Ripai *et al.*[14] menyimpulkan bahwa kombinasi *Silhouette Index*, *Davies-Bouldin Index*, dan *Calinski-Harabasz Index* tetap

diperlukan untuk mendapatkan hasil evaluasi yang paling komprehensif, karena masing-masing metrik mengukur aspek yang berbeda dari kualitas *clustering* secara keseluruhan.

2.7 Metrik Perbandingan dan Stabilitas *Clustering*

2.7.1 *Adjusted Rand Index* (ARI)

Adjusted Rand Index (ARI) adalah metrik evaluasi eksternal yang mengukur tingkat kesesuaian antara dua hasil pengelompokan dengan mempertimbangkan faktor kebetulan (*chance agreement*). Metrik ini menghitung kemiripan antara dua partisi data dengan cara mengevaluasi jumlah pasangan data yang ditempatkan bersama atau dipisahkan secara konsisten di kedua partisi, kemudian membandingkannya dengan ekspektasi acak yang dihitung berdasarkan model hipergeometrik [13]. Nilai ARI memiliki rentang dari negatif satu -1 hingga positif satu $+1$. Berdasarkan skala interpretasi yang digunakan oleh [13], nilai $\geq 0{,}90$ dikategorikan sebagai kesesuaian yang baik sekali, nilai $\geq 0{,}80$ dikategorikan baik, nilai $\geq 0{,}65$ dikategorikan sedang, dan nilai $< 0{,}65$ dikategorikan buruk.

Dalam penerapannya pada data akademik, Alkautsar *et al.*[13] menunjukkan bahwa *cluster* yang terbentuk secara geometri belum tentu mencerminkan pengelompokan yang bermakna secara substantif apabila tidak diverifikasi menggunakan metrik evaluasi eksternal. Kecenderungan nilai ARI yang rendah pada eksperimen tersebut mengindikasikan bahwa terdapat diskrepansi antara struktur *cluster* berbasis kedekatan fitur dengan kategorisasi aktual yang dimiliki oleh data. Temuan ini menegaskan pentingnya evaluasi eksternal untuk mengukur sejauh mana hasil *clustering* selaras dengan label referensi yang tersedia, serta menyoroti bahwa ukuran jarak geometris saja tidak cukup untuk menangkap makna substantif dari pengelompokan data. Dalam penelitian ini, ARI digunakan untuk menguji stabilitas hasil *clustering* terhadap variasi inisialisasi *centroid* (*random seed*) yang berbeda pada nilai K yang sama. Nilai ARI yang tinggi dan konsisten di seluruh percobaan mengindikasikan bahwa struktur *cluster* yang dihasilkan tidak bergantung secara kritis pada titik inisialisasi awal.

2.7.2 *Normalized Mutual Information* (NMI)

Normalized Mutual Information (NMI) adalah metrik yang mengukur jumlah informasi yang dibagikan (*mutual information*) antara dua hasil

pengelompokan, yang kemudian dinormalisasi agar nilainya berada pada rentang yang dapat dibandingkan secara konsisten antar pasangan pengelompokan dengan jumlah *cluster* yang berbeda-beda. Rahmat *et al.*[25] mendefinisikan NMI sebagai standar pengukuran kualitas informasi yang digunakan untuk mencapai keseimbangan antara pemilihan jumlah *cluster* dan kualitas hasil pengelompokan secara bersamaan. Nilai NMI berkisar antara 0 hingga 1, di mana nilai mendekati 1 mengindikasikan bahwa kedua pengelompokan saling berbagi informasi secara penuh (hampir identik), sedangkan nilai mendekati 0 menunjukkan bahwa kedua hasil pengelompokan tidak memiliki keterkaitan informasi yang berarti.

Rahmat *et al.*[25] memanfaatkan NMI untuk mengevaluasi kualitas pengelompokan pada berbagai konfigurasi jumlah *cluster*. Berdasarkan eksperimen yang dilakukan, metrik ini terbukti sensitif terhadap perubahan jumlah *cluster*, di mana nilai NMI yang tinggi mengindikasikan adanya keselarasan informasi yang kuat antara struktur *cluster* yang terbentuk dengan label referensi yang tersedia. Temuan tersebut menunjukkan bahwa NMI dapat dijadikan panduan yang objektif dalam menentukan konfigurasi *clustering* yang paling optimal, karena metrik ini tidak hanya mengukur kemiripan buta antar partisi, tetapi juga menangkap seberapa besar ketidakpastian label kelas yang dapat dijelaskan oleh struktur *cluster* yang terbentuk.

