Jurnal INSAN (Journal of Information Systems Management Innovation)
Volume 5 No. 1 Juni 2025
E-ISSN: 2777-1385

Penentuan Harga Diskon Optimal Di Empat Provinsi Dengan K-Means Di Yayasan Achmad Zaky Foundation

Arvano Salma Fatimatus Zahra[1], Syaiful Anwar[2], M. Haddiel Fuad[3]

[1,2,3] Universitas Bina Sarana Informatika
Jalan Kramat Raya No. 98, Kec. Senen, Jakarta Pusat, Indonesia
e-mail: [1]19236091@bsi.ac.id, [2]syaiful.sfa@bsi.ac.id, [3]mhfuad@gmail.com

Artikel Info : Diterima : 05-03-2025 | Direvisi : 10-05-2025 | Disetujui : 20-06-2025

Abstrak - Penelitian ini menyelidiki pengoptimalan strategi diskon di e-commerce menggunakan algoritma pengelompokan K-Means. Berfokus pada data transaksi dari empat provinsi, penelitian ini bertujuan untuk meningkatkan pemahaman tentang proses sains data, visualisasi data, dan pembelajaran mesin. Metode K-Means, yang diterapkan secara manual dan secara komputasi menggunakan Python, menghasilkan dua klaster: tingkat pembakaran yang tinggi dan rendah. Penelitian ini menyarankan untuk menyesuaikan diskon berdasarkan nilai transaksi, yang mengarah pada pemanfaatan anggaran yang lebih baik dan peningkatan pendapatan.

Kata Kunci : *E-Commerce*, Pengoptimalan Diskon, K-Means

Abstracts *– The study investigates optimizing discount strategies in e-commerce using the K-Means clustering algorithm. Focusing on transaction data from four provinces, it aims to enhance understanding of data science processes, data visualization, and machine learning. The K-Means method, applied manually and computationally using Python, resulted in two clusters: high and low burn rates. The research suggests adjusting discounts based on transaction values, leading to better budget utilization and increased revenue.*

Keywords : E-Commerce, *Discount Optimization*, *K-Means*

PENDAHULUAN

Internet telah mengubah kehidupan sehari-hari dan cara orang berpikir, yang berdampak pada keputusan konsumen untuk membeli barang (Indartini & Rachma, 2023). Bermunculannya toko melalui *online* atau biasa disebut dengan *e-commerce* terjadi karena adanya perkembangan teknologi yang pesat yang memudahkan mereka dalam hal jual beli baik menggunakan blog, media sosial maupun *website* (Huda et al., 2023). Lembaga riset *e-commerce* dari Jerman, ECDB, menyebut Indonesia menjadi negara dengan proyeksi pertumbuhan *e-commerce* tertinggi di dunia pada 2024. Tingkat pertumbuhannya menyentuh 30,5%. Proyeksi itu lebih tinggi hampir tiga kali lipat dari rerata global yang sebesar 10,4% (Santika, 2024).

Dalam persaingan e-commerce yang semakin kompetitif, perusahaan memberikan diskon besar-besaran untuk menarik perhatian konsumen. Namun, pendekatan diskon ini belum tentu efisien karena kebutuhan dan perilaku konsumen berbeda antar daerah. Oleh karena itu, permasalahan yang diangkat adalah bagaimana menentukan strategi harga diskon yang optimal di empat provinsi Indonesia berdasarkan pengelompokan data transaksi menggunakan algoritma K-Means.

Penulis akan menggali fenomena persaingan yang semakin ketat di dunia *e-commerce*, terlihat dari berbagai platform yang berlomba-lomba untuk menarik perhatian konsumen melalui promosi yang menarik, terutama dengan memberikan harga diskon besar-besaran. Namun, penulis juga akan mempertimbangkan perbedaan harga yang terjadi di setiap provinsi, karena daerah-daerah tersebut memiliki karakteristik dan kebutuhan yang berbeda. Oleh karena itu, penulis akan fokus untuk menemukan strategi diskon yang optimal dari setiap provinsi.

Untuk menemukan solusi dari permasalahan tersebut, penulis mengikuti Studi Independen Bersertifikat (SIB) dengan trek *data science* di Startup Campus. Startup Campus merupakan program persiapan kerja terbaik di bawah naungan Yayasan Bakti Achmad Zaky. Program ini akan mendampingi peserta dari awal hingga siap kerja di perusahaan ternama dalam dan luar negeri. (S. Campus, 2024).

Tujuan dari penelitian ini adalah untuk mengelompokkan data transaksi dari empat provinsi (Jawa Barat, Bali, Nusa Tenggara Barat, dan Maluku Utara) menggunakan algoritma K-Means berdasarkan tiga fitur utama: gross_amount, discounts, dan burn rate percentage. Menentukan nilai diskon rata-rata untuk setiap klaster yang

terbentuk (high burn rate dan low burn rate). Memberikan rekomendasi nilai diskon optimal untuk masing-masing provinsi berdasarkan hasil klasterisasi.

Penelitian ini hanya menggunakan data dari empat provinsi: Jawa Barat, Bali, Nusa Tenggara Barat, dan Maluku Utara. Jumlah data transaksi di semua provinsi disamakan melalui teknik down sampling mengikuti jumlah terkecil (55.865 baris data). Pemodelan hanya menggunakan tiga fitur utama gross_amount, discounts, burn_rate_percentage. Algoritma yang digunakan adalah K-Means Clustering dengan dua klaster.

