

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Penelitian-penelitian sejenis ini telah dilakukan sebelumnya, beberapa penelitian terdahulu yang mendasari penelitian ini antara lain.

Afriyanti dan Retnoningsih, (2022) melakukan penelitian yang memanfaatkan Algoritma FP-Growth untuk diterapkan pada aplikasi Sistem Rekomendasi Buku Perpustakaan dan hasil keluaran berupa rekomendasi buku sejenis dan rekomendasi buku yang paling sering dipinjam (*top five*) yang sesuai. Hasil *rules* dapat dilihat jenis buku mana yang menjadi prioritas atau tidak, prioritas untuk ditambahkan di perpustakaan karena banyak yang meminjam, dan transaksi peminjaman hasil rekomendasi dapat menjadi acuan bagi pengelolaan perpustakaan apabila akan menyusun tata letak rak buku.

Penelitian lainnya tentang Algoritma FP-Growth adalah untuk mengetahui hasil penjualan dari *sparepart* motor sport yang paling banyak terjual di PT. Selamat Lestari Mandiri Cibadak yang dilakukan oleh Suhada, dkk (2020). Dimana penerapan Algoritma FP-Growth untuk analisis pola pembelian konsumen sangat bermanfaat bagi perusahaan tersebut, karena PT. Selamat Lestari Mandiri Cibadak akan mengetahui *sparepart* mana yang banyak dibeli dan membantu dalam pemesanan *sparepart* pada kantor pusat.

Silitonga dan Windarto (2022) melakukan penelitian menggunakan Algoritma FP-Growth untuk mengetahui jenis obat yang muncul secara

bersamaan berdasarkan data transaksi resep dokter pada Apotek French Farma. Hasil dari penelitian ini berupa *association rule* yang dapat dijadikan masukan bagi pihak apotek dalam menentukan pola letak barang pada apotek tersebut.

Anas dkk., (2022) melakukan penelitian tentang perbandingan Algoritma FP-Growth dan Apriori dalam mengelola data transaksi toko. Berdasarkan beberapa skenario pengujian, kedua algoritma tersebut memiliki kelebihan dan kekurangan masing-masing. Algoritma Apriori memiliki waktu eksekusi 0,07 detik lebih lama, karena proses scanning lebih mendalam sedangkan menggunakan FP-Growth scanning tidak terlalu dalam sehingga memiliki waktu yang relatif lebih cepat, yaitu 0,2 detik. Namun, hasil kombinasi *item* yang diperoleh Algoritma Apriori lebih bervariasi dibandingkan dengan algoritma FP-Growth.

Penelitian lainnya tentang perbandingan Algoritma Apriori dan FP-Growth juga dilakukan oleh Patel dan Yadav (2022). Hasil dari penelitian ini adalah Algoritma FP-Growth membutuhkan waktu lebih sedikit dari pada Algoritma Apriori. Hal ini dikarenakan Algoritma FP-Growth hanya melakukan scanning database sebanyak dua kali sedangkan Algoritma Apriori melakukan scanning database satu kali untuk setiap kandidat *itemset* yang dihasilkan. Waktu eksekusi kedua algoritma adalah 19.988207817077637 detik (Algoritma Apriori) dan 15.845721960067749 detik (Algoritma FP-Growth).

Aplikasi yang dibangun dalam penelitian ini berguna untuk menerapkan Algoritma FP-Growth untuk mengetahui pola pembelian konsumen untuk mempermudah pihak swalayan dalam membuat keputusan ketika menyusun strategi penjualan, seperti menentukan tata letak barang, dan menentukan paket pembelian.

