PREDIKSI PELANGGAN CHURN MENGGUNAKAN ALGORITMA RANDOM FOREST DENGAN SEGMENTASI K-MEANS DALAM INDUSTRI RETAIL OTOMOTIF

(Studi Kasus: KALLA TOYOTA)

MAYLINDA EKA CHRISTY
D121201050

PROGRAM STUDI SARJANA TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS HASANUDDIN
GOWA
2024

PREDIKSI PELANGGAN CHURN MENGGUNAKAN ALGORITMA RANDOM FOREST DENGAN SEGMENTASI K-MEANS DALAM INDUSTRI RETAIL OTOMOTIF

(Studi Kasus: KALLA TOYOTA)

MAYLINDA EKA CHRISTY

D121 20 1050

PROGRAM STUDI SARJANA TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS HASANUDDIN

GOWA

2024

PREDIKSI PELANGGAN CHURN MENGGUNAKAN ALGORITMA RANDOM FOREST DENGAN SEGMENTASI K-MEANS DALAM INDUSTRI RETAIL OTOMOTIF

(Studi Kasus: KALLA TOYOTA)

MAYLINDA EKA CHRISTY

D121 20 1050

Skripsi

sebagai salah satu syarat untuk mencapai gelar sarjana

Program Studi Teknik Informatika

pada

PROGRAM STUDI TEKNIK INFORMATIKA

DEPARTEMEN TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS HASANUDDIN

GOWA

2024

LEMBAR PENGESAHAN SKRIPSI

PREDIKSI PELANGGAN CHURN MENGGUNAKAN ALGORITMA RANDOM FOREST DENGAN SEGMENTASI K-MEANS DALAM INDUSTRI RETAIL OTOMOTIF

(Studi Kasus: KALLA TOYOTA)

MAYLINDA EKA CHRISTY
D121201050

Skripsi,

telah dipertahankan di depan Panitia Ujian Sarjana pada 2 Oktober 2024 dan dinyatakan telah memenuhi syarat kelulusan
Pada

Program Studi Sarjana Teknik Informatika
Departemen Teknik Informatika
Fakultas Teknik
Universitas Hasanuddin
Gowa

Mengesahkan:	Mengetahui:
Pembimbing	Ketua Program Studi
Dr. Ir. Ingrid Nurtanio, M.T.	Prof. Dr. Ir. Indrabayu, S. T., M. T., M. Bus., Sys., IPM, AsEAN. Eng.
NIP 19610813 198811 2 001	NIP 19750716 200212 1 004

PERNYATAAN KEASLIAN SKRIPSI DAN PELIMPAHAN HAK CIPTA

Dengan ini saya menyatakan bahwa, skripsi berjudul "Prediksi pelanggan churn menggunakan algoritma random forest dengan segmentasi k-means dalam industri retail otomotif (studi kasus: KALLA TOYOTA)" adalah benar karya saya dengan arahan dari pembimbing ibu Dr. Ir. Ingrid Nurtanio, M.T. sebagai Pembimbing Utama. Karya ilmiah ini belum diajukan dan tidak sedang diajukan dalam bentuk apa pun kepada perguruan tinggi mana pun. Sumber informasi yang berasal atau dikutip dari karya yang diterbitkan maupun tidak diterbitkan dari penulis lain telah disebutkan dalam teks dan dicantumkan dalam Daftar Pustaka skripsi ini. Apabila di kemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan skripsi ini adalah karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut berdasarkan aturan yang berlaku.

Dengan ini saya melimpahkan hak cipta (hak ekonomis) dari karya tulis saya berupa skripsi ini kepada Universitas Hasanuddin.

Makassar, 22 Agustus 2024

...nda Eka Christy

NIM D121201050

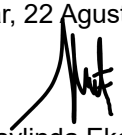
Ucapan Terima Kasih

Puji syukur ke hadirat Tuhan Yesus Kristus, atas berkat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul "Prediksi Pelanggan *Churn* Menggunakan Algoritma Random Forest Dengan Segmentasi K-Means Dalam Industri Retail Otomotif (Studi Kasus: Kalla Toyota)" ini dengan baik sebagai salah satu syarat dalam menyelesaikan jenjang Strata-1 di Departemen Teknik Informatika, Fakultas Teknik, Universitas Hasanuddin. Banyak pihak yang telah memberikan dukungan dan bantuan selama proses ini. Oleh karena itu, dengan segala kerendahan hati, penulis ingin mengucapkan terima kasih kepada:

1. Tuhan Yesus Kristus, atas segala berkat dan karunia-Nya yang senantiasa tercurah kepada penulis.
2. Keluarga penulis, Bapak Daud Sugiarto dan Ibu Martina Lapu, yang selalu mendoakan, memberikan semangat, serta dukungan materi dan non-materi yang tiada henti kepada penulis.
3. Ibu Dr. Ir. Ingrid Nurtanio, M.T., selaku pembimbing, yang dengan sabar dan penuh perhatian telah meluangkan waktu, tenaga, pikiran, dan fasilitas untuk membimbing penulis dalam menyelesaikan tugas akhir ini.
4. Segenap dosen dan staf Departemen Teknik Informatika, Fakultas Teknik, Universitas Hasanuddin, yang telah membantu dan membimbing penulis selama masa perkuliahan.
5. Saudara penulis, Grescia dan Anastasia, yang senantiasa menghibur dan memberikan semangat kepada penulis dalam menyelesaikan tugas akhir ini.
6. Teman-teman Ngok, Mahasiswa Tanpa Lab, Mahasiswa Lab Seis, Rekan MSIB dan PPGT JBK yang memberi dukungan dan membantu penulis dalam menyelesaikan tugas akhir ini.
7. Jane, Muti, Thesa, Miqa, paula, Tiwi dan teman-teman dekat penulis yang tidak dapat disebutkan satu per satu atas kebersamaan, bantuan, dan semangat yang diberikan.
8. Seluruh pihak yang tidak dapat disebutkan satu per satu yang telah meluangkan waktu, tenaga, dan pikiran selama penyusunan skripsi ini.
9. *Last but not least, I wanna thank me, I wanna thank me for believing in me, I wanna thank me for doing all this hard work, I wanna thank me for having no days off, I wanna thank me for never quitting.*

Penulis menyadari bahwa skripsi ini masih jauh dari sempurna. Oleh karena itu, kritik dan saran yang membangun sangat penulis harapkan untuk perbaikan di masa yang akan datang. Semoga tugas akhir ini dapat memberikan pengetahuan dan manfaat bagi penulis dan pembaca.

Makassar, 22 Agustus 2024



Maylinda Eka Christy

ABSTRAK

MAYLINDA EKA CHRISTY. **Prediksi Pelanggan *Churn* Menggunakan Algoritma Random Forest Dengan Segmentasi K-Means Dalam Industri Retail Otomotif (Studi Kasus: Kalla Toyota)** (dibimbing oleh Ingrid Nurtanio).

Latar Belakang. Kalla Toyota, yang telah beroperasi hampir 70 tahun dan menjadi bagian penting dalam industri otomotif, masih belum secara efisien menerapkan segmentasi pelanggan pada bengkelnya. Padahal, segmentasi yang tepat terbukti meningkatkan loyalitas dan mengurangi *churn* (perpindahan pelanggan), sebagaimana diakui oleh 90% pelaku industri. **Tujuan**. Penelitian ini bertujuan untuk mengidentifikasi segmen pelanggan Kalla Toyota menggunakan K-Means serta memprediksi kelas pelanggan dengan algoritma *Random Forest*. **Metode**. Penelitian dibagi menjadi dua tahap: 1) Segmentasi pelanggan menggunakan CLTV (Customer lifetime value) yaitu metrik yang mengukur nilai finansial dari hubungan jangka panjang antara bisnis dan pelanggan dan K-Means; 2) Prediksi kelas pelanggan berdasarkan segmentasi dengan *Random Forest*. **Hasil.** Segmentasi pelanggan menghasilkan lima kelompok: 1) Potensi *Churn* Tinggi; 2) Pelanggan Baru; 3) Potensi Loyalitas Tinggi; 4) *Churn* Menengah ke Atas; 5) Loyalitas Menengah ke Atas. Nilai Silhouette Score sebesar 0.53 mendukung validitas segmentasi, sementara akurasi prediksi kelas menggunakan *Random Forest* mencapai 87%. **Kesimpulan**. Segmentasi menggunakan K-Means dan prediksi kelas dengan *Random Forest* efektif dalam mengidentifikasi potensi *churn* pelanggan, membantu perusahaan mengoptimalkan strategi retensi.

Kata Kunci: otomotif; segmentasi pelanggan; *churn*; *k-means*; cltv; *random forest*

ABSTRACT

MAYLINDA EKA CHRISTY. **Customer Churn Prediction Using the Random Forest Algorithm with K-Means Segmentation in the Automotive Retail Industry (Case Study: Kalla Toyota*)*** (*supervised by* Ingrid Nurtanio).

Background. Kalla Toyota, a company that has been operating for nearly 70 years and plays a significant role in the automotive industry, has yet to efficiently implement customer segmentation in its service workshops. Effective segmentation has been proven to enhance customer loyalty and reduce churn, as recognized by 90% of industry professionals. **Aim**. This study aims to identify customer segments at Kalla Toyota using the K-Means algorithm and predict customer churn classes using the Random Forest algorithm. The research is divided into two stages: 1) Customer segmentation using CLTV (Customer Lifetime Value), a metric that measures the financial value of the long-term relationship between a business and its customers, and K-Means; 2) Customer churn prediction based on segmentation using Random Forest. **Results.** The customer segmentation process identified five clusters: 1) High Churn Potential; 2) New Customers; 3) High Loyalty Potential; 4) Mid-to-High Churn; 5) Mid-to-High Loyalty. A Silhouette Score of 0.53 supports the validity of the segmentation, while the customer class prediction using the Random Forest model achieved an accuracy of 87%. **Conclusion.** The segmentation using K-Means and customer churn prediction with Random Forest effectively identify potential churn risks, allowing the company to optimize its retention strategies.

Keywords: automotive; customer segmentation; churn; k-means; CLTV; random forest

DAFTAR ISI

DAFTAR TABEL

Nomor urut Halaman

DAFTAR GAMBAR

Nomor urut Halaman

DAFTAR LAMPIRAN

Nomor urut Halaman

DAFTAR SINGKATAN DAN ARTI SIMBOL

Singkatan	Arti dan Penjelasan
RFM	Recency, Frequency, Monetary
CRM	Customer Relationship Management
CLTV	Customer Lifetime Value
BG/NBD	Beta-Geometric Negative Binomial Distribution
KDD	Knowledge Discovery In Databases
PCA	Principal component analysis
DR	Data reduction
SSE	Sum of Square Error
DBI	Davies-Bouldin Index
ANOVA	Analysis of Variance
SSW	Sum of Square Within
SST	Total Sum of Squares
dfB	Degrees of Freedom Between
dfW	Degrees of Freedom Within
MSB	Mean Square Between
MSW	Mean Square Within

BAB I
PENDAHULUAN

1.1 Latar Belakang

PT HADJI KALLA, sebuah perusahaan dengan lebih dari 70 tahun pengalaman dalam berbagai sektor bisnis, termasuk kontribusinya yang signifikan dalam pembangunan ekonomi Indonesia, khususnya di wilayah Indonesia bagian timur. Salah satu sektor bisnis yang dijalankan oleh PT HADJI KALLA yaitu retail otomotif (KALLA TOYOTA). PT HADJI KALLA berusaha sebaik mungkin untuk memberikan jasa layanannya agar dapat bersaing dan dapat mempertahankan pelanggan serta loyalitas pelanggan.

Survei yang dilakukan oleh penelitian Sheng Zhang (S. Zhang et al., 2019) menunjukkan bahwa perusahaan otomotif berhasil mempertahankan 84% pelanggan selama tahun pertama garansi, namun mengalami penurunan menjadi 29% pada tahun kedua, dan hanya 8% pada tahun kelima. Hal ini menandakan rendahnya pemeliharaan preventif dari konsumen, meskipun masih dalam garansi, berdampak buruk pada kepuasan, loyalitas, retensi pelanggan, dan kinerja produk. Dalam konteks ini, perusahaan perlu menganalisis cara mempertahankan pelanggan. Salah satu pendekatan yang dapat dilakukan adalah dengan melakukan analisis terkait segmentasi dari pelanggan dan juga analisis prediksi *churn* pada pelanggan berdasarkan masa hidup pelanggan (Kamil et al., 2023).

Segmentasi pasar, diperkenalkan oleh Smith pada tahun 1956 (Smith, 1956) merupakan strategi penting dalam pemasaran yang membahas diferensiasi produk dan segmentasi sebagai cara perusahaan memperoleh keunggulan. Segmentasi adalah alat krusial dalam pengambilan keputusan pemasaran, baik di lingkungan domestik maupun global (Farris & Buzzell, 1979; Nezampour et al., 2022). Aspek-aspek kunci dari penerapan segmentasi pelanggan mencakup prinsip-prinsip pemasaran (Hooley et al., n.d.), memaksimalkan keuntungan, perencanaan strategi, dan pengambilan keputusan terkait penempatan produk, harga, dan kualitas (Wedel & Kamakura, 2000). Pemahaman gaya hidup, nilai, dan karakteristik konsumen (Mcdonald, 2010). Dalam konsep segmentasi ini memungkinkan perusahaan mengenali perilaku pelanggan yang loyal atau berpotensi melakukan perpindahan (*churn*).

Churn, atau perpindahan pelanggan, terjadi ketika pelanggan beralih ke penyedia jasa lain (Masarifoğlu & Buyuklu, 2019). Inilah menjadi fokus penting dalam mencapai tujuan *Customer Relationship Management* (CRM) untuk mempertahankan pelanggan, meningkatkan loyalitas, dan membangun hubungan berkelanjutan (Hardjono & San, 2017). Untuk menjaga pelanggan, perusahaan perlu menganalisis kapan pelanggan berpotensi melakukan *churn* dan mengevaluasi probabilitas masa hidup pelanggan (CLTV). CLTV sendiri adalah metrik yang mengukur nilai seorang pelanggan bagi perusahaan selama hubungannya. Analisis CLTV memungkinkan perusahaan mengidentifikasi pelanggan dengan risiko tinggi *churn*, serta memberikan wawasan tentang basis pelanggan, mengelompokkan pelanggan menjadi kategori bernilai tinggi, menengah, dan rendah (Alvi et al., 2021). Untuk memprediksi *churn*, perusahaan dapat

mengimplementasikan analisis pada data transaksi pelanggan, memungkinkan mereka mengambil tindakan proaktif untuk mempertahankan pelanggan (Hanif et al., 2017).

Dalam Penelitian oleh Mohammadzadeh et al. melibatkan pelacakan perilaku pasien selama 3 tahun di tiga klinik rumah sakit sektor publik. Mereka menggunakan model RFM dengan fitur-fitur *recency*, *frequency*, dan *monetary* untuk mengidentifikasi kelas pelanggan dan membangun model prediksi *churn*, dengan tujuan melihat potensi loyalitas pelanggan (Mohammadzadeh et al., 2017). Wardani et al. juga melakukan analisis RFM untuk mengelompokkan pelanggan berdasarkan karakteristiknya, menggunakan fitur *recency*, *frequency*, dan *monetary* untuk membangun model prediksi *churn*, terutama dengan algoritma *Random Forest*. *Random Forest* adalah algoritma klasifikasi dalam data mining yang termasuk dalam kategori *ensemble learning*, yang menggabungkan beberapa model untuk membentuk model yang lebih stabil dan kuat (Ashfania et al., 2022). Dalam penelitian oleh Ullah dan rekan-rekannya, hasilnya menunjukkan bahwa *Random Forest* memiliki tingkat akurasi sebesar 88,63% dan nilai *F1-Score* sebesar 88,20% (Ullah et al., 2019).

Pada penelitian ini secara khusus bertujuan untuk melakukan analisis kelas pelanggan dengan menggunakan *K-Means* lalu selanjutnya melakukan analisis prediksi pelanggan *churn* dengan menggunakan algoritma *Random Forest* pada KALLA TOYOTA.

1.2 Teori

1.2.1 *Customer Relationship Management*

CRM merupakan salah satu alat manajemen bisnis untuk membangun metode untuk mendapatkan informasi yang efektif yang berpusat pada pelanggannya serta berpusat pada solusi perencanaan sumber daya manusia (Vicedo et al., 2020). Adapun tujuan utama dari CRM adalah untuk peningkatan kinerja suatu organisasi atau perusahaan untuk dapat mencapai hasil yang maksimal terhadap bisnisnya (Guerola-Navarro et al., 2022). Penelitian modern tentang manajemen bisnis dan penggunaan teknologi informasi mempertimbangkan perlunya melakukan pengelolaan informasi penting yang akan dijadikan sebagai pengambilan keputusan bisnis. Menetapkan kerangka strategis untuk manajemen hubungan pelanggan terhadap pengelolaan informasi yang menyeluruh serta berorientasi pada harapan pelanggan, untuk mengarahkan pengambilan keputusan yang baik serta strategi pemasaran yang terkoordinasi serta efektif dalam rangka menarik dan mempertahankan pelanggan perusahaan yang memiliki potensi paling menguntungkan (Payne & Frow, 2005). Sehingga untuk inilah tujuan dari CRM sebagai alat data untuk membangun strategi bisnis yang berorientasi pada pelanggan (Ayyagari, 2019).

Penelitian oleh Gil-Gomez juga mengatakan bahwa dalam dalam transformasi digital pada masyarakat dan pasar secara global dan dinamis, penting untuk suatu perusahaan membangun CRM sebagai alat utama dalam kemajuan bisnis (Gil-Gomez et al., 2020), serta e CRM juga memiliki potensi menjadi salah satu solusi teknologi manajemen di bidang manajemen bisnis modern (Adiwijaya et al., 2017).