Melalui penelitian ini, diharapkan dapat memberikan kontribusi bagi para pelaku bisnis *online* dalam mengoptimalkan promosi mereka dan meningkatkan keuntungan mereka secara efektif di setiap daerah.

METODE PENELITIAN

Metode penelitian di bawah ini akan menjelaskan terkait metode pengumpulan data dan metode pengolahan data.

1. Metode Pengumpulan Data

Metode pengumpulan data yang digunakan untuk penelitian ini adalah:

A. Observasi
Penulis akan mengamati *dataset* yang disediakan oleh Startup Campus. Data ini dapat mencakup informasi tentang penjualan, produk terlaris, dan tren pembelian.. Pengamatan ini dapat meliputi analisis perilaku konsumen, preferensi konsumen terhadap produk tertentu, dan pengaruh strategi diskon terhadap keputusan pembelian.

B. Studi Literatur
Penulis akan melakukan studi literatur yang meliputi sumber-sumber seperti LMS Startup Campus, jurnal akademik, buku, artikel *online*, dan video *youtube* yang terkait dengan penelitian.

2. Metode Pengolahan Data

Dalam pengerjaan proyek akhir studi independen bersertifikat mengenai penentuan harga diskon optimal pada setiap provinsi yang terpilih, penulis menggunakan algoritma K-Means. Dalam mengolah *dataset*, penulis melakukan beberapa tahapan *data science*, antara lain:

A. *Data Collecting* (Mengumpulkan Data)
Data collecting merupakan proses menghimpun dan mengevaluasi informasi atau data dari berbagai sumber guna mencari jawaban atas masalah eksplorasi, menjawab pertanyaan, memperkirakan masalah, serta melihat tren dan peluang. Tahapan ini sangat penting dalam semua jenis eksplorasi, analisis, dan pengambilan keputusan, termasuk di bidang ilmu sosial, bisnis, dan kesehatan (Simplilearn, 2023).

B. *Data Preprocessing* (Pra Pemrosesan Data)
Data preprocessing adalah serangkaian teknik yang digunakan untuk membersihkan, mentransformasi, dan mempersiapkan data mentah sebelum dilakukan analisis lebih lanjut (Fan et al., 2021). Teknik ini meliputi langkah-langkah seperti menangani data yang hilang (*missing values*), deteksi data ekstrim (*outlier*), normalisasi data, transformasi data, ekstraksi fitur (*feature engineering*), dan pemilihan fitur (*feature selection*). *Data preprocessing* diperlukan untuk memastikan kualitas data yang baik dan hasil analisis yang akurat serta dapat diandalkan.

C. *Modeling* (Pemodelan)
Dalam proses *modeling*, data dipergunakan untuk membuat model yang dapat digunakan untuk menganalisis data, membuat prediksi, dan mencapai tujuan proyek *data science* (Costa & Aparicio, 2020).. Terdapat dua teknik pendekatan model, yaitu *supervised learning* dan *unsupervised learning*. Pendekatan model yang digunakan oleh penulis adalah *unsupervised learning* menggunakan teknik klastering dengan algoritma k-means.

Prinsip utama dari teknik ini adalah menyusun K buah partisi/pusat massa (*centroid*)/rata-rata (*mean*) dari sekumpulan data. Adapun tujuan dari pengelompokan data ini adalah untuk meminimalkan fungsi objektif yang diset dalam proses pengelompokan, yang pada umumnya berusaha meminimalkan variasi dalam suatu kelompok dan memaksimalkan variasi antar kelompok (Sulistiyawati & Supriyanto, 2021).

Langkah-langkah melakukan *clustering* dengan metode K-Means (Sulistiyawati & Supriyanto, 2021) sebagai berikut:

1) Tentukan nilai k sebagai jumlah klaster yang ingin dibentuk.
2) Inisialisasi k pusat klaster ini bisa dilakukan dengan berbagai cara, namun yang paling sering dilakukan adalah dengan cara *random* yang diambil dari data yang ada.
3) Menghitung jarak setiap data input terhadap masing – masing *centroid* menggunakan rumus jarak Euclidean (*Euclidean Distance*) hingga ditemukan jarak yang paling dekat dari setiap data dengan *centroid*. Berikut adalah persamaan *Euclidean Distance* :

$De = \sqrt{(x_i - s_i)^2 + (y_i - t_i)^2}$
De adalah *Euclidean Distance*,
I adalah banyaknya obyek,
(x, y) adalah koordinat obyek, dan
(s, t) adalah koordinat *centroid*

4) Mengklasifikasikan setiap data berdasarkan kedekatannya dengan *centroid* (jarak terkecil).
5) Memperbaharui nilai *centroid*. Nilai *centroid* baru diperoleh dari rata-rata cluster yang bersangkutan dengan menggunakan rumus:

$$v_{ij} = \frac{1}{N_i}\sum_{k=0}^{N_i} X_{kj}$$

v_{ij} adalah *centroid* rata-rata *cluster* ke-i untuk variabel ke-j,
N_i adalah jumlah data yang menjadi anggota *cluster* ke-i,
i, k adalah indeks dari *cluster*,
j adalah indeks dari variabel, dan
X_{kj} adalah nilai data ke-k yang ada di dalam *cluster* tersebut untuk variabel ke-j.