Tabel 2.1 Perbandingan penelitian

No.	Penulis	Objek	Metode	Hasil
1.	Afriyanti dan Retnoningsih, (2022)	Perpustakaan SMPN 31 Bekasi	Algoritma FP-Growth	Menghasilkan aplikasi Sistem Rekomendasi Buku Perpustakaan. Hasil keluaran berupa rekomendasi buku sejenis dan rekomendasi buku yang paling sering dipinjam (<i>top five</i>) yang sesuai.
2.	Satia Suhada, Daniel Ratag, Gunawan, Dede Wintana, Taufik Hidayatulloh (2020)	PT. Selamat Lestari Mandiri Cibadak	Algoritma FP-Growth	Hasil penjualan dari sparepart motor sport yang paling banyak terjual di PT. Selamat Lestari Mandiri Cibadak bisa diketahui dengan menggunakan Algoritma <i>FP-Growth</i> . <i>Sparepart</i> yang

No.	Penulis	Objek	Metode	Hasil
				memenuhi minimum <i>support</i> dan <i>minimum confidence</i> serta yang banyak terjual adalah OLI MPX2 10W30 SL 0,8L IDE, BRAKESHOE, dan RACESTEERINGKIT.
3.	Desi Asima Silitonga, Agus Perdana Windarto (2022)	Apotek French Farma Sidikalang	Algoritma FP-Growth	Metode Algoritma FP-Growth terbukti mampu mengetahui hasil analisis yang muncul secara bersamaan berdasarkan data penjualan obat di Apotek French Farma. Hasil implementasi menggunakan <i>software WEKA3.9.5</i> sebagai alat bantu untuk membuktikan bahwa perhitungan data manual hasilnya sama dengan pengujian dilakukan menggunakan <i>software WEKA3.9.5</i> .
4.	Syukron Anas, Nelson Rumui,	Algoritma Apriori dan	Algoritma Apriori dan	Algoritma Apriori memiliki waktu eksekusi yang lebih

No.	Penulis	Objek	Metode	Hasil
	Andi Roy, dan Pujo Hari Saputro (2022)	FP-Growth	FP-Growth	lama, karena proses scanning lebih mendalam sedangkan menggunakan FP-Growth scanning tidak terlalu dalam sehingga memiliki waktu yang relatif lebih cepat. Namun, hasil kombinasi <i>item</i> yang diperoleh Algoritma Apriori lebih bervariasi dibandingkan dengan algoritma FP-Growth.
5.	Harvendra Kumar Patel dan K. P. Yadav (2022)	Algoritma Apriori dan FP-Growth	Algoritma Apriori dan FP-Growth	Algoritma FP-Growth membutuhkan waktu lebih sedikit dari pada Algoritma Apriori. Hal ini dikarenakan Algoritma FP-Growth hanya melakukan scanning database sebanyak dua kali sedangkan Algoritma Apriori melakukan scanning database satu kali untuk setiap kandidat <i>itemset</i> yang dihasilkan.

No.	Penulis	Objek	Metode	Hasil
6.	Bagas Aditia (2024)	Swalayan Surya Jaya	Algoritma FP-Growth	Menghasilkan aplikasi analisis pola pembelian konsumen sehingga bisa membantu pemilik swalayan dalam mengatur tata letak barang dan menentukan paket pembelian.

2.2 Dasar Teori

Dalam penelitian terdapat beberapa teori yang digunakan untuk memahami definisi, serta pengertian dasar dari kebutuhan aplikasi yang akan dibangun.

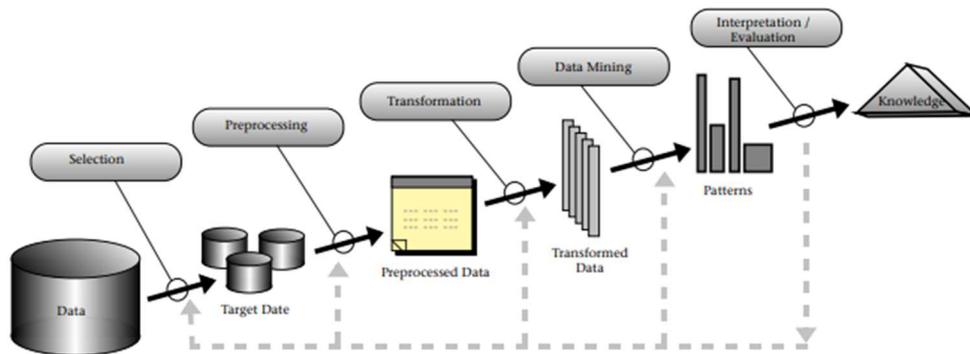
2.2.1 Data Mining

Data mining adalah proses penggalian informasi dan pola yang bermanfaat dari data yang sangat besar. Proses ini bertujuan untuk menemukan informasi baru dan bermanfaat dari data besar yang tersimpan, seperti pola, hubungan antar variabel, dan anomali yang tidak terlihat secara kasat mata. *Data mining* mencakup pengumpulan data, ekstraksi data, analisis data, dan statistik data (Arhami dan Nasir, 2020).

