

BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

Penelitian terdahulu digunakan sebagai landasan penelitian saat ini. Penelitian terdahulu memberikan pemahaman mendalam mengenai gambaran penelitian yang telah dilakukan sebelumnya. Adanya penelitian terdahulu memungkinkan peneliti untuk mengidentifikasi celah pengetahuan yang masih belum terpenuhi. Dengan memahami maksud dan tujuan penelitian sebelumnya, penelitian ini dapat diarahkan secara lebih terfokus, menghindari duplikasi, dan memastikan kontribusi unik terhadap bidang pengetahuan yang ada.

Dalam mencari penelitian terdahulu, dilakukan pencarian jurnal utama untuk *association rule mining* pada data transaksi. Langkah ini penting dalam memperoleh wawasan yang mendalam dalam bidang ini. Melalui penelusuran literatur yang cermat, peneliti dapat mengidentifikasi kerangka kerja, metode, dan pendekatan yang telah terbukti efektif dalam analisis asosiasi, serta aplikasinya pada konteks data transaksi. Dengan mempelajari temuan dan pendekatan yang diterbitkan dalam jurnal utama, pengetahuan dari penelitian sebelumnya dapat diintegrasikan ke dalam metodologi penelitian saat ini. Tujuannya untuk meningkatkan kualitas dan relevansi penelitian.

U M W N
U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

Tabel 2. 1 Tabel Penelitian Terdahulu

Judul Jurnal	Nama Jurnal	Penulis / Tahun	Metode	Hasil	Kesimpulan
<i>Comparison of Apriori, Apriori-TID and FP-Growth Algorithms in Market Basket Analysis at Grocery Stores</i> [18]	<i>The IJICS (International Journal of Informatics and Computer Science)</i>	Idris, A., Sampetoding, E., Ardhana V., Maritsa, I., Sakri, A., Ruslan, H., Manapa, E. / 2022	Penelitian ini menganalisis dan menilai kinerja algoritma <i>Apriori</i> , <i>FP-Growth</i> , dan <i>Apriori-TID</i> untuk mencari <i>frequent itemsets</i> dan aturan asosiasi dari data transaksi.	Hasil penelitian ini menjelaskan bahwa <i>Apriori</i> , <i>FP-Growth</i> , dan <i>Apriori-TID</i> memiliki evaluasi performa yang serupa, yaitu menghasilkan 162 aturan asosiasi, <i>confidence</i> tertinggi sebesar 0.373. Waktu komputasi tercepat adalah <i>FP-Growth</i> dengan waktu komputasi 75 ms. Di sisi lain, <i>memory usage</i> paling efisien adalah algoritma <i>Apriori</i> dan <i>Apriori-TID</i> sebesar 8.3 MB.	Berdasarkan penelitian ini, dapat disimpulkan bahwa algoritma <i>Apriori</i> , <i>Apriori-TID</i> , dan <i>FP-Growth</i> dapat digunakan untuk menganalisis data transaksi. <i>FP-Growth</i> memiliki waktu komputasi paling cepat, sedangkan <i>Apriori</i> memiliki waktu komputasi paling lama. Di sisi lain, <i>memory usage</i> paling efektif adalah <i>Apriori</i> dan <i>Apriori-TID</i>
<i>Association Rule Mining Algorithms: A Comparative Review</i> [20]	<i>International Research Journal of Engineering and Technology</i>	Sharma, A., Ganpati, A. / 2021	Penelitian ini membandingkan algoritma <i>Apriori</i> , <i>Apriori-TID</i> , <i>FP-Growth</i> , dan ECLAT menggunakan evaluasi <i>memory utilization</i> , <i>execution time</i> , dan <i>demerit</i> untuk masing-masing algoritma.	Hasil penelitian ini menjelaskan bahwa algoritma ECLAT memiliki waktu eksekusi paling cepat dibandingkan algoritma lainnya. Dalam konteks <i>memory utilization</i> , algoritma ECLAT dan <i>FP-Growth</i> menggunakan <i>memory</i>	Berdasarkan penelitian ini, dapat disimpulkan bahwa setiap algoritma memiliki fungsi, kelebihan, dan kekurangannya masing-masing. Peneliti dapat menerpakan algoritma ini pada data transaksi dengan mempertimbangkan

Judul Jurnal	Nama Jurnal	Penulis / Tahun	Metode	Hasil	Kesimpulan
				yang lebih efisien dibandingkan <i>Apriori</i> . Dalam konteks <i>merit</i> dan <i>demerit</i> , <i>Apriori</i> menghasilkan banyak <i>frequent itemsets</i> namun konsumsi <i>memory</i> yang besar, <i>FP-Growth</i> melakukan <i>scanning database</i> yang efisien namun tidak cocok untuk <i>dataset</i> yang besar, dan <i>ECLAT</i> memiliki waktu eksekusi yang cepat namun menggunakan <i>memory</i> yang tidak efisien	<i>execution time, usage of memory, merit, dan demerit</i> masing-masing algoritma.
<i>Clustering Optimization in RFM Analysis Based on K-Means</i> [21]	<i>Indonesian Journal of Electrical Engineering and Computer Science</i>	Gustriansyah, R., Suhandi, N., Antony, F. / 2019	Penelitian ini menganalisis kinerja algoritma <i>K-Means</i> dalam konteks mengelompokkan produk dengan karakteristik yang sama. Dilakukan juga beberapa metode untuk menguji kinerja algoritma, seperti <i>Elbow Method</i> , <i>Davies-Bouldin Index</i> , dan <i>Calinski-Harabasz Index</i> .	Hasil penelitian menunjukkan bahwa algoritma <i>K-Means</i> dapat mengelompokkan produk dengan baik dan meningkatkan akurasi manajemen stok produk. Nilai <i>Davies-Bouldin Index</i> dan <i>Calinski-Harabasz Index</i> secara berturut-turut adalah 0.89 dan 1600.	Berdasarkan penelitian ini, dapat disimpulkan bahwa algoritma <i>K-Means</i> dapat mengelompokkan produk dengan baik, sehingga setiap kluster memiliki karakteristik yang sama. Hal ini dapat meningkatkan akurasi manajemen stok produk.

Judul Jurnal	Nama Jurnal	Penulis / Tahun	Metode	Hasil	Kesimpulan
<i>Enhanced Expectation-Maximization (EM) Clustering through Gaussian Mixture Models (GMM) [22]</i>	<i>International Journal of Innovative Technology and Exploring Engineering (IJITEE)</i>	Nagarjuna, R., Sai, R., Babu, R. / 2019	Penelitian ini berfokus pada pengelompokan data menggunakan <i>Gaussian Mixture Model</i> .	Hasil penelitian menunjukkan bahwa metode GMM memiliki efektivitas yang tinggi dalam pengelompokan data ilmiah, memberikan hasil yang baik dalam mengolah data yang diberikan, dengan tingkat ketepatan yang sangat baik, mencapai 99% untuk kedua <i>dataset</i> yang digunakan, yakni <i>dataset</i> jamur dan <i>dataset</i> MIST.	Berdasarkan penelitian ini, dapat disimpulkan bahwa bahwa metode GMM ini memiliki kinerja yang lebih baik daripada <i>K-Means</i> .
<i>Sequential Pattern Mining Using Apriori and FP-Growth Algorithm [23]</i>	<i>Journal of Data Acquisition and Processing</i>	Wakchaure S., Vishwakarma R., Abdul A. / 2023	Penelitian ini menganalisis kualitas dan performa antara algoritma <i>Apriori</i> dan <i>FP-Growth</i> dalam konteks waktu eksekusi dan konsumsi memori.	Hasil penelitian menunjukkan bahwa kedua algoritma dapat dengan baik menganalisis pola yang sering muncul pada <i>dataset</i> . Dari sisi <i>time execution</i> untuk <i>minimum support</i> 0.2, algoritma <i>FP-Growth</i> memiliki waktu eksekusi selama 268ms, sedangkan <i>Apriori</i> membutuhkan 278ms. Dari sisi waktu eksekusi data dengan jumlah kecil sebesar 500 data,	Berdasarkan penelitian ini, dapat disimpulkan bahwa algoritma <i>FP-Growth</i> dan <i>Apriori</i> dapat digunakan untuk analisis <i>dataset</i> besar. Namun dari sisi kecepatan, <i>FP-Growth</i> lebih cepat dibandingkan <i>Apriori</i> .

Judul Jurnal	Nama Jurnal	Penulis / Tahun	Metode	Hasil	Kesimpulan
				algoritma <i>Apriori</i> unggul dengan waktu eksekusi 15ms. Dari segi waktu komputasi, algoritma <i>FP-Growth</i> unggul untuk semua <i>minimum support</i> yang diuji.	
<i>Comparison Between Apriori and FP-Growth Algorithms on Inventory Model of Item Availability</i> [24]	Jurnal IPTEKS Terapan	Husna R., Hendra Y., Akbar M. / 2020	Penelitian ini membandingkan/ komparasi kualitas dan performa untuk algoritma ARM <i>Apriori</i> dan <i>FP-Growth</i> .	Hasil penelitian menunjukkan bahwa algoritma <i>FP-Growth</i> menghasilkan hasil asosiasi yang lebih baik dibandingkan <i>Apriori</i> . <i>Apriori</i> menghasilkan aturan asosiasi dengan <i>generate</i> kandidat <i>itemset</i> , sedangkan <i>FP-Growth</i> tidak perlu <i>generate</i> kandidat <i>itemset</i> .	Berdasarkan penelitian ini, dapat disimpulkan bahwa algoritma <i>FP-Growth</i> lebih baik dibandingkan <i>Apriori</i> dalam konteks konsumsi <i>space, time, memory,</i> dan <i>process</i> .
<i>Evaluation of Apriori, FP-Growth and ECLAT Association Rule Mining Algorithms</i> [19]	<i>International Journal of Health Sciences</i>	Srinadh, V. / 2022	Penelitian ini membandingkan/ komparasi kualitas dan performa untuk algoritma ARM <i>Apriori</i> , <i>FP-Growth</i> , dan ECLAT.	Hasil penelitian menunjukkan bahwa algoritma ECLAT lebih baik daripada <i>FP-Growth</i> dan <i>Apriori</i> dalam konteks kecepatan waktu eksekusi. Namun, jika dilihat dari konteks yang lainnya, ketiga algoritma ini memiliki pendekatan dan cara	Berdasarkan penelitian ini, dapat disimpulkan bahwa algoritma ECLAT membutuhkan waktu eksekusi yang paling singkat. Namun, disarankan juga untuk mencoba algoritma yang lain, seperti <i>FP-Growth</i> dan <i>Apriori</i> karena ketiga algoritma ini memiliki hasil dan

Judul Jurnal	Nama Jurnal	Penulis / Tahun	Metode	Hasil	Kesimpulan
				kerja yang unik, sehingga tidak ada yang mengungguli satu dengan yang lainnya.	pendekatan yang unik antara satu dengan yang lainnya.
<i>Improvement of k-Means Clustering Performance on Disease Clustering using Gaussian Mixture Model</i> [25]	<i>Journal of System and Management Sciences</i>	Santoso, H., Haw, S. / 2023	Penelitian ini membandingkan dua algoritma <i>clustering</i> , yaitu <i>K-Means</i> dan <i>Gaussian Mixture Model</i> (GMM).	Hasil penelitian menunjukkan bahwa algoritma GMM lebih baik dalam mengatasi masalah dimana sebuah <i>data point</i> harus bisa menjadi <i>member</i> untuk beberapa <i>cluster</i> . <i>K-Means</i> hanya bisa melakukan <i>plotting</i> untuk satu <i>data point</i> ke dalam satu kluster. Namun, <i>K-Means</i> unggul dari segi evaluasi menggunakan <i>silhouette method</i> . Sehingga, <i>K-Means</i> lebih cocok untuk membuat kluster berdasarkan karakteristik dataset <i>disease</i> ini.	Berdasarkan penelitian ini, dapat disimpulkan bahwa kedua algoritma ini memiliki keunggulan masing-masing. Dimana <i>K-Means</i> mampu menghasilkan <i>high intra-cluster</i> dan <i>low inter-cluster similarity</i> . Dari sisi pembentukan kluster yang kompleks, GMM lebih cocok karena algoritma ini dapat menempatkan sebuah <i>data point</i> ke dalam beberapa kluster.
<i>Unsupervised Seismic Facies Classification using K-means and Gaussian Mixture Modeling</i> [26]	<i>A CGG GeoSoftware Company</i>	Russell, B. / 2020	Penelitian ini mencari tahu kemampuan dua algoritma <i>clustering</i> , yaitu <i>K-Means</i> dan <i>Gaussian Mixture Model</i> (GMM).	Hasil penelitian menunjukkan bahwa algoritma GMM dan <i>K-Means</i> dapat dengan baik membuat kluster menggunakan <i>dataset</i> yang kompleks.	Berdasarkan penelitian ini, dapat disimpulkan bahwa algoritma GMM dan <i>K-Means</i> cocok digunakan untuk membuat kluster yang berkualitas, bahkan

Judul Jurnal	Nama Jurnal	Penulis / Tahun	Metode	Hasil	Kesimpulan
					untuk <i>dataset</i> yang kompleks.
<i>Clustering Cloud Workloads: K-Means vs Gaussian Mixture Model</i> [27]	<i>Procedia Computer Science</i>	Patel, E., Kushwaha, D. / 2023	Penelitian ini dilakukan untuk membandingkan efektivitas pengelompokan antara <i>K-Means</i> dan GMM.	Hasil penelitian menunjukkan bahwa GMM memberikan pengelompokan beban kerja yang lebih bermakna dibandingkan dengan <i>K-Means</i> , memungkinkan karakterisasi yang lebih detail terhadap kebutuhan.	Berdasarkan penelitian ini, dapat disimpulkan bahwa penggunaan GMM dapat meningkatkan pemahaman terhadap pola penggunaan sumber daya beban kerja, namun demikian, GMM memiliki kinerja yang lebih lambat dibandingkan dengan <i>K-Means</i> .
<i>Market Basket Analysis with K-Means and FP-Growth Algorithm as Citra Mustika Pandawa Company</i> [28]	<i>MALCOM: Indonesian Journal of Machine Learning and Computer Science</i>	Setyorini, S. G., Sari, E. K., Elita, L. R., Putri, S. A.	Penelitian ini melakukan kombinasi antara algoritma ARM, yaitu <i>FP-Growth</i> dan <i>K-Means</i> untuk menghasilkan aturan asosiasi pada setiap klaster.	Hasil penelitian menunjukkan bahwa diperoleh 5 klaster berdasarkan nilai <i>Davies-Bouldin Index</i> . Jumlah aturan asosiasi yang ditemukan adalah 4 aturan menggunakan 30% dan 50% <i>support</i> dan 50% <i>confidence</i> .	Berdasarkan penelitian ini, dapat disimpulkan bahwa algoritma kombinasi <i>FP-Growth</i> dan <i>K-Means</i> dapat digunakan untuk menganalisis data transaksi dan menghasilkan aturan asosiasi.