1.2.2 Segmentasi Pelanggan

Segmentasi pelanggan merupakan proses penting dan krusial yang melibatkan pembagian pelanggan menjadi beberapa kelompok kategori dengan masing-masing kelompok memiliki karakteristiknya sendiri. Memahami kebutuhan dan preferensi setiap segmen, dapat memberikan informasi penting yang dapat membantu perusahaan dalam meningkatkan efisiensi pemasaran terhadap pelanggannya, baik berupa rancangan kampanye pada masing-masing segmen maupun promo yang lebih relevan terhadap pelanggan (Joshua & Preethi, 2022). Selain itu melakukan segmentasi memungkinkan bisnis untuk mengidentifikasi pelanggan yang memiliki tingkat loyalitas yang tinggi dan mengembangkan strategi untuk mempertahankan pelanggan yang menguntungkan bagi Perusahaan (Kumari, 2022). Sehingga dengan melakukan segmentasi pelanggan dapat membantu perusahaan untuk memperoleh keunggulan dengan memahami bagaimana kebutuhan pelanggannya dengan membangun strategi tepat sasaran terhadap segmen pelanggan. Peter Madzík menjelaskan secara umum bahwa terdapat empat tipe segmentasi pada pelanggan, yaitu (Madzík et al., 2021):

1. Segmentasi Demografis: Segmentasi Demografis merupakan salah satu jenis segmentasi yang paling umum dan segmentasi yang paling dasar. Segmentasi ini mengacu pada pengelompokkan pelanggan berdasarkan perbedaan informasi objektif pada pelanggan, seperti usia, jenis kelamin, umur, pendapatan, agama, tingkat Pendidikan bahkan status perkawinan.
2. Segmentasi Psikografis: Segmentasi psikografis merupakan segmentasi yang hampir mirip dengan segmentasi demografis, namun lebih berkaitan dengan sifat dari pelanggan (mental dan emosional). Tidak sama dengan demografis yang mudah untuk diamati, segmentasi ini memerlukan pemahaman terhadap pelanggan mengenai preferensi, motif dan kebutuhan pelanggan. Beberapa contoh karakteristik dari segmentasi ini yaitu kepribadian, hobi, gaya hidup, minat dan keyakinan.
3. Segmentasi Geografi: Segmentasi geografi merupakan segmentasi yang membagi kelompok pelanggannya berdasarkan lokasi atau wilayahnya, seperti iklim, budaya, bahasa dan tingkat kepadatan penduduk. Segmentasi inu membantu perusahaan dalam memahami kebutuhan pelanggannya berdasarkan karakteristik wilayah tinggalnya.
4. Segmentasi Perilaku: Segmentasi perilaku merupakan proses pengelompokan pelanggan berdasarkan dengan perilaku atau kebiasaan yang ditunjukkan pelanggan pada saat berinteraksi dengan perusahaan atau produk, seperti kebiasaan pada saat berbelanja, kebiasaan saat menjelajahi produk, umpan balik yang diberikan, loyalitas terhadap merek yang digunakan. Kebanyakan analisis perilaku ini dikumpulkan melalui interaksi pelanggan dengan website dari bisnis.

1.2.3 *Churn*

Churn, istilah yang mengacu pada perpindahan pelanggan, *churn* terjadi ketika pelanggan beralih ke penyedia jasa lainnya (Masarifoğlu & Buyuklu, 2019). Salah satu fokus utama dalam *Customer Relationship Management* (CRM) untuk mempertahankan pelanggan, meningkatkan loyalitas, dan membangun hubungan berkelanjutan. *Churn* sendiri telah didefinisikan dengan berbagai cara diberbagai bidang industri. Definisi *churn* dalam kamus juga dikenal sebagai periode tidak aktif yang berkepanjangan (Periáñez et al., 2016). Tetapi kriteria *churn* atau 'tidak aktif' dan 'berkepanjangan' memiliki perbedaan menurut masing-masing bidang. Ketidakkonsistenan seperti ini sering ditemukan karena semakin banyak layanan di zaman modern yang mengadopsi persyaratan berlangganan yang fleksibel mengikuti persaingan bisnis. Di masa lalu, perpindahan pada pelanggan terjadi jelas ketika pelanggan melakukan pembatalan kontrak, tetapi dalam masa modern sekarang baik internet maupun layanan ritel, perpindahan pelanggan terjadi karena rendahnya biaya investasi pelanggan (Buckinx & Van den Poel, 2005; Gattermann-Itschert & Thonemann, 2022; Ma, 2009). Oleh karena itu, dalam analisis perpindahan pelanggan, *churn* dapat dibagi menjadi 2 bidang bisnis yaitu *churn* pada bisnis kontraktual dan churn pada bisnis non-kontraktual (Ahn et al., 2020).

Kriteria *churn* yang pertama adalah *churn* pada bisnis kontraktual. *Churn* pada bisnis ini mengacu pada penghentian dimana pelanggan tidak memperpanjang kontrak meskipun tanggal perpanjang kontrak telah tercapai (Y. Chen et al., 2018). Penghentian kontrak ini biasa terjadi karena pelanggan kehilangan minat pada layanan yang diberikan, hal ini juga biasa terjadi ketika pelanggan menutup rekening perbankan atau ketika berpindah ke operator lain, dari satu layanan ke layanan lainnya. Sehingga *churn* pada bisnis dengan basis kontraktual cenderung ditemukan pada layanan dengan tarif tetap seperti layanan layanan *streaming* musik dan film.

Kriteria *churn* kedua adalah *churn* pada bisnis non-kontraktual. Secara umum, dalam kondisi non-kontraktual, pelanggan dapat meninggalkan layanan/kontrak tanpa Batasan waktu. Dalam sudut pandang operasional layanan, kriteria pelanggan *churn* akan dibuat dan kemudian pelanggan yang memenuhi kriteria tersebut dikategorikan sebagai pelanggan *churn* (Ahn et al., 2020). Ketika periode ketidakaktifan atau perubahan perilaku pelanggan melebihi ambang batas, pelanggan akan dianggap sebagai pelanggan yang *churn* (Lee et al., 2018).

1.2.4 RFM

Dalam dunia bisnis, salah satu analisis yang digunakan untuk mengkategorikan suatu pelanggan kedalam kelompok perilaku yang serupa yaitu berdasarkan nilai RFM (Christy et al., 2021). RFM adalah singkatan dari *Recency, Frequency* dan *Monetary Value*, metode ini digunakan untuk membuat analisis segmentasi pelanggan dengan membentuk beberapa kelas tertentu (Laksono & Wulansari, 2021). Analisis RFM sangat bergantung pada data transaksi yang telah

dilakukan oleh pelanggan. Berikut beberapa penjelasan tentang singkatan dari RFM.

1. *Recency* atau keterkinian, data yang digunakan untuk mengidentifikasi pelanggan yang telah bertransaksi dalam waktu dekat. Pelanggan yang belum lama bertransaksi cenderung bereaksi terhadap penawaran yang baru.
2. *Frequency* atau frekuensi, data yang memberikan informasi tentang banyaknya pembelian yang dilakukan oleh pelanggan. Pelanggan yang lebih sering untuk bertransaksi cenderung memberikan penilaian positif terhadap bisnis.
3. *Monetary* value atau nilai uang, data yang mengacu pada nilai moneter yang dikeluarkan oleh pelanggan selama bertransaksi.

1.2.5 CLTV (Customer lifetime value)

Setelah CRM menjadi isu dalam dunia bisnis, perusahan saat ini sudah mulai bersaing pada tingkat pelanggan (Sun et al., 2023; Tsou & Huang, 2018). Para penelitian terdahulu mengajukan konsep tentang nilai yang dirasakan oleh pelanggan (*customer perceived value*), yaitu bagaimana pelanggan melihat manfaat dan nilai yang dirasakan saat memperoleh produk atau layanan sesuai dengan biaya yang mereka keluarkan (Ha & (Shawn) Jang, 2010). Saat ini, ada 2 pandangan untuk mendefinisikan value dari pelanggan. Salah satunya yaitu mendefinisikan nilai dari pelanggan sesuai dengan perspektif pelanggan (Hiray & Anjum, 2022). Nilai pelanggan bisa dikatakan sebagai nilai seumur hidup pelanggan (CLTV) dalam siklus hidup (transaksi) pelanggan untuk perusahaan.

Dalam kasus hubungan perusahaan dengan pelanggan yang bersifat kontrak, pelanggan hanya perlu melakukan kontrak bisnis jangka panjang dengan pihak perusahaan dan melakukan aktivitas bisnis yang seperti biasa. Ketika kedua belah pihak tidak memperbarui kontraknya maka hal ini dianggap sebagai pelanggan yang telah *churn*. Dalam kasus kontrak bisnis ini kehilangan pelanggan pada perusahan dapat diamati dengan sangat jelas. Nilai umur pelanggan (CLTV) dapat dihitung dengan memperkirakan arus kas yang akan dibawa oleh pelanggan ke perusahaan di masa depan. Namun berbeda dengan bisnis yang kontraktual, dimana pelanggan dapat melakukan transaksi pada bisnis lain disaat yang sama. Ketika pelanggan menghentikan suatu transaksi pada perusahan, hal ini tidak langsung mengkategorikan pelanggan menjadi pelanggan yang *churn*, hal ini terjadi karena pelanggan masih memiliki kemungkinan untuk kembali lagi untuk melakukan transaksi pada perusahaan (Sun et al., 2023).

Dalam hubungan non-kontrak, perusahaan sering kehilangan pelanggan tanpa disadari. Oleh karena itu, perlu untuk menghitung *Customer Lifetime Value* (CLTV) dengan memperkirakan probabilitas atau frekuensi pembelian di masa depan berdasarkan model RFM dan data lainnya. Model RFM yang diusulkan oleh Arthur Hughes pada tahun 1990-an membentuk dasar teori untuk menghitung CLTV (Abbasimehr & Shabani, 2019). RFM berfokus pada tiga variabel utama: Kesegaran (*Recency*), Frekuensi (*Frequency*), dan Moneter (*Money*). Model

probabilistik dan *machine learning* telah digunakan untuk memperkirakan CLTV dalam hubungan non-kontraktual, serta untuk segmentasi jenis pelanggan. Penemuan Pengetahuan dalam Basis Data digunakan untuk menemukan segmentasi nilai pelanggan. Berbagai metode penghitungan CLTV masa depan, seperti model distribusi dua BG/NBD dan model distribusi *gamma-gamma heterogen parameter*, membantu meningkatkan keakuratan nilai pelanggan.

Model BG/NBD digunakan untuk menggambarkan perilaku pembelian berulang dalam konteks hubungan pelanggan non kontraktual. Artinya, pengguna bisa membeli produk kapan saja tanpa batasan waktu. Model ini dapat menggunakan data historis transaksi pengguna untuk memprediksi waktu transaksi dan tingkat *turnover* setiap pengguna di masa depan sedangkan *model gamma-gamma* merupakan jumlah rata-rata konsumsi pelanggan (Banggang et al., 2020). Berikut formula dari kedua model probabilitas tersebut (Sun et al., 2023).

Model BG/NBD

$$E(Y(t)|X=x,t_x,T,r,\alpha,a,b)=\frac{a+b+x-1}{a-1}\times \frac{\left[1-\left(\frac{\alpha+T}{\alpha+T+t}\right)^{r+x}{}_1^2F\left(r+x,b+x;a+b+x-1;\frac{t}{\alpha+T+t}\right)\right]}{1+\delta_{(x>0)}\frac{a}{b+x-1}\left(\frac{\alpha+T}{\alpha+t_x}\right)^{r+x}} \quad (1)$$

dimana,

$E(Y(t))$ = Nilai ekspektasi dari jumlah transaksi pada periode waktu t

x = Jumlah transaksi yang diamati pada periode kalibrasi.

t_x = keterkinian transaksi pelanggan

T =umur pelanggan (Waktu dari tanggal pembelian pertama hingga tanggal waktu observasi) (Panjang periode kalibrasi.)

r = Parameter distribusi Gamma untuk tingkat transaksi (jumlah transaksi per waktu)

α = parameter skala dari distribusi gamma untuk tingkat transaksi

a = parameter distribusi beta untuk tingkat pelanggan berhenti

b = parameter skala dari distribusi beta untuk tingga berhenti

$1-\left(\frac{\alpha+T}{\alpha+T+t}\right)^{r+x}$ = menangkap probabilitas transaksi di masa depan.

${}_1^2F$ = fungsi hipergeometrik untuk menggeneralisasi koefisien binomial

$1+\delta_{(x>0)}$ = fungsi delta Kronecker yang bernilai 1 jika x > 0, jika tidak maka 0.

Model Gamma-Gamma

$$E(M|p,q,\gamma,m_x,x) = \left(\frac{q-1}{px+q-1}\right)\frac{\gamma p}{q-1} + \left(\frac{px}{px+q-1}\right)m_x \quad (2)$$

dimana,

E = Nilai probabilitas profit yang diharapkan

p = parameter tingkat frekuensi transaksi yang diharapkan

q = parameter untuk mengatur variasi dalam tingkap frekuensi transaksi.

γ = rata-rata pendapatan atau margin yang dihasilkan per transaksi

m_x = total margin atau pendapatan yang telah diamati hingga saat observasi.

x = jumlah transaksi yang diamati hingga waktu observasi

Dengan rumus CLTV dari hasil model BG/NBD dan model Model Gamma-Gamma adalah sebagai berikut:

$$CLV = \frac{E(Y)\times E(M)}{(1+p)^t} \quad (3)$$

dimana,

t = periode waktu yang dihitung,

p = tingkat diskonto

1.2.6 *Data Mining*

Penambangan data atau yang biasa dikenal sebagai *data mining* merupakan proses untuk menemukan informasi pengetahuan pada suatu kumpulan data yang berjumlah banyak (Koul, 2020). Secara sederhana *data mining* untuk menemukan pola-pola pada suatu kumpulan data besar, menganalisis informasi pada data tersebut dengan menggunakan teknik matematis untuk menghasilkan informasi yang membantu dalam perkembangan bisnis, seperti memprediksi perilaku pelanggan, mengidentifikasi pola-pola tren penjualan serta mengurangi resiko yang dapat merugikan Perusahaan (Olufemi Ogunleye, 2022; Sreejit Ramakrishnan, 2023; Xie & Zhang, 2022).

Proses data mining mengacu pada keseluruhan proses untuk menghasilkan pengetahuan yang berguna. Proses ini biasa disebut sebagai *Knowledge Discovery from Data* (KDD), yang merupakan rangkaian iterative untuk menghasilkan suatu insight atau pengetahuan dari kumpulan data yang dimiliki, yaitu (C. Wang, 2019):

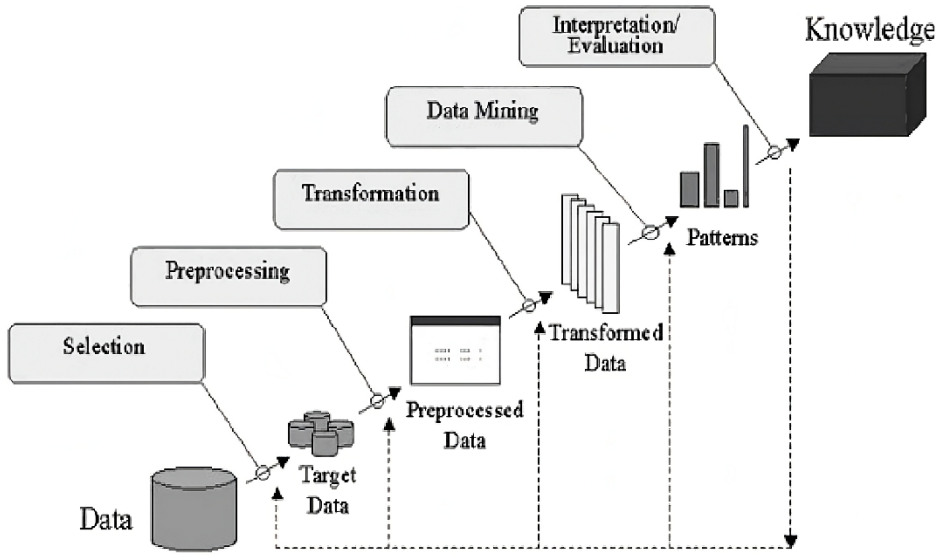


Fig 1

Source: Knowledge discovery from data process, 2023. Diakses dari https://www.semanticscholar.org/paper/SALES-LEVEL-ANALYSIS-USING-THE-ASSOCIATION-METHOD-Samuel-Sani/6519d9ccf1bcae168d6469e9db583c38f2b08c98

Gambar 1. Knowledge discovery from data process

Secara keseluruhan untuk menemukan dan menafsirkan pola dari data melibatkan penerapan berulang dari proses KDD seperti pada langkah-langkah berikut (C. Wang, 2019):

1. **Ruang Lingkup Data:** Pada tahap awal dari proses KDD adalah memahami konteks dari data yang akan digunakan. Proses ini untuk mengidentifikasi target proses dari KDD yang akan mengarah pada bagaimana data ini akan diolah sehingga menghasilkan suatu informasi yang bermanfaat.
2. **Membuat Target Dari Data:** Pada tahap ini merupakan salah satu bagian penting dalam proses KDD untuk melihat variabel apa saja yang bisa digunakan untuk analisis dan memberikan informasi yang bermanfaat. Pada proses ini data akan dipilih berdasarkan fokus analisis yang akan dilakukan. Sehingga, dalam memilih data yang akan digunakan haruslah bersifat representative terhadap analisis yang akan dilakukan.
3. **Data Preprocessing:** Model dari data serta pola yang terdapat pada data memiliki peranan penting untuk menghasilkan informasi yang bermanfaat pada perusahaan. Sehingga data yang digunakan haruslah memiliki kualitas yang baik. Adapun itu data haruslah akurat dan konsisten (Ruddle, 2023), data juga harus lengkap, dimana tidak ada data yang hilang. Hal ini untuk mempertahankan kinerja dari machine learning (H. Nugroho, 2023; Soni et al., 2023). Hal-hal ini perlu diperhatikan karena kualitas data akan secara signifikan mempengaruhi kinerja mesin dalam proses pengambilan

keputusan. Namun menemukan data yang sudah baik dalam dunia nyata sangatlah sulit, sehingga perlu dilakukan persiapan data atau yang biasa dikenal dengan data *preprocessing*. Untuk mempersiapkan data agar dapat digunakan, berikut merupakan langkah-langkahnya yaitu (Joshi & Patel, 2021):

a. *Data Cleaning*

Banyak yang mengira bahwa *data cleaning* merupakan hal yang sama dengan data *preprocessing*. Namun, nyatanya *data cleaning* merupakan salah satu bagian yang ada di dalam proses data *preprocessing*. Dalam tahap *data cleaning*, bagian data yang tidak relevan, tidak sempurna, tidak lengkap atau biasa yang disebut "data kotor" akan diganti, dimodifikasi atau bahkan dihapus dengan teknik tertentu. Proses *data cleaning* dapat dilakukan dengan beberapa teknik sebagai berikut:

1) Data yang memiliki nilai yang tidak lengkap atau biasa yang disebut dengan *missing value* dapat di hapus barisnya, namun perlu diingat teknik ini tidak disarankan dalam kasus dimana data yang digunakan tidak berjumlah banyak.
2) Mengisi nilai yang hilang dengan menggunakan mean, median, atau modus. Biasanya teknik ini digunakan untuk data dengan tipe numerik, dimana data yang hilang akan diisi berdasarkan dengan mean, median, atau modusnya.

b. *Data Transformation*

Data Transformation adalah proses untuk mengubah atau mengkonversi data dari satu format ke format uang lain. Hal ini biasa dilakukan untuk memanipulasi data agar dapat digunakan sesuai dengan keperluan, selain itu juga proses transformasi biasanya digunakan jika dalam ada memiliki tipe yang berbeda dengan tipe data yang dapat dibaca oleh mesin (Pandey et al., 2023). Berikut beberapa metode yang dapat digunakan untuk transformasi data:

1) *Smoothing (Remove Noise from Dataset)*

Smoothing adalah proses transformasi data untuk meningkatkan kualitas pada data dengan menghilangkan *noise* atau pola tidak terstruktur pada data. Biasanya proses *smoothing* dilakukan untuk menganalisis deret waktu, atau pola tren yang terkadang memiliki data yang sangat turun dan sangat rendah sehingga mesin sulit untuk mempelajarinya (Mohamed et al., 2023).