Sebagaimana ARI, NMI juga digunakan dalam penelitian ini untuk menguji stabilitas hasil *clustering* terhadap variasi inisialisasi acak pada *K-Means*. Meskipun memiliki tujuan yang serupa, penggunaan ARI dan NMI secara bersamaan dalam penelitian ini bertujuan untuk memperkuat keyakinan terhadap hasil yang diperoleh melalui dua sudut pandang yang berbeda. ARI mengukur kesesuaian berdasarkan kesepakatan pasangan data (*pairwise*), sehingga lebih peka terhadap perubahan kecil dalam komposisi *cluster*, sedangkan NMI mengukur kesesuaian berdasarkan pengurangan ketidakpastian informasi, sehingga lebih bersifat global dan tidak terlalu terpengaruh oleh ukuran *cluster* yang tidak seimbang. Dengan demikian, konsistensi nilai yang tinggi pada kedua metrik dapat menjadi indikator yang lebih meyakinkan mengenai keandalan dan stabilitas suatu konfigurasi *cluster* yang dihasilkan oleh *K-Means*.

2.8 Data Pengeluaran Per Kapita

Pengeluaran per kapita merupakan salah satu indikator ekonomi yang umum digunakan untuk menggambarkan tingkat kesejahteraan dan daya beli masyarakat

di suatu wilayah. Secara sederhana, pengeluaran per kapita menunjukkan rata-rata pengeluaran yang dilakukan oleh setiap individu dalam periode tertentu, baik untuk kebutuhan makanan maupun non-makanan. Data ini sering dijadikan acuan karena mampu memberikan gambaran nyata mengenai pola konsumsi masyarakat. Dalam penelitian ini, data yang digunakan berasal dari Badan Pusat Statistik (BPS), yang menyajikan informasi pengeluaran per kapita secara rinci hingga tingkat kabupaten/kota, sehingga dapat mencerminkan kondisi ekonomi masyarakat secara lebih spesifik [26].

Tingkat pengeluaran per kapita memiliki hubungan yang erat dengan daya beli masyarakat. Wilayah dengan nilai pengeluaran per kapita yang lebih tinggi umumnya menunjukkan kondisi ekonomi yang lebih baik, sehingga kemampuan konsumsi masyarakatnya juga cenderung lebih besar. Sebaliknya, wilayah dengan tingkat pengeluaran yang rendah dapat mengindikasikan keterbatasan daya beli. Perbedaan ini menjadi penting dalam konteks segmentasi, karena setiap wilayah memiliki karakteristik konsumsi yang berbeda. Dengan memahami pola tersebut, analisis segmentasi dapat membantu dalam mengelompokkan wilayah berdasarkan kondisi ekonominya, sehingga strategi atau kebijakan yang diambil dapat disesuaikan dengan kebutuhan masing-masing kelompok [27].

Selain itu, data pengeluaran per kapita dari BPS juga memiliki peran penting dalam berbagai analisis pembangunan. Data ini digunakan sebagai salah satu indikator dalam pengukuran kesejahteraan, termasuk dalam perhitungan Indeks Pembangunan Manusia (IPM), serta dalam kajian terkait kondisi ekonomi masyarakat dan perkembangan sektor usaha mikro dan kecil [1, 2]. Dalam penelitian berbasis clustering, data pengeluaran per kapita dimanfaatkan sebagai variabel utama untuk mengelompokkan wilayah berdasarkan pola konsumsi masyarakat. Namun, dalam penelitian ini data pengeluaran per kapita BPS diposisikan sebagai pembanding, bukan sebagai variabel utama dalam memetakan segmen konsumen ke wilayah. Dengan cakupan data yang luas dan terstandarisasi, data BPS tetap dapat memberikan gambaran makro yang objektif mengenai kondisi ekonomi wilayah secara umum, sebagaimana diuraikan lebih lanjut pada Subbab 4.6.5.