6) Melakukan perulangan dari langkah 2 hingga 5, sampai anggota tiap *cluster* tidak ada yang berubah.

D. *Evaluation* (Evaluasi)

Evaluasi adalah proses untuk melihat performa dari pola yang dihasilkan suatu algoritma yang diterapkan di dalam pemodelan tertentu.

E. *Data Visualization* (Visualisasi Data)

Menurut Statistical Analysis System, visualisasi data adalah metode untuk menampilkan informasi dengan cepat menggunakan format grafik atau gambar.. Selain itu, bisa untuk memudahkan pengambil keputusan dalam memahami konsep yang sulit atau mengidentifikasi pola-pola baru (Latifatunnisa, 2022).

HASIL DAN PEMBAHASAN

Pada bagian ini, dijelaskan hasil dan pembahasan dari penelitian penentuan harga diskon optimal di empat provinsi menggunakan k-means. Hasil dari setiap tahapan sebagai berikut:

1. Data Understanding

Dataset yang diolah merupakan *E-commerce Dataset*. *Dataset* ini disediakan oleh Startup Campus yang terdiri dari 4 jenis *dataset*. Gambaran dari setiap *dataset* dapat dijelaskan sebagai berikut:

a. ***user.csv***

Merujuk pada *https://bit.ly/userdata_e-commerce*, *dataset* ini berisi informasi terperinci mengenai pengguna, meliputi data demografis dan riwayat transaksi yang membantu memahami perilaku pengguna. *Dataset user* terdiri dari 427.486 baris data dan 8 atribut yang akan dijelaskan di Tabel *1.* Atribut Pada Dataset UserTabel 1.

Tabel 1. Atribut Pada *Dataset User*

No.	Nama Atribut	Informasi	Tipe Data
1.	*id*	6 angka pertama melambangkan kode provinsi, kabupaten, dan kecamatan	*Object*
2.	*full_name*	Nama lengkap	*Object*
3.	*gender*	Jenis kelamin	*Object*
4.	*money_spent*	Jumlah uang yang dibelanjakan oleh *user*	*Float*
5.	*refund*	Jumlah uang yang dikembalikan	*Float*
6.	*wallet_balance*	Jumlah uang yang disimpan di dompet digital	*Float*
7.	*join_date*	Tanggal *user* bergabung dengan *e-commerce*	*Object*
8.	*birth*	Tanggal lahir *user*	*Object*

b. **location_reference.csv**

Merujuk pada *https://bit.ly/locationdata_e-commerce*, *dataset* ini menyediakan rinci lokasi geografis yang penting untuk analisis regional. Rincian lokasi di-*breakdown* ke level provinsi, kabupaten/kota, dan kecamatan. *Dataset location_reference* terdiri 7407 baris data dan 6 atribut yang akan dijelaskan pada Tabel 2.

Tabel 2. Atribut Pada *Dataset* Lokasi

No.	Nama Atribut	Informasi	Tipe Data
1.	nama provinsi	Berisi 39 provinsi di Indonesia	*Object*
2.	kode provinsi	Kode unik setiap provinsi	*Integer*
3.	nama kabupaten	Berisi nama kabupaten dari setiap provinsi	*Object*
4.	kode kabupaten	Kode unik setiap kabupaten	*Object*

5.	nama kecamatan	Berisi nama kecamatan dari setiap kabupaten	*Object*
6.	kode kecamatan	Kode unik setiap kecamatan	*Object*

c. **Folder Trx**

Merujuk pada *https://bit.ly/trxdata_e-commerce*, *dataset* trx memuat catatan historis transaksi yang mendetail untuk menganalisis tren pembelian dan pengaruh promosi di setiap provinsi. *Dataset* ini memiliki 7 atribut yang dijelaskan pada Tabel 3.

Tabel 3. Atribut Pada *Dataset* Trx

No.	Nama Atribut	Informasi	Tipe Data
1.	*id*	Kode unik setiap transaksi	*Object*
2.	*user_id*	Kode unik *user*	*Object*
3.	*product_id*	Kode unik produk	*Integer*
4.	*gross_amount*	Berisi jumlah penghasilan kotor	*Float*
5.	*discounts*	Berisi harga diskon yang digunakan	*Float*
6.	*transaction_date*	Berisi tanggal *user* bertransaksi	*Object*

Fokus penelitian penulis hanya mengambil 4 provinsi saja, yaitu Jawa Barat, Bali, Nusa Tenggara Barat, dan Maluku Utara. Rincian jumlah data dari keempat provinsi tersebut bisa terlihat pada Tabel 4.

Tabel 4. Rincian Provinsi Terpilih

Kode Provinsi	Provinsi	Jumlah Baris
32	Jawa Barat	1.919.165
51	Bali	80.851
52	Nusa Tenggara Barat	356.694
82	Maluku Utara	55.865

Karena perbedaan jumlah baris yang sangat banyak, penulis melakukan *down sampling* mengikuti jumlah baris yang paling sedikit, yaitu sejumlah 55.865 baris. Nantinya, *dataset* transaksi di provinsi Jawa Barat, Bali, dan Nusa Tenggara Barat akan dikurangi baris datanya mengikuti jumlah baris data di provinsi Maluku Utara.

d. **product_reference.csv**

Merujuk pada https://bit.ly/productdata_e-commerce, *dataset* ini menawarkan gambaran lengkap tentang jajaran produk untuk menghubungkan preferensi regional dengan tawaran spesifik. *Dataset product_reference* memiliki 18 jenis produk dengan 2 atribut yaitu *id* dan *product_name*. Rincian dataset *product_reference* dapat terlihat pada Tabel 5.