Dilakukan pencarian jurnal utama untuk *clustering* dan evaluasi hasilnya. Langkah ini penting dalam memperkuat landasan teoritis dan metodologis penelitian. Melalui analisis terhadap penelitian-penelitian terdahulu, peneliti dapat memahami berbagai pendekatan *clustering* yang telah dikembangkan dan diuji, serta mengevaluasi kinerja dan kecocokannya dalam berbagai konteks aplikasi. Dengan memahami kelebihan dan kekurangan metode-metode *clustering* yang ada, pendekatan yang lebih baik dapat dikembangkan sesuai dengan tujuan penelitian. Tabel penelitian terdahulu dapat dilihat pada Tabel 2.1.

Jurnal utama yang digunakan pada penelitian ini adalah jurnal yang ditulis oleh Idris, A., dkk. yang berjudul “*Comparison of Apriori, Apriori-TID and FP-Growth Algorithms in Market Basket Analysis at Grocery Stores*” [18]. Berdasarkan penelitian tersebut, ditemukan bahwa *Apriori*, *FP-Growth*, dan *Apriori-TID* memiliki evaluasi performa yang serupa, yaitu menghasilkan 162 *frequent itemsets*, 176 aturan asosiasi, dan maksimum *confidence* sebesar 0.373. Waktu komputasi tercepat adalah *FP-Growth* dengan waktu komputasi 75 ms (0.075 detik). Maksimum *memory usage* paling efisien adalah algoritma *Apriori* dan *Apriori-TID* sebesar 8.3 MB.

Perbedaan antara penelitian ini dengan penelitian utama adalah pengimplementasian algoritma ARM yang dikombinasi dengan *clustering* pada *dataset*. Tujuannya untuk mencoba pendekatan lain untuk memaksimalkan performa algoritma ARM. Selain itu, penelitian ini juga memberikan *insights* lain, seperti karakteristik pelanggan untuk setiap kluster. Hal ini tentunya dapat memberikan gambaran yang lebih komprehensif.

Perbedaan penelitian ini dengan penelitian utama juga dapat terlihat pada *dataset* yang digunakan. *Dataset* penelitian ini dan penelitian terdahulu berbeda. *Dataset* yang digunakan pada penelitian sebelumnya adalah data riwayat transaksi pada toko kelontong, sedangkan *dataset* penelitian ini adalah data transaksi pulsa. Pada *dataset* pulsa, peneliti harus menerapkan metode *pre-processing* lain, seperti *group and filter* untuk mengubah data transaksi pulsa yang memiliki 1 produk per transaksi menjadi beberapa produk per transaksi.

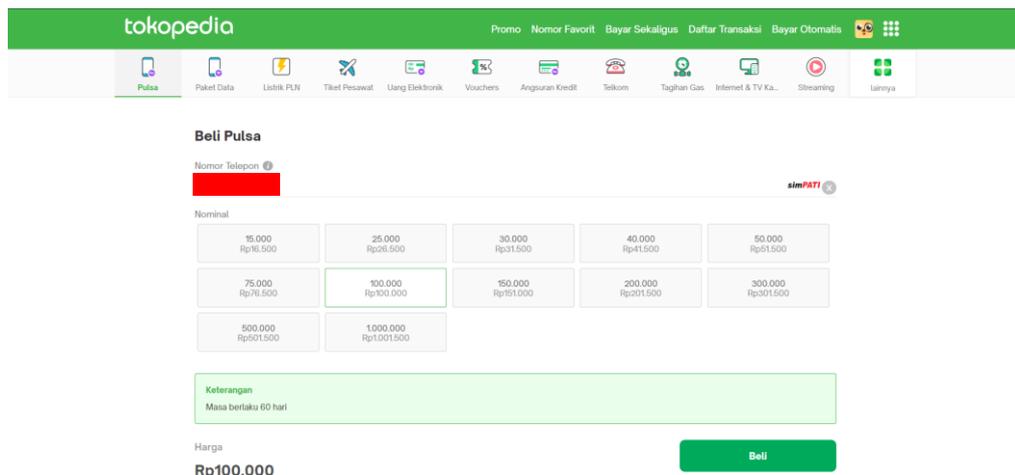
2.2 Tinjauan Teori

2.2.1 Pulsa

Pulsa merupakan sarana yang diperlukan untuk menghubungkan individu dalam berkomunikasi, baik secara langsung maupun jarak jauh [29]. Selain itu, pulsa dapat dianggap sebagai sistem penghitungan yang digunakan untuk menentukan biaya layanan kepada pelanggan. Fungsinya mengetahui penggunaan biaya dalam berkomunikasi melalui telepon, mengirim pesan singkat, berkomunikasi melalui aplikasi pesan instan. Selain itu, pulsa juga dapat digunakan dalam bermain *game online*. Pulsa seringkali juga digunakan sebagai alat pembayaran untuk layanan listrik. Dalam konteks layanan telekomunikasi, pulsa menjadi salah satu metode pembayaran yang paling umum digunakan oleh pengguna untuk mengakses layanan tersebut. Pengguna membeli pulsa dan menggunakannya untuk membayar layanan yang mereka gunakan sehari-hari.

Sejarah perkembangan pulsa telekomunikasi dimulai dari era telepon koin pada awal abad ke-20, dimana pengguna harus memasukkan koin ke dalam telepon untuk melakukan panggilan [30]. Seiring berkembangnya teknologi, khususnya dalam industri telekomunikasi, konsep pulsa berkembang menjadi metode pembayaran yang lebih fleksibel dan praktis. Pulsa telah berkembang menjadi sistem pembayaran yang terintegrasi dengan berbagai layanan telekomunikasi, termasuk layanan seluler dan internet.

Dewasa ini, peran pulsa bagi masyarakat sangatlah krusial. Pulsa sangat relevan dengan kebutuhan era modern yang dipenuhi dengan mobilitas tinggi dan aktivitas digital. Pulsa memfasilitasi komunikasi jarak jauh, memungkinkan individu untuk tetap terhubung dengan keluarga, teman, atau rekan kerja, bahkan di tempat yang terpencil atau sulit dijangkau oleh jaringan telepon kabel [31]. Selain itu, pulsa juga menjadi kunci akses bagi banyak orang untuk mengakses internet melalui *smartphone*, yang merupakan sumber informasi, hiburan, dan sarana produktivitas yang penting dalam kehidupan sehari-hari.



Gambar 2. 1 Pengisian Pulsa pada Platform E-Commerce Tokopedia

Gambar 2.1 menunjukkan penggunaan *platform e-commerce* untuk transaksi pulsa. Dengan berkembangnya *platform e-commerce* dan teknologi digital, pembelian pulsa menjadi semakin mudah, aman, dan nyaman bagi pengguna. Melalui *e-commerce*, bisnis dapat dengan mudah menjangkau pelanggan dari berbagai kalangan. *E-commerce* dapat menganalisis profil, rentang harga, dan keunikan masing-masing pelanggan [32]. Sebelum adanya *e-commerce*, pengisian pulsa memerlukan kunjungan ke konter fisik atau *outlet* khusus, namun sekarang prosesnya dapat dilakukan secara *online* melalui berbagai *platform e-commerce* terkemuka.

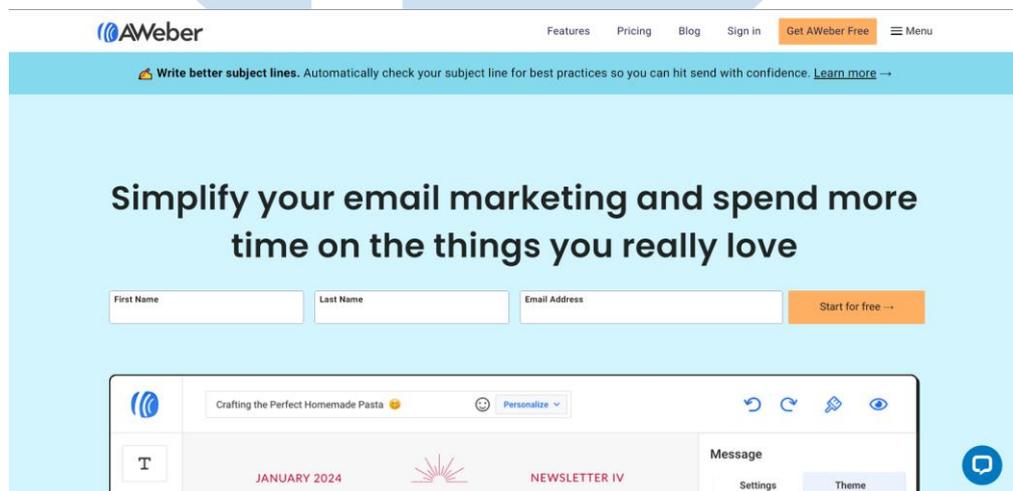
Melalui *platform* ini, pengguna dapat dengan cepat dan mudah membeli pulsa tanpa harus meninggalkan rumah atau kantor mereka. Selain itu, pengisian pulsa melalui *platform e-commerce* juga memberikan keuntungan tambahan, seperti adanya promo khusus, diskon, atau *cashback* yang tersedia secara eksklusif bagi pengguna. Dengan demikian, proses pembelian pulsa secara *online* tidak hanya meningkatkan aksesibilitas bagi pengguna, tetapi juga menciptakan pengalaman yang lebih nyaman dan menguntungkan dalam pengisian pulsa [33].

Transformasi ke pulsa massal menandai perubahan signifikan dalam industri telekomunikasi, dimana pulsa tidak lagi hanya menjadi cara untuk membayar layanan secara individual, tetapi juga menjadi komoditas yang diperdagangkan dalam jumlah besar. Dengan adopsi *platform e-commerce* dan

teknologi digital lainnya, pengisian pulsa tidak lagi terbatas pada konter fisik, melainkan dapat dilakukan secara *online* melalui berbagai *platform* [34]. Ini memungkinkan akses yang lebih mudah dan lebih cepat bagi pengguna untuk mendapatkan pulsa, serta membuka peluang baru bagi pelaku industri untuk menciptakan inovasi dalam layanan dan model bisnis mereka.

2.2.2 *Newsletter*

Newsletter (e-commerce) merupakan alat pemasaran yang digunakan untuk mengirimkan informasi terkini, promosi, dan berita kepada pelanggan atau target pasar dalam jumlah besar melalui *email* secara berkala [35]. Definisi ini menunjukkan bahwa *newsletter* bertujuan untuk menjaga pelanggan tetap memperoleh informasi tentang perkembangan terbaru perusahaan atau produknya. *Newsletter* dapat membentuk hubungan yang kuat antara perusahaan dan pelanggan, sekaligus menjadi saluran efektif untuk menyampaikan pesan pemasaran [36].



Gambar 2. 2 Contoh *Web Application* untuk *Newsletter* AWeber

Gambar 2.2 menunjukkan penintegrasian aplikasi *newsletter*. Perkembangan *newsletter* dapat ditinjau seiring dengan kemajuan teknologi informasi dan popularitas internet. Sebelumnya, *newsletter* sering dikirimkan dalam bentuk cetak, tetapi dengan berkembangnya internet, pengiriman melalui *email* menjadi lebih umum dan efisien. Munculnya *platform-platform* digital membuat pembuatan dan distribusi *newsletter* semakin mudah, memberikan

kesempatan bagi pelaku bisnis untuk lebih aktif berkomunikasi dengan audiens mereka [37].