2) *Aggregation (Preparing Data in Abstract Format)*

Proses agregasi data mengacu pada pengumpulan data mentah dari berbagai sumber yang kemudian diformat ulang untuk disajikan dalam satu dataset yang lebih ringkas. Hal ini dilakukan untuk merangkum data menjadi satu sehingga mudah untuk dipahami oleh manusia atau mesin (T. Wen, 2020).

3) *Discretization (Transforming Continues Data in to Some Interval)*

Metode diskritisasi adalah proses dimana mengubah data yang memiliki variabel kontinu menjadi diskrit atau interval kategori tertentu

(Huang et al., 2023). Proses ini sering digunakan dalam analisis untuk mempermudah dalam pengelolaan, terutama pada model mesin yang memiliki keefektifan yang baik jika menggunakan data kategori dibandingkan kontinu seperti proses klasifikasi, penggunaan algoritma pohon keputusan sehingga pengembangan model dapat lebih efektif dan efisien.

4) ***Normalization (Transforming Data in to Given Range)***

Normalisasi data adalah proses yang digunakan untuk menyesuaikan nilai data numerik dalam kumpulan data ke skala umum, tanpa mendistorsi perbedaan rentang nilainya (X. Wang et al., 2024). Normalisasi biasanya dilakukan untuk memperkecil rentang pada suatu data sehingga proses pada mesin tidak terlalu berat. Salah satu metode normalisasi yang paling banyak digunakan adalah *StandardScaler (Z-score)* dan *Min-Max Normalization.*

Z-score atau yang biasa dikenal dengan *StandardScaler* adalah metode normalisasi yang mengubah data menjadi rata-rata (mean) 0 dan standar deviasi 1. Metode *StandardScaler* banyak digunakan pada kasus data yang memiliki distribusi yang normal dan lebih tahan terhadap *outliers*. Sedangkan metode *Min-Max Normalization* adalah metode yang mengubah data menjadi rentang tertentu seperti 0 dan 1. Metode ini biasanya digunakan dalam kasus data yang memiliki nilai input yang terbatas atau ketika data memiliki fitur dengan satuan dan skala yang berbeda, metode ini juga sangat sensitif terhadap *outliers* pada data (Blessing & Klaus, 2023).

5) ***Label Encoding***

Dalam *pra-pemrosesan* data, seringkali proses *Label Encoding* digunakan keefektifan proses pembelajaran sehingga dapat mengurangi waktu analisis manusia dan menghasil file penjelasan yang lebih kecil (Manai et al., 2023). *Label Encoding* sendiri merupakan metode yang mengubah data kategorik – data yang direpresentasikan sebagai label teks- menjadi format numerik (LaRose & Coyle, 2020). Berbeda dengan *Label Encoding* yang tiap labelnya memiliki urutan alaminya (peringkat tinggi sampai rendah), *One-Hot Encoding* tidak memiliki urutan yang hirarki. Proses pada *One-Hot Encoding* mengacu pada penggambaran variabel kategori sebagai vektor biner. Dalam metode ini setiap kategori akan diubah menjadi vector biner yang memiliki panjang yang sama sesuai dengan jumlah kategori (K. Zhang et al., 2023).

4. ***Data Reduction*****:** Data reduksi merupakan teknik yang digunakan untuk mengurangi jumlah data dengan menggabungkan data, menghapus data atau menggunakan metode reduksi dimensi untuk menyederhanakan data tanpa kehilangan informasi yang penting dan mudah untuk dianalisis. Selain itu teknik reduksi data dapat membantu dalam proses pembelajaran mesin melalui pemilihan fitur yang memiliki pengaruh yang besar (Fernandes et al., 2024). Salah satu teknik dari reduksi fitur yang paling banyak digunakan

adalah *Principal Component Analysis* (PCA). PCA merupakan teknik reduksi dimensi efektif yang cara kerjanya membangun fitur-fitur relevan melalui kombinasi dari fitur aslinya (Tsoulfidis & Athanasiadis, 2022).

5. ***Choosing The Data Mining Task*:** Pada proses ini, data yang telah melewati serangkaian proses awal kemudian data siap untuk digunakan. Maka selanjutnya yaitu proses pemilihan teknik *data mining* yang akan digunakan. Saat ini sudah ada beberapa tugas-tugas khusus yang dapat dilakukan oleh mesin seperti untuk klasifikasi, prediksi dan *clustering* sehingga dapat membantu dalam proses analisis (C. Wang, 2019). Pemilihan teknik *data mining* juga haruslah disesuaikan dengan karakteristik dari data yang digunakan seperti memperhatikan bagaimana distribusi datanya atau ukuran data yang akan digunakan (Anand Ramalingam et al., 2023). Selain itu pemilihan teknik data mining juga dilihat pada tugas spesifik apa yang akan dilakukan oleh mesin baik berupa *Classification Tasks* atau *Regression Tasks* dan juga bagaimana tingkat kompleksitas model terhadap data, sehingga pembelajaran yang akan dilakukan dapat berjalan dengan efisien (Singh & Rout, 2023; .V et al., 2023).
6. ***Choosing the data mining algorithms*:** Pada proses ini, tugas dari data mining yang telah ditentukan pada tahap sebelumnya kemudian pada prose pemilihan algoritma data mining haruslah melibatkan pemahaman tentang tipe data dan bagaimana ukuran dan serta kompleksitas dari data yang digunakan (X. Chen et al., 2023; Lu et al., 2023). Selain di tinjau dari datanya, hal yang perlu diperhatikan juga adalah terkait tugas yang akan dijalankan oleh algoritma, ini mencakup bagaimana algoritma akan bekerja sesuai dengan analisis yang diinginkan (Sajid & Amin, 2023).
7. ***Data Mining*:** Pada proses *data mining* ini mengacu pada pencarian pola-pola menarik pada hasil data yang telah dianalisis dengan menggunakan berbagai pendekatan algoritma dan metode dalam menghasilkan suatu informasi yang bermanfaat, seperti melihat bagaimana hasil klasifikasi algoritma terhadap data, bagaimana aturan pohon keputusan yang terbentuk atau bagaimana hasil prediksi masa akan datang terhadap data (C. Wang, 2019).
8. ***Interpreting mined patterns*:** Setelah mendapatkan pola dari hasil *data mining* selanjutnya adalah menafsirkan pola yang tersebut, ini merupakan hal yang penting untuk memahami dan memanfaat hasil wawasan yang diperoleh dari data mining (C. Wang, 2019).
9. ***Consolidating discovered knowledge*:** Pada tahap terakhir yaitu, menyatukan wawasan yang telah didapatkan melalui penggabungan sistem, menuliskan dalam dokumentasi atau menyajikan hasil analisis kepada pemangku kepentingan (C. Wang, 2019)

1.2.7 *Clustering*

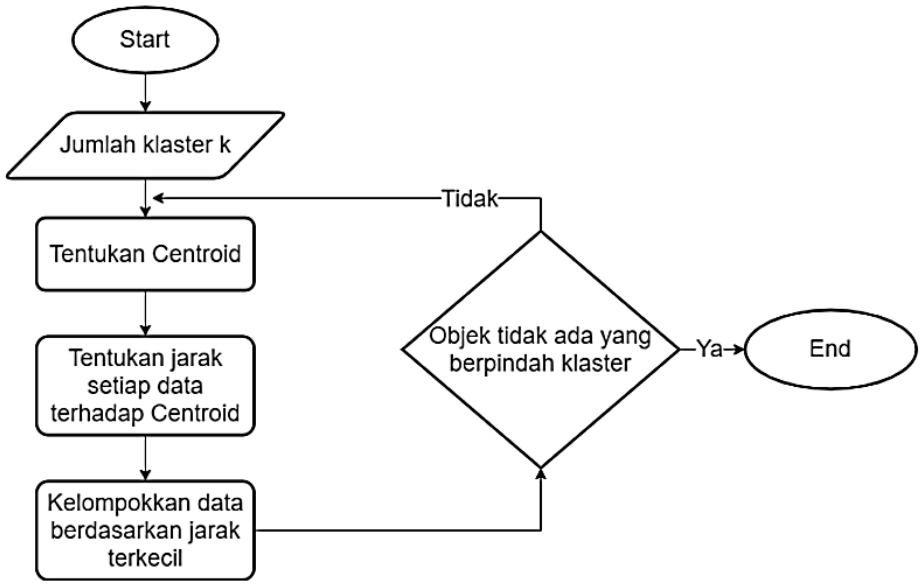
Clustering adalah salah satu teknik data analisis untuk mengelompokkan sekumpulan objek membentuk suatu kelompok yang memiliki makna yang serupa (Bavykina, 2022). Konsep *clustering* sudah ada dan diperkenalkan pada abad ke-20 oleh Tiedeman pada tahun 1955 yang mendefinisikan *cluster* sebagai komponen dalam *Mixture modelling* dalam memodelkan data menjadi kelompok-kelompok tertentu yang sebelumnya tidak terdefinisikan (McNicholas, 2016).

Clustering sendiri dapat digambarkan sebagai teknik *unsupervised learning* dimana objek dikelompokkan kesamaannya. Berikut beberapa metode atau tipe dari *clustering*.

1. ***Partitional Clustering*,** pengelompokkan yang membagi data menjadi sub kumpulan (*cluster*) yang tidak tumpang tindih agar setiap data berada pada tepat satu sub kumpulan. Contoh algoritma pada tipe ini yaitu K- means dan Clara (Jin & Han, 2010; Liu et al., 2023; Rodriguez et al., 2019)
2. *Linkage Clustering,* teknik clustering ini juga biasa dikenal sebagai *Hierarchical Clustering* yang membentuk *cluster* hirarki dengan cara aglomerasi (*bottom-up*) atau memecah-belah (*top-down*) (Rodriguez et al., 2019).
3. ***Model-Based Clustering*,** pada tipe ini proses clustering mengasumsikan model probabilistik terhadap data untuk menemukan kesamaan antara data dan model. Contoh dari tipe ini yaitu GMM (*Gaussian Mixture Models*) yang mengasumsikan bahwa data dihasilkan dari campuran distribusi *gaussian* (Gormley et al., 2023; Rodriguez et al., 2019).
4. ***Spectral Methods,*** pengelompokan spektral merupakan proses pengelompokan yang menggunakan nilai *eigen matriks* kesamaan untuk melakukan reduksi terhadap dimensi sebelum menerapkan algoritma untuk pengelompokan standar. Metode ini mengubah data menjadi grafik dengan menggunakan spektral untuk mengidentifikasi *cluster* (Rodriguez et al., 2019; J. Wen et al., 2020).
5. ***Density-Based Clustering*,** model yang *cluster* dengan melihat wilayah yang memiliki kepadatan tinggi, yang dimana data ini dipisahkan oleh wilayah yang relatif sedikit. Model ini biasanya digunakan untuk mengidentifikasi suatu anomali pada data, contoh algoritma pada tipe ini yaitu DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*) atau OPTICS (*Ordering Points To Identify The Clustering Structure*) (Bhuyan & Borah, 2013; Rodriguez et al., 2019).
6. ***Subspace Clustering*,** metode ini mengelompokkan data dengan mencari subruang yang mendasari dalam ruang dimensi atau ruang data untuk mendapatkan subruang dan hasil pengelompokan yang sesuai. Kebanyakan metode ini digunakan untuk pengenalan wajah, segmentasi gerakan, pengenalan emosi ucapan dan data yang berdimensi tinggi lainnya (Qu et al., 2023; Rodriguez et al., 2019).

1.2.8 *K-Means*

Algoritma *K-Means* merupakan algoritma pengelompokan yang secara diusulkan oleh beberapa peneliti berbeda seperti Steinhaus (Steinhaus, 1956), Lloyd (Lloyd, 1982), Jancey (Jancey, 1966) dan MacQueen (Cam & Neyman, 1967). Algoritma *K-Means* mengelompokkan data berdasarkan kedekatan data satu sama lain dengan mencari minimum jarak titik data ke pusat data (Marisa et al., 2021). Dari hasil variasi penelitian, para peneliti menunjukkan ada empat tahapan dalam prose pembentukan cluster yaitu sebagai berikut (Marisa et al., 2021; Sud et al., 2020):



Source: Flowchart K-Means, 2021. Diakses dari https://ejournal.unikama.ac.id/index.php/jst/article/view/5270

Gambar 2. Flowchart k-means

Berdasarkan flowchart pada gambar 2 dijelaskan sebagai berikut

1. Tentukan jumlah *cluster* awal Langkah awal pada saat melakukan analisis dengan menggunakan *K-Means* adalah dengan menentukan jumlah *cluster* yang akan digunakan. penentuan jumlah cluster ini menggunakan elbow method untuk melihat jumlah cluster optimal yang terbentuk.
2. Setelah menentukan jumlah *cluster*, selanjutnya yaitu menentukan pusat data (*Centroid*) yang akan digunakan untuk setiap *cluster*, dengan menggunakan *Euclidean distance* lalu mengelompokkan setiap data point ke *centroid* terdekat.
3. Menghitung ulang posisi *centroid* sebagai rata-rata dari data point untuk setiap *cluster* yang terbentuk.
4. Mengulangi Langkah 2 dan 3 hingga *centroid* tidak lagi berubah secara signifikan terhadap *data point*.

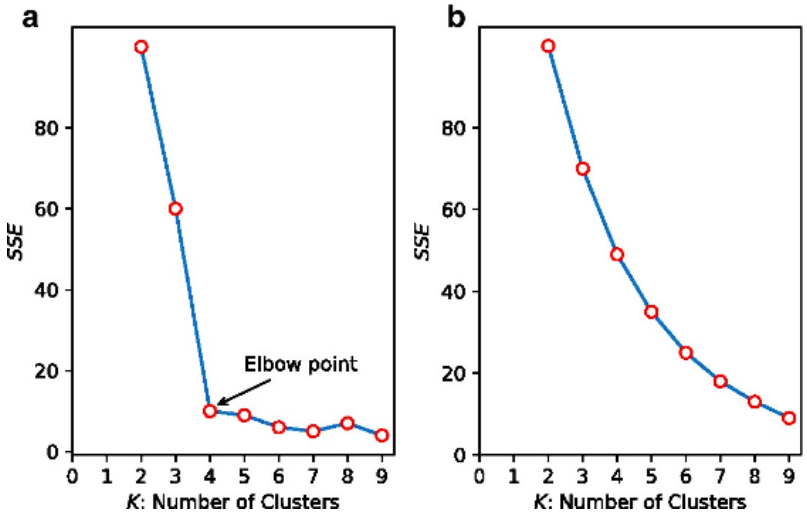
1.2.9 *Principal Component Analysis* (PCA)

Principal Component Analysis atau yang biasa disingkat dengan PCA merupakan teknik *dimensionality-reduction* (DR) yang banyak digunakan dalam mereduksi dimensi kumpulan data yang memiliki fitur dengan jumlah yang banyak menjadi data dengan fitur yang kecil (Hasan & Abdulazeez, 2021). Teknik dalam reduksi data dengan PCA sendiri melibatkan kombinasi linear dari fitur-fitur untuk membangun subruang utama, dimana subruang ini menampung sebagian besar vektor fitur data. PCA juga menggunakan varians sebagai tolak ukur informasi terhadap subruang, kualitas subruang yang di pilih di lihat dengan membandingkan varians pada keseluruhan dari dataset (Marukatat, 2023). Berikut langkah-langkah dari proses PCA (Marukatat, 2023):

1. Standarisasi Data, langkah awal dari proses PCA adalah normalisasi data atau standarisasi data untuk memastikan bahwa setiap fitur atau variabel memiliki bobot yang sama.
2. Menghitung Matriks Kovarians, pada tahap ini akan dihitung matriks kovarians untuk melihat varian dan kovarian pada data, hal ini dilakukan untuk melihat bagaimana korelasi antara masing-masing fitur.
3. Menghitung vektor eigen dan nilai eigen, selanjutnya yaitu menghitung eigenvector dan eigenvalue dari matriks kovarian. Dalam matriks kovarians eigenvector merupakan arah dari variabel terbesar, sedangkan *eigenvalue* menunjukkan besarnya varian dalam arah *eigenvector*.
4. Pengurutan *eigenvector* dan *eigenvalue*, pada proses ini *eigenvector* akan disortir berdasarkan nilai *eigenvalue* dari besar ke kecil. Komponen dengan *eigenvalue* terbesar pertama akan menunjukkan bahwa itu merupakan komponen utama, nilai *eigenvalue* terbesar kedua akan menunjukkan bahwa itu merupakan komponen utama kedua dan seterusnya.

