

BAB II

LANDASAN TEORI

2.1. Penelitian Terkait

Dalam penyusunan penelitian ini, penulis menjadikan beberapa penelitian terdahulu yang relevan sebagai referensi dan landasan teoretis. Tinjauan ini bertujuan untuk memposisikan penelitian saat ini di antara studi-studi yang sudah ada serta mengidentifikasi kontribusi yang dapat diberikan. Adapun beberapa penelitian terkait yang relevan adalah sebagai berikut:

Penelitian oleh (Wulansari & Chulkamdi, 2022) dengan judul "Penerapan *Algoritma Apriori* untuk Menentukan Tata Letak Menempatkan Barang Dagangan Toko Mekar Sari di Blitar" menampilkan ranking produk dengan pengurutan jumlah transaksi terbanyak menuju ke jumlah transaksi terkecil. Hasil penelitian menunjukkan bahwa *algoritma Apriori* yang diterapkan melalui *Data mining* dapat menentukan penataan barang untuk menentukan barang terlaris yang dapat membantu mengembangkan aturan asosiasi dengan analisis pola frekuensi tinggi (*frequent pattern mining*). Tujuan dari penelitian ini adalah untuk menentukan tata letak suatu produk yang dapat mendukung peningkatan penjualan di toko kelontong Mekar Sari.

Selanjutnya, penelitian yang dilakukan oleh (Kumalasari et al., 2020) berjudul "Implementasi *Algoritma Apriori* untuk Menemukan Pola Pembelian Konsumen pada Perusahaan Retail" menunjukkan bahwa *algoritma Apriori* sesuai diterapkan untuk menemukan pola pembelian

konsumen pada data transaksi penjualan. Data transaksi penjualan dapat dimanfaatkan oleh pihak manajemen toko retail untuk menganalisis kebiasaan belanja konsumen terkait barang-barang yang sering dibeli secara bersamaan dengan menerapkan analisis keranjang belanja (*market basket analysis*) menggunakan teknik *Data mining*. Aturan asosiasi yang dihasilkan dapat digunakan sebagai pendukung dalam pengambilan keputusan oleh manajemen toko.

Penelitian relevan lainnya oleh (Saenafulloh & Stefani, 2023) menunjukkan bahwa peningkatan penjualan toko kelontong dapat dicapai dengan memperhatikan persediaan stok barang. Selain itu, rekomendasi produk kepada pelanggan dapat diberikan berdasarkan analisis data *support* kombinasi dua item atau aturan asosiasi pada hasil akhir. Penerapan layout item yang strategis juga menjadi faktor penting dalam menarik perhatian pelanggan. Item yang nilai *confidence* yang tinggi dan lift ratio diatas satu, dapat ditempatkan di posisi yang strategis. Item-item yang memiliki keterkaitan erat dalam pembelian disarankan ditempatkan berdekatan untuk mempengaruhi keputusan pembelian pelanggan. Dengan menggunakan metode Apriori penjual dapat menemukan pola pembelian pelanggan dan hubungan antarproduk, sehingga membantu dalam pengambilan keputusan terkait pengelolaan stok dan penataan produk di toko.

Kemudian, penelitian oleh (Sari & Wahyuni, 2024) menunjukkan bahwa Penelitian ini menggunakan metode *Data mining* dengan *algoritma Apriori* untuk menganalisis data transaksi penjualan di Koperasi ABC

Tama. *Algoritma Apriori* bekerja dengan menggunakan prinsip “*downward-closure*” atau “*anti-monotone*,” di mana *Itemset* yang tidak memenuhi ambang frekuensi minimum akan dieliminasi beserta semua superset-nya. Dalam penelitian ini, penulis menggunakan data transaksi penjualan dari Koperasi ABC Tama sebagai bahan utama untuk analisis. Setelah data transaksi penjualan terkumpul, data tersebut kemudian diubah ke dalam format Microsoft Excel agar lebih mudah diproses. Selanjutnya, data yang telah diubah ini diimpor ke dalam aplikasi RapidMiner untuk dianalisis menggunakan *Algoritma Apriori* untuk menemukan aturan asosiasi antara barang-barang yang sering dibeli bersama.