Pentingnya *newsletter* bagi strategi pemasaran suatu perusahaan tidak dapat diabaikan. *Newsletter* memberikan saluran komunikasi langsung dengan pelanggan, memungkinkan perusahaan untuk membangun hubungan yang lebih personal. Selain itu, *newsletter* juga dapat menjadi alat yang efektif untuk meningkatkan kesadaran merek, memberikan informasi tentang penawaran khusus atau promosi, serta memberikan nilai tambah dengan menyampaikan konten yang relevan dan bermanfaat.

Seiring dengan evolusi teknologi, penggunaan data analysis dalam menyusun *newsletter* menjadi semakin penting. Dengan memanfaatkan hasil *insight* dari data analisis terkait produk yang sering dibeli masyarakat, perusahaan dapat membuat *newsletter* yang lebih personal dan relevan [38]. Melalui data analisis, pelaku bisnis dapat mengidentifikasi pola pembelian, preferensi produk, dan kebutuhan pelanggan. *Newsletter* yang ideal harus mampu memanfaatkan data ini untuk memberikan rekomendasi *sales* yang cerdas, menyesuaikan konten dengan minat pelanggan, dan menciptakan pengalaman yang lebih personal bagi pelanggan.

Dengan menerapkan strategi *newsletter* yang cerdas dan terkini, perusahaan dapat memaksimalkan potensi untuk meningkatkan retensi pelanggan, mendapatkan kepercayaan konsumen, dan pada akhirnya, mencapai tujuan pemasaran yang diinginkan. *Newsletter* yang diarahkan oleh hasil analisis data tidak hanya memberikan informasi yang relevan, tetapi juga dapat meningkatkan efektivitas kampanye pemasaran secara keseluruhan, tetapi juga membuka peluang untuk pertumbuhan bisnis yang berkelanjutan.

2.2.3 Sistem Rekomendasi Penjualan

Sistem rekomendasi penjualan adalah teknik analisis data dalam bidang *data mining* yang digunakan untuk mengidentifikasi hubungan atau asosiasi antara *item* atau variabel dalam *dataset* [39]. Tujuan utama dari sistem rekomendasi penjualan adalah untuk menemukan aturan asosiasi yang menarik

dan bermanfaat dari data transaksional atau transaksi. Dalam konteks penjualan atau bisnis, sistem rekomendasi penjualan dapat digunakan untuk mengidentifikasi pola pembelian yang terjadi secara bersamaan atau berkorelasi, yang kemudian dapat digunakan untuk membuat rekomendasi penjualan yang lebih cerdas dan tepat.

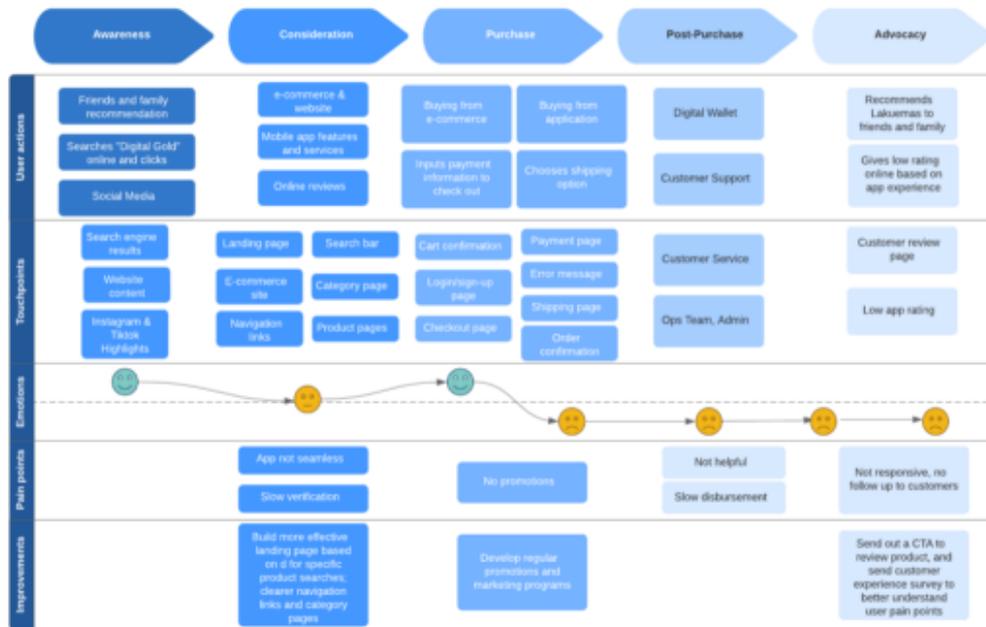
Dalam bisnis, rekomendasi penjualan yang tepat adalah kunci untuk meningkatkan penjualan dan memuaskan pelanggan. Dengan menggunakan sistem rekomendasi penjualan, perusahaan dapat menganalisis pola pembelian pelanggan dan mengidentifikasi hubungan antara produk yang sering dibeli bersama [40]. Hal ini memungkinkan perusahaan untuk dapat membuat rekomendasi penjualan yang lebih personal dan relevan, yang pada gilirannya dapat meningkatkan konversi penjualan dan retensi pelanggan. Dengan demikian, kualitas strategi pemasaran produk dan jasa yang dilakukan oleh perusahaan menjadi lebih tinggi dan menghasilkan *output* maksimal.

Manfaat *insight* dari sistem rekomendasi penjualan bagi perusahaan pulsa di divisi pemasaran *newsletter* sangatlah signifikan. Dengan menggunakan *insight* yang diperoleh dari analisis data transaksi, tim pemasaran dapat membuat *newsletter* yang lebih personal dan relevan. Dengan mengetahui pola pembelian pelanggan, mereka dapat menyesuaikan konten *newsletter* dengan preferensi produk masing-masing pelanggan, serta menawarkan promosi atau diskon yang sesuai dengan kebutuhan mereka [41]. Dengan demikian, *newsletter* dapat menjadi alat pemasaran yang lebih efektif dan dapat meningkatkan keterlibatan pelanggan.

2.2.4 Analisis Data Transaksi

Analisis data transaksi adalah proses mengumpulkan, memproses, dan menganalisis data yang dihasilkan dari transaksi bisnis atau keuangan [42]. Data transaksi mencakup informasi tentang pembelian produk atau layanan oleh pelanggan, termasuk detail seperti produk yang dibeli, jumlahnya, harga, waktu transaksi, dan informasi pelanggan lainnya. Analisis data transaksi bertujuan

untuk memahami pola pembelian, perilaku pelanggan, dan tren pasar yang dapat memberikan wawasan berharga bagi perusahaan.



Gambar 2. 3 Customer Experience Journey [43]

Pentingnya melakukan analisis rekomendasi *sales* terhadap data transaksi sangatlah signifikan dalam strategi pemasaran dan penjualan. Pada Gambar 2.3, perusahaan dapat mengidentifikasi pola pembelian pelanggan, produk yang paling diminati, serta kesempatan untuk meningkatkan penjualan dengan memahami *customer experience journey* [43]. Analisis data transaksi juga memungkinkan perusahaan untuk membuat keputusan yang lebih cerdas dalam mengembangkan strategi pemasaran, menentukan harga produk, dan mengoptimalkan layanan pelanggan.

Proses analisis data transaksi untuk mencari produk yang dibeli secara bersamaan oleh pelanggan dan meningkatkan penjualan disebut *Market Basket Analysis* (MBA). MBA adalah bagian dari kategori *Association Rules*. *Association Rule* merupakan ekspresi implikasi yang ditulis sebagai $A \rightarrow B$, di mana A dan B adalah kumpulan item yang tidak saling beririsan (*disjoint*), sehingga $A \cap B = \emptyset$ [44].

Manfaat untuk *newsletter* pemasaran adalah strategi yang digunakan untuk meningkatkan nilai dari *newsletter* yang dikirim kepada pelanggan dengan menawarkan konten atau produk tambahan yang relevan dan menarik. Dengan menerapkan strategi *up-sell* dalam *newsletter* pemasaran, perusahaan dapat menawarkan produk atau layanan yang lebih eksklusif atau premium kepada pelanggan yang sudah pernah membeli [45]. Hal ini tidak hanya meningkatkan nilai dari *newsletter* itu sendiri, tetapi juga memberikan kesempatan bagi perusahaan untuk meningkatkan pendapatan dari penjualan produk atau layanan tambahan kepada pelanggan yang sudah ada.

Insight yang baik dari analisis data transaksi meliputi pemahaman yang mendalam tentang preferensi produk pelanggan, pola pembelian yang terjadi secara bersamaan, dan tren pasar yang sedang berkembang. Pentingnya analisis data transaksi dalam mengidentifikasi tren pasar yang sedang berkembang dan mengantisipasi pergeseran kebutuhan konsumen [46]. Dengan memahami pola pembelian pelanggan, perusahaan dapat membuat strategi pemasaran yang lebih terarah dan relevan, serta menyesuaikan komunikasi dan penawaran produk dengan kebutuhan dan minat pelanggan. *Insight* ini juga dapat membantu perusahaan mengidentifikasi peluang untuk mengembangkan produk baru atau meningkatkan produk yang ada sesuai dengan permintaan pasar.

Melalui analisis data transaksi, perusahaan dapat mengoptimalkan strategi pemasaran, meningkatkan efisiensi operasional, dan meningkatkan kepuasan pelanggan. Dengan memanfaatkan data transaksi secara efektif, perusahaan dapat mengidentifikasi peluang untuk meningkatkan penjualan, memperluas pangsa pasar, dan memperkuat hubungan dengan pelanggan. Dengan demikian, analisis data transaksi adalah alat yang sangat berharga dalam meningkatkan kinerja bisnis dan mencapai tujuan strategis perusahaan.

2.3 Framework, Algoritma, dan Metode Evaluasi

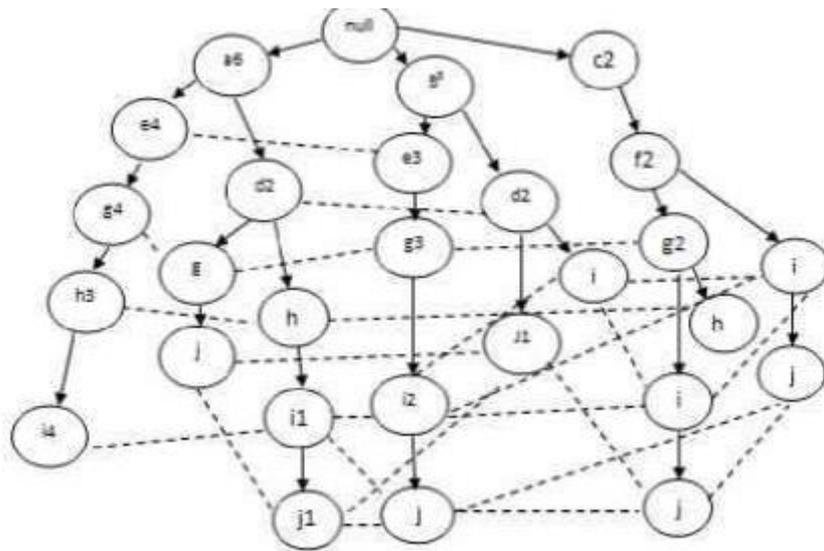
2.3.1 Framework

2.3.1.1 Association Rule Mining

Association Rule Mining (ARM) adalah teknik dalam *data mining* yang bertujuan untuk mengidentifikasi pola atau hubungan yang tersembunyi dalam data transaksional atau dataset besar [47]. Teknik ini digunakan untuk menemukan asosiasi antara item-item dalam dataset, seperti kecenderungan bahwa jika suatu item muncul, maka item lainnya juga cenderung muncul bersamaan. Dengan metrik seperti *support*, *confidence*, dan *lift*, ARM memberikan wawasan baru yang bernilai dalam pengambilan keputusan bisnis, seperti meningkatkan strategi pemasaran, mengoptimalkan penempatan produk, atau memprediksi perilaku pelanggan di masa depan [48]. Oleh karena itu, analisis ARM memungkinkan organisasi untuk memperoleh pemahaman yang lebih dalam tentang pola-pola dalam data mereka, dan menggunakan informasi ini untuk mengambil keputusan yang lebih terinformasi dan efektif

Dalam pengimplementasian ARM, langkah awal yang krusial adalah persiapan data. Data transaksi harus diproses untuk diubah ke dalam format yang sesuai, seperti format transaksi atau daftar *itemset*, sehingga cocok untuk analisis. Proses ini mencakup pembersihan data dari anomali atau nilai yang hilang, serta menyaring *item* yang jarang muncul untuk mengoptimalkan kinerja algoritma.

Setelah data siap, langkah selanjutnya adalah menerapkan algoritma ARM. Salah satu algoritma yang umum digunakan adalah algoritma *Apriori*. Algoritma *apriori* bekerja dengan mencari *itemset* yang sering muncul bersama-sama dalam transaksi. Proses ini melibatkan pencarian kandidat *itemset*, menghitung *support* dari *itemset-itemset* tersebut, dan kemudian memfilter *itemset* yang tidak memenuhi *threshold support* yang ditentukan [49].