Tabel 5. Rincian *Dataset product_reference*

id	product_name
1	Man Fashion
2	Woman Fashion
3	Food & Drink
4	Ride Hailing
5	Keperluan Rumah Tangga
6	Travel
7	Keperluan Anak
8	Elektronik
9	Other
10	Transportasi (Kereta, Pesawat, Kapal)
11	Top Up Game
12	Otomotif
13	Pulsa
14	Kesehatan
15	Investasi
16	Sewa Motor/Mobil
17	Hotel
18	Tagihan (WiFi, PLN)

2. Data Preprocessing

Pada tahap *data preprocessing*, terdapat beberapa kegiatan yang dilakukan seperti melakukan *formatting data*, menangani *missing values*, membersihkan *data noisy*, menggabungkan *dataset*, dan normalisasi data.

a. Penanganan *Missing Values*

Data yang hilang ditemukan di dalam *dataset* trx, tepatnya pada atribut *gross_amount* dan *discounts*. Pada atribut *gross_amount*, *missing values* dihapus karena jumlahnya sedikit dan tidak

menyebabkan bias yang signifikan. Sedangkan pada atribut *discounts*, *missing values* diisi dengan nilai 0, karena asumsinya transaksi yang diskonnya kosong artinya transaksi tidak menggunakan diskon.

b. Penanganan Data *Noise*

Pada *dataset* trx juga ditemukan data *noisy* berupa nilai negatif, tepatnya di atribut *gross_amount* dan *discounts*. Nilai negatif tersebut akhirnya dihapus untuk memastikan validitas dan akurasi data. Selain itu, ditemukan pula nilai negatif pada *dataset user*, tepatnya di atribut *wallet_balance* dan *refund*. Penanganan yang dilakukan sama seperti pada *handling noise data* di *dataset* trx.

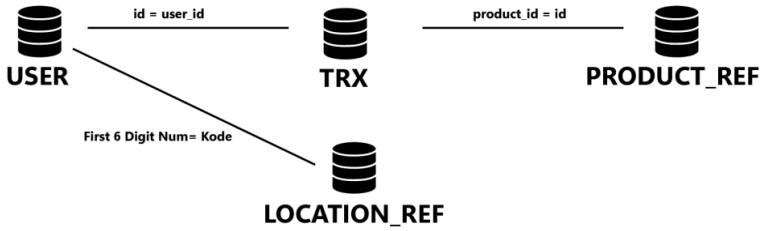
c. Pemformatan Data

Selanjutnya, dilakukan proses *formatting data*. Hampir seluruh *dataset* dilakukan *formatting data* karena banyak tipe data yang kurang sesuai. Seperti pada *dataset location_reference*, kode kabupaten dan kecamatan seharusnya memiliki tipe data *integer* karena isinya berupa angka. Akan tetapi, di *dataset* awal tipe datanya berupa *object*. Sehingga perlu dilakukan pengubahan tipe data.

d. Penggabungan *Dataset*

Setelah pembersihan data, langkah pertama adalah menggabungkan dataset transaksi dari empat provinsi. Kemudian, dataset transaksi digabungkan dengan dataset produk menggunakan *left join*, dan id_produk dihapus untuk menghindari duplikasi. Selanjutnya, *dataset user* digabungkan dengan *dataset* lokasi melalui ekstraksi kode kecamatan dari *id user*. Terakhir, *dataset* transaksi-produk digabungkan dengan dataset user-lokasi, menghasilkan *dataset* siap olah.

Ilustrasi penggabungan *dataset* dapat dilihat pada Gambar 1.



Sumber: (M. B. 6 S. Campus, 2024)

Gambar 1. Ilustrasi Penggabungan *Dataset*

e. *Feature Engineering*

Dataset yang sudah bersih akan dibuat fitur-fitur baru. Tahapan ini disebut dengan *feature engineering*. Ada berbagai fitur baru yang dibuat seperti *age, age status, money spent status, refund status, wallet balance status, transaction date detail (day, day name, date, month, month name, year), discount flag, discount burn rate percentae,* dan *burn rate z-score*. Untuk penjelasan fitur-fitur baru yang sudah dibuat dapat dilihat pada