Terakhir, penelitian dari (Wijayanti et al., 2022) menunjukkan bahwa metode yang digunakan pada pengembangan strategi tata letak barang pada Toko “Ina 2” menggunakan *algoritma Apriori* ini adalah SDLC (*System Development Life Cycle*) dengan pendekatan model waterfall. Pada pembuatan pengembangan strategi tata letak barang pada Toko “Ina 2” menggunakan *algoritma Apriori* membutuhkan beberapa desain yaitu desain Diagram Konteks, DFD (*Data Flow Diagram*) yang menggunakan notasi Yourdan dan De Marco, Tampilan Antarmuka Sistem, dan ERD. Kemudian, mengimplementasikan source code bahasa pemrograman PHP. Pengujian sistem menggunakan UAT (*User Acceptance Test*) untuk mengetahui kelemahan dari sistem, apakah data yang dihasilkan sesuai dengan data yang dimasukkan setelah dieksekusi atau tidak. Selain itu juga untuk meminimalisir kesalahan pada aplikasi sebelum digunakan pengguna.

Dibandingkan dengan penelitian-penelitian sebelumnya, penelitian ini tidak hanya menerapkan *algoritma Apriori* untuk menemukan asosiasi antarproduk, tetapi juga menghubungkan hasil analisis langsung dengan perancangan ulang tata letak barang secara implementatif di toko kelontong sehingga menghasilkan rekomendasi yang lebih aplikatif dan dapat diukur efektivitasnya. Dengan demikian, penelitian ini tidak hanya mereplikasi penerapan *algoritma Apriori*, tetapi juga mengisi celah penelitian terdahulu yang belum banyak menghubungkan analisis Apriori dengan perubahan nyata pada penataan produk di toko kelontong.

Originalitas penelitian ini terletak pada pendekatan integratif yang tidak hanya menganalisis pola pembelian menggunakan *algoritma Apriori*, tetapi juga menghubungkan hasil analisis tersebut secara langsung dengan perancangan strategi penataan ulang barang yang dapat diimplementasikan pada kondisi nyata toko kelontong. Penelitian ini menghadirkan kebaruan karena memadukan proses mining data, interpretasi aturan asosiasi, dan evaluasi perubahan tata letak dalam satu alur penelitian yang utuh, sehingga tidak berhenti pada analisis teoritis atau pengembangan sistem semata. Dengan demikian, penelitian ini menawarkan kontribusi praktis yang lebih konkret, yakni menghasilkan rekomendasi tata letak berbasis data yang dapat diuji dampaknya terhadap kenyamanan konsumen dan peningkatan penjualan

2.2. Tinjauan Teoritis

2.2.1. Algoritma Apriori

Algoritma Apriori merupakan bagian dari sebuah metode *Association rule* yang berfungsi untuk menemukan korelasi di antara item dalam dataset transaksional (Avini et al., 2024). *Association rules* pada *algoritma Apriori* dibentuk dengan cara menganalisis pola data yang sering muncul (*frequent pattern*) dan dengan menggunakan parameter *support* dan *confidence* untuk mengidentifikasi hubungan yang paling penting dan sesuai (Talumewo et al., 2025). Berikut ini standar atau kriteria penting yang diperlukan untuk membentuk *rules* dalam penggunaan *algoritma Apriori* (Jufri. et al., 2025):

a. Support

Support merupakan suatu ukuran yang menunjukkan seberapa besar tingkat dominasi suatu item/item set dari keseluruhan transaksi. Ukuran ini akan menentukan apakah suatu item/item set layak untuk dicari *confidence* selanjutnya atau tidak. Nilai *support* sebuah item diperoleh dengan rumus berikut:

$$\text{Support (A)} = \frac{\text{Jumlah Transaksi mengandung A}}{\text{Total Transaksi}}$$

Sedangkan nilai *support* dari 2 item diperoleh dari:

$$\text{Support (A} \cap \text{B)} = \frac{\text{Jumlah Transaksi mengandung A dan B}}{\text{Total Transaksi}}$$

b. *Confidence*

Confidence merupakan suatu ukuran yang menunjukkan hubungan antar 2 (dua) variabel item secara kondisional. Nilai *confidence* dari aturan $A \rightarrow B$ diperoleh rumus berikut:

$$\text{Confidence } P(B|A) = \frac{\text{Jumlah Transaksi mengandung } A \text{ dan } B}{\text{Jumlah Transaksi mengandung } A}$$

Terdapat dua proses utama yang dilakukan dalam *algoritma Apriori*, antara lain (Isa et al., 2022):