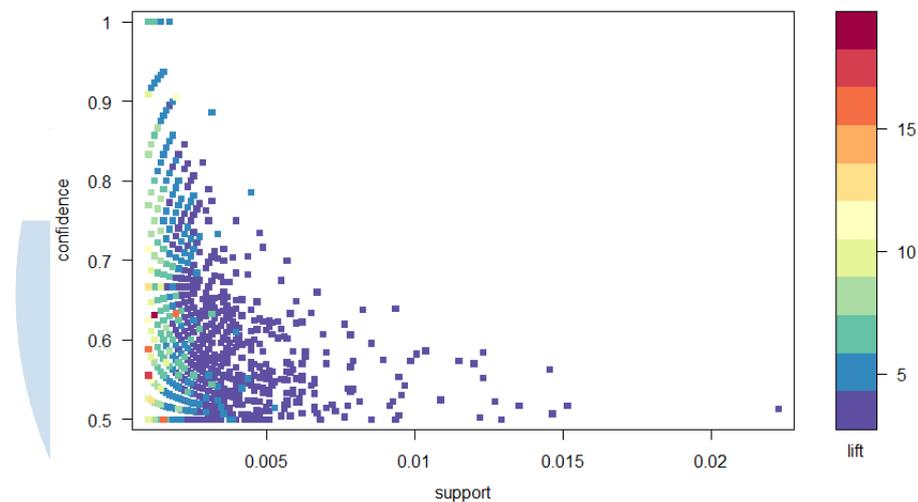
Gambar 2. 4 Decision Tree *FP-Tree* [50]

Selain *Apriori*, terdapat juga algoritma *FP-Growth* yang sering digunakan. Pada Gambar 2.4, algoritma *FP-Growth* menggunakan struktur data pohon *FP-Tree* untuk mencari pola asosiasi dalam *dataset* [50]. Prosesnya melibatkan pembangunan pohon *FP-Tree* dari *dataset* transaksi, ekstraksi pola asosiasi dari pohon *FP-Tree*, dan menghasilkan aturan asosiasi berdasarkan pola yang ditemukan. Keunggulan *FP-Growth* terletak pada kemampuannya untuk mengatasi masalah keterbatasan memori yang sering terjadi pada algoritma *Apriori*, terutama pada *dataset* yang besar [51].

Selain *Apriori* dan *FP-Growth*, algoritma ARM yang populer digunakan adalah ECLAT. Algoritma ECLAT (*Equivalence Class Clustering and Bottom-Up Lattice Traversal*) adalah sebuah algoritma yang digunakan untuk mengekstraksi aturan asosiasi dari data transaksional. ECLAT menggunakan pendekatan *bottom-up* untuk menghasilkan aturan asosiasi dengan mengeksplorasi *lattice*, yang merupakan representasi struktur hirarkis dari kumpulan item yang muncul dalam transaksi [52].

Salah satu keunggulan utama dari algoritma ECLAT adalah kemampuannya untuk menangani *dataset* yang besar dengan efisien. Dengan menggunakan struktur data *TID-list* dan pendekatan *bottom-up*,

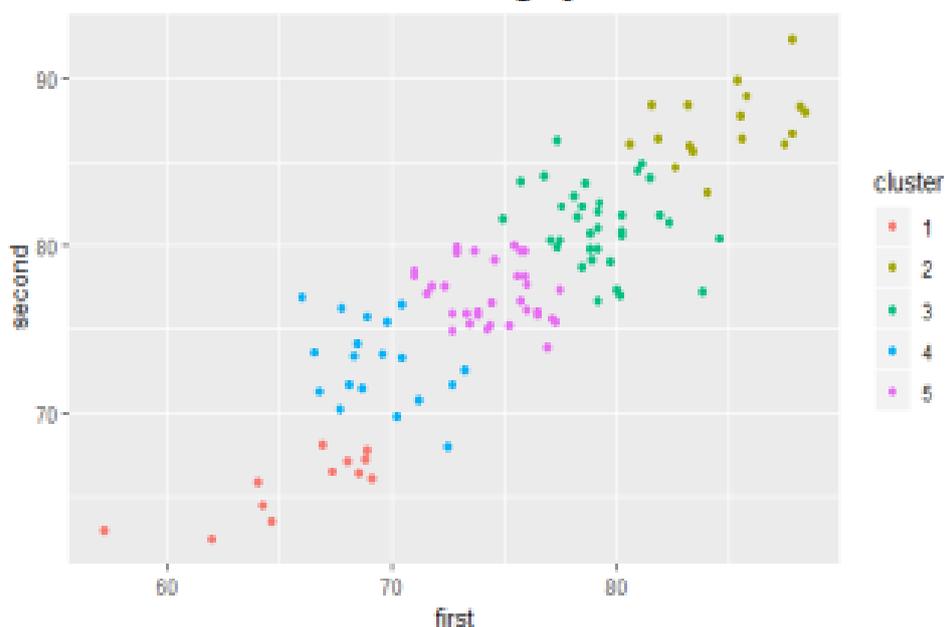
ECLAT dapat menghindari pembangunan keseluruhan matriks itemset seperti yang dilakukan oleh algoritma lainnya seperti Apriori.



Gambar 2. 5 *Support, Confidence, dan Lift* [53]

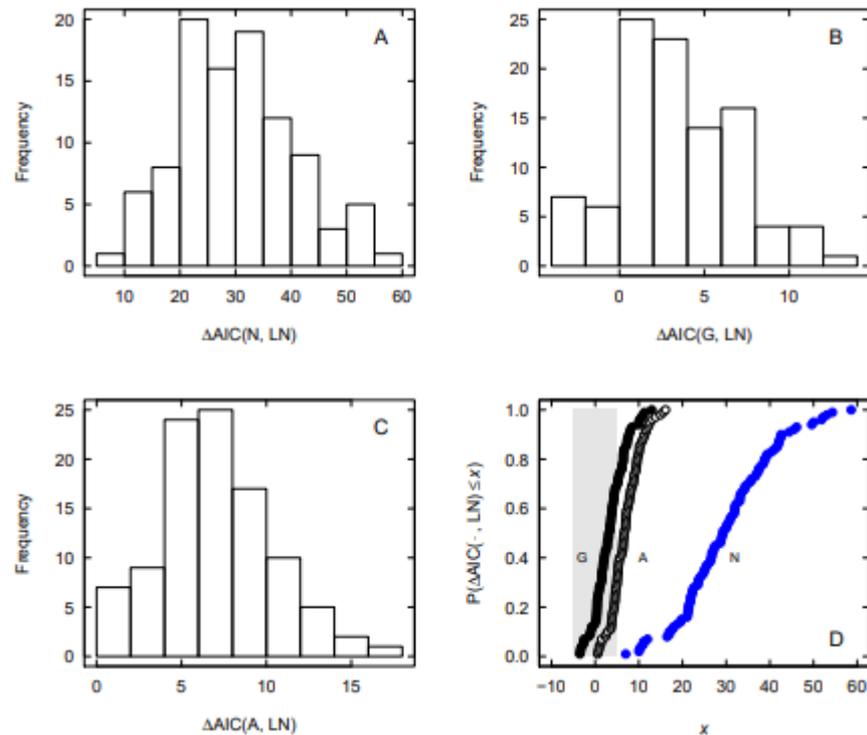
Setelah aturan asosiasi dihasilkan, langkah terakhir adalah evaluasi dan interpretasi aturan tersebut. Ini melibatkan pengukuran kualitas aturan berdasarkan parameter seperti *confidence*, *lift*, dan *support* seperti pada Gambar 2.5. Aturan yang relevan dan berkualitas dapat digunakan untuk membuat rekomendasi penjualan yang lebih cerdas dan terarah bagi perusahaan. Dengan memahami langkah-langkah teknis ini, pengimplementasian ARM dapat dilakukan secara efektif untuk mengungkap pola-pola pembelian yang signifikan dalam data transaksi.

2.3.1.2 Clustering



Gambar 2. 6 Algoritma Clustering [54]

Setelah membentuk model ARM tunggal, langkah selanjutnya adalah menggabungkan model ARM dengan teknik *clustering*. Algoritma *clustering* adalah metode analisis data yang digunakan untuk mengelompokkan objek atau data ke dalam kelompok-kelompok yang serupa berdasarkan kesamaan karakteristik atau atribut tertentu [54]. Tujuan utama dari algoritma clustering adalah untuk mengidentifikasi pola atau struktur tersembunyi dalam data dan memisahkannya ke dalam kelompok-kelompok yang berarti seperti pada Gambar 2.6. Tujuannya adalah untuk meningkatkan kualitas model dari segi keakuratan asosiasi.

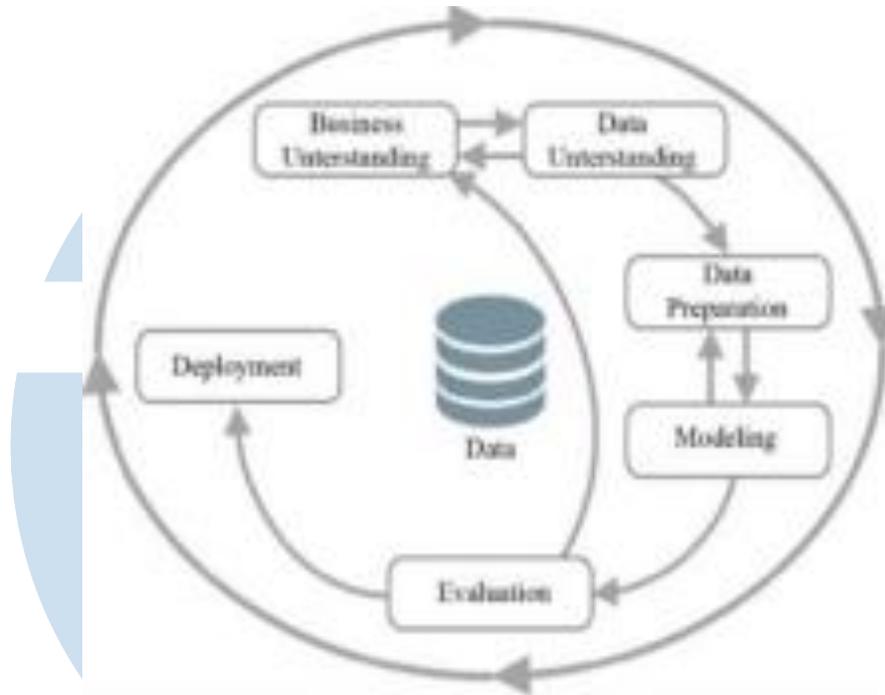


Gambar 2. 7 Metode Evaluasi Kluster AIC [55]

Proses ini mempertimbangkan pembentukan kluster dari dataset yang ada, dimana setiap kluster kemudian akan menerapkan model ARM secara terpisah. Proses pembentukan kluster juga melibatkan penggunaan beberapa metode, seperti *Elbow Method*, *Akaike Information Criterion (AIC)*, *Bayesian Information Criterion (BIC)*, *Davies-Bouldin Index (DBI)*, dan *Callinski-Harabasz Index (CHI)* untuk mendapatkan parameter yang optimal. Pada Gambar 2.7, metode AIC digunakan untuk mencari jumlah kluster yang optimal.

Pembentukan kluster dengan metode ini memungkinkan identifikasi pola asosiasi yang lebih akurat antara item dalam setiap kluster. Dengan mengintegrasikan model ARM ke dalam kluster yang dihasilkan, diharapkan bahwa model asosiasi yang diperoleh akan menjadi lebih adaptif terhadap karakteristik data yang lebih kompleks. Strategi ini juga memungkinkan analisis yang lebih mendalam terhadap hubungan antara item-item dalam setiap kluster, yang dapat digunakan untuk pengambilan keputusan yang lebih baik dalam berbagai konteks aplikatif.

2.3.1.3 CRISP-DM



Gambar 2. 8 CRISP-DM [54]

CRISP-DM (*Cross-Industry Standard Process for Data Mining*) adalah sebuah metodologi yang digunakan untuk mengatur dan memandu proses analisis data [56]. Pada Gambar 2.8, metodologi ini terdiri dari serangkaian langkah-langkah yang terstruktur, mulai dari pemahaman bisnis hingga evaluasi hasil. Metodologi ini fleksibel dan dapat disesuaikan dengan kebutuhan proyek dan lingkungan kerja. Berikut adalah penjelasan umum dari setiap tahap dalam CRISP-DM [56]:

A. *Business Understanding*

Business understanding merupakan tahap awal dalam proses CRISP-DM dimana tujuan bisnis dari proyek analisis data dipahami dengan mendalam. Fokus utamanya adalah pada pemahaman terhadap masalah atau kesempatan bisnis yang ingin diselesaikan atau dimanfaatkan melalui analisis data. Tujuan-tujuan bisnis ini harus jelas dan terukur agar proses analisis dapat diarahkan dengan tepat sesuai dengan kebutuhan organisasi.