1.2.10 *Elbow Method dan Sum of Squared Errors* (SSE)

Pada proses data mining ini mengacu pada pencarian pola-pola menarik pada hasil data yang telah dianalisis dengan menggunakan berbagai pendekatan algoritma dan metode dalam menghasilkan suatu informasi yang bermanfaat, seperti melihat bagaimana hasil klasifikasi algoritma terhadap data, bagaimana aturan pohon keputusan yang terbentuk atau bagaimana hasil prediksi masa akan datang terhadap data



Source: Elbow point and number of cluster, 2021. Diakses dari https://www.researchgate.net/publication/348457366_A_Quantitative_Discriminant_Method_of_Elbow_Point_for_the_Optimal_Number_of_Clusters_in_Clustering_Algorithm

Gambar 3. Elbow point and number of cluster

SSE sendiri merupakan metrik yang akan menghitung jumlah total variasi dalam data dan diimplementasikan dalam bentuk *cluster* yang terbentuk, SSE ini akan mengukur seberapa baik data yang telah dikelompokkan pada suatu *cluster* (Humaira & Rasyidah, 2020). Penurunan nilai SSE yang signifikan pada pembentukan *cluser* merupakan indikasi bahwa memberi tambahan pada jumlah cluster bukanlah suatu hal yang dapat memberikan hasil *cluster* yang lebih baik, sehingga perlu di pertimbangkan penambahan jumlah *cluster* jika tidak sesuai dengan titik sikunya (Refialy et al., 2021). Adapun persamaan SSE ditunjukan pada rumusan dibawah ini (Syakur et al., 2018).

$$SSE = \sum_{K=1}^{K} \sum_{x_i \in S_K} (X_i - C_k)^2 \qquad (4)$$

dimana,

k = jumlah *cluster*

S_K = Set (kumpulan) dari semua titik data yang termasuk dalam *cluster* K.

X_i = titik data yang termasuk dalam *cluster* S_K.

C_K = Pusat *cluster* ke-k.

1.2.11 *Euclidean Distances*

Euclidean distance merupakan metode perhitungan jarak antara dua titik dalam ruang *euclidean*, yaitu ruang datar konvensional yang digunakan secara

teratur dalam kehidupan sehari-hari seperti ruang 2D atau 3D (nD). Metode ini merupakan bentuk secara umum untuk mengukur jarak sebenarnya antara dua titik atau lebih. Perhitungan jarak *Euclidean* antara dua titik *centroid* yaitu sebagai berikut (Syakur et al., 2018).

$$d(x,y) = \|x-y\| = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}; i = 1,2,3,\ldots,n \quad (5)$$

dimana,

$d(x,y)$ = jarak antara objek x dan y.

$\|x-y\|$ = notasi untuk norma *euclidean* (jarak *euclidean*) antara dua vektor x dan y

$\sum_{i=1}^{n}(x_i - y_i)^2$ = penjumlahan kuadrat perbedaan komponen yang sesuai dari x dan y

x_i = komponen ke-i dari vektor x

y_i = komponen ke-i dari vektor y

n = jumlah dimensi atau jumlah komponen dalam vektor x dan y

1.2.12 Silhouette Score

Silhouette Score adalah metode yang digunakan untuk menunjukkan seberapa dekat suatu objek dengan clusternya dibandingkan dengan *cluster* lainnya (Januzaj et al., 2023). Penelitian oleh Mohammed Salihoun (Salihoun, 2020), rata-rata analisis *silhouette* yang dilakukan sudah secara tepat menunjukkan jumlah *cluster* yang optimal. Nilai yang diperoleh skor yang diperoleh dalam analisis *silhouette* berkisar -1 hingga +1, semakin tinggi nilai maka semakin dekat pula objek itu dengan clusternya, sebaliknya semakin kecil nilainya maka semakin jauh jarak objek dengan clusternya (Januzaj et al., 2023). Berikut persamaan untuk menghitung skor *silhouette* untuk setiap titik data yang dihitung menggunakan rata-rata jarak antar *cluster* (a) dan rata-rata jarak cluster terdekat (b) (N. Nugroho & Adhinata, 2022):

$$S(i) = \frac{b(i)-a(i)}{max(a(i),b(i))} \quad (6)$$

dimana,

$S(i)$ = nilai *silhouette coefficient* pada object ke-i

(i) = titik data ke-i

$a(i)$ = rata-rata jarak titik data i dengan anggota dengan semua data dalam *cluster* yang sama.

$b(i)$ = rata-rata jarak antara data ke-I dengan semua titik data dalam *cluster* terdekat yang bukan clusternya

Dalam penilaian skor *silhouette coefficient*, terdapat beberapa interpretasi nilai yaitu sebagai berikut.

Tabel 1. Tabel skor *silhouette coefficient*

Nilai *Silhouette Score Coefficient*	Interpretasi
0.71 – 1.00	Struktur yang dihasilkan kuat
0.51 – 0.70	Struktur yang dihasilkan baik
0.26 – 0.50	Struktur yang dihasilkan lemah
≤ 0.25	Tidak terstruktur

Source: N. Nugroho & Adhinata (2022)

1.2.13 *Davies-Bouldin Index (DBI)*

Indeks *Davies-Bouldin* adalah metrik yang digunakan untuk mengevaluasi atau mengukur kualitas clustering dalam analisis data. Adapun tujuan dari metrik ini adalah untuk mengukur seberapa baik *cluster* yang dihasilkan algoritma. clustering untuk memisahkan kelompok data yang berbeda dan mendekati pusat clusternya (Umagapi et al., 2023). Indeks *Davies-Bouldin* untuk sekumpulan *cluster* didefinisikan sebagai berikut:

$$DBI = \frac{1}{N}\sum_{i=1}^{N} max_{j \neq i}(R_{ij}) \quad (7)$$

dengan penjabaran R_{ij} yang merupakan nilai *Davies-Bouldin Index* antara dua *cluster* C_i dan C_j adalah sebagai berikut:

$$R_{ij} = \frac{S_i + S_j}{d_{ij}} \quad (8)$$

dimana,

R_{ij} = nilai *Davies-Bouldin Index* antar *cluster* C_i dan C_i.

S_i = ukuran sebaran data dalam *cluster* C_i yang dihitung dengan varian atau jarak rata-rata antara titik data dalam pusat cluster m_i.

d_{ij} = jarak antara pusat *cluster* m_i dan m_j.

DBI = nilai *Davies-Bouldin Index* keseluruhan.

N = jumlah total *cluster*.

Nilai *Davies-Bouldin Index* yang lebih cenderung rendah akan menunjukkan kualitas dari hasil *cluster* yang lebih baik, hal ini menunjukkan bahwa *cluster* yang terbentuk cenderung lebih terpisah dan lebih dekat dengan pusat clusternya.

1.2.14 *Analysis of Variance* (ANOVA)

ANOVA, yang merupakan singkatan dari "*Analysis of Variance*" adalah metode statistik yang digunakan untuk membandingkan rata-rata antara dua atau lebih kelompok. Prinsip dasar ANOVA melibatkan dua jenis varian: varian antar kelompok, yang mengukur perbedaan antara rata-rata setiap kelompok dengan rata-rata keseluruhan, dan varian dalam kelompok, yang mengukur variasi di dalam masing-masing kelompok (Septiadi & Ramadhani, 2020). Uji ANOVA dilakukan dengan menggunakan hipotesis sebagai berikut.

1. *Null hypothesis (H0)*: Rata-rata semua kelompok adalah sama atau identik.
2. *Alternative hypothesis (H1)*: Setidaknya ada satu rata-rata kelompok yang berbeda dari kelompok lainnya.

Ada berbagai jenis ANOVA, salah satunya adalah *One-Way* ANOVA. One-Way ANOVA digunakan ketika terdapat tiga atau lebih variabel bebas (kelompok) dan satu variabel terikat kontinu (Kim, 2017). Model dasar ANOVA satu arah dimulai dengan menghitung jumlah kuadrat antar (SSB) untuk mengukur selisih antar kelompok, seperti pada persamaan di bawah ini (Sullivan, 2019).

$$SSB = \sum n_i \left(\overline{X}_i - \overline{X}_G\right)^2 \quad (9)$$

dimana,

n_i : jumlah sampel dalam grup i
$\overline{X}_i$: rata-rata sampel untuk grup I
$\overline{X}_G$: rata-rata keseluruhan

Selanjutnya, dihitung variabilitas dalam grup atau *Sum of Square Within* (SSW) seperti pada persamaan dibawah ini:

$$SSW = \sum \left(\overline{X}_{ij} - \overline{X}_i\right)^2 \quad (10)$$

dimana,

$\overline{X}_{ij}$: nilai observasi j dalam grup i
$\overline{X}_i$: rata-rata sampel untuk grup I

Selanjutnya, dihitung variabilitas total dalam data atau *Total Sum of Squares* (SST) seperti pada persamaan berikut:

$$SST = SSB + SSW \quad (11)$$

Selanjutnya, dihitung *Degrees of Freedom Between* (dfB) seperti pada persamaan berikut:

$$dfB = k - 1 \quad (12)$$

dimana,

k : Jumlah Grup

Selanjutnya, dihitung *Degrees of Freedom Within* (dfW) seperti pada persamaan berikut:

$$dfW = N - k \quad (13)$$

dimana,

N : Jumlah total observasi

Selanjutnya, dihitung *Mean Square Between* (MSB) seperti pada persamaan berikut

$$MSB = \frac{SSB}{dfB} \quad (14)$$

Selanjutnya, dihitung *Mean Square Within* (MSW) seperti pada persamaan berikut:

$$MSW = \frac{SSW}{dfW} \quad (15)$$

Selanjutnya, dihitung F-Statistik yang membandingkan varians antar kelompok dengan varians dalam kelompok seperti pada persamaan berikut:

$$F = \frac{MSB}{MSW} \quad (16)$$

Selanjutnya, *p-value* diperoleh dengan membandingkan F-statistik yang dihitung dengan distribusi F, yang dapat dilakukan menggunakan tabel distribusi F atau perangkat lunak statistik. *P-value* mengindikasikan probabilitas untuk mendapatkan hasil observasi atau yang lebih ekstrim jika Hipotesis Nol benar. Jika *p-value* lebih kecil dari tingkat signifikansi yang ditentukan (misalnya, 0,05), maka Hipotesis Nol ditolak. Ini menunjukkan bahwa setidaknya ada satu kelompok dengan rata-rata yang berbeda secara signifikan. Sebaliknya, jika *p-value* lebih besar dari tingkat signifikansi, maka tidak ada cukup bukti untuk menolak Hipotesis Nol, yang berarti tidak dapat disimpulkan bahwa terdapat perbedaan signifikan antara rata-rata kelompok (Kim, 2017; Septiadi & Ramadhani, 2020).

1.2.15 Uji *Chi-Square*

Uji *chi-square* biasanya digunakan untuk menguji hubungan antara variabel kategori dan distribusinya dalam kelompok (Kadhim & Jassim, 2021). uji *Chi-square* efektif untuk pemilihan fitur kategoris dan diferensiasi kelompok bila diterapkan dengan K-means, sehingga meningkatkan kemampuan algoritma dalam mendeteksi perbedaan yang signifikan antar kelompok (Ferreira-Santos & Pereira Rodrigues, 2018). Uji *Chi-Square* (*Chi-Square Test*) digunakan untuk mengetahui apakah ada hubungan atau perbedaan yang signifikan antara dua variabel kategorikal. Biasanya, uji ini digunakan untuk menguji hipotesis nol (H_0) yang menyatakan bahwa tidak ada hubungan atau perbedaan antara variabel. Berikut adalah langkah-langkah umum untuk melakukan uji *Chi-Square* (Singhal & Rana, 2015), dengan hipotesis sebagai berikut:

1. Hipotesis nol (H_0): Tidak ada hubungan antara variabel yang diuji.
2. Hipotesis alternatif (H_1): Ada hubungan antara variabel yang diuji.

Pada tahap awal frekuensi harapan dihitung berdasarkan asumsi bahwa tidak ada hubungan antara variabel. Rumus untuk frekuensi harapan adalah sebagai berikut:

$$E_{ij} = \frac{(R_i \times C_j)}{N} \quad (17)$$

dimana,

E_{ij} = frekuensi harapan untuk sel di baris ke-i dan kolom ke-j.

R_i = jumlah total baris ke-i.

C_j = jumlah total kolom ke-j.

N = jumlah total observasi.

Setelah menghitung frekuensi harapan, nilai *Chi-Square* dihitung menggunakan rumus:

$$x^2 = \sum \frac{(O_{ij} - E_{ij})}{E_{ij}} \quad (18)$$

dimana,

O_{ij} = frekuensi observasi pada sel di baris ke-i dan kolom ke-j

E_{ij} = frekuensi harapan pada sel di baris ke-i dan kolom ke-j.

Selanjutnya yaitu menghitung derajad kebebasan, derajat kebebasan (*Degrees of Freedom, df*) dalam uji *Chi-Square* memiliki peran penting dalam menentukan seberapa besar variabilitas yang diizinkan dalam penghitungan statistik uji (C.-M. J. Wang, 1994). Derajat kebebasan dihitung dengan rumus sebagai berikut:

$$df = (r - 1)(c - 1) \quad (19)$$

dimana,

r = jumlah baris.

c = jumlah kolom.

Bandingkan nilai *Chi-Square* yang dihitung dengan nilai kritis dari tabel distribusi *Chi-Square* berdasarkan tingkat signifikansi dan derajat kebebasan. Jika nilai *Chi-Square* yang dihitung lebih besar dari nilai kritis, maka hipotesis nol ditolak. Jika H_0 ditolak, ada perbedaan atau hubungan signifikan antara variabel-variabel tersebut, Jika H_0 diterima, tidak ada perbedaan atau hubungan signifikan.

1.2.16 Uji *Post Hoc* Menggunakan Uji *Tukey HSD*

Uji *post hoc*, berasal dari kata Latin "setelah ini", adalah analisis statistik yang dilakukan setelah uji awal telah dilakukan. Uji ini berguna untuk mengidentifikasi perbedaan spesifik antar kelompok, terutama ketika hipotesis nol uji omnibus, seperti ANOVA, ditolak. Uji *post hoc*, juga dikenal sebagai tes perbandingan berganda, digunakan untuk menentukan kelompok mana yang berbeda secara signifikan satu sama lain. Salah satu Uji *post hoc* yang digunakan adalah uji HSD *Tukey*. Uji ini penting untuk memeriksa perbedaan antara rata-rata beberapa kelompok sambil mengendalikan tingkat kesalahan eksperimen (Agbangba et al., 2024).

Uji *Tukey's Honestly Significant Difference (HSD)* merupakan uji *post hoc* yang sering digunakan untuk menilai signifikansi perbedaan antara pasangan rata-rata kelompok. Uji ini membandingkan semua kemungkinan pasangan rata-rata dan didasarkan pada distribusi rentang yang disederhanakan. Dalam analisis statistik, uji HSD menggunakan beberapa fitur seperti perbedaan rata-rata (*Mean Difference*), *p-value* yang telah dikoreksi (*p-adj*), serta batas bawah dan atas, memberikan pemahaman yang lebih komprehensif dan nuansa lebih dalam terhadap hasil penelitian. Uji *Tukey's* HSD, perbedaan rata-rata, *p-adj*, dan interval kepercayaan semuanya berkontribusi pada pengambilan keputusan mengenai signifikansi statistik. Sementara Tukey's HSD dan *p-adj* secara langsung menangani signifikansi statistik perbandingan, perbedaan rata-rata dan interval kepercayaan memberikan informasi tentang besar kecilnya perbedaan dan ketepatan estimasi tersebut. Dengan menggunakan semua informasi ini secara bersamaan, peneliti dapat membuat interpretasi yang lebih kaya dan lebih informatif mengenai data mereka (Nanda et al., 2021).

Perbedaan rata-rata memberikan informasi tentang ukuran perbedaan antara grup. Ini membantu dalam memahami signifikansi praktis dari temuan tersebut, bukan hanya signifikansi statistiknya. Sebagai contoh, perbedaan yang signifikan secara statistik mungkin tidak memiliki arti praktis yang signifikan jika perbedaannya sangat kecil, seperti yang ditunjukkan pada persamaan berikut (Howell, 2010).

$$Perbedaan\ Rataan = \overline{x}_2 - \overline{x}_1 \tag{20}$$

dimana,

$\overline{x}_1$: Rata-rata sampel untuk kelompok pertama,

$\overline{x}_2$: Rata-rata sampel untuk kelompok kedua

P-value terkoreksi (*p-adj*) mengatasi masalah pengujian multipel yang dapat meningkatkan risiko kesalahan tipe I (*false positives*). Ini memberikan ukuran yang lebih hati-hati dan akurat tentang signifikansi statistik dari perbandingan, sehingga kesimpulan yang diambil dari data menjadi lebih dapat diandalkan. Berbeda dengan rumus sederhana yang digunakan dalam perhitungan statistik lainnya, *p-adj* dalam uji *Tukey* HSD dihitung berdasarkan distribusi probabilitas khusus yang berkaitan dengan perbedaan rata-rata antar pasangan grup. Prinsipnya adalah menganalisis perbedaan antara semua pasangan grup dalam satu langkah untuk mengendalikan tingkat kesalahan tipe I secara keseluruhan pada level yang diinginkan (biasanya α = 0,05). P-adj menunjukkan probabilitas bahwa perbandingan antara grup akan menghasilkan nilai *q* yang tinggi atau lebih, jika hipotesis nol benar (tidak ada perbedaan). Nilai *q* yang dihitung kemudian dibandingkan dengan tabel distribusi *q Tukey*, yang memberikan nilai kritis berdasarkan jumlah grup dan total observasi. Jika nilai q yang dihitung melebihi nilai kritis, maka *p-adj* akan menunjukkan signifikansi statistik. Rumus untuk *q* ditunjukkan pada persamaan berikut (Abdi & Williams, 2019; Howell, 2010).

$$q = \frac{Perb\ rataan\ terbesar}{Estimasi\ Standar\ Kesalahan} \tag{21}$$

dimana,

q : Nilai statistik yang dihitung, yang kemudian dibandingkan dengan nilai kritis dari distribusi q Tukey untuk menentukan signifikansi

MSE: *Mean Square Error* dari ANOVA

Batas Bawah dan Atas adalah interval kepercayaan yang menunjukkan rentang nilai di mana perbedaan sebenarnya antara kelompok berada, dengan tingkat kepercayaan tertentu (biasanya 95%). Ini memberikan gambaran tentang ketidakpastian terkait estimasi perbedaan rata-rata dan penting untuk menilai keandalan temuan tersebut. Perhitungan Batas Bawah dan Atas dimulai dengan menghitung estimasi standar kesalahan, seperti yang ditunjukkan pada persamaan (Abdi & Williams, 2019; Howell, 2010).

$$Estimasi\ Standar\ Kesalahan = \sqrt{MSE\left(\frac{1}{n_1}+\frac{1}{n_2}\right)} \qquad (22)$$

dimana,

$n_i\ dan\ n_j$: Ukuran sampel dari dua kelompok yang dibandingkan

Selanjutnya, perhitungan Batas Bawah atau Atas menggunakan persamaan berikut:

$$Batas\ atas\ atau\ bawah = Perb\ Rataan\ \pm (q_{\propto} x\ Estimasi\ Standar\ Kesalahan)$$

dimana,

simbol $\pm$: Menunjukkan bahwa untuk mendapatkan batas bawah, dikurangkan nilai setelahnya, dan untuk batas atas, ditambahkan nilai setelahnya

$q_{\propto}$: Nilai kritis dari distribusi *q Tukey* untuk level signifikansi α yang diinginkan (misalnya, 0.05) berdasarkan derajat kebebasan total dan jumlah perbandingan yang dilakukan,

1.2.17 *Random Forest*

Random forest merupakan algoritma *clustering* yang paling sederhana dan mudah digunakan, dan algoritma ini banyak digunakan untuk melakukan proses deduktif dan klasifikasi (Erlin et al., 2022). *Random Forest* merupakan kumpulan algoritma pembelajaran yang menggabungkan beberapa pohon keputusan untuk meningkatkan akurasi hasil analisis (Kiangala & Wang, 2021).