1. *Join* (Penggabungan) yaitu pada proses ini setiap item dikombinasikan dengan item yang lainnya sampai tidak terbentuk kombinasi lagi. Untuk menemukan L_k , suatu set dari kandidat k -*Itemset* dihasilkan dengan cara menjoinkan L_{k-1} dengan dirinya sendiri. Set kandidat hasil *join* ini nanti akan dinotasikan sebagai C_k . Adapun aturan dari *join* ini adalah setiap kandidat yang dihasilkan tidak boleh mengandung kandidat yang kembar antara satu dengan yang lainnya.
2. *Prune* (Pemangkasan) yaitu pada proses ini, hasil dari item yang telah dikombinasikan tadi lalu dipangkas dengan menggunakan *minimum support* yang telah ditentukan oleh user. Semua $(k-1)$ -*Itemset* yang tidak *frequent* tidak mungkin dapat menjadi subset dari *frequent* k -*Itemset*. Oleh karena itu, jika ada $(k-1)$ subset dari kandidat k -*Itemset* yang tidak termasuk dalam L_{k-1} , maka kandidat tidak mungkin *frequent* juga dan oleh karena itu dapat dihapus dari C_k .

Langkah-langkah *algoritma Apriori* untuk mendapatkan *rules* yang diinginkan, antara lain (Rahayu et al., 2024):

1. Tentukan minimum *support*.
2. Iterasi 1: Sistem men-scan database untuk mendapat kandidat 1-*Itemset* (himpunan item yang terdiri dari 1 item) dan menghitung nilai *supportnya*. Kemudian nilai *support* tersebut dibandingkan dengan *minimum support (MinSupport)* yang telah ditentukan, jika nilainya lebih besar atau sama dengan *minimum support* maka *Itemset* tersebut termasuk dalam large *Itemset*. *Itemset* yang tidak termasuk dalam large *Itemset* tidak diikuti dalam iterasi selanjutnya.
3. Iterasi 2: Sistem akan menggunakan hasil large *Itemset* pada iterasi pertama (L1) untuk membentuk kandidat *Itemset* kedua (L2). Pada iterasi selanjutnya sistem akan menggunakan hasil large *Itemset* pada iterasi selanjutnya akan menggunakan hasil large *Itemset* pada iterasi sebelumnya (Lk-1) untuk membentuk kandidat *Itemset* berikut (Lk). Sistem akan menggabungkan Lk-1 dengan Lk-1 untuk mendapatkan Lk, seperti pada iterasi sebelumnya sistem akan menghapus kombinasi *Itemset* yang tidak termasuk dalam large *Itemset*
4. Setelah dilakukan operasi gabung, maka pasangan *Itemset* baru hasil proses join tersebut dihitung *supportnya*.
5. Proses pembentuk kandidat yang terdiri dari proses join dan pangkas akan terus dilakukan hingga himpunan kandidat *Itemsetnya* null, atau sudah tidak ada lagi kandidat yang akan dibentuk.

6. Setelah itu, dari hasil *frequent Itemset* tersebut dibentuk *Association rules* yang memenuhi nilai *support* dan *confidence* yang telah ditentukan.

2.2.2. *Data mining*

Data mining merupakan serangkaian proses untuk menggali nilai tambah berupa informasi yang selama ini tidak diketahui secara manual dari status basis data. Informasi yang dihasilkan didapatkan dengan mengekstraksi dan mengenali pola yang penting atau menarik dari data yang terdapat pada basis data. *Data mining* digunakan untuk mencari pengetahuan yang terdapat dalam basis data dengan jumlah yang besar. *Data mining* adalah proses pencarian pengetahuan dari data dalam jumlah yang besar dengan menggunakan pengenalan pola, statistik dan teknik matematika (Putro, 2021).

Pada dasarnya *Data mining* memiliki keterkaitan yang erat dengan analisa data dan penggunaan perangkat lunak untuk mencari kesamaan dan pola dalam sekumpulan data. Analisa *Data mining* berjalan pada data yang cenderung terus membesar dan teknik terbaik yang dapat digunakan berorientasi pada data berukuran sangat besar untuk mendapatkan keputusan dan kesimpulan yang layak (Putro, 2021).

Data mining memiliki peran utama dalam membantu untuk memahami data, menemukan pola tersembunyi dan mengambil keputusan strategis. Peran utama *Data mining* adalah sebagai berikut (Rachbini & Zulkifli, 2025):

1. Estimasi (*Estimation*) yaitu proses proses memperkirakan nilai suatu variabel berdasarkan karakteristik dan pola yang ada.
2. Peramalan (*Forecasting*) yaitu berfokus untuk memperkirakan nilai atau tren di masa depan berdasarkan data historis.
3. Klasifikasi (*Classification*) yaitu tugas *Data mining* yang memiliki tujuan mengelompokkan data ke dalam kategori tertentu berdasarkan atributnya.
4. Klasterisasi (*Clustering*) yaitu suatu proses pengelompokan data tanpa adanya kategori yang telah ditentukan sebelumnya
5. Asosiasi (*Association*) yaitu proses menemukan aturan atau pola yang sering muncul bersama dalam dataset.