B. *Data Understanding*

Data Understanding berkaitan dengan memahami data yang akan digunakan dalam analisis. Hal ini melibatkan eksplorasi data untuk mengetahui karakteristiknya, pola-pola yang mungkin terdapat di dalamnya, serta hubungan antar variabel. Tujuan dari tahap ini adalah untuk mendapatkan pemahaman yang mendalam tentang data yang akan digunakan dalam proses analisis.

C. *Data Preparation*

Data Preparation adalah tahap dimana data dipersiapkan untuk analisis lebih lanjut. Proses ini melibatkan pembersihan data dari noise, duplikat, atau nilai yang hilang, serta transformasi data ke dalam format yang sesuai untuk pemodelan. Persiapan data yang baik sangat penting karena kualitas dan kebersihan data akan mempengaruhi hasil analisis secara keseluruhan.

D. *Modeling*

Modeling melibatkan pembangunan model prediktif atau deskriptif berdasarkan data yang telah dipersiapkan sebelumnya. Model-model ini digunakan untuk mengungkap pola, membuat prediksi, atau mengidentifikasi hubungan dalam data. Memilih teknik pemodelan yang sesuai dan membangun model dengan parameter yang optimal adalah kunci keberhasilan tahap ini.

E. *Evaluation*

Evaluation dilakukan untuk menilai kualitas dan kinerja model yang telah dibangun. Tahap ini melibatkan pengujian model terhadap data yang tidak digunakan dalam pembangunan, serta analisis hasil untuk memahami keefektifan dan keandalannya. Evaluasi yang baik diperlukan untuk memastikan bahwa model dapat digunakan dengan tepat dalam konteks bisnis yang sesungguhnya.

F. *Deployment*

Deployment adalah tahapan terakhir dari CRISP-DM dimana model yang telah dievaluasi diterapkan dalam lingkungan

operasional organisasi. Ini mencakup langkah-langkah untuk mempersiapkan *production*, mengimplementasikan model ke dalam sistem atau proses operasional yang relevan, serta pemantauan dan pemeliharaan model yang diimplementasikan. Penyebaran yang sukses adalah kunci untuk mendapatkan nilai bisnis dari analisis data yang dilakukan.

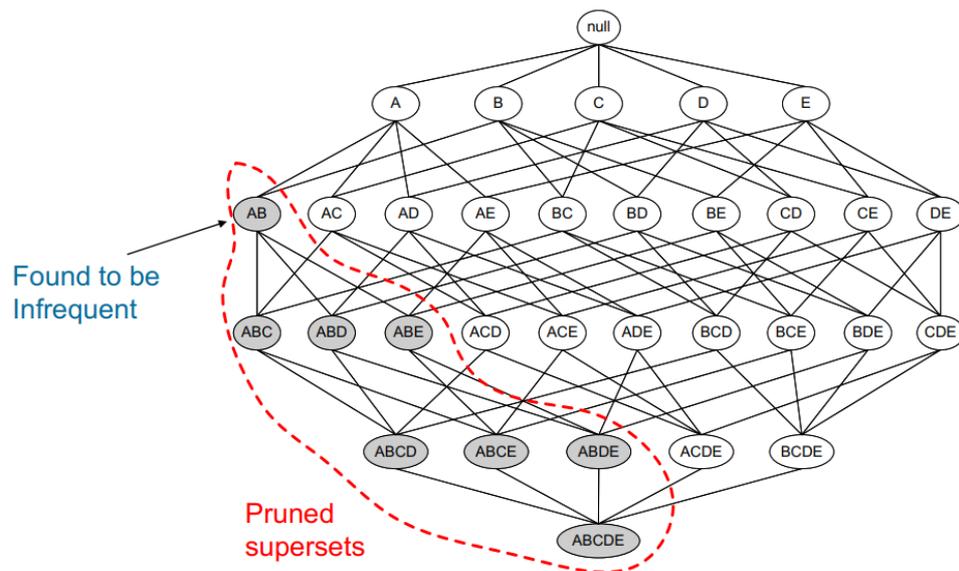
2.3.2 Algoritma

2.3.2.1 *Apriori*

Apriori adalah algoritma yang digunakan dalam *data mining* untuk menemukan pola asosiasi di antara sejumlah besar *item* yang sering muncul bersama dalam kumpulan data transaksi [57]. Secara khusus, *Apriori* digunakan untuk menemukan kumpulan *item* yang sering muncul bersama-sama dalam transaksi (*itemset*), serta aturan asosiasi di antara *item-item* tersebut. Algoritma ini berguna dalam menganalisis perilaku pembelian pelanggan, rekomendasi produk, dan penyesuaian stok.

Selain itu, *Apriori* menggunakan pendekatan yang disebut "*apriori principle*", yang menyatakan bahwa jika sebuah *itemset* jarang muncul dalam kumpulan data, maka subset dari *itemset* tersebut juga jarang muncul [58]. Dengan kata lain, jika sebuah aturan asosiasi tidak memenuhi persyaratan *support* yang ditentukan sebelumnya, maka aturan tersebut dianggap tidak relevan. Proses ini memungkinkan *Apriori* untuk secara efisien mengeksplorasi ruang pencarian yang besar dan menemukan pola-pola yang signifikan dalam data transaksi dengan kompleksitas waktu yang cukup rendah. Ini membuat *Apriori* menjadi salah satu algoritma yang paling umum digunakan dalam analisis pola asosiasi.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 2. 9 Apriori Breadth-first Search [59]

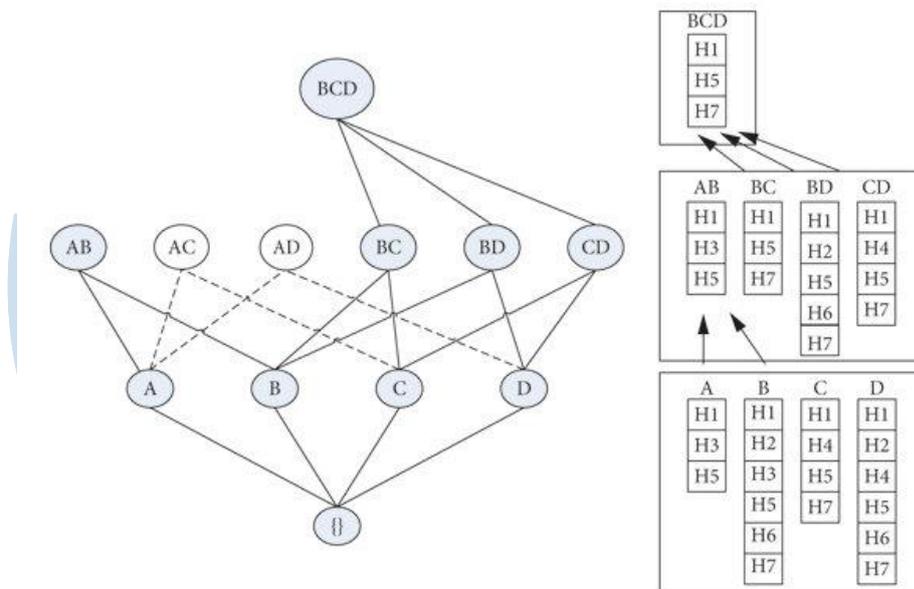
Pada Gambar 2.9, dijelaskan bahwa *Apriori* memanfaatkan pendekatan *breadth-first search* dan struktur data *hash tree* untuk mencari jumlah kandidat *itemset* secara akurat dan efisien. Dengan kata lain, *Apriori* menggunakan pendekatan *bottom-up* untuk menentukan *itemset* yang paling sering muncul.

Algoritma *Apriori* beroperasi dalam beberapa langkah, termasuk menciptakan kandidat *itemset*, menghitung *support* untuk setiap kandidat, memfilter kandidat berdasarkan *min-support* yang ditentukan sebelumnya, dan menghasilkan aturan asosiasi berdasarkan *confidence* yang ditentukan [60]. Dengan demikian, *Apriori* memberikan kerangka kerja yang kuat untuk mengidentifikasi pola pembelian yang signifikan dalam data transaksi.

2.3.2.2 ECLAT

ECLAT (*Equivalence Class Transformation*) adalah algoritma pencarian pola asosiasi yang digunakan dalam *data mining*. Algoritma ini bertujuan untuk menemukan kumpulan *item* yang sering muncul bersama-sama dalam kumpulan data transaksi. ECLAT berbeda dengan *Apriori* karena tidak menggunakan prinsip anti-monoton dalam mengeksplorasi

ruang pencarian. Sebaliknya, ECLAT menggunakan pendekatan berbasis penggabungan ekivalen yang lebih efisien dalam menemukan pola-pola yang signifikan.



Gambar 2. 10 ECLAT [61]

Pada Gambar 2.10 ditunjukkan proses pencarian dari algoritma ECLAT. Algoritma ECLAT menggunakan pendekatan *depth-first search* untuk menghitung *support* (seberapa sering munculnya sebuah itemset dalam basis data transaksi) dan menentukan keterkaitan antara itemset yang berbeda. Dalam gambar tersebut, dapat dilihat beberapa tahap proses ECLAT, yaitu pembentukan *itemsets*, pengulangan, dan *pruning*. Melalui proses ini, algoritma ECLAT secara efisien dapat menemukan semua itemset yang sering muncul tanpa perlu melakukan banyak pencarian secara melebar seperti yang dilakukan oleh algoritma *Apriori*.

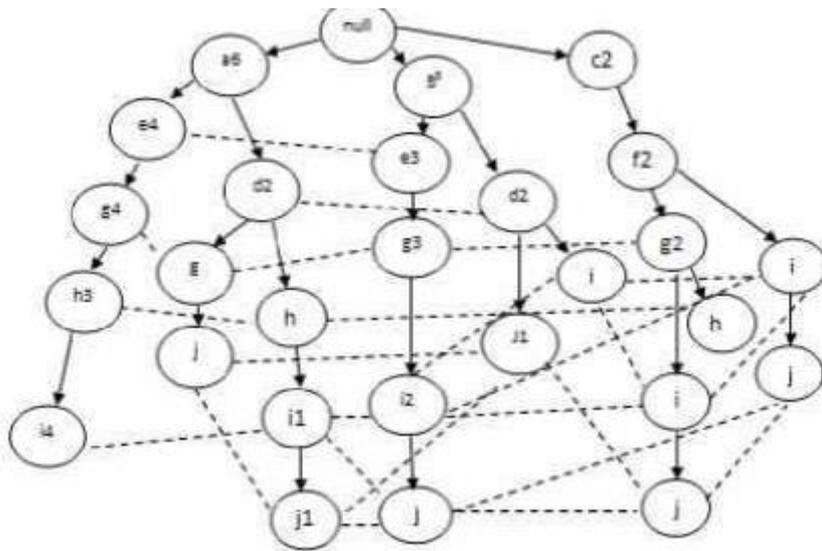
Rumus umum yang digunakan dalam ECLAT adalah untuk menghitung *support* dari sebuah *itemset* [62]. *Support* diukur sebagai proporsi dari jumlah transaksi yang mengandung *itemset* tersebut dari total jumlah transaksi.

Dalam konteks ECLAT, algoritma tersebut secara iteratif menciptakan kumpulan *item* yang berpotensi dan kemudian menghitung *support* untuk setiap *itemset*. *Itemset-itemset* yang memiliki *support* di atas batas minimum yang ditetapkan sebelumnya dianggap signifikan dan disimpan untuk langkah selanjutnya. Proses ini berlanjut hingga tidak ada lagi *itemset* yang dapat digabungkan atau sampai semua *itemset* relevan telah ditemukan. Dengan demikian, ECLAT memberikan cara yang efisien untuk menemukan pola asosiasi dalam data transaksi dengan menghindari pembangunan kandidat secara eksponensial seperti pada algoritma *Apriori*.

2.3.2.3 *FP-Growth*

FP-Growth (Frequent Pattern Growth) adalah algoritma populer dalam *data mining* yang digunakan untuk menemukan pola-pola yang sering muncul dalam kumpulan data transaksional atau transaksi [63]. Algoritma ini dikembangkan untuk mengatasi keterbatasan dari algoritma *Apriori*, terutama dalam hal kinerja ketika menghadapi *dataset* yang besar. *FP-Growth* bekerja dengan cara membangun struktur data yang disebut *FP-Tree (Frequent Pattern Tree)* dari *dataset* yang diberikan. *FP-Tree* menyimpan pola-pola yang sering muncul secara kompak dalam struktur pohon, yang kemudian digunakan untuk menemukan pola-pola yang sering muncul (*frequent patterns*).