Tabel 6. Penjelasan Atribut Baru Dari *Feature Engineering*

No.	Nama Atribut	Informasi	Tipe Data
1.	*age*	Usia *user*	*Integer*
2.	*age_status*	Status usia (*Young, Adult, Middle-Aged, Senior*)	*Object*
3.	*money_spent_status*	Status uang yang dihabiskan (*Low, Medium, High, Very High*)	*Object*
4.	*refund_status*	Status pengembalian uang (*Low, Medium, High, Very High*)	*Object*
5.	*wallet_balance_status*	Status uang yang tersimpan di dompet digital (*Low, Medium, High, Very High*)	*Object*
6.	*day*	Hari transaksi dalam angka	*Integer*
7.	*day_name*	1: *Tuesday,* 2: *Wednesday*, 3: , 4: *Friday*, 5: *Saturday*, 6: *Sunday*	*Object*
8.	*date*	Tanggal transaksi	*Integer*
9.	*month*	Bulan transaksi dalam angka	*Integer*
10.	*month_name*	1: *January*, 2: *February*, 3: *March*, 4: *April*, 5: *May*, 6: *June*, 7: *July*, 8: *August*, 9: *September*, 10: *October*, 11: *November*, 12: *December*	*Object*
11.	*year*	Tahun transaksi	*Integer*
12.	*flag_discount*	Untuk menandai transaksi memiliki diskon atau tidak	*Integer*
12.	*flag_discount_status*	0: *No Discount*, 1: *Has Discount*	*Object*
13.	*burn_rate_percentage*	Persentase yang menandakan seberapa besar diskon yang diberikan relatif terhadap nilai transaksi total.	*Float*
14.	*z_score_burn_rate*	Ukuran statistik yang menunjukkan seberapa jauh nilai *burn rate* dari rata-rata (*mean*) dalam satuan standar deviasi	*Float*

f. *Feature Selection*

Tahapan selanjutnya yaitu dilakukan proses pemilihan fitur untuk pemodelan. Dalam tahapan ini, penulis menggunakan *heatmap correlation* untuk melihat korelasi antar fitur dan analisis VIF untuk

mengatasi masalah multikolinearitas dengan menghilangkan fitur yang saling berkorelasi tinggi. Sehingga didapatkan bahwa fitur yang relevan untuk pemodelan *clustering* adalah *gross_amount, discounts*, dan *burn rate percentage*.

Hasil akhir dari tahap *data preprocessing* adalah didapatkan data bersih yang digunakan untuk pemodelan sebanyak 219.677 data dengan 3 fitur yang akan digunakan, yaitu *gross_amount, discounts,* dan *burn rate percentage*.

g. Normalisasi Data

Tahap terakhir yaitu proses normalisasi data menggunakan teknik *min-max normalization*. Proses ini dilakukan supaya hubungan antar data tetap sama antara nilai asli dengan nilai normalisasi. Teknik ini akan menghasilkan data dengan rentang 0 sampai 1, sehingga bisa meminimalkan *outlier* data.

Normalisasi dilakukan di ketiga fitur dengan rumus:

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

x_{scaled} adalah data hasil normalisasi.
x adalah data awal sebelum normalisasi.
x_{min} adalah data yang memiliki nilai terkecil.
x_{max} adalah data yang memiliki nilai terbesar.

Perbandingan data sebelum dan setelah normalisasi dapat dilihat pada Tabel 7 dan **Error! Reference source not found.**. Penulis mencantumkan 10 data teratas dari keseluruhan data penelitian.

Tabel 7. Data Sebelum dan Sesudah Normalisasi

	discounts	*gross_amount*	*burn_rate_percentage*		*discounts*	*gross_amount*	*burn_rate_percentage*
	0	273900	0		0	0,122402467	0
S	30000	58800	51,02040816	**S**	0,03386769	0,026276981	0,510204082
E	0	59900	0	**E**	0	0,026768557	0
B	10700	41000	26,09756098	**S**	0,012079476	0,018322385	0,26097561
E	0	2700	0	**U**	0	0,001206596	0
L	0	6000	0	**D**	0	0,002681325	0
U	0	28100	0	**A**	0	0,012557537	0
M	0	282500	0	**H**	0	0,126245699	0
	24100	24100	100		0,027207044	0,010769987	1
	0	24900	0		0	0,011127497	0

3. Modeling

Untuk menentukan harga diskon yang optimal di setiap provinsi, pemodelan yang digunakan adalah *clustering* menggunakan algoritma K-Means. Data memiliki 219.677 baris dengan 3 atribut yang terpilih untuk pemodelan, yaitu *discounts, gross_amount,* dan *burn_rate_percentage*.

a. Penghitungan Manual Dengan Microsoft Excel

Penulis mencoba untuk mengimplementasikan algoritma K-Means secara manual di Microsoft Excel dengan jumlah klaster sebanyak 2 klaster. Tahapan pembentukan klaster dengan K-Means untuk penghitungan secara manual dijabarkan sebagai berikut:

1) Menentukan Jumlah Klaster (k)

Pada penelitian ini, penulis menentukan klaster yang akan dibentuk sebanyak 2 klaster, yaitu klaster dengan *burn rate* yang tinggi dan klaster dengan *burn rate* yang rendah.

2) Menentukan Pusat Klaster (*Centroid*)

Pada tahapan penentuan *centroid* awal, *centroid* dipilih secara acak. Rincian *centroid* yang terpilih tercantum pada Tabel 8.

Tabel 8. *Centroid* Awal

Cluster	Titik *Cluster*	*discounts*	*gross_amount*	*burn_rate_percentage*
1	Data ke-1	0	0,122402467	0
2	Data ke-9	0,027207044	0,010769987	1

3) Menghitung Jarak Data ke *Centroid* (Iterasi 1)

Setiap data harus dihitung jaraknya terhadap titik pusat klasternya. Penelitian ini menggunakan teknik menghitung jarak Euclidean (*Euclidean Distance*). Contohnya pada data ke-1, penghitungan jarak antara data dengan klaster 1 adalah:

$De = \sqrt{(0 - 0)^2 + (0{,}122402467 - 0{,}122402467)^2 + (0 - 0)^2} = 0$

Sedangkan untuk penghitungan jarak antara data dengan klaster 2 adalah:

$De = \sqrt{(0 - 0{,}027207044)^2 + (0{,}122402467 - 0{,}010769987)^2 + (0 - 1)^2} = 0$

4) Mengelompokkan Data Sesuai Jarak yang Terpendek ke *Centroid*

Untuk menentukan pembagian klaster, bisa dilihat dari jarak yang terpendek ke *centroid*. Jika semakin dekat data dengan titik pusat klaster, maka data masuk ke satu klaster dengan titik pusat tersebut. 10 data teratas dari hasil penghitungan tercantum pada Tabel 9.