Random forest menggunakan rata-rata untuk meningkatkan keakuratan hasil prediksi dan mengontrol redundansi atau overfitting. Ketika *bootstrap=True* (default), ukuran sampel diukur dengan parameter *max_sample*. Jika tidak, seluruh kumpulan data akan digunakan untuk membangun setiap pohon. Ketika *Random Forest* digunakan untuk klasifikasi data, rumus *indeks Gini* yang ditunjukkan pada persamaan yang digunakan untuk menentukan node cabang dari pohon keputusan. Model ini menggunakan kelas dan probabilitas untuk

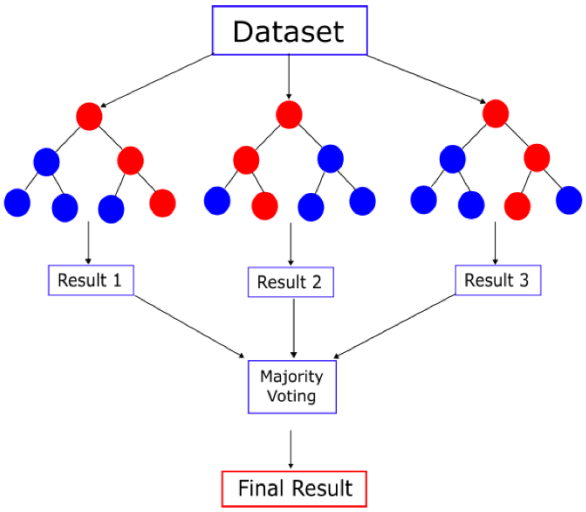
menentukan *Gini* setiap cabang pada suatu node dan cabang mana yang akan muncul (Erlin et al., 2022).

$$Gini = 1 - \sum_{i=1}^{c}(p_i)^2 \quad (23)$$

Dimana p_i mewakili frekuensi kelas yang ditemukan dalam kumpulan data dan c mewakili jumlah kelas. Selain *Gini*, entropi digunakan untuk menentukan bagaimana node bercabang di pohon keputusan. Rumus entropi ada pada persamaan berikut:

$$Entropy = \sum_{i=1}^{c} -p_i \times \log_2(p_i) \quad (24)$$

Entropi menggunakan probabilitas yang dihasilkan untuk menentukan bagaimana sebuah node harus bercabang. Berbeda dengan indeks *Gini*, indeks ini lebih bersifat matematis karena fungsi aljabar yang digunakan dalam perhitungannya.



Source: Struktur Random Forest, 2023. Diakses dari https://www.mdpi.com/2075-1729/13/4/1031

Gambar 4. Struktur Random Forest

Subset yang dipilih secara acak, memperoleh prediksi dari setiap pohon keputusan, melakukan pemungutan suara pada setiap hasil prediksi, dan memilih hasil prediksi terbaik berdasarkan voting terbanyak (Erlin et al., 2022).

1.2.18 Matrik Evaluasi

Memilih metode pengukuran kinerja yang tepat untuk evaluasi algoritma adalah hal yang penting. Algoritma prediksi atau klasifikasi yang dilatih pada kumpulan data latih mungkin saja dapat memberikan tingkat akurasi yang tinggi, namun sangat bias terhadap sebagian data uji yang diberikan. Kinerja yang tepat dari algoritma yang digunakan akan memberikan kemampuan adaptasi secara efisien terhadap data baru. Tujuan utamanya adalah mendapatkan sebanyak mungkin *true positif* (TP) dan *true negative* (TN), diimbangi dengan meminimalkan

false negatif (FN) sebanyak mungkin (Erlin et al., 2022). Adapun beberapa matriks yang digunakan untuk melihat bagaimana kinerja dari model yang digunakan yaitu sebagai berikut (Aria et al., 2021).

1. *Accuracy*, mengacu pada kinerja klasifikasi secara keseluruhan. Namun pengukuran langsung dapat menyesatkan jika datanya tidak seimbang karena bobot lebih diberikan kepada kelas mayoritas dibandingkan kelas minoritas, sehingga sulit bagi classifier untuk berkinerja baik pada kelas minoritas. Adapun rumus dari matriks ini yaitu:

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FM} \quad (25)$$

2. *Recall/Sensitivity*, *Recall* atau sensitivitas mengukur keakuratan kelas positif. Sensitivitas mengevaluasi kemampuan mengklasifikasikan berdasarkan kelas positif atau minoritas. Adapun rumus dari matriks ini yaitu:

$$Recall = \frac{TP}{TP+FN} \quad (26)$$

3. *Precision*, ukuran seberapa akurat model tersebut. Nilai koefisien yang lebih akurat menunjukkan klasifikasi yang lebih baik. Ini secara langsung mengukur proporsi prediksi bagus dan sempurna. Berikut rumus yang digunakan untuk matriks precision

$$\text{Precision} = \frac{TP}{TP+FP} \quad (27)$$

4. *Score*, ukuran kinerja yang menggabungkan *precision* dan *recall* menjadi satu untuk menghasilkan metrik yang seimbang. *F1 Score* memberikan gambaran mengenai seberapa baik model kita dalam mengklasifikasikan kelas-kelas pada data secara akurat. Berikut rumus yang digunakan untuk menghitung *F1 Score*

$$F1\ Score = 2 \times \frac{Recall \times precision}{Recall + precision} \quad (28)$$

1.3 Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan adapun rumusan masalah dari penelitian ini, yaitu sebagai berikut:

1. Bagaimana hasil implementasi K-Means untuk membentuk segmen dan mengidentifikasi kelompok terhadap pelanggan bengkel pada PT HADJI KALLA?
2. Bagaimana hasil implementasi *Random Forest* dalam memprediksi kelas pada pelanggan PT HADJI KALLA?

1.4 Tujuan Penelitian

Tujuan penelitian merupakan jawaban yang berdasar pada rumusan masalah, diantaranya:

1. Untuk mengetahui bagaimana hasil implementasi K-means untuk membentuk segmen dan melihat kelompok yang terbentuk pada pelanggan pada PT HADJI KALLA.
2. Untuk mengetahui hasil prediksi kelas pada pelanggan PT HADJI KALLA menggunakan algoritma *Random Forest*.

1.5 Manfaat Penelitian

Penelitian ini diharapkan memberikan manfaat sebagai berikut :

1. Memberikan hasil prediksi potensial kelas pelanggan yang mungkin berpindah ke pesaing atau menghentikan penggunaan produk atau layanan.
2. Menyediakan salah satu indikator dalam menargetkan pelanggan yang berpotensi berpindah, meningkatkan efektivitas kampanye promosi, dan mengurangi risiko kehilangan pangsa pasar.

1.6 Ruang Lingkup

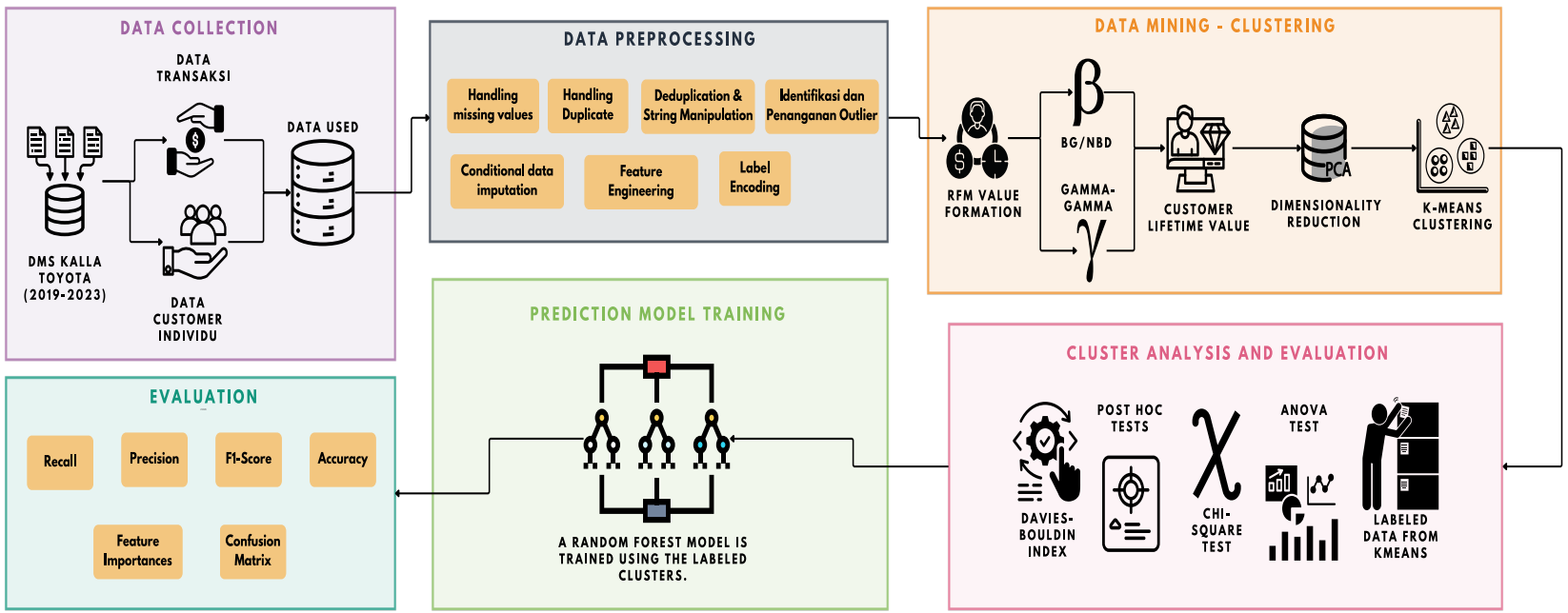
Ruang lingkup pada penelitian ini diantaranya:

1. Data yang digunakan adalah data historis 5 tahun terakhir dari transaksi pelanggan pada dealer bengkel KALLA TOYOTA.
2. Data pelanggan yang diambil adalah data pelanggan personal yang telah melakukan pembayaran sesuai dengan *invoice* yang tersedia dan juga data pelanggan yang mendapatkan *free* jasa.
3. Penelitian ini berfokus pada penggunaan algoritma K-Means untuk membentuk segmentasi pelanggan dan dilanjutkan dengan *Random Forest* untuk dilakukan prediksi pelanggan yang *churn*.
4. *Output* dari hasil penelitian ini adalah memberikan model prediksi pelanggan yang memiliki kemungkinan akan berpindah (*churn*) dari bengkel Kalla Toyota

BAB II
METODE PENELITIAN

2.1 *Pipeline Data Analysis*

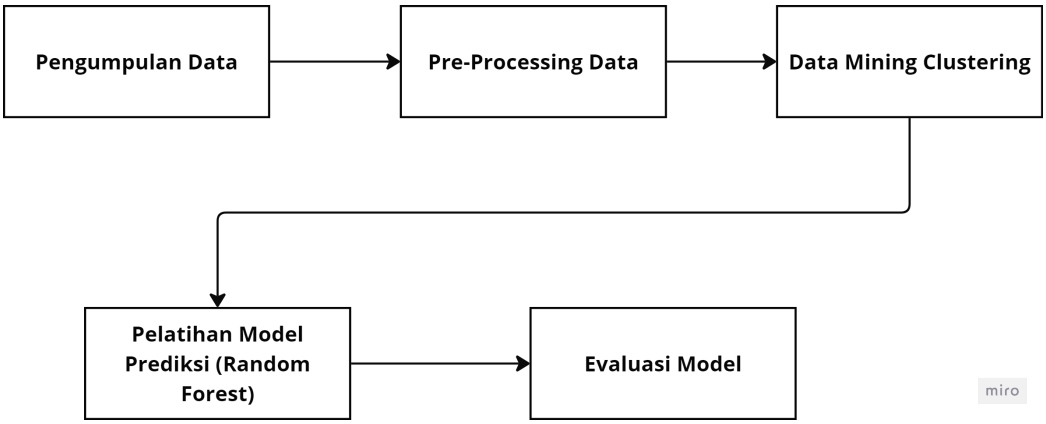
Berdasarkan gambar yang disediakan, *pipeline* analisis data ini menunjukkan alur kerja komprehensif yang terdiri dari beberapa tahapan utama yang dirancang untuk mengolah data transaksi dan data pelanggan individu dari DMS Kalla Toyota periode 2019-2023. Berikut adalah *Pipeline Data Analysis* yang menjelaskan alur analisis data secara terstruktur dari pengumpulan data hingga evaluasi model prediksi.



Gambar 5. Pipeline Analysis

2.2 Tahapan Penelitian

Tahapan Penelitian “Prediksi Pelanggan *Churn* Menggunakan Al*goritma Random Forest* Dengan Segmentasi K-Means Dalam Industri Retail Otomotif” dapat dilihat pada Gambar berikut:

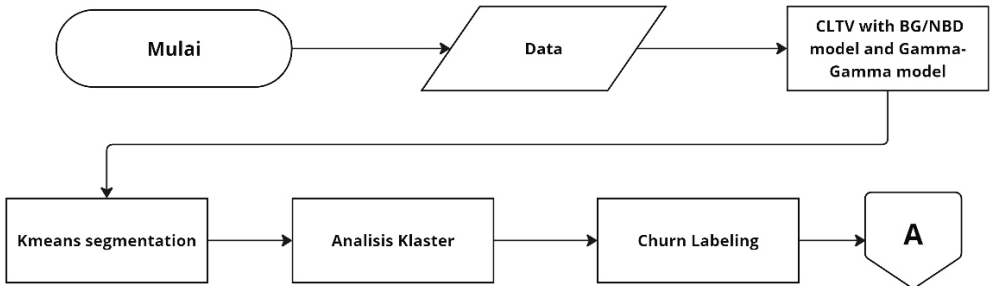


Gambar 6. Tahapan penelitian

Berikut merupakan keterangan dari alur penelitian yang digunakan diantaranya:

a. Pengumpulan Data
   Sistem dimulai dengan tahap pengumpulan data dari *Dealer Management System* (DMS) Kalla Toyota. Data yang digunakan adalah data pelanggan bengkel mobil di dealer Kalla Toyota selama lima tahun terakhir, mulai dari tahun 2019 hingga 2023. Tujuan pengumpulan data ini adalah untuk melihat adanya pemberian layanan gratis untuk mobil baru. Setelah itu dilakukan *preprocessing* data sebelum memasuki proses *data mining* dalam penentuan *cluster*.
b. *Preprocessing* Data
   Tahap *Preprocessing* data merupakan langkah penting dalam menganalisis data dan membuat model pembelajaran mesin. Prosesnya dimulai dengan mengimpor dataset dari berbagai sumber seperti file CSV, Excel, *database*, atau API. Setelah data diimpor, langkah selanjutnya adalah memverifikasi dan memahami struktur data dengan memeriksa baris pertama dan terakhir kumpulan data, serta memeriksa tipe data setiap kolom dan menghitung jumlah baris dan kolom memahami. ukuran kumpulan data. Menyelesaikan *missing value* adalah langkah penting berikutnya, dimana data yang hilang dapat dihilangkan atau diisi menggunakan metode seperti *mean*, *median*, atau *mode*. Normalisasi atau standarisasi data kemudian dilakukan untuk memastikan semua fitur berada dalam skala yang sama, sehingga berkontribusi secara proporsional terhadap analisis atau model.

c. *Data Mining Clustering*



Gambar 7. Tahapan kmeans

Selanjutnya akan dilakukan analisis cluster dengan algoritma *K Means* untuk membentuk *cluster* yang sesuai dengan preferensi pelanggan.