Gambar 2. 11 *FP-Growth Complete Tree* [50]

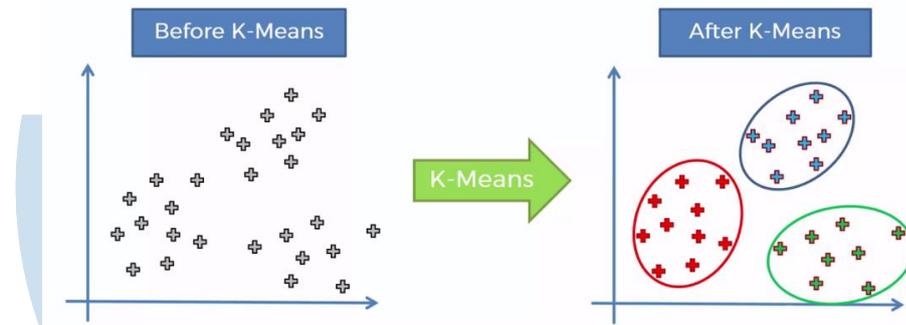
Gambar 2.11 menunjukkan *FP-Growth complete tree*. *FP-tree* adalah struktur pohon yang memiliki satu *root* berlabel *null* dengan *child* berupa *subtrees* dari awalan item dan sebuah tabel *header item* yang sering muncul. Setiap *node* dalam *subtree* ini memiliki tiga bidang, yaitu nama item, jumlah transaksi yang diwakili oleh jalur menuju *node* tersebut, dan *hubungan* *node* yang menghubungkan ke *node* berikutnya dengan nama item yang sama atau *null* jika tidak ada. Tabel *header item* sering muncul memiliki dua bidang, yaitu nama *item* dan kepala *hubungan node*.

Contoh skenario terbaik terjadi ketika semua transaksi memiliki kumpulan *item* yang sama, dimana ukuran *FP-tree* hanya akan menjadi satu cabang. Skenario terburuk terjadi ketika setiap transaksi memiliki kumpulan *item* unik, sehingga ruang yang diperlukan untuk menyimpan pohon lebih besar dari ruang yang digunakan untuk menyimpan set data asli karena *FP-tree* memerlukan ruang tambahan untuk menyimpan pointer antar *node* dan penghitung untuk setiap item.

Rumus utama dalam *FP-Growth* adalah untuk menghasilkan *frequent itemsets* dan menemukan pola-pola yang sering muncul dalam *dataset* [63]. Rumus ini menggunakan struktur *FP-Tree* yang efisien untuk mengatasi keterbatasan kinerja yang dialami oleh algoritma *Apriori*, terutama pada

dataset yang besar dan memiliki banyak *item*. Dengan memanfaatkan struktur *FP-Tree*, algoritma *FP-Growth* mampu menghasilkan *frequent itemsets* dengan cepat dan efisien, sehingga cocok untuk digunakan dalam analisis data transaksional dalam skenario bisnis yang nyata.

2.3.2.4 *K-Means*



Gambar 2. 12 Sebelum dan Sesudah *K-Means Clustering* [64]

K-Means merupakan salah satu algoritma *clustering* yang paling populer dalam analisis data. Seperti pada Gambar 2.12, tujuan utama dari *K-Means* adalah mengelompokkan data ke dalam sejumlah k kluster, dimana setiap titik data tergabung ke dalam kluster yang memiliki pusat atau *centroid* yang paling dekat dengannya. Algoritma ini beroperasi dengan mengiterasi antara dua langkah utama, yaitu inisialisasi *centroid* awal dan menetapkan setiap titik data ke kluster yang memiliki *centroid* terdekat, lalu menghitung kembali posisi *centroid* berdasarkan titik-titik yang baru ditugaskan tersebut. Proses ini terus diulang hingga tidak ada lagi perubahan signifikan dalam posisi *centroid* atau konvergensi tercapai.

Tahapan algoritma *K-Means clustering* secara umum adalah sebagai berikut [65]:

1. Pertama, jumlah kluster (k) dipilih secara acak untuk menjadi pusat dari masing-masing kluster.
2. Kemudian, jarak antara pusat kluster dan setiap titik data dihitung menggunakan metode *euclidean distance*, yang merupakan

pengukuran jarak geometris antara dua titik dalam ruang *euclidean*.

Teori *euclidean distance* ditunjukkan pada Rumus 2.1:

$$D(i, j) = \sqrt{(x_{1i} - x_{1j})^2 + (x_{2i} - x_{2j})^2 + \dots + (x_{ki} - x_{kj})^2}$$

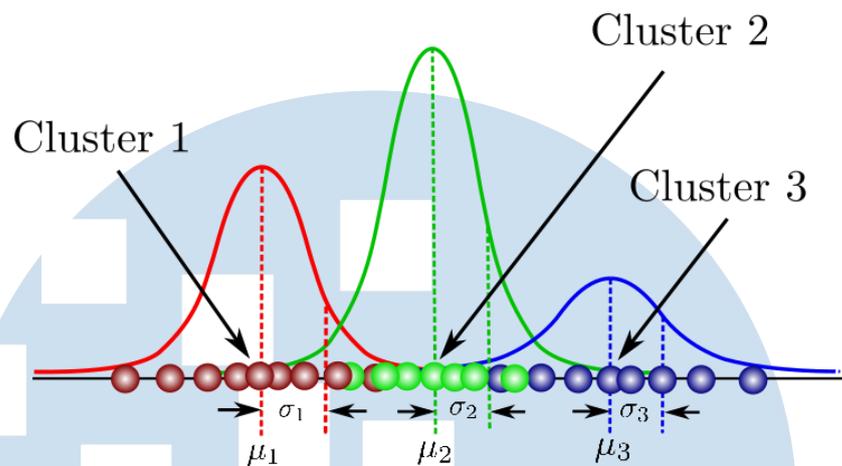
Rumus 2. 1 Rumus *Euclidean Distance*

3. Setelah itu, setiap titik data ditempatkan ke dalam kluster yang memiliki pusat kluster terdekat berdasarkan perhitungan jarak sebelumnya.
4. Pusat kluster baru kemudian dihitung berdasarkan rata-rata posisi semua titik data yang termasuk dalam kluster tersebut.
5. Proses penentuan pusat kluster dan penempatan data ke dalam kluster terus diulangi hingga tidak ada perubahan lagi dalam posisi pusat kluster, menunjukkan konvergensi algoritma *K-Means*.

2.3.2.5 *Gaussian Mixture Model (GMM)*

Gaussian Mixture Model (GMM) adalah pendekatan probabilistik untuk *clustering* data yang didasarkan pada asumsi bahwa data dihasilkan dari beberapa distribusi *Gaussian* yang berbeda. Dalam GMM, setiap kluster direpresentasikan oleh sebuah distribusi *Gaussian* dengan parameter sendiri, yaitu *mean* (rata-rata) dan *covariance* (kovariansi). Tujuan utama dari GMM adalah untuk memodelkan distribusi data secara tepat dengan memperkirakan parameter-parameter distribusi *Gaussian* yang terbaik untuk setiap kluster. Algoritma GMM melibatkan proses iteratif dimana pada setiap iterasi, model diperbarui dengan memperkirakan parameter distribusi *Gaussian* yang paling sesuai dengan data yang diamati.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 2. 13 GMM Clustering [66]

Pada Gambar 2.13, terlihat bahwa ada tiga fungsi *Gaussian*, sehingga $C = 3$. Setiap *Gaussian* menjelaskan data yang terkandung dalam masing-masing dari tiga kluster yang ada. Koefisien campuran adalah probabilitas. Total dari semua koefisien campuran harus sama dengan 1, yang menunjukkan bahwa keseluruhan distribusi adalah gabungan dari ketiga fungsi *Gaussian* tersebut. Setiap koefisien merepresentasikan proporsi data yang dijelaskan oleh masing-masing *Gaussian* dalam model.

Tahapan algoritma GMM *clustering* secara umum adalah sebagai berikut [65]:

1. Menginisialisasi secara acak nilai-nilai μ_k , σ_k , dan π_k untuk setiap kluster. Di sini, μ merujuk pada *mean*, σ pada *varians*, dan π pada koefisien campuran, sementara k merupakan angka yang menunjuk ke suatu mixture dalam distribusi *gaussian*, dan k ekuivalen sebagai nilai yang menunjuk ke suatu kluster.
2. Melakukan evaluasi dilakukan terhadap hasil *log-likelihood* dengan menggunakan parameter μ_k , σ_k , dan π_k . C adalah kluster, p adalah probabilitas, i adalah angka yang menunjuk ke suatu distribusi gaussian dan x adalah distribusi *gaussian*. Misalkan cluster C_k diwakili oleh distribusi *gaussian* (μ_k , σ_k), maka probabilitas x_i apa pun milik cluster C_k dihitung dengan persamaan. Perhitungannya ditunjukkan pada Rumus 2.2:

$$\rho(C_k|x_i) = \frac{\rho(x_i|C_k) * \rho(C_k)}{\rho(x_i)}$$

Rumus 2. 2 Rumus Menghitung Persamaan Probabilitas Terhadap Kluster

3. Perhitungan dilanjutkan dengan menghitung nilai kemungkinan (*likelihood*). Perhitungannya ditunjukkan pada Rumus 2.3:

$$\rho(x_i|C_k) = \frac{1}{\sqrt{2\pi_k\sigma}} * \exp\left(\frac{-(x_i - \mu_k)^2}{2\sigma^2}\right)$$

Rumus 2. 3 Rumus Menghitung Nilai Kemungkinan (*Likelihood*)

Nilai evaluasi (*evidence*) ditunjukkan pada Rumus 2.4:

$$\rho(x_i) = \sum_k \rho(x_i|C_k) * \rho(C_k)$$

Rumus 2. 4 Rumus Menghitung Nilai Evaluasi (*Evidence*)

4. Setelah itu, nilai-nilai μ_k , σ_k , dan $\rho(C_k)$ diubah melalui proses perhitungan Rumus 2.5:

$$\mu_k = \frac{\sum_k \rho(C_k|x_i) * x_i}{\sum_i \rho(C_k|x_i)}$$

$$\sigma_k = \frac{\sum_i \rho(C_k|x_i) * (x_i - \mu_k)^2}{\sum_i \rho(C_k|x_i)}$$

$$\pi_k = \frac{\sum_i \rho(C_k|x_i)}{n}$$

Rumus 2. 5 Rumus Transformasi Nilai μ_k , σ_k , dan $\rho(C_k)$

5. Langkah-langkah evaluasi dan penyesuaian parameter dilakukan secara berulang sampai kriteria konvergensi terpenuhi. Kriteria konvergensi biasanya ditentukan oleh nilai ambang tertentu untuk perubahan *mean* dan *varians* dalam setiap iterasi berurutan.

2.3.2.6 Label Encoder

Encoder	Maxwell	ISBSG	Nasa93	USP05
OH	93	58	85	302
LE	24	11	22	14
TE	24	11	22	14
CE	24	11	22	14

Gambar 2. 14 Label Encoded Data [67]

Label encoder merupakan teknik pemrosesan data yang digunakan untuk mengubah nilai kategorik menjadi numerik. *Label encoder* biasanya digunakan untuk memproses data kategorik yang ingin diolah sebagai numerik unik yang dimulai dari 0 atau 1 dan terus bertambah untuk setiap kategori [67]. *Label encoder* cocok digunakan untuk menangani masalah beberapa algoritma memerlukan *input* numerik untuk memproses data. *Label encoder* merubah data kategorik, seperti pria/wanita, lokasi, warna, menjadi nilai numerik. Pria dan wanita dapat di-*encode* menjadi 0 dan 1. Pada Gambar 2.14, dilakukan *encoding* pada data kategorikal menjadi numerikal menggunakan beberapa *encoder* yang berbeda.

2.3.2.7 Standard Scaler

Standard Scaler merupakan salah satu teknik persiapan data yang digunakan dalam analisis data dan pembelajaran mesin untuk menstandarisasi fitur-fitur numerik dalam dataset. Tujuan dari *standard scaler* adalah untuk mengubah distribusi nilai dari setiap fitur sehingga memiliki *mean* 0 dan *standard deviation* 1. Proses ini dilakukan dengan mengurangi rata-rata dari setiap fitur dan membaginya dengan deviasi standarnya [68]. Hasil dari transformasi dari proses *standard scaler* adalah *z-score*. Rumus perhitungan *z-score* dapat dilihat pada Rumus 2.6:

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

Rumus 2. 6 Rumus Perhitungan Z-Score

Variabel x merupakan nilai asli dari data, μ adalah nilai *mean*, dan σ adalah nilai *standard deviation*. Penerapan *standard scaler* sangat berguna ketika fitur-fitur dalam *dataset* memiliki skala yang berbeda-beda. Misalnya, jika satu fitur memiliki rentang nilai yang sangat besar dibandingkan dengan fitur lainnya, hal ini dapat menjadi anomali dalam proses pembelajaran mesin. Dengan menggunakan *standard scaler*, perbedaan skala ini dinormalisasi sehingga model dapat memberikan bobot yang sama pada setiap fitur selama proses pembelajaran.

2.3.2.8 *Principal Component Analysis (PCA)*

Principal Component Analysis (PCA) merupakan teknik untuk mengurangi dimensi pada data. Tujuan utama penerapan PCA adalah untuk mengurangi jumlah *features* dalam *dataset* yang kompleks. Secara umum, PCA mencari kombinasi linear dari fitur-fitur yang ada untuk menghasilkan komponen baru yang disebut *principal components* [69]. Penerapan PCA sangat cocok untuk menghilangkan ketergantungan yang berlebihan antar fitur-fitur dan mengidentifikasi fitur penting. PCA juga sangat baik dalam mengurangi potensi *overfitting*.