Tabel 9. Hasil Iterasi 1

Jarak C1	Jarak C2	*Cluster*
0	1,006579373	1
0,520283898	0,490086597	2
0,09563391	1,000497965	1
0,281223836	0,739217783	1
0,121195871	1,000415754	1
0,119721142	1,000402744	1
0,10984493	1,00037164	1
0,003843232	1,007012842	1
1,006579373	0	2
0,11127497	1,000370107	1

Jumlah data pada setiap klaster tercantum pada Tabel 10.

Tabel 10. Jumlah Data Setiap Klaster di Iterasi 1

Cluster	∑ Anggota
1	183662
2	36015

Pada penelitian ini, penghitungan dilanjutkan sampai ke iterasi 8 karena untuk mengecek kesesuaian jumlah data.

5) Mengelompokkan Data Sesuai Jarak yang Terpendek ke *Centroid* Pada Iterasi 8

10 data teratas dari hasil penghitungan iterasi 8 tercantum pada Tabel 11.

Tabel 11. Hasil Iterasi 8

Jarak C1	Jarak C2	*Cluster*
0,098533597	0,862577692	1
0,482949447	0,345443202	2
0,028485409	0,856288225	1
0,233133912	0,594995704	1
0,039130875	0,856409906	1
0,038133659	0,856382147	1
0,032408062	0,856261687	1
0,102219101	0,863051007	1
0,972244019	0,144818186	2
0,033116256	0,856272077	1

Jumlah data pada setiap klaster tercantum pada Tabel 12.

Tabel 12. Jumlah Data Setiap Klaster di Iterasi 8

Cluster	∑ Anggota Iterasi 7	∑ Anggota Iterasi 8
1	181312	181312
2	38365	38365

Kesimpulan dari penghitungan manual K-Means adalah iterasi dilakukan sebanyak 8 kali dengan klaster 1 yang melambangkan tingkat *burn rate* tinggi memiliki 181.312 data dan klaster 2 yang melambangkan tingkat *burn rate* rendah memiliki 38.365 data.

b. Pemodelan Komputasi dengan Python Via Google Colab

Pada bagian ini akan membahas tentang penerapan algoritma K-Means dalam menganalisis data transaksi menggunakan Python di Google Colab. Penulis akan mengelompokkan data transaksi berdasarkan fitur-fitur seperti *discounts, gros_amount,* dan *burn_rate_percentage*. Langkah-langkah yang dilakukan adalah sebagai berikut:

1) Pertama, penulis memilih fitur-fitur yang akan digunakan untuk *clustering* dengan membuat objek baru `*features*` yang hanya berisi kolom `*discounts*`, `*gross_amount*`, dan `*burn_rate_percentage*` dari *dataset* `df. Kode program ditulis sebagai berikut:

```
features = df[['discounts', 'gross_amount', 'burn_rate_percentage']]
```

2) Selanjutnya, model K-Means diinisiasi dengan jumlah klaster yang diinginkan, dalam contoh ini adalah 2 klaster. Parameter `*random_state*` digunakan untuk membuat hasil *clustering reproducible*. Kode program ditulis sebagai berikut:

```
kmeans = KMeans(n_clusters=2, random_state=42)
```

3) Kemudian, model K-Means akan dilatih dengan data `*features*` menggunakan metode `*fit()*`. Kode program ditulis sebagai berikut:

```
kmeans.fit(features)
```

Pada tahap ini, algoritma K-Means akan mencari pusat (*centroid*) dari setiap klaster berdasarkan data fitur yang diberikan. Proses ini dilakukan secara iteratif hingga pusat klaster tidak berubah lagi atau mencapai jumlah iterasi maksimum.

4) Setelah model K-Means dilatih, label klaster ditambahkan ke dalam *dataset* `df` dengan membuat kolom baru *'cluster'* yang berisi label klaster untuk setiap baris data. Kode program ditulis sebagai

berikut:

```
df['cluster'] = kmeans.labels_
```

Label klaster ini merepresentasikan kelompok mana setiap baris data akan ditempatkan berdasarkan hasil *clustering*.