1) CLTV (Customer lifetime value)

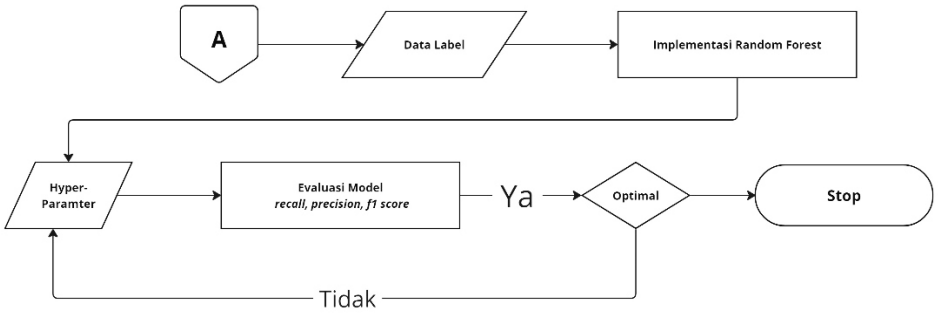
Tahap awal adalah membentuk model CLTV. Dalam hubungan non kontraktual, hilangnya pelanggan tidak dapat diobservasi secara langsung oleh perusahaan. Salah satu pendekatan yang dapat dilakukan adalah perhitungan nilai seumur hidup pelanggan (CLTV) yang berfokus pada memperkirakan probabilitas atau frekuensi pembelian berulang pelanggan di masa depan berdasarkan model frekuensi dan moneter terkini (RFM) dan data lainnya. Sarjana Arthur Hughes mengusulkan model RFM pada tahun 1990an, yang meletakkan dasar teoritis untuk penghitungan nilai seumur hidup pelanggan (Abbasimehr & Shabani, 2019). Pada tahap ini akan dilakukan analisis dengan model BG/NBD (Banggang et al., 2020), model ini digunakan untuk menggambarkan perilaku transaksi berulang dalam konteks hubungan pelanggan non kontraktual. Artinya, pengguna bisa membeli produk kapan saja tanpa batasan waktu. Model ini dapat menggunakan data historis transaksi pengguna untuk memprediksi waktu transaksi dan tingkat *turnover* (*churn*) setiap pengguna di masa depan. Setelah menggunakan model BG/NBD untuk memprediksi frekuensi pembelian dan probabilitas *churn*, dilanjutkan dengan model *Gamma-Gamma* untuk memperkirakan nilai transaksi rata-rata. Model *Gamma-Gamma* digunakan untuk menangani perbedaan besar dalam nilai transaksi di antara pelanggan yang telah bertransaksi. Kemudian menggabungkan prediksi frekuensi pembelian dan nilai transaksi rata-rata untuk menghasilkan perkiraan CLTV. Hasil ini akan memberikan gambaran tentang seberapa berharga setiap pelanggan dari segi pendapatan jangka panjang dan memberikan wawasan tentang potensi pelanggan yang loyal. Setelah itu dilanjutkan dengan penggunaan algoritma K-Means untuk melihat bagaimana hasil segmentasi karakteristik dari masing-masing pelanggan dalam hal nilai seumur hidup dan perilaku pembelian dengan menggunakan data yang ada. Model ini digunakan untuk menggambarkan perilaku transaksi berulang dalam konteks hubungan pelanggan non kontraktual. Artinya, pengguna bisa membeli produk kapan saja tanpa batasan waktu. Model ini dapat menggunakan data historis transaksi pengguna untuk memprediksi waktu transaksi dan tingkat *turnover* (*churn*) setiap pengguna di masa depan. Setelah menggunakan model BG/NBD untuk memprediksi frekuensi pembelian

dan probabilitas *churn*, dilanjutkan dengan model *Gamma-Gamma* untuk memperkirakan nilai transaksi rata-rata. Model *Gamma-Gamma* digunakan untuk menangani perbedaan besar dalam nilai transaksi di antara pelanggan yang telah bertransaksi. Kemudian menggabungkan prediksi frekuensi pembelian dan nilai transaksi rata-rata untuk menghasilkan perkiraan CLTV. Hasil ini akan memberikan gambaran tentang seberapa berharga setiap pelanggan dari segi pendapatan jangka panjang dan memberikan wawasan tentang potensi pelanggan yang loyal.

2) *K-Means segmentation*, analisis *cluster* dan data labeling

Setelah itu dilanjutkan dengan penggunaan algoritma *K Means* untuk melihat bagaimana hasil segmentasi karakteristik dari masing-masing pelanggan dalam hal nilai seumur hidup dan perilaku pembelian dengan menggunakan data yang ada.

d. Model Prediksi



Gambar 8. Tahapan random forest

Setelah menyelesaikan proses pembentukan *cluster* dan pemberian label kepada pelanggan, langkah selanjutnya adalah mengimplementasikan data kedalam analisis prediksi yaitu implementasi algoritma *Random Forest* (*supervised learning*), untuk melihat bagaimana aturan-aturan yang terbentuk serta melihat bagaimana model dapat memprediksi pelanggan yang akan mengalami perpindahan (*churn*).

e. Evaluasi

Pada tahap ini, aplikasi yang telah diuji dievaluasi untuk menilai kinerjanya berdasarkan beberapa aspek:

- Evaluasi nilai fitur yang memainkan peran dan mempengaruhi prediksi kelas pelanggan.
- Penilaian akurasi *Random Forest* dengan menggunakan metode *cross-validation*.
- Penilaian model uji menggunakan *confusion matrix* berdasarkan recall, *precision*, *f1 score*, dan akurasi.

2.3 Waktu dan Lokasi Penelitian

Penelitian ini mulai dilaksanakan sejak disetujuinya proposal penelitian ini, yaitu 19 Maret 2024 hingga pada proses pelaporan hasil penelitian ini pada 28 Agustus 2024. Penelitian ini dilakukan di Laboratorium Animasi dan Multimedia Teknik Informatika, Fakultas Teknik, Universitas Hasanuddin. Adapun lokasi pengumpulan yaitu di Wisma Kalla, Jl. DR. Ratulangi No.8.

2.4 Instrumen Penelitian

Adapun instrumen penelitian yang digunakan dalam penelitian sebagai berikut:

1. Hardware
   a. Sistem Operasi : Windows 11 Home
   b. Prosesor : AMD Ryzen 7 7840HS with Radeon 780M Graphics~3.80 GHz
   c. RAM : 16GB
   d. Penyimpanan : SSD internal 512GB dan SSD external 512GB
2. Software
   a. Google Colab
   b. Visual Studio Code
   c. Mysql
3. Bahasa Pemrograman yang digunakan yaitu Python

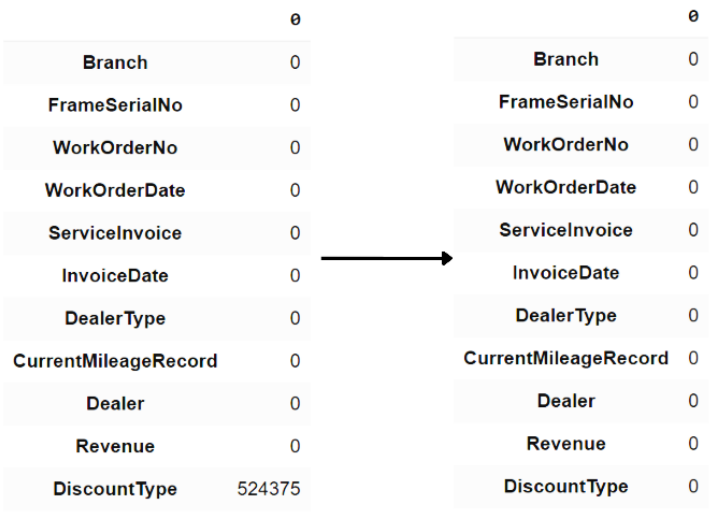
2.5 Perancangan Sistem

Pada bagian ini akan dijelaskan secara rinci perancangan sistem, sesuai dengan tahapan penelitian yang telah dijabarkan diatas.

2.4.1 *Preprocessing*

1. *Handling Missing Value*

Proses ini menangani data yang hilang atau tidak lengkap dalam dataset. Ini merupakan langkah penting dalam analisis data untuk memastikan integritas dan keakuratan model yang dibangun.



Gambar 9. *Handling missing value*

Adapun penjelasan dari masing-masing atribut yang ada pada data yaitu sebagai berikut:

Tabel 2. Atribut penelitian

Attribute	Information	Meaning
Branch	Cabang Kalla Toyota	Cabang-cabang penyebaran dealer Kalla Toyota
FrameSerialNo	ID Kendaraan	Kode unik pada masing masing kendaraan yang datang ke bengkel
WorkOrderNo	Nomor Pengerjaan	Berisi kode unik untuk tiap pekerjaan pada pelanggan
WorkOrderDate	Tanggal penerbitan	Tanggal yang menunjukkan penerbitan dari pekerjaan pada WorkOrderNo
ServiceInvoice	Faktur Tagihan	Faktur pencatatan biaya dari services yang dilakukan
InvoiceDate	Tanggal Penerbitan Faktur	Tanggal penerbitan faktur transaksi
DealerType	Tipe Dealer	Berisikan informasi tentang dealer yang merupakan Mitra dari bagian Kalla Toyota (Internal) dan dealer yang bukan dari Kalla Toyota (Eksternal)

CurrentMileageRecord	Kilometer Tempuh	Catatan mengenai jumlah mil atau kilometer yang telah ditempuh oleh kendaraan
Dealer	Sebaran Bengkel	Berisikan nama nama unit cabang bengkel
Revenue	Jumlah Pembayaran	Berisikan Jumlah pembayaran pelanggan
DiscountType	No Diskon	Merupakan nomor kategori dari program diskon

2. *Data Filtering*

Data Filtering adalah proses dalam analisis data di mana subset data dipilih atau diekstraksi dari dataset yang lebih besar berdasarkan kriteria tertentu. Tujuan dari data *filtering* adalah untuk fokus pada bagian data yang relevan dengan analisis yang sedang dilakukan, sambil menghilangkan data yang tidak sesuai atau tidak diperlukan. Dalam konteks penelitian ini, *data filtering* digunakan untuk memisahkan antara pelanggan individu dan pelanggan korporat yang berada di bawah naungan perusahaan yang telah berlangganan.

```
df_com = df_data.merge(customer, on='FrameSerialNo', how='inner')

# Menampilkan DataFrame hasil
df_com.head(5)
```

	Unnamed: 0	Branch_x	FrameSerialNo	WorkOrderNo_x	WorkOrderDate	ServiceInvoice	InvoiceDate_x	DealerType	VehicleUnit	WorkOrderStatus	CurrentMileageRecord	VehicleModel	Dealer
0	18658	SERUI GENERAL REPAIR	MHKM5EA3JGK009432	20102/SWO/18/04/00175	2018-04-05 11:00:17	20102/INV/19/02/00259	2019-02-09 10:29:48	Internal	DD-1290-UU	Settlement	49448.0	AVANZA	HK GSO
1	1673016	KOLAKA GENERAL REPAIR	MHKM5EA3JGK009432	21602/SWO/23/02/00011	2023-02-01 09:42:15	21602/INV/23/02/00029	2023-02-01 16:58:24	Internal	DD-1290-UU	Waiting For Payment	105472.0	AVANZA	HK GSO
2	1853671	LUWUK BANGGAI GENERAL REPAIR	MHKM5EA3JGK009432	22102/SWO/20/01/00418	2020-01-31 09:19:06	22102/INV/20/01/00405	2020-01-31 10:27:35	External	DN-990-CE	Settlement	62512.0	AVANZA	PT. HASRAT ABADI
3	20561	SERUI GENERAL REPAIR	MHKM5EA3JHK061292	20102/SWO/18/04/01120	2018-04-28 11:40:39	20102/INV/19/01/00865	2019-01-23 09:29:23	Internal	DD-8255-XY	Settlement	19937.0	AVANZA	HK GSO
4	1335675	MAMUJU GENERAL REPAIR	MHKM5EA3JHK061292	21002/SWO/19/10/00200	2019-10-10 10:20:24	21002/INV/19/10/00187	2019-10-10 14:05:38	Internal	DD-1804-YU	Settlement	60139.0	AVANZA	HK GSO

Gambar 10. Dataframe pelanggan

Pada proses ini dilakukan dengan perintah *merge* untuk penggabungan dua *DataFrame*, yaitu df_data dan *customer*, berdasarkan kolom FrameSerialNo. Hanya baris-baris yang memiliki nilai FrameSerialNo yang sama di kedua DataFrame yang akan disertakan dalam DataFrame.

3. Pembentukan Fitur RFM

Pada proses ini dilakukan perhitungan tiga metrik utama—*Recency* (kapan terakhir kali pelanggan bertransaksi), *Frequency* (seberapa sering pelanggan bertransaksi), dan *Monetary* (berapa banyak yang telah dibelanjakan pelanggan). Nilai-nilai ini akan digunakan sebagai input dalam perhitungan CLTV, membantu untuk lebih akurat mengestimasi nilai jangka panjang dari setiap pelanggan berdasarkan perilaku pembelian mereka.

```
today_date = dt.datetime(2023, 1, 31)

cltv_df = df_com.groupby('FrameSerialNo').agg({
    'InvoiceDate_x': [
        lambda x: (x.max() - x.min()).days,  # recency
        lambda x: (today_date - x.min()).days  # T
    ],
    'ServiceInvoice': lambda x: x.nunique(),  # frequency
    'Revenue': lambda x: x.sum()  # monetary
})
```

FrameSerialNo	recency	T	frequency
MHKE8FA3JJK004893	159.571429	165.000000	51
MHKA4DA3JHJ116452	190.428571	195.000000	53
MHKM1BA2JDK024039	187.428571	192.714286	48
MR0KB8CD2K1120946	177.285714	185.714286	46
MHKA4GA5JHJ007153	180.000000	193.285714	48
MHKA4DA5JKJ002033	154.857143	166.000000	39
MHF11KF7000022251	152.857143	158.285714	37
MHKM5EA3JKK145011	162.714286	170.428571	38
MR0ES8BB8G0061654	160.714286	185.857143	50
MHKM1BA3JCK111629	199.714286	205.714286	41

Gambar 11. Pembentukan atribut RFM

Perintah diatas menghitung fitur RFM untuk setiap pelanggan dalam DataFrame *df_com*. Dengan *today_date* ditetapkan sebagai 31 Januari 2023, proses ini mengelompokkan data berdasarkan `FrameSerialNo` dan menghitung tiga metrik: *Recency* diukur dengan selisih hari antara transaksi terakhir dan pertama, *Frequency* dihitung sebagai jumlah transaksi unik

yang dilakukan, dan *Monetary* adalah total pendapatan dari pelanggan. Data ini digunakan untuk analisis lebih lanjut dan perhitungan CLTV.

4. Data Labeling

Pada proses ini dilakukan *labeling* data atau kategori pada data berdasarkan kriteria tertentu. Pada proses ini, data yang tidak terstruktur atau tidak berlabel diorganisir dan diberi kategori yang dapat mempermudah analisis.

```
dalam_kota = ['URIP SUMOHARJO GENERAL REPAIR', 'URIP SUMOHARJO BODY PAINT', 'SERUI GENERAL REPAIR','ALAUDDIN GENERAL REPAIR']

# Fungsi untuk mengkategorikan cabang
def categorize_branch(branch):
    if branch in dalam_kota:
        return 'DalamKota'
    else:
        return 'LuarKota'

# Terapkan fungsi untuk membuat kolom baru 'kategori lokasi'
df_use_filtered['ServiceLocationCategory'] = df_use_filtered['Branch_x'].apply(categorize_branch)
df_use_filtered['DealerBeliLocationCategory'] = df_use_filtered['Branch_y'].apply(categorize_branch)
```

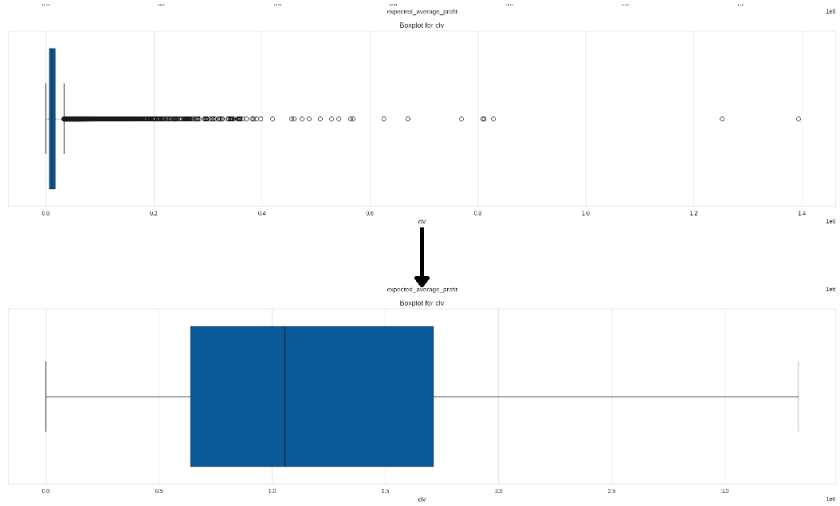
ServiceLocationCategory	DealerBeliLocationCategory
LuarKota	LuarKota
DalamKota	LuarKota
DalamKota	DalamKota
LuarKota	LuarKota
LuarKota	LuarKota

Gambar 12. Labeling data pelanggan

Pada proses diatas dilakukan dengan menggunakan fungsi *categorize_branch* untuk mengkategorikan cabang-cabang berdasarkan apakah mereka berada dalam daftar lokasi dalam kota makassar atau luar kota makassar. Fungsi ini diterapkan pada kolom-kolom tertentu dalam DataFrame untuk membuat kolom baru yang berisi label lokasi ('DalamKota' atau 'LuarKota'). Proses ini memudahkan analisis lebih lanjut dengan memberikan informasi terstruktur tentang lokasi cabang.

5. *Outlier Detection and Treatment*

Pada proses ini di mana nilai-nilai ekstrim atau *outlier* dalam data diidentifikasi dan diolah. Proses ini melibatkan menghitung batas bawah dan atas berdasarkan *Interquartile Range* (IQR) untuk setiap kolom numerik. Nilai-nilai yang berada di luar batas ini kemudian diganti dengan nilai batas terdekat, sehingga data menjadi lebih konsisten dan analisis menjadi lebih stabil.



Gambar 13. Handling outlier

6. Label Encoding

Pada proses ini di mana nilai-nilai kategorikal dalam kolom data diubah menjadi angka untuk memudahkan pemrosesan dalam model pembelajaran mesin. Pada proses ini, setiap kategori unik diubah menjadi nilai numerik yang sesuai menggunakan objek *LabelEncoder*.