2.3.3 Metode Evaluasi

2.3.3.1 *Support and Confidence*

Model evaluasi *support* dan *confidence* digunakan dalam analisis *association rule mining* untuk mengevaluasi seberapa kuatnya hubungan antara *item-item* dalam *dataset* transaksional [70]. *Support* mengukur seberapa sering suatu aturan asosiasi (*association rule*) muncul dalam *dataset*, sementara *confidence* mengukur seberapa sering aturan asosiasi tersebut terbukti benar. Perhitungan nilai *support* ditunjukkan pada Rumus 2.7:

$$\text{Support}(A \rightarrow B) = \frac{\text{Number of transaction A and B}}{\text{Total number of transaction}}$$

Rumus 2.7 Rumus Menghitung *Support*

Support (S) mengukur frekuensi kemunculan suatu aturan asosiasi dalam *dataset* transaksional. *Support* berkisar antara 0 hingga 1, dimana nilai 1 menunjukkan bahwa setiap transaksi dalam *dataset* mengandung kedua *item*, sementara nilai yang lebih rendah menunjukkan tingkat kemunculan yang lebih rendah. Perhitungan nilai *confidence* ditunjukkan pada Rumus 2.8:

$$Confidence(A \rightarrow B) = \frac{Support(A \cup B)}{Support(A)}$$

Rumus 2. 8 Rumus Menghitung *Confidence*

Confidence (C) mengukur seberapa sering *item* Y muncul dalam transaksi yang juga mengandung *item* X. Ini menunjukkan seberapa kuatnya hubungan antara X dan Y [70]. *Confidence* juga memiliki rentang nilai antara 0 hingga 1. Nilai 1 menunjukkan bahwa setiap kali *item* X muncul dalam transaksi, *item* Y juga muncul dalam transaksi tersebut.

Sebagai contoh, terdapat aturan asosiasi {Pulsa} → {Data Paket} dengan *support* 0.4 dan *confidence* 0.8. Ini berarti bahwa 40% dari total transaksi dalam *dataset* mengandung *item* Pulsa dan Data Paket. Dari transaksi yang mengandung Pulsa, 80% dari transaksi tersebut juga mengandung Data Paket.

2.3.3.2 *Lift*

Lift adalah salah satu metrik penting dalam analisis *association rule mining* yang mengukur seberapa kuatnya hubungan antara dua *item* dalam sebuah aturan asosiasi [71]. *Lift* mengindikasikan jumlah kemungkinan terjadinya hubungan antara dua *item* tersebut dibandingkan dengan jika hubungan tersebut terjadi secara acak. Semakin tinggi nilai *Lift*, semakin signifikan hubungan antara kedua *item* tersebut. Berikut adalah penjelasan teknis beserta rumusnya.

Lift adalah ukuran dari seberapa banyak *item* B dipengaruhi oleh keberadaan *item* A dalam suatu transaksi. *Lift* menggambarkan "*lift*" atau

peningkatan dalam kemunculan item B ketika *item* A ada[71]. Perhitungan nilai *lift* ditunjukkan pada Rumus 2.9:

$$Lift(A \rightarrow B) = \frac{Confidence(A \rightarrow B)}{Support(A)}$$

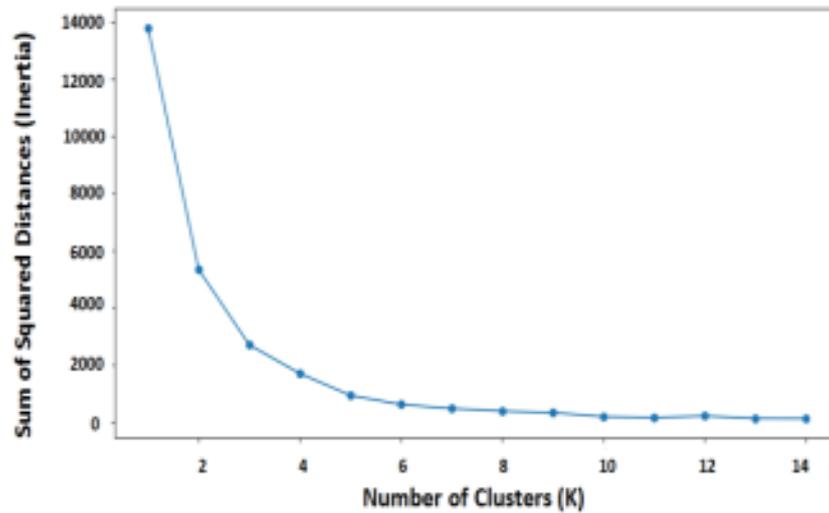
Rumus 2. 9 Rumus Menghitung Lift

Lift dapat diinterpretasikan sebagai rasio antara *confidence* aturan asosiasi $A \rightarrow B$ dan *support item* B. Jika $Lift > 1$, itu menunjukkan bahwa keberadaan *item* X dalam transaksi meningkatkan kemungkinan munculnya *item* B. Jika $Lift = 1$, itu menunjukkan bahwa keberadaan *item* A dalam transaksi tidak memiliki pengaruh terhadap *item* B. Jika $Lift < 1$, itu menunjukkan bahwa keberadaan *item* A dalam transaksi mengurangi kemungkinan munculnya *item* B.

Sebagai contoh, terdapat aturan asosiasi {Pulsa} \rightarrow {Data Paket} dengan *confidence* 0.8 dan *support* 0.4. *Lift* dari aturan asosiasi tersebut adalah $\frac{0.8}{0.4} = 2$. Ini menunjukkan bahwa keberadaan *item* Pulsa dalam transaksi meningkatkan kemungkinan munculnya *item* Data Paket sebanyak 2 kali lipat.

2.3.3.3 Elbow Method

Elbow method merupakan salah satu teknik evaluasi model *clustering* yang umum digunakan untuk menentukan jumlah kluster optimal. Metode *elbow* memanfaatkan pengukuran *inertia* untuk mengevaluasi kualitas kluster [72]. *Inertia* dalam *elbow* dikenal juga dengan sebutan *within-cluster sum of squares* (WCSS). *Inertia* menilai dan mengukur seberapa jauh titik-titik sebuah kluster menyebar dari pusatnya. Dalam visualisasi *elbow method*, nilai *inertia* ditampilkan sebagai fungsi dari jumlah kluster yang digunakan dalam analisis.

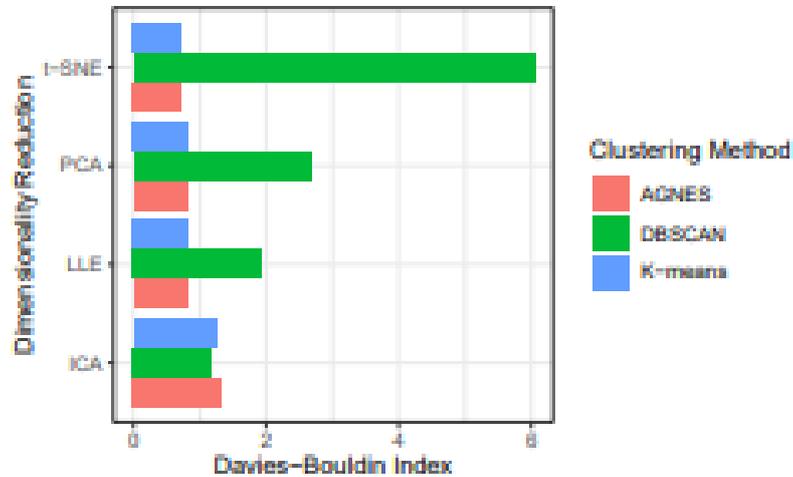


Gambar 2. 15 *Elbow Method Inertia* [72]

Cara mengetahui jumlah kluster optimal menggunakan *elbow method* adalah dengan memperhatikan pergerakan nilai *inertia*. Pada Gambar 2.15, semakin besar jumlah kluster, maka nilai *inertia* akan semakin menurun. Penurunan nilai *inertia* akan melambat pada suatu titik tertentu dan kurva akan membentuk grafik seperti siku. Titik perlambatan *inertia* disebut sebagai *elbow point*, dimana penambahan kluster setelah titik tersebut tidak menunjukkan penurunan *inertia* yang signifikan. Nilai *elbow point* ini yang dijadikan jumlah kluster optimal dalam menentukan jumlah kluster pada algoritma *clustering*.

UMN
 UNIVERSITAS
 MULTIMEDIA
 NUSANTARA

2.3.3.4 Davies-Bouldin Index (DBI)



Gambar 2. 16 Davies-Bould Index [73]

Davies-Bouldin Index (DBI) adalah salah satu metrik evaluasi yang digunakan untuk mengevaluasi kualitas partisi kluster yang dihasilkan oleh algoritma *clustering*. Tujuan utama DBI adalah untuk mengukur seberapa baik kluster terpisah satu sama lain dan seberapa mirip karakteristik setiap masing-masing kluster. Dalam konteks umum, DBI sering digunakan sebagai bagian dari proses pemilihan model atau *parameter tuning*. Data yang diberikan dapat memiliki karakteristik tertentu, seperti kepadatan atau dispersi yang berbeda di antara kluster-klasternya. Gambar 2.16 menunjukkan bentuk *plot* DBI yang dapat membantu pemilihan jumlah kluster optimal untuk mewakili struktur data dengan baik. Semakin rendah nilai *index*, semakin bagus kluster tersebut.

$$DBI = \frac{1}{n} \sum_{i=1}^n \max_{j \neq i} \left(\frac{\sigma_i + \sigma_j}{d(c_i, c_j)} \right)$$

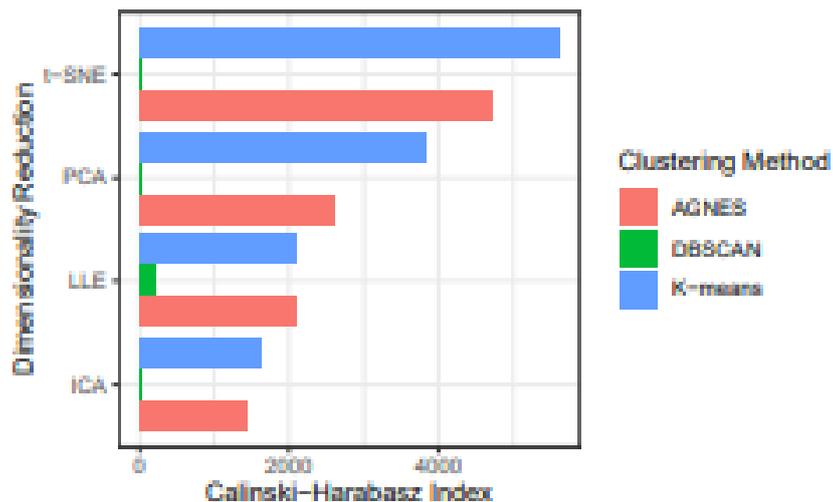
Rumus 2. 10 Rumus Perhitungan DBI

Rumus perhitungan nilai DBI dapat dilihat pada Rumus 2.10 [73]. n adalah jumlah kluster, σ_i adalah jarak rata-rata setiap titik dalam kluster i ke pusatnya c_i , dan $d(c_i, c_j)$ adalah jarak antara pusat c_i dan c_j dari kluster i dan j . Ini menjelaskan bagaimana menghitung *Calinski-Harabasz Index* (CHI),

dimana CHI adalah rasio antara *varians* antar-kluster dan *varians* dalam kluster. *Varians* dalam kluster dihitung dengan mempertimbangkan jarak rata-rata setiap titik dalam kluster ke pusatnya, sementara *varians* antar-kluster dihitung dengan mempertimbangkan jarak antara pusat-pusat kluster.

Cara kerja DBI adalah dengan mengukur rasio antara dua hal, yakni jarak antara pusat kluster dengan ukuran kluster sendiri (kekompakan) dan jarak antara pusat kluster dengan kluster lainnya (pemisahan). Nilai DBI yang lebih rendah menunjukkan partisi kluster yang lebih baik, dimana kluster-kluster terpisah dengan baik dan mirip secara internal.

2.3.3.5 Callinski-Harabasz Index (CHI)



Gambar 2. 17 Callinski-Harabasz Index [73]

Callinski-Harabasz Index (CHI) adalah metrik evaluasi yang digunakan untuk mengukur kualitas partisi kluster yang dihasilkan oleh algoritma *clustering*. Tujuan utama pengimplementasian CHI adalah untuk memberikan penilaian terhadap seberapa baik kluster terpisah satu sama lain dan seberapa kesamaan mereka di dalam kluster masing-masing. Gambar 2.17 menunjukkan contoh hasil *plot* CHI untuk menentukan jumlah kluster optimal. Semakin tinggi nilai *index*, semakin bagus kluster tersebut. Secara umum, CHI merupakan metrik evaluasi yang intuitif dan sering digunakan

dalam praktik *data clustering*. CHI dapat menjadi alat yang berguna dalam mengevaluasi partisi kluster dan membantu dalam pemilihan model *clustering* yang optimal.