5) Keseluruhan kode program untuk pemodelan dengan k-means sebagai berikut:

```
features = df[['discounts', 'gross_amount', 'burn_rate_percentage']]
# Inisialisasi model K-Means dengan jumlah klaster sebanyak 2
kmeans = KMeans(n_clusters=2, random_state=42)
# Melakukan clustering
kmeans.fit(features)
# Menambahkan label klaster ke dalam data
df['cluster'] = kmeans.labels_
df
```

4. Evaluasi Model

Model akan dievaluasi menggunakan nilai DBI. DBI atau *Davies-Bouldin Index* digunakan untuk mengukur kekompakan dan pemisahan antara klaster, yang mana nilai yang lebih rendah menunjukkan *clustering* yang lebih baik. Penghitungan akan dilakukan baik dengan Excel maupun Python.

a. Penghitungan Manual Dengan Microsoft Excel

Untuk menghitung nilai DBI, tahap pertama yang harus dilakukan adalah menghitung nilai SSW. SSW yang merupakan kepanjangan dari *Sum of Square Within cluster* digunakan untuk mengetahui keterikatan anggota klaster di dalam satu klaster. Rumus perhitungan nilai SSW sebagai berikut:

$$SSW_i = \frac{1}{m_i}\sum_{j=i}^{m_i} d(x_j, c_i)$$

m_i = jumlah data dalam klaster ke-i
x = data dalam klaster
d(x,c) = jarak data ke *centroid*
x_j = data pada klaster tersebut
c_i = *centroid* klaster ke-i

Karena jumlah klaster yang terbentuk ada 2 klaster, maka masing-masing klaster perlu dihitung SSW-nya. Hasil penghitungan SSW dari kedua klaster dicantumkan pada Tabel 13.

Tabel 13. Hasil Penghitungan SSW

SSW 1	SSW 2
0,06546272	0,830295601

Tahap selanjutnya adalah menghitung nilai SSB. SSB yang merupakan kepanjangan dari *Sum of Square Between cluster* digunakan untuk mengetahui perbedaan antara satu klaster dengan klaster lainnya. Rumus perhitungan nilai SSB sebagai berikut:

$$SSB_{ij} = d(c_i, c_j)$$

c_i = klaster satu
c_j = klaster lainnya
$d(c_i,c_j)$ = jarak antara *centroid* satu dengan lainnya

Karena jumlah klaster yang terbentuk ada 2 klaster, maka nilai SSB yang dihasilkan hanya satu saja, karena hanya melihat hubungan antara klaster 1 dan klaster 2. Hasil perhitungan SSB dari kedua klaster dicantumkan pada Tabel 14.

Tabel 14. Hasil Penghitungan SSB

	Centroid	
SSB	1	2
1	0	0,827985724
2	0,827985724	0

Tahap ketiga adalah mencari rasio untuk mengetahui seberapa bagus nilai perbandingan antara klaster satu dengan klaster lainnya.

Rumus perhitungan rasio sebagai berikut:

$$R_{ij} = \frac{SSW_i + SSW_j}{SSB_{ij}}$$

R_{ij} = Rasio antara klaster i dan j
SSW_i = nilai SSW pada klaster 1
SSW_j = nilai SSW pada klaster 2
SSB_{ij} = separasi dari klaster 1 dan 2

Jika diterapkan menggunakan rumus di atas, perhitungannya sebagai berikut:

$$R_{ij} = \frac{0{,}06546272 + 0{,}830295601}{0{,}827985724} = 1{,}081852375$$

Barulah di tahap terakhir ini nilai DBI bisa ditentukan, dengan rumus sebagai berikut:

$$DBI = \frac{1}{k}\sum_{i=1}^{k} max_{i \neq j}(R_{ij})$$

k = Klaster yang ada
R_{ij} = Rasio antara klaster i dan j
max = Rasio antar klaster yang terbesar
Jika diterapkan menggunakan rumus di atas, perhitungannya sebagai berikut:

$$DBI = \frac{1}{2} \times 1{,}081852375 = 0{,}540926187$$

b. Pemodelan Komputasi dengan Python Via Google Colab

Evaluasi model dengan bahasa pemrograman Python di Google Colab akan memeriksa nilai DBI dan Silhouette Score. Berbagai jumlah klaster dibandingkan untuk mendapatkan nilai DBI dan Silhouette Score yang terbaik. Keseluruhan kode program untuk evaluasi model sebagai berikut:

```
# Coba berbagai jumlah kluster
range_n_clusters = [2, 3, 4, 5, 6, 7]
best_n_clusters = 0
best_silhouette_score = -1
best_dbi_score = float('inf')
best_clusters = None
for n_clusters in range_n_clusters:
    kmeans = KMeans(n_clusters=n_clusters, random_state=42)
    clusters = kmeans.fit_predict(features)
    silhouette_avg = silhouette_score(features, clusters)
    dbi_score = davies_bouldin_score(features, clusters)
    print(f'For n_clusters = {n_clusters}, Silhouette Score = {silhouette_avg}, DBI = {dbi_score}')
    if silhouette_avg > best_silhouette_score:
        best_n_clusters = n_clusters
        best_silhouette_score = silhouette_avg
        best_dbi_score = dbi_score
        best_clusters = clusters
# Gunakan jumlah kluster terbaik
modeling_df['cluster'] = best_clusters
# Menampilkan hasil clustering
print(f'\nBest number of clusters: {best_n_clusters}')
print(f'Best Silhouette Score: {best_silhouette_score}')
print(f'Best DBI: {best_dbi_score}')
print(modeling_df)
# Interpretasi: Menganalisis setiap cluster untuk menentukan strategi diskon yang tepat
# Menampilkan statistik deskriptif untuk setiap cluster
for cluster in modeling_df['cluster'].unique():
    print(f"\nCluster {cluster}")
    print(modeling_df[modeling_df['cluster'] == cluster].describe())
```

Gambar 2. Kode program evaluasi model

Hasil dari evaluasi model yang dilakukan di Google Colab dapat terlihat pada Tabel 15.