	FrameSerialNo	ServiceLocationCategory	DealerBeliLocationCategory	DealerType
0	,HKM1CA4JEK070700	LuarKota	LuarKota	Internal
1	.	DalamKota	LuarKota	Internal
2	0107870	DalamKota	DalamKota	External
3	085346096412	LuarKota	LuarKota	Internal
4	0F2318975	LuarKota	LuarKota	Internal

	FrameSerialNo	ServiceLocationCategory	DealerBeliLocationCategory	DealerType
0	,HKM1CA4JEK070700	1	1	1
1	.	0	1	1
2	0107870	0	0	0

Gambar 14. Label encoding nilai kategorikal

7. Pembentukan Nilai CLTV

Dalam proses analisis CLTV menggunakan model BG/NBD, data disiapkan dengan mengatur ulang kolom dan melakukan beberapa transformasi. Pertama, kolom diperbarui untuk memastikan hanya variabel yang relevan yang digunakan, yaitu *recency*, T, *frequency*, dan *monetary*. Analisis dilakukan hanya pada transaksi yang memiliki nilai moneter positif. Variabel *monetary* dihitung ulang sebagai pendapatan rata-rata per transaksi dengan membagi total pendapatan dengan frekuensi transaksi. Selanjutnya, kolom *recency* dan T yang awalnya dalam satuan hari, diubah menjadi minggu untuk memberikan perspektif waktu yang lebih panjang dalam analisis dan membuat data lebih mudah untuk dianalisis dalam konteks model CLTV.

```
cltv_df.columns = cltv_df.columns.droplevel(0)
cltv_df.columns = ['recency', 'T', 'frequency', 'monetary']
cltv_df = cltv_df[cltv_df['monetary'] > 0]

# asumsi bahwa nilai moneter adalah pendapatan rata-rata per transaksi
cltv_df['monetary'] = cltv_df['monetary'] / cltv_df['frequency']

# ubah hari ke minggu
cltv_df['recency'] = cltv_df['recency'] / 7
cltv_df['T'] = cltv_df['T'] / 7
```

Gambar 15. Perhitungan variable RFM

Setelah mempersiapkan data dengan mengubah *recency* dan T menjadi satuan minggu dan menghitung *monetary* sebagai rata-rata pendapatan per transaksi, model *Beta Geo Fitter* (BGF) dilakukan untuk analisis CLTV. Model ini diinisialisasi dengan koefisien penaliser untuk meningkatkan stabilitas estimasi parameternya. Proses *fitting* dilakukan pada data *frequency, recency,* dan T, dan menghasilkan parameter model yang mencakup alpha, beta, r, dan a, yang masing-masing menggambarkan karakteristik perilaku pembelian pelanggan dalam dataset tersebut.

```
bgf = BetaGeoFitter(penalizer_coef=1e-06)
bgf.fit(cltv_df['frequency'],
        cltv_df['recency'],
        cltv_df['T'])

<lifetimes.BetaGeoFitter: fitted with 84814 subjects, a: 0.48, alpha: 347.66, b: 10.19, r: 17.99>
```

Gambar 16. Model betageometrik

Setelah menggunakan model *GammaGammaFitter* untuk memprediksi pendapatan rata-rata dari pelanggan berdasarkan frekuensi dan nilai moneter, kita bisa mengidentifikasi pelanggan yang paling menguntungkan. Model ini menghitung *expected_average_profit* untuk setiap pelanggan, yang merefleksikan keuntungan rata-rata yang diperkirakan per transaksi dari pelanggan tersebut di masa depan. Hasilnya,

kita dapat menyortir dan menampilkan 10 pelanggan paling menguntungkan berdasarkan nilai *expected_average_profit* yang telah dihitung, memberikan wawasan berharga bagi strategi pemasaran dan keputusan bisnis yang lebih terinformasi.

```
ggf = GammaGammaFitter(penalizer_coef=0.01)
ggf.fit(cltv_df['frequency'], cltv_df['monetary'])


# top 10 most profitable customers
ggf.conditional_expected_average_profit(cltv_df['frequency'],
                                        cltv_df['monetary']).sort_values(ascending=False).head(10)


cltv_df["expected_average_profit"] = ggf.conditional_expected_average_profit(cltv_df['frequency'],
                                                                             cltv_df['monetary'])

cltv_df.sort_values("expected_average_profit", ascending=False).head(10)
```

recency	T	frequency	monetary	expected_purc_1_week	expected_purc_1_month	expected_average_profit
0.0	23.857143	1	7.410916e+07	0.043749	0.174417	1.298670e+08
0.0	6.428571	1	5.814239e+07	0.050195	0.200083	1.018872e+08
0.0	118.714286	1	5.665815e+07	0.003010	0.012008	9.928626e+07
0.0	153.428571	1	4.503214e+07	0.000759	0.003027	7.891315e+07
0.0	26.714286	1	4.443787e+07	0.042464	0.169301	7.787177e+07
0.0	152.285714	1	4.164764e+07	0.000793	0.003165	7.298223e+07
0.0	5.000000	1	3.812725e+07	0.050633	0.201826	6.681321e+07

Gambar 17. Model gamma-gamma

Dalam kode ini, fungsi *customer_lifetime_value* dari pustaka *lifetimes* digunakan untuk mengestimasi nilai seumur hidup (CLTV) pelanggan selama 12 bulan berdasarkan data seperti frekuensi pembelian, waktu terakhir pembelian, total periode observasi dalam minggu, dan pendapatan moneter rata-rata per transaksi. Dengan parameter time=12, CLTV dihitung untuk satu tahun ke depan, dengan penggunaan frekuensi mingguan dan tingkat diskon sebesar 0.01 untuk menyesuaikan nilai masa depan uang. Hasil perhitungan diurutkan dan 20 pelanggan dengan nilai CLTV tertinggi ditampilkan, menawarkan gambaran tentang pelanggan yang paling bernilai selama periode tersebut.

```
cltv = ggf.customer_lifetime_value(bgf,
                                   cltv_df['frequency'],
                                   cltv_df['recency'],
                                   cltv_df['T'],
                                   cltv_df['monetary'],
                                   time=12,   # 12 month
                                   freq="W", # Weekly
                                   discount_rate=0.01)

cltv.head()
cltv = cltv.reset_index()
cltv.sort_values(by= "clv", ascending=False).head()

# 50 most valuable customers in a 12-month
cltv.sort_values(by="clv", ascending=False).head(20)
```

	FrameSerialNo	clv
63437	MHKE8FB3JNK063938	2.639329e+08
21005	MHFM1CA4JAK042400	2.370257e+08
56753	MHKE8FA3JMK067554	1.567562e+08
559	JTFSS22P0K0184584	1.536713e+08
13044	MHFGX8GS7K0506318	1.530248e+08
17811	MHFK23F39J2047904	1.457461e+08
9429	MHFE2CJ3JGK113743	1.270347e+08
14491	MHFJW8EM1J2352740	1.185943e+08

Gambar 18. Pelanggan dengan nilai CLTV

Dari gambar diatas pelanggan dengan nomor seri "MHKE8FB3JNK063938". Pelanggan ini memiliki nilai CLTV sebesar 263.932.900, yang menunjukkan nilai ekonomi yang sangat tinggi berdasarkan interaksi mereka sebelumnya. Ini bisa mencerminkan dari pembelian yang sering, nilai transaksi yang tinggi, atau keduanya. Memiliki CLTV yang tinggi seperti ini menunjukkan bahwa pelanggan ini adalah salah satu kontributor utama terhadap pendapatan perusahaan dan mungkin harus menjadi fokus utama dalam upaya retensi dan pemasaran karena

potensi mereka untuk menghasilkan pendapatan yang signifikan di masa depan.

```
cltv_final = cltv_df.merge(cltv, on="FrameSerialNo", how="left")
cltv_final.sort_values(by="clv", ascending=False).head(10)

# Crossing customer information
cltv_final[cltv_final["FrameSerialNo"] == 'MR0KB8CD2M1126555']
cltv_final[cltv_final["FrameSerialNo"] == 'MR0DS8CD6H0262675']

# apply them scaler to understanding clearly
scaler = MinMaxScaler(feature_range=(0, 1))
scaler.fit(cltv_final[["clv"]])
cltv_final["scaled_clv"] = scaler.transform(cltv_final[["clv"]])
cltv_final.sort_values(by="scaled_clv", ascending=False).head()
```

expected_average_profit	clv	scaled_clv
1.298670e+08	2.639329e+08	1.000000
1.018872e+08	2.370257e+08	0.898053
6.681321e+07	1.567562e+08	0.593924
7.787177e+07	1.536713e+08	0.582236
6.544228e+07	1.530248e+08	0.579787

Gambar 19. Penggabungan data CLTV kedalam dataframe

Selanjutnya data CLTV digabungkan dengan *dataframe* utama berdasarkan FrameSerialNo untuk menyatukan informasi pelanggan dengan nilai CLTV pelanggan. Selanjutnya, data disortir berdasarkan nilai CLTV dalam urutan menurun untuk mengidentifikasi 10 pelanggan dengan nilai CLTV tertinggi. Data tersebut kemudian dinormalisasi menggunakan *MinMaxScaler*, yang mengubah skala nilai CLTV ke rentang 0 hingga 1, memudahkan perbandingan relatif antara pelanggan. Berikut perhitungan sederhana pada model CLTV dengan data yang digunakan sebagai berikut:
Parameter BG/NBD

- $a = 0.48$
- $\alpha = 347.66$
- $b = 10.19$
- $r = 17.99$

Dengan data pelanggan yaitu memiliki *recency* 28.285714, nilai T yaitu 33.0, nilai *frequency* yaitu 2, nilai *monetary* yaitu 1,138,891.00 dan dengan memberikan CLTV pada 1 minggu kedepan. Dengan nilai *hypergeometric* yaitu ${}_1^2F = (19.99, 12.19; 11.67; 0.002471210398853358) = 1.0530204$

yang di ambil dari fungsi *generalized hypergeometric functions*, karena perhitungannya yang cukup kompleks untuk dilakukan secara manual.

1. Menghitung ekspektasi jumlah transaksi dengan model bg/nbd

$$E(Y(t)|X = x, t_x, T, r, \alpha, a, b)$$

$$= \frac{a+b+x-1}{a-1} \times \frac{\left[1-\left(\frac{\alpha+T}{\alpha+T+t}\right)^{r+x} {}_2^1F\left(r+x, b+x; a+b+x-1; \frac{t}{\alpha+T+t}\right)\right]}{1+\delta_{(x>0)}\frac{a}{b+x-1}\left(\frac{\alpha+T}{\alpha+t_x}\right)^{r+x}}$$

$$= \frac{0.48+10.19+2-1}{0.48-1} \times \frac{\left[1-\left(\frac{347.66+33.0}{347.66+33.0+1}\right)^{17.99+2} {}_2^1F(1.0530204668122196)\right]}{1+\frac{0.48}{10.19+2-1}\left(\frac{347.66+33.0}{347.66+28.285714}\right)^{17.99+2}}$$

$$\approx 0.0065$$

2. Menghitung ekspektasi nilai moneter $E(M)$ dengan model *gamma-gamma* dengan nilai $p = 2.13$, $q = 0.08$, $\gamma = 2.05$:

$$E(M|p, q, \gamma, m_x, x) = \left(\frac{q-1}{px+q-1}\right)\frac{\gamma p}{q-1} + \left(\frac{px}{px+q-1}\right)m_x$$

$$= \left(\frac{0.08-1}{2.13+2-1}\right)\frac{2.05\times 2.13}{0.08-1} + \left(\frac{2.13\times 2}{2.13+2-1}\right)1{,}138{,}891.00$$

$$\approx 1{,}453{,}362.41$$

3. Menghitung CLTV untuk 1 minggu ke depan

$$CLV(t=1) = \frac{E(Y_1)\times E(M_1)}{(1-p)^t} = \frac{0.0065\times 1{,}453{,}362.41}{(1-0.01)^1} \approx 9{,}354.31$$

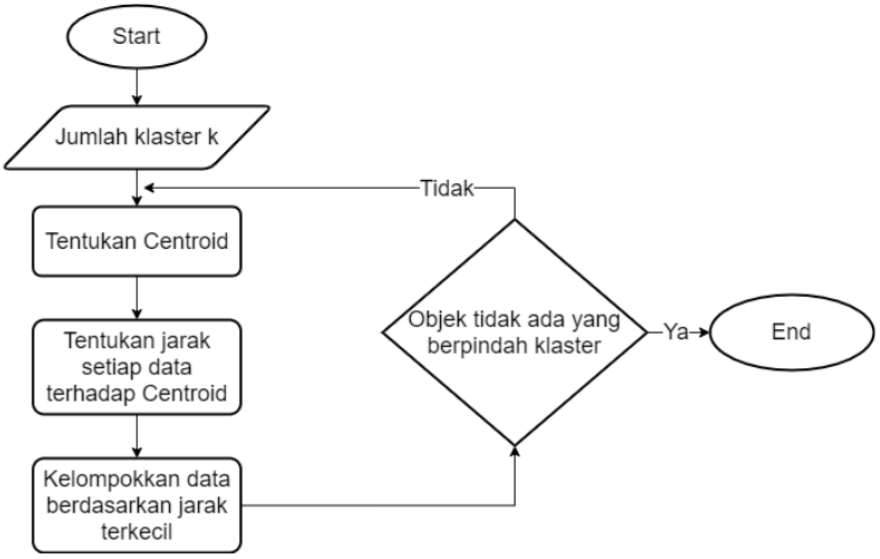
Nilai *Customer Lifetime Value* (CLTV) yang diharapkan untuk pelanggan pelanggan tersebut dalam 1 minggu ke depan adalah Rp 9,354.31. Jika dalam data yang akan dianalisis adalah 6 atau 12 bulan kedepan maka harus dilakukan perhitungan terpisah untuk setiap minggu (t = 1 hingga 24) dan menjumlahkannya.

8. PCA

Setelah mendapatkan nilai CLTV masing-masing pelanggan, langkah selanjutnya adalah memanfaatkan *Principal Component Analysis* (PCA) untuk mengolah data lebih lanjut. PCA dapat digunakan untuk mereduksi dimensi data yang mungkin memiliki banyak fitur, sehingga memungkinkan analisis yang lebih sederhana namun tetap mempertahankan variansi maksimum dari data asli. Dengan mereduksi dimensi, kita dapat mengidentifikasi pola atau struktur yang tidak terlihat dalam dimensi yang lebih tinggi.

2.4.2 *Model Clustering K-Means*

Berikut tahap *Clustering* seperti pada Gambar



Gambar 20. Tahapan prose k means

Berikut penjelasan lebih lanjut mengenai proses *clustering* pengunjung seperti pada Gambar 16:

a. Dataset yang di input meliputi *Service Location Category*, Dealer Beli *Location Category*, DealerType, *Current Mileage Record*, *DiscountType*, *recency*, T, *frequency*, *monetary*, dan CLTV Titik data setiap data ditunjukkan pada Gambar berikut

	PC1	PC2	PC3	PC4	PC5	PC6	FrameSerialNo
0	0.017875	-0.370309	0.250058	0.247421	-0.632246	0.037669	,HKM1CA4JEK070700
1	0.615276	0.196493	0.088056	0.097061	-0.539341	0.139986	.
2	0.833657	0.835100	0.106783	0.736664	0.376657	-0.165528	0107870
3	-0.777966	-0.095547	0.056964	-0.110134	-0.006936	-0.007423	085346096412
4	0.426443	-0.576644	-0.249421	-0.018504	-0.099308	-0.028870	0F2318975

Gambar 21. Dataset hasil PCA

b. Jumlah *cluster* optimal dihitung menggunakan Metode *Elbow* dengan mengukur *Inertia* dan *Silhouette Score*, mempertimbangkan patahan yang terlihat pada grafik *Elbow*. Pencarian jumlah cluster dilakukan dalam rentang 2 hingga 11 *cluster*. Jumlah *cluster* optimal yang ditemukan adalah

sebanyak 5 *cluster*. Berikut ini adalah perhitungan inersia pada titik data pertama X_i dalam cluster 0, dengan pusat *cluster* C_K, tanpa melibatkan penjumlahan titik data dan *cluster* lainnya.

$$SSE = \sum_{K=1}^{K} \sum_{x_i \in S_K} (X_i - C_k)^2$$

$$SSE_{\_\text{titikdata0\&cluster0}} = (X_i - C_k)^2$$

$$= (0.01787476 - (-0.23821136))^2 + (-0.37030902 - 1.05687017)^2 + (0.25005795 - (-0.1006629))^2 - + (0.24742074 - (-0.04603965))^2 + (-0.63224558 - 0.07752369)^2 + (0.03766893 - 0.04749762)^2$$

$$= 0.61452659$$

Berdasarkan perhitungan titik data 0 pada *cluster* 0 memiliki inertia sekitar 0.61452659.