Cara kerja CHI mirip dengan *Davies-Bouldin Index* (DBI), dimana CHI juga menggunakan rasio antara keterpisahan dalam kluster dengan dispersi antar-kluster. Namun, CHI menggunakan pendekatan dan perhitungan yang berbeda untuk mencapai tujuan ini.

$$CHI = \frac{B}{W} \times \frac{N - k}{k - 1}$$

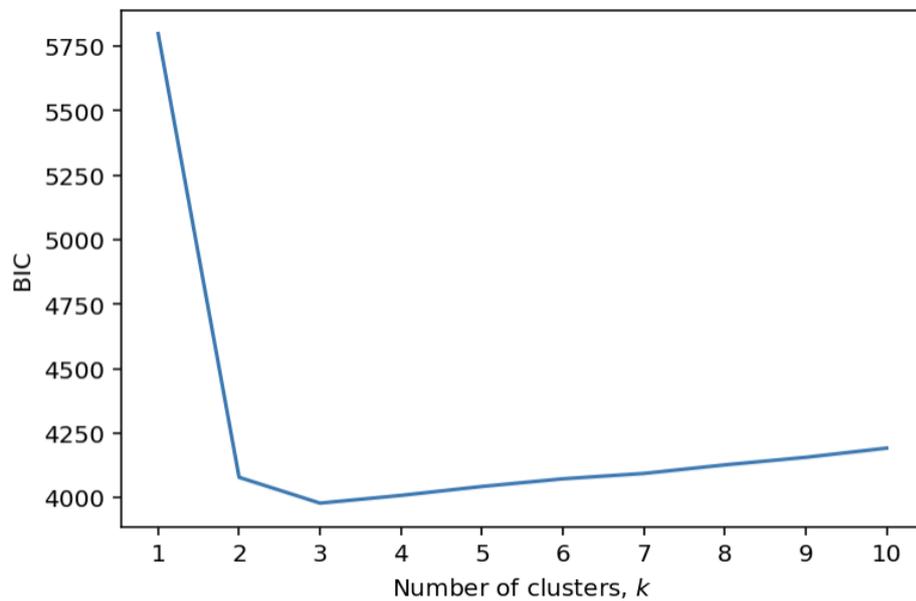
Rumus 2. 11 Rumus Perhitungan CHI

Rumus perhitungan nilai DBI dapat dilihat pada Rumus 2.11[73]. *B* adalah *varians* antar-kluster, yaitu *varians* antara pusat-pusat kluster, *W* adalah *varians* dalam kluster, *N* adalah jumlah total titik data, dan *k* adalah jumlah kluster.

Dalam menghitung nilai CHI, dilakukan perhitungan *varians* dalam kluster (*within-cluster variance*) dan *varians* antar kluster (*between-cluster variance*). Kemudian, CHI dihitung sebagai rasio antara *varians* antar kluster dengan *varians* dalam kluster, dikalikan dengan faktor normalisasi. Nilai CHI yang lebih tinggi menunjukkan partisi kluster yang lebih baik, dimana kluster-kluster terpisah dengan baik dan kompak secara internal.

2.3.3.6 Akaike Information Criterion (AIC)

Akaike Information Criterion (AIC) adalah metode evaluasi yang digunakan dalam statistika dan pembelajaran mesin untuk memilih model yang paling baik dari beberapa model alternatif. Metode ini dikembangkan oleh Hirotugu Akaike. Tujuan utama penerapan AIC adalah untuk memperkirakan kualitas relatif dari sebuah model dengan memperhitungkan keseimbangan antara kemampuan prediktif model dan kompleksitasnya.



Gambar 2. 18 *Akaike Information Criterion* [55]

Gambar 2.18 menunjukkan *plot* AIC untuk menentukan jumlah kluster optimal. Cara kerja AIC adalah dengan mengevaluasi seberapa baik model dapat menjelaskan data yang dianalisis dengan mempertimbangkan jumlah parameter yang digunakan dalam model-model tersebut. AIC mengukur kualitas model dengan mencari keseimbangan antara kebaikan *fit* (*how well the model fits the data*) dan *parsimony* (*how simple the model is*). Model yang baik harus mampu memberikan penjelasan yang baik terhadap data, sementara juga memiliki kompleksitas yang rendah agar tidak terjadi *overfitting* pada data [55].

$$AIC = 2L_{ML} + 2k$$

Rumus 2. 12 Rumus Perhitungan AIC

Dalam rumus perhitungan AIC pada Rumus 2.12. L_{ML} merupakan *log likelihood* yang diperoleh setelah menerapkan model dengan *maximum likelihood*. k adalah jumlah parameter. Untuk jumlah sampel yang lebih kecil, perhitungan dapat disesuaikan dengan menggunakan Rumus 2.13.

$$AIC_c = 2L_{ML} + \frac{2k(k+1)}{n-k-1}$$

Rumus 2. 13 Rumus Perhitungan AIC untuk Sampel Kecil

Jika kesalahan dalam *fitting* secara normal didistribusikan, maka $-L_{ML} = x^2 ML$, dimana $x^2 ML$ adalah *standard x^2* yang dievaluasi menggunakan estimasi *maximum likelihood parameter*.

2.3.3.7 Bayesian Information Criterion (BIC)

Bayesian Information Criterion (BIC) adalah metode evaluasi yang digunakan dalam statistika untuk memilih model yang paling baik dari beberapa model alternatif. Metode ini dikembangkan berdasarkan pendekatan *bayesian* dan sering digunakan dalam pemodelan statistik, termasuk dalam pemilihan model dalam analisis regresi dan *clustering*.

Tujuan utama dari BIC adalah untuk memperkirakan kualitas relatif dari sebuah model dengan memperhitungkan keseimbangan antara kecocokan *fit model* terhadap data dan kompleksitasnya. Metode BIC memberikan pemberat lebih besar terhadap kompleksitas model daripada AIC. Hal ini dilakukan untuk mencegah pemilihan model yang terlalu kompleks yang mungkin *overfitting* pada data [55].

$$BIC = -2L_{ML} + k \ln(n)$$

Rumus 2. 14 Rumus Perhitungan BIC

Pada Rumus 2.14, perhitungan BIC memiliki variabel L yang merupakan *maximum log likelihood* dan k merupakan jumlah parameter dari model. Salah satu keuntungan utama dari BIC adalah bahwa metode ini secara langsung memberikan pemberat terhadap kompleksitas model, yang bermanfaat terutama ketika memilih di antara model yang berbeda.

2.3.3.8 *Testing*

Testing adalah proses penting dalam pengembangan perangkat lunak yang bertujuan untuk memastikan kualitas dan kehandalan produk atau layanan. Melalui pengujian, kesalahan atau *bug* dapat diidentifikasi dan diperbaiki sebelum produk diluncurkan ke pasar. Hal ini memastikan bahwa produk yang disajikan kepada pengguna memiliki performa yang optimal, fungsionalitas yang diharapkan, serta memenuhi standar keamanan dan kinerja yang ditetapkan.

Software Integration Testing (SIT) adalah tahap penting dalam siklus pengujian perangkat lunak dimana komponen-komponen yang telah dikembangkan diuji secara bersama-sama untuk memastikan bahwa mereka berinteraksi dengan baik. Fokus utama dari SIT adalah untuk mengidentifikasi masalah integrasi antara komponen-komponen tersebut secara eksternal, sehingga memastikan bahwa sistem berfungsi secara kohesif dan sesuai dengan spesifikasi yang diharapkan [74]. Dalam SIT, pengujian dilakukan di tingkat unit atau modul yang lebih kecil dengan tujuan memvalidasi antarmuka antara komponen-komponen tersebut.

User Acceptance Testing (UAT) merupakan tahap pengujian yang melibatkan pengguna akhir atau pihak yang mewakili pengguna akhir. Tujuan dari UAT adalah untuk memastikan bahwa sistem atau produk yang dikembangkan sesuai dengan kebutuhan dan ekspektasi pengguna. Melalui UAT, pengguna dapat menguji fungsionalitas, kegunaan, dan kinerja produk dalam lingkungan yang mendekati situasi nyata penggunaan. Hasil dari UAT memungkinkan tim pengembang untuk mengevaluasi apakah produk telah memenuhi standar yang diharapkan dan siap untuk dirilis ke pengguna akhir.

2.4 Software dan Tools Penelitian

2.4.1 Python

Python adalah bahasa pemrograman tingkat tinggi yang sering digunakan dalam berbagai bidang, termasuk pengembangan perangkat lunak, analisis data, kecerdasan buatan, pengembangan web, dan banyak lagi. *Python* didesain dengan sintaksis yang mudah dipahami dan bersifat mudah dibaca, sehingga cocok untuk pemula dan profesional dalam pengembangan perangkat lunak.

Kegunaan *Python* sangat luas dan beragam. Dalam konteks skripsi ini, *Python* dapat digunakan untuk berbagai keperluan, terutama dalam analisis data. Dengan berbagai pustaka dan alat yang tersedia, *Python* memungkinkan para peneliti untuk melakukan analisis data yang kompleks, termasuk pemrosesan data, visualisasi, dan pemodelan [75]. Selain itu, *Python* juga dapat digunakan untuk mengimplementasikan berbagai algoritma machine learning dan *data mining*, termasuk algoritma *association rule mining* yang relevan dalam penelitian ini.

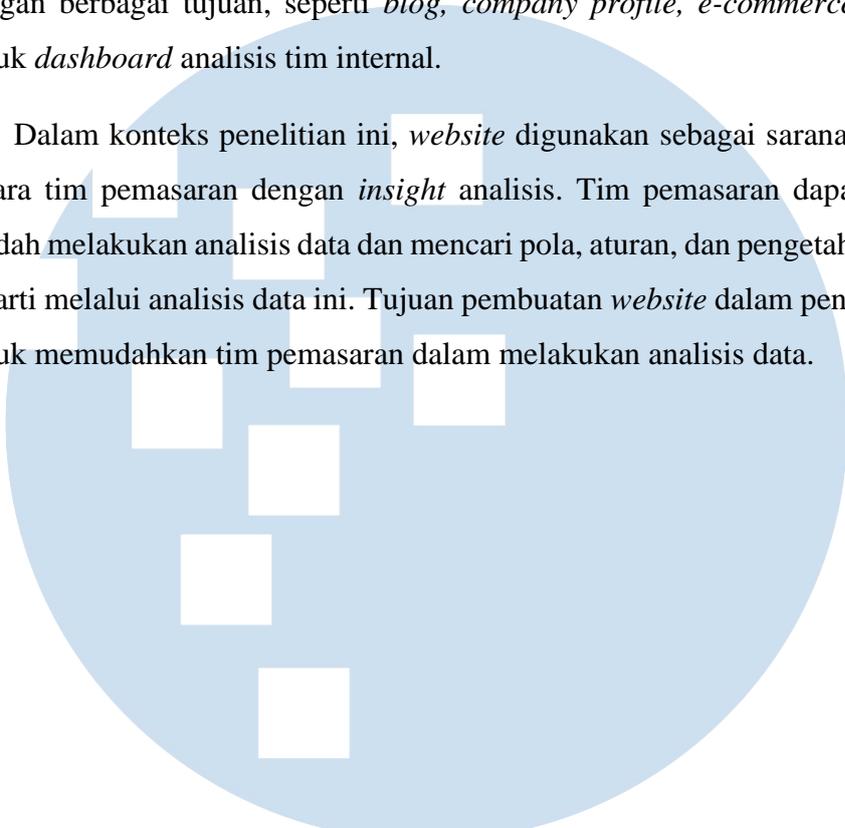
Python sangat cocok untuk skripsi ini karena kelebihan-kelebihan yang dimilikinya. Pertama, *Python* memiliki komunitas yang besar dan aktif, sehingga mudah untuk menemukan bantuan dan sumber daya yang diperlukan. Kedua, *Python* memiliki berbagai pustaka dan kerangka kerja yang kuat untuk analisis data, seperti *pandas*, *numpy*, dan *scikit-learn*, yang mempercepat proses pengembangan dan memungkinkan peneliti untuk fokus pada inti dari penelitian mereka. Ketiga, *Python* bersifat lintas *platform*, artinya dapat dijalankan di berbagai sistem operasi, sehingga memudahkan kolaborasi peneliti yang menggunakan *platform* yang berbeda. Dengan demikian, *Python* merupakan pilihan yang sangat baik untuk skripsi ini karena kemudahannya dalam analisis data dan implementasi algoritma yang diperlukan.

2.4.2 Website

Website merupakan kumpulan halaman web yang terkait dan dapat diakses oleh pengguna melalui internet. Setiap halaman *web* memiliki alamat unik yang disebut URL (*Uniform Resource Locator*). Setiap *web* dapat berisi kumpulan

teks, gambar, CTA (*call-to-action*), *video*, dan lain lain. *Website* dapat dibuat dengan berbagai tujuan, seperti *blog*, *company profile*, *e-commerce*, bahkan untuk *dashboard* analisis tim internal.

Dalam konteks penelitian ini, *website* digunakan sebagai sarana interaksi antara tim pemasaran dengan *insight* analisis. Tim pemasaran dapat dengan mudah melakukan analisis data dan mencari pola, aturan, dan pengetahuan yang berarti melalui analisis data ini. Tujuan pembuatan *website* dalam penelitian ini untuk memudahkan tim pemasaran dalam melakukan analisis data.



UMMN

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A