Tabel 15. Hasil Evaluasi Model Menggunakan Google Colab

Jumlah Klaster	Nilai DBI	Silhouette Score
2	0.5330153652610891	0.8481181462974069
3	0.5877981638364146	0.7674725227500699
4	0.6146986834182813	0.7208375412536774
5	0.6556533845423488	0.6809920001211783
6	0.7003585369274026	0.6461465528073544
7	0.7215526184534607	0.6284222204208759

Nilai DBI untuk performa model yang bagus adalah yang mendekati nilai 0, sedangkan Silhouette Score untuk performa model yang bagus adalah yang mendekati nilai 1. Jadi, dari hasil perhitungan dengan Excel maupun komputasi dengan Google Colab, dapat disimpulkan bahwa data yang memiliki 2 klaster memiliki performa model yang lebih baik dibandingkan jumlah klaster yang lebih banyak.

5. Hasil Rekomendasi Harga Diskon Dari 4 Provinsi Terpilih

Sebagai hasil akhir dari penelitian untuk menentukan harga diskon optimal pada setiap provinsi yang terpilih berdasarkan klasterisasi empat provinsi menggunakan k-means, penulis menambahkan informasi mengenai rekomendasi harga diskon untuk setiap klaster di masing-masing provinsi.

Business Recomendations

	Province Name	Cluster	Avarage Discount Value Suggested
1.	BALI	Low Burn Rate	Rp48.4K
2.	MALUKU UTARA	Low Burn Rate	Rp13.91K
3.	NUSA TENGGARA BARAT	Low Burn Rate	Rp21.41K
4.	JAWA BARAT	Low Burn Rate	Rp44.9K
5.	JAWA BARAT	High Burn Rate	Rp8.31K
6.	NUSA TENGGARA BARAT	High Burn Rate	Rp4.84K
7.	MALUKU UTARA	High Burn Rate	Rp3.12K
8.	BALI	High Burn Rate	Rp5.26K

1 - 8 / 8

Gambar 3. Rekomendasi Harga Diskon

KESIMPULAN

Kesimpulan dari penelitian ini menunjukkan bahwa pemodelan harga diskon optimal di setiap provinsi menggunakan algoritma K-Means dengan tiga atribut berhasil membentuk dua klaster, yaitu klaster high burn rate dan low burn rate. Klaster high burn rate menunjukkan volume transaksi dan penghasilan yang lebih tinggi, tetapi dengan penggunaan anggaran diskon yang besar dan efektivitas keuntungan yang lebih rendah (4,5 kali lipat). Sebaliknya, klaster low burn rate menghasilkan keuntungan yang lebih tinggi secara proporsional (7 kali lipat dari anggaran diskon) meskipun dengan jumlah transaksi dan pengeluaran per pembeli yang lebih rendah. Oleh karena itu, evaluasi strategi diskon diperlukan agar penggunaan anggaran lebih optimal dan dapat meningkatkan efisiensi serta penghasilan dari setiap transaksi.

REFERENSI

Campus, M. B. 6 S. (2024). *Guideline Final Project Data Science & AI Startup Campus Batch 6*. 1–47.

Campus, S. (2024). *Startup Campus*.

Costa, C. J., & Aparicio, J. T. (2020). POST-DS: A Methodology to Boost Data Science. *Iberian Conference on Information Systems and Technologies, CISTI*, *2020-June*(June), 24–27. https://doi.org/10.23919/CISTI49556.2020.9140932

Fan, C., Chen, M., Wang, X., Wang, J., & Huang, B. (2021). A Review on Data Preprocessing Techniques Toward Efficient and Reliable Knowledge Discovery From Building Operational Data. *Frontiers in Energy Research*, *9*(March), 1–17. https://doi.org/10.3389/fenrg.2021.652801

Huda, M. Q., Kusumaningtyas, R. H., Aini, Q., Hidayah, N. A., & Yulian, I. (2023). Analisis Validitas dan Reliabilitas Sosial Budaya dan Organisasi terhadap Adopsi E-Commerce UMKM Tangerang Selatan. *Applied Information System and Management (AISM)*, *6*(1), 1–6. https://doi.org/10.15408/aism.v6i1.25198

Indartini, M., & Rachma, N. (2023). Analisis Pengaruh Website Design Quality, E-Service Quality Dan Online Customer Review Terhadap Keputusan Pembelian Konsumen Pada E-Commerce Sociolla. *JAMER : Jurnal Akuntansi Merdeka*, *4*(1), 11–21. https://doi.org/10.33319/jamer.v4i1.94

Latifatunnisa, H. (2022). *Visualisasi Data: Jenis, Fungsi Penting, Contoh, dan Tools*. RevoUpedia.

Santika, E. F. (2024). *ECDB: Proyeksi Pertumbuhan e-Commerce Indonesia Tertinggi Sedunia pada 2024*.

Simplilearn. (2023). *What Is Data Collection: Methods, Types, Tools*.

Sulistiyawati, A., & Supriyanto, E. (2021). Implementasi Algoritma K-means Clustring dalam Penetuan Siswa Kelas Unggulan. *Jurnal Tekno Kompak*, *15*(2), 25. https://doi.org/10.33365/jtk.v15i2.1162