Beberapa aplikasi dari *Data mining* yang banyak digunakan adalah sebagai berikut (Suradiradja & Farizy, 2024):

1. WEKA

Waikato-Environment for Knowledge-Analysis (WEKA) merupakan software atau perangkat lunak yang menggunakan teknik atau algoritma machine learning untuk melakukan proses *Data mining*. WEKA akan menampilkan informasi mengenai hasil *Data mining* dalam bentuk chart atau gambar agar mudah dipahami.

2. *Rapid Miner*

Rapid Miner merupakan software atau perangkat lunak yang digunakan untuk melakukan kegiatan *Data mining*. Dengan aplikasi ini, dapat melakukan text mining yaitu untuk melakukan analisis berupa teks,

mengeksktraksi pola dan menggabungkan dengan menggunakan metode statistika, database dan *artificial intelligence*.

3. *Rattle*

Aplikasi *rattle* dimanfaatkan untuk keperluan *Data mining* dengan mengandalkan kekuatan 'R' *Statistical Software*. *Rattle* juga dimanfaatkan untuk sarana dalam mempelajari bahasa pemrograman R dan dapat melakukan analisis statistik serta membuat suatu model.

4. *Orange*

Orange merupakan aplikasi yang dapat menampilkan visualisasi data yang cukup bagus. Aplikasi ini menggunakan bahasa pemrograman. Tiap komponen yang ada pada *orange* disebut *widgets* yang digunakan untuk menampilkan tabel data dan selanjutnya menampilkan elemen data, memilih fitur, membaca data dan sebagainya.

5. KNIME

KNIME merupakan suatu platform yang terintegrasi untuk kebutuhan pembuatan laporan dan analisis data. Aplikasi KNIME menampilkan informasi berupa analisis data pelanggan dan analisis data finansial. KNIME mudah untuk digunakan dan juga di dalam pengoperasiannya memerlukan waktu yang cukup singkat.

2.2.3. *Association rules*

Association rules adalah teknik dalam data mining yang digunakan untuk menemukan aturan asosiatif antara item dalam sebuah kombinasi *Itemset* (Buulolo, 2020). *Association rule* sering disebut sebagai *Market*

Basket Analysis untuk menemukan hubungan antara himpunan dalam item-item tertentu. *Market Basket Analysis* adalah metode untuk menganalisis pola belanja konsumen dan mengetahui item-item yang sering dibeli bersamaan. *Association rule* juga dapat diterapkan pada database dengan dimensi yang cukup besar yang terdiri dari data transaksi (Adinugroho & Sari, 2018)

Association rule akan menemukan hubungan penting antar item dalam setiap transaksi, hubungan tersebut dapat menandakan kuat atau tidaknya suatu aturan dalam asosiasi. *Association rule* dapat digunakan untuk mengidentifikasi item-item yang mungkin dibeli secara bersamaan dengan item lain. Untuk menemukan aturan asosiasi sesuai yang diharapkan, maka harus menemukan nilai dari *support* yang telah ditetapkan. Nilai *support* dapat ditemukan dengan mencari semua aturan yang jumlah *support* $\geq$ minimum *support*. Nilai ini kemudian digunakan untuk menemukan nilai *confidence* yang ditentukan dari nilai *support* suatu aturan dalam sebuah transaksi (Isa et al., 2022).

Apabila suatu *Itemset* jarang muncul dalam transaksi (*infrequent*), maka kandidat yang tidak memenuhi nilai *support* $\geq$ minimum *support* tersebut harus segera dipangkas tanpa perlu menghitung *confidence*-nya. Strategi umum yang digunakan oleh banyak algoritma penggalian aturan asosiasi yaitu memecahkan masalah ke dalam dua pekerjaan utama, antara lain (Isa et al., 2022):

1. *Frequent Itemset Generation* bertujuan untuk mencari semua *Itemset* yang memenuhi ambang batas *minimum support* yang disebut *Itemset frequent* atau *Itemset* yang sering muncul.
2. *Rules Generation* bertujuan untuk mengekstrak aturan dengan *confidence* tinggi dari *Itemset frequent* yang ditemukan dalam langkah sebelumnya disebut aturan yang kuat (*strong rules*)