Berikut merupakan perhitungan *silhouette score* pada titik data 0 (b) pada *cluster* 0 dan belum melibatkan *datapoint* dan *cluster* lain.

$$S(i) = \frac{b(i) - a(i)}{max(a(i), b(i))}$$

$$= \frac{1.5361328916314365 - 1.1347763895904937}{1.5361328916314365}$$

$$\approx 0.2612772008381942$$

Berdasarkan perhitungan titik data 0 pada *cluster* 0 memiliki *silhouette score* 0,24.

c. Berikut inisialisasi posisi *centroid* pada gambar dibawah ini

```
Centroids:
[[-0.23821136  1.05687017 -0.1006629  -0.04603965  0.07752369  0.04749762]
 [-0.69512043 -0.12408918  0.04125548 -0.14109328  0.05421068  0.01514969]
 [ 0.39514948 -0.51133336 -0.16798823 -0.00475841 -0.17193865 -0.01716353]
 [ 0.38149737 -0.36667564  0.55131395  0.53138956  0.3539722  -0.01519344]
 [ 0.93470331  0.74247402 -0.03703204  0.04475297 -0.09235576 -0.03739467]]
```

Gambar 22. Centroid dataset

Pada *cluster* 0, PCA 1 berpusat pada -0.23821136 dan PCA 2 berpusat pada 1.05687017. Untuk nilai PCA 3 hingga PCA 6 posisinya adalah -0.1066629, -0.04603965, 0.87752369, dan 0.84749762. Hal ini menggambarkan sebaran dan susunan data pada *cluster* 0 dalam ruang yang direduksi menjadi enam dimensi, memperlihatkan variasi dan hubungan antar karakteristik dalam *cluster* tersebut.

d. Jarak antar *centroid* dan titik data individu dihitung dengan menggunakan metode jarak *Euclidean*. Untuk menentukan jarak dari *centroid* setiap *cluster* ke titik data pertama (index 0), berikut perhitungan jarak *centroid* pada titik data 0

Jarak Titik Data 0 ke *Centroid* 0:

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

$$= \sqrt{\begin{matrix}\left(0.01787476-(-0.23821136)\right)^2+\\(-0.37030902-1.05687017)^2+\\\left(0.25005795-(-0.1006629)\right)^2+\\(0.24742074-(-0.04603965))^2\\+(-0.63224558-0.07752369)^2+\\(0.03766893-0.04749762)^2\end{matrix}}$$

$$\approx 1.6780$$

Jarak Titik Data 0 ke *Centroid* 1:

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

$$= \sqrt{\begin{matrix}\left(0.01787476-(-0.69512043)\right)^2+\\(-0.37030902-(-0.12408918))^2+\\(0.25005795-0.04125548)^2+\\(0.24742074-(-0.14109328))^2\\+(-0.63224558-0.05421068)^2+\\(0.03766893-0.01514969)^2\end{matrix}}$$

$$\approx 1.1114$$

Jarak Titik Data 0 ke *Centroid* 2:

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

$$= \sqrt{\begin{matrix}(0.01787476-0.39514948)^2+\\(-0.37030902-(-0.51133336))^2+\\\left(0.25005795-(-0.16798823)\right)^2+\\(0.24742074-(-0.00475841))^2\\+\left(-0.63224558-(-0.17193865)\right)^2+\\(0.03766893-(-0.01716353))^2\end{matrix}}$$

$$\approx 0.7845$$

Jarak Titik Data 0 ke *Centroid* 3:

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

$$= \sqrt{\begin{matrix}(0.01787476 - 0.38149737)^2 + \\ (-0.37030902 - (-0.36667564))^2 + \\ (0.25005795 - 0.55131395)^2 + \\ (0.24742074 - 0.53138956)^2 \\ +(-0.63224558 - 0.3539722)^2 + \\ (0.03766893 - (-0.01519344))^2\end{matrix}}$$

≈ 1.1301

Jarak Titik Data 0 ke *Centroid* 4:

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

$$= \sqrt{\begin{matrix}(0.01787476 - 0.93470331)^2 + \\ (-0.37030902 - 0.74247402)^2 + \\ \left(0.25005795 - (-0.03703204)\right)^2 + \\ (0.24742074 - 0.04475297)^2 \\ +\left(-0.63224558 - (-0.09235576)\right)^2 + \\ (0.03766893 - (-0.03739467))^2\end{matrix}}$$

≈ 1.5810

Berdasarkan hasil perhitungan tersebut, jarak titik data 0 memiliki posisi terdekat dengan *centroid* 2, sehingga titik data ini dimasukkan ke *cluster* 2.

e. Selanjutnya, setiap sampel akan diarahkan ke *cluster* yang memiliki *centroid* terdekat berdasarkan jarak terkecil dari sampel tersebut ke *centroid*.
f. Algoritma *K-Means* terus melakukan pembaruan pada posisi *centroid* hingga posisi *centroid* menjadi stabil, yang berarti tidak atau hanya sedikit perubahan yang terjadi pada posisi *centroid*.
g. Hasil *clustering* disimpan ke dalam sebuah *dataframe*, memudahkan pengelolaan dan analisis data yang telah diklasifikasikan ke dalam kelompok-kelompok berdasarkan kesamaan karakteristik.

2.4.3 *Performa Kmeans*

Perhitungan *K-Means* dievaluasi melalui metode *Silhouette Score*, dimana nilai skor di atas 0,5 mengindikasikan pengelompokan data yang efektif. Sementara itu, nilai Index *Davies-Bouldin* (DBI) yang rendah menunjukkan efisiensi pengelompokan yang lebih baik. Evaluasi ini menjadi standar dalam menguji efektivitas pengelompokan data. Berikut adalah perhitungan evaluasi DBI antara cluster 0 terhadap cluster 1 dan cluster 2.

Rasio dengan *cluster* 1

$$DBI = \frac{1}{N}\sum_{i=1}^{N} max_{j \neq i}\left(\frac{S_0 + S_1}{d_{01}}\right)$$

$$= \frac{0.604494750941409 + 0.6207608599941916}{1.7122793217910781}$$

$$\approx 0.7155$$

Rasio dengan *cluster* 2

$$DBI = \frac{1}{N}\sum_{i=1}^{N} max_{j \neq i}\left(\frac{S_0 + S_2}{d_{02}}\right)$$

$$= \frac{0.604494750941409 + 0.4461954698286953}{1.2054833479950788}$$

$$\approx 0.8714$$

Langkah-langkah yang sama untuk menghitung nilai DBI dengan evaluasi terhadap setiap *cluster* (0, 1, 2, 3, dan 4) dengan mengidentifikasi rasio maksimum antara *cluster* tersebut dengan *cluster* lainnya. Setelah mendapatkan nilai rasio maksimum dari masing-masing *cluster*, rata-ratakan nilai-nilai ini untuk mendapatkan nilai DBI secara keseluruhan. Metode ini membantu menilai seberapa efektif pengelompokan dalam memisahkan data ke dalam *cluster* yang berbeda.

2.4.4 *Uji ANOVA*

Uji ANOVA digunakan untuk memeriksa perbedaan signifikan pada semua fitur antar semua *cluster*. Fitur yang memiliki p-value lebih dari 0,05 menyatakan penolakan hipotesis nol, yang menyatakan bahwa rata-rata antar kelompok tidak sama, menandakan bahwa model *K-Means* berhasil melakukan pengelompokan dengan efektif. Sebaliknya, *p-value* di bawah 0,05 berarti hipotesis nol diterima, menunjukkan tidak ada bukti yang cukup untuk memastikan bahwa *K-Means* efektif dalam mengelompokkan data. Untuk menghitung nilai *p-value*, langkah pertama adalah menentukan *f-value*, yang memerlukan perhitungan SSB (*Sum of Squares Between groups*). Perhitungan SSB melibatkan agregasi SSB dari setiap titik data di setiap *cluster*. Berikut adalah ilustrasi perhitungan SSB untuk fitur frekuensi di *cluster* pertama.

$$SSB = \sum n_i\left(\overline{X}_i - \overline{X}_G\right)^2$$
$$= 10807(3.5950772647358193 - 4.926637739942461)$$
$$= 19161.38700372532$$

Kemudian, untuk menentukan SSW (*Sum of Squares Within groups*) secara total, kita menjumlahkan hasil SSW dari semua titik data dalam masing-masing *cluster*. Contoh berikut adalah perhitungan, berfokus pada fitur frekuensi.

$$SSW = \sum\left(\overline{X}_{ij} - \overline{X}_i\right)^2$$
$$= (1.0 - 3.5950772647358193)^2$$
$$\approx 6.742282216365236$$

Selanjutnya, dfB dan dfW dihitung sebagai berikut.

$$dfB = k - 1$$
$$= 5 - 1$$
$$= 4$$
$$dfW = N - k$$
$$= 84{,}812 - 5$$
$$= 84{,}807$$

Selanjutnya, setelah SSB, SSW, dfB, dan dfW dihitung secara keseluruhan, maka selanjutnya menghitung MSB dan MSW sebagai berikut.

$$MSB = \frac{SSB}{dfB}$$
$$= \frac{133{,}528.3953}{4} = 33{,}382.0988$$
$$MSW = \frac{SSW}{dfW}$$
$$= \frac{750{,}581.1447}{84{,}807} = 8.8559$$
$$F = \frac{MSB}{MSW}$$
$$= \frac{33{,}382.0988}{8.8559} = 3{,}773.5867$$

Selanjutnya, p-value dihitung dengan membandingkan nilai *f-value* dengan *f-distribusi* pada tabel distribusi *f* sehingga didapatkan *p-value* pada fitur frekuensi dengan nilai 1.1102230246251565e-16. Hal tersebut menunjukkan *p-value* <0,05 sehingga menolak hipotesis nol.

2.4.5 *Uji Post Hoc Menggunakan Uji Tukey HSD*

Dalam analisis mendalam, peneliti memanfaatkan Uji *Tukey* HSD untuk mengidentifikasi dimana perbedaan antara setiap cluster berada secara signifikan. Jika hipotesis nol (H0) ditolak, ini menunjukkan bahwa ada perbedaan signifikan antara *cluster* yang dianalisis. Sebaliknya, jika H0 tidak ditolak, berarti tidak ada perbedaan signifikan antara *cluster* tersebut. *Tukey's* HSD dan *p-adj* langsung digunakan untuk mengukur signifikansi statistik dari perbandingan ini, sedangkan perbedaan rata-rata dan interval kepercayaan menyediakan gambaran mengenai magnitudo perbedaan serta presisi dari estimasi ini. Berikut adalah penghitungan perbedaan rata-rata antara *cluster* 0 dan cluster 1 pada fitur frekuensi, dengan mengasumsikan *Mean Squared Error* (MSE) sama dengan *Mean Squared Within* (MSW).

$$Perbedaan\ Rataan = \overline{x}_2 - \overline{x}_1$$
$$= 3.732466747279323 - 3.5950772647358193$$
$$= 0.13738948254350358$$

Selanjutnya, estimasi standar kesalahan pada *cluster* 0 dan *cluster* 1 dihitung sebagai berikut

$$Estimasi\ Standar\ Kesalahan = \sqrt{MSE\left(\frac{1}{n_1} + \frac{1}{n_2}\right)}$$
$$= \sqrt{8.850462163628434\,\frac{1}{29772} + \frac{1}{10807}}$$
$$= 0.03341004527683278$$

Selanjutnya, nilai *q* dihitung, yang kemudian dibandingkan dengan nilai kritis dari distribusi *q Tukey* untuk mendapatkan nilai *p-adj*. Berikut perhitungan nilai *q* pada *cluster* 0 (kelompok 1) dengan *cluster* 1 (kelompok 2).

$$q = \frac{Perb\ rataan\ terbesar}{Estimasi\ Standar\ Kesalahan}$$
$$= \frac{0.13738948254350358}{0.03341004527683278}$$
$$\approx 4.11222078285456$$

Selanjutnya, *p-adj* dihitung dan diperoleh nilai 0.0004 dan sangat sulit dihitung secara manual dan harus menggunakan software. Konsepnya *p-adj* membandingkan nilai *q* yaitu 4.11222078285456 dengan nilai $q\propto$ yaitu 3.7153464418465902, dimana *q*>$q\propto$ berarti fitur frekuensi memiliki nilai signifikan pada *cluster* 0 dan *cluster* 1 yang benar menolak H0. Selanjutnya, Batas Bawah dan Atas dihitung pada *cluster* 0 dan *cluster* 1 pada fitur jenis kelamin sebagai berikut.

$$Batas\ bawah = Perb\ Rataan\ \pm (q_{\propto} x\ Estimasi\ Standar\ Kesalahan)$$
$$= 0.13738948254350358$$
$$-(3.7153464418465902 \times 0.03341004527683278)$$
$$= 0.043259589702289437$$
$$Batas\ atas = Perb\ Rataan + (q_{\propto} x\ Estimasi\ Standar\ Kesalahan)$$
$$= 0.13738948254350358$$
$$-(3.7153464418465902 \times 0.03341004527683278)$$
$$= 0.2615193753847177$$

Untuk *cluster* 0 dan *cluster* 1 atau kelompok 1 dan 2, Batas Bawah adalah 0.0463 dan Batas Atas adalah 0.2285. Interval ini tidak melintasi nol, menunjukkan perbedaan signifikan.

2.4.6 *Implementasi Random Forest Classifier*

Pada tahap awal penerapan algoritma *random forest*, dilakukan dengan sejumlah kolom yang sebelumnya telah digunakan dalam proses *K Means* dan akan dinormalisasi terlebih dahulu.

Proses ini melibatkan pemilihan kolom-kolom yang sebelumnya digunakan dalam *K Means* untuk dinormalisasi. Fitur-fitur diambil dari kolom-kolom tersebut, sementara label target diambil dari kolom *Cluster_PCA*. Data fitur kemudian dinormalisasi menggunakan *MinMaxScaler* untuk memastikan nilainya berada dalam rentang 0 hingga 1. Setelah itu, label target dikonversi menjadi kategori, dan baik fitur-fitur yang dinormalisasi maupun label target diubah menjadi array *numpy* untuk memudahkan pemrosesan lebih lanjut.

```
columns_to_normalize = ['LocationCategory', 'DealerType',
                        'CurrentMileageRecord', 'recency',
                        'T', 'frequency', 'monetary',
                        'expected_average_profit', 'clv']

# Memisahkan fitur dan label target
X = df_final[columns_to_normalize]
y = df_final['Cluster_MinMax']

# Normalisasi data hanya pada fitur
scaler = MinMaxScaler(feature_range=(0,1))
X_scaled = scaler.fit_transform(X)
X_scaled_df = pd.DataFrame(X_scaled, columns=columns_to_normalize)

# Konversi label target menjadi kategori
y_cat = y.astype('category')

X_arr = X_scaled_df.to_numpy()
y_arr = y_cat.to_numpy()
```

Gambar 23. Proses awal random forest

Selanjutnya data akan dibagi menggunakan metode *k-fold* dengan k=5 untuk pembentukan kelompok. Pemilihan nilai k ini secara empiris telah terbukti memberikan estimasi kesalahan pengujian yang optimal, menghindari bias yang berlebihan atau varian yang terlalu tinggi. Dimana perbedaan nilai k akan mempengaruhi nilai akurasi dan komputasi model, Dimana k=5 dapat mengurangi biaya komputasi yang efisien saat bekerja dalam kumpulan data yang besar. sebagaimana dijelaskan dalam literatur oleh Kohavi dan Nadarajan (Kohavi, 1995; Nadarajan & Sulaiman, 2023)

```
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import KFold

#kfold split data-5

kf = KFold(n_splits= 5)

fold_number = 1

for train_index, test_index in kf.split(X_arr):
  X_train = X.iloc[train_index]
  X_test = X.iloc[test_index]
  y_train = y_cat.iloc[train_index]
  y_test = y_cat.iloc [test_index]
```

Gambar 24. Proses split data dengan k fold

Setelah data dipisahkan, langkah selanjutnya adalah melakukan validasi nilai data untuk mencegah *overfitting* dan *underfitting*. Selain itu, proses ini dilanjutkan dengan mencari parameter optimal untuk model *Random Forest*, yang akan melatih data secara acak guna menghasilkan model yang paling baik.

```
train_scores, test_scores = list(), list()

#parameter kedalaman pohon
values = [i for i in range(1, 13)]


for i in values:
  model = RandomForestClassifier(max_depth=i)
  model.fit(X_train, y_train)
  train_yhat = model.predict(X_train)
  train_acc = accuracy_score(y_train, train_yhat)
  train_scores.append(train_acc)
  test_yhat = model.predict(X_test)
  test_acc = accuracy_score(y_test, test_yhat)
  test_scores.append(test_acc)
```

```
from sklearn.model_selection import RandomizedSearchCV as RSCV

param_grid = {'n_estimators': np.arange(50, 200, 10),
              'max_features': np.arange(0.1, 1, 0.1),
              'max_depth': [3, 5, 7, 9, 12],
              'max_samples': [0.3, 0.5, 0,8]}
model = RSCV(RandomForestClassifier(), param_grid, n_iter=15). fit(X_train, y_train)
model = model.best_estimator_
```

Gambar 25. Validasi k fold dan pencarian parameter terbaik

Parameter terbaik yang telah diperoleh akan diterapkan pada algoritma *Random Forest*, memungkinkan model untuk melakukan klasifikasi pada setiap data dengan lebih akurat.

2.4.7 *Evaluasi Model Random Forest*

Tahap terakhir dari *Random Forest* adalah evaluasi model, yang bertujuan untuk menilai kinerja model *machine learning* dalam memprediksi kelas pelanggan. Evaluasi model akan dilakukan dengan cara berikut.

1. Evaluasi Fitur, Mengevaluasi fitur-fitur untuk mengetahui mana yang paling berpengaruh dalam menentukan kelas setiap pelanggan.

```
model.feature_importances_
imp_df = pd.DataFrame({
    'Varname': X_train.columns,
    'Imp': model.feature_importances_
})

imp_df.sort_values(by='Imp', ascending=False)
```

Gambar 26. Evaluasi pengaruh fitur

2. *Cross-Validation*, Menggunakan teknik *k-fold cross-validation* untuk membagi data menjadi beberapa bagian, melakukan pelatihan dan pengujian berulang kali, lalu menghitung rata-rata akurasi dari setiap iterasi.

```
scores = cross_val_score(rfc, X_scaled_df, y_cat, cv = 5)
# Menampilkan hasil akurasi untuk setiap fold
for i, score in enumerate(scores):
    print(f"Fold {i+1}: Accuracy = {score:.4f}")

# Menampilkan rata-rata akurasi dan standar deviasi
print(f"Mean Accuracy: {scores.mean():.4f}")
print(f"Standard Deviation: {scores.std():.4f}")
```

Gambar 27. Validasi akurasi dari k fold

3. *Confusion Matrix*, Menilai hasil prediksi menggunakan *confusion matrix* memungkinkan pengukuran metrik penting seperti *precision*, *recall*, akurasi, dan *F1 Score*. Dalam bagian ini, *recall* akan menjadi acuan utama untuk memastikan bahwa model dapat dengan baik mendeteksi pelanggan yang berpotensi berpindah atau *churn* berdasarkan kelasnya.

```
print(classification_report(y_train, y_pred_train))
accuracy = accuracy_score(y_train, y_pred_train)
print("Akurasi train:", accuracy)

print(classification_report(y_test, y_pred_test))
accuracy = accuracy_score(y_test, y_pred_test)
print("Akurasi test :", accuracy)
```

Gambar 28. Confusion matrix

Dengan perhitungan pada masing masing metrik *cluster* 0 pada data train adalah sebagai berikut:

$$Recall_0 = \frac{TP}{TN + FP} = \frac{8455}{8455 + 977} \approx 0.896$$

$$\text{Precision}_0 = \frac{TP}{TP + FP} = \frac{8455}{8455 + 1221} \approx 0.874$$

$$F1Score_0 = 2 \times \frac{Recall \times precision}{Recall + precision} = 2 \times \frac{0.896 \times 0.874}{0.896 + 0.874} \approx 0.885$$

$$Akurasi_0 = \frac{8455 + 57297}{8455 + 1221 + 977 + 57297} \approx 0.969$$