Data mining juga biasa dikenal dengan *Knowledge Discovery in Database* (KDD) yang merupakan proses penggalian informasi bermanfaat dari data yang sangat besar. Proses ini melibatkan beberapa tahapan, seperti pemilihan dan

pengumpulan data, *data preprocessing*, *data transformation*, *data mining*, dan *interpretation/evaluation*.

A. Proses Data Mining



Gambar 2.1 Tahap-tahap yang menyusun proses KDD (Fayyad dkk., 1996)

Proses data mining melibatkan serangkaian tahapan yang bertujuan untuk mengekstrak pengetahuan yang berharga dari data. Berikut adalah tahapan dalam proses data mining sesuai pada Gambar 2.1:

1. *Data Selection* (Seleksi Data)

Tahap awal dari proses KDD adalah memilih dataset yang relevan dengan tujuan penelitian atau analisis yang akan dilakukan. Proses ini memastikan bahwa data yang digunakan untuk penggalian pengetahuan merupakan data yang relevan, berkualitas, dan sesuai dengan tujuan analisis yang ingin dicapai (Han dan Kamber, 2006).

2. *Data Preprocessing* (Pra Pemrosesan Data)

Tahap ini meliputi operasi pembersihan data, seperti identifikasi dan penanganan data yang hilang, outlier, atau noise dalam data, integrasi data dilakukan untuk menggabungkan data dari berbagai sumber menjadi satu dataset yang konsisten, reduksi data untuk mengurangi jumlah atribut dan memilih atribut yang paling relevan atau yang memiliki dampak yang signifikan terhadap tujuan analisis (Arhami dan Nasir, 2020).

3. *Data Transformation* (Transformasi Data)

Dalam tahap ini data ditransformasikan dan dikonsolidasi kedalam bentuk yang sesuai dengan kebutuhan analisis (Han dan Kamber, 2006).

4. *Data Mining* (Penambangan Data)

Merupakan proses penerapan teknik *data mining* yang sesuai dengan tujuan analisis. Teknik *data mining* yang sering digunakan antara lain, *classification, clustering, regression, dan association rules*. Teknik yang diterapkan akan bergantung pada tujuan analisis dan jenis data yang digunakan (Han dan Kamber, 2006).

5. *Interpretation/Evaluation Patterns* (Interpretasi/Evaluasi Pola)

Setelah menerapkan teknik *data mining*, tahap selanjutnya adalah mengevaluasi pola yang benar-benar penting dan menginterpretasikan perolehan pola dalam bentuk yang mudah dimengerti oleh pengguna atau pihak yang berkepentingan (Arhami dan Nasir, 2020).

6. *Knowledge* (Pengetahuan)

Merupakan tahapan terakhir yang melibatkan pengetahuan secara langsung, menggabungkan pengetahuan dengan sistem lain untuk tindakan

lanjutan, atau mendokumentasikan dan melaporkannya kepada pihak yang berkepentingan (Arhami dan Nasir, 2020).

B. Analisis Asosiasi

Analisis asosiasi adalah teknik dalam *data mining* yang digunakan untuk menemukan aturan asosiatif antara suatu kombinasi *item*. Karena teknik ini sering digunakan untuk menganalisis isi keranjang belanja para konsumen yang melakukan transaksi, analisis asosiasi juga dikenal dengan nama *Market Basket Analysis* (Kusrini dan Luthfi, 2009).

Hasil dari analisis asosiasi adalah aturan asosiasi (*association rule*), aturan ini umumnya dinyatakan dalam bentuk jika “kejadian sebelumnya” kemudian “konsekuensinya” (*If antecedent, then consequent*). Jika dimisalkan, $A \rightarrow B$. Yang artinya jika konsumen membeli *itemset* A, maka ada kemungkinan konsumen tersebut membeli *itemset* B (Ristianingrum dan Sulastrri, 2017).

Dalam menentukan suatu *association rule*, terdapat ukuran yang menyatakan sebuah aturan dianggap penting atau menarik (*interestingness measure*) yang didapat dari hasil pengolahan atau perhitungan tertentu (Ristianingrum dan Sulastrri, 2017). Untuk mengukur *interestingness measure*, dapat digunakan variabel berikut:

a) *Support*

Support adalah suatu ukuran yang menunjukkan rata-rata popularitas *item/itemset* dalam *dataset* (Ristianingrum dan Sulastrri, 2017). Jika terdapat

aturan “*if A then B*”, maka *support* adalah seberapa populer *item/itemset* A dan B dibeli secara bersamaan dari keseluruhan transaksi yang ada.

$$\text{Support}(A, B) = P(A \cap B)$$

$$P(A \cap B) = \frac{\sum \text{Transaksi mengandung A dan B}}{\sum \text{Transaksi}}$$

Contoh: Jika terdapat kombinasi pembelian "roti dan mentega" di sebuah toko, dan kombinasi ini muncul di 20 dari 100 transaksi, maka nilai *Support*-nya adalah 20% (karena 20 transaksi mengandung "roti dan mentega" dari total 100 transaksi).

b) *Confidence*

Confidence adalah suatu ukuran yang menunjukkan hubungan antara kedua *item* secara *conditional* (berdasarkan suatu kondisi tertentu) (Suhada dkk., 2020). Misal seberapa sering konsumen membeli *item/itemset* B jika konsumen tersebut membeli *item/itemset* A.

$$\text{Confidence}(A \rightarrow B) = \frac{\sum \text{Transaksi mengandung A dan B}}{\sum \text{Transaksi mengandung A}}$$

Contoh: Jika terdapat aturan asosiasi "roti maka mentega" dengan *confidence* 80%, itu berarti ketika seseorang membeli roti, kemungkinan besar dia juga akan membeli mentega adalah sebanyak 80%.

c) *Lift Ratio*

Lift ratio adalah suatu ukuran yang menunjukkan seberapa kuat aturan asosiasi (*association rule*) yang telah terbentuk. Nilai *lift ratio* yang lebih besar dari 1 berarti *item/itemset* B kemungkinan akan dibeli jika *item/itemset* A dibeli. Sedangkan jika nilai *lift ratio* kurang dari sama dengan 1 maka

item/itemset A dan B tidak memiliki asosiasi sama sekali (Arhami dan Nasir, 2020).

$$Lift\ Ratio\ (A \rightarrow B) = \frac{Support\ (A,B)}{Support\ (A)*Support(B)}$$

2.2.2 Algoritma Frequent Pattern Growth

Algoritma FP-Growth (*Frequent Pattern-Growth*) adalah salah satu algoritma dalam analisis asosiasi untuk menemukan *item* yang sering muncul atau disebut juga *frequent itemset*. Algoritma ini merupakan pengembangan dari Algoritma Apriori. Algoritma FP-Growth memiliki keuntungan dalam efisiensi waktu komputasi, terutama ketika diterapkan pada dataset yang besar dengan banyak *item* (Afriyanti dan Retnoningsih, 2022). Algoritma FP-Growth menggunakan struktur *FP-Tree* untuk mengidentifikasi *frequent itemset* secara efisien, sehingga tidak perlu melakukan pembangkitan kandidat-kandidat *itemset* yang memakan waktu seperti yang dilakukan oleh Algoritma Apriori. (Patel dan Yadav, 2022).

Langkah-langkah dari Algoritma FP-Growth adalah sebagai berikut (Murti dkk., 2019) :

1. Memperoleh data transaksi.
2. Menentukan nilai minimum *support* dan minimum *confidence*.
3. Menyusun tabel *generated frequent itemset*.
4. Menyusun tabel *fp-list*.
5. Pembentukan *fp-tree*.

6. Pembangkitan *conditional pattern base*.
7. Pembangkitan *conditional FP-Tree*.
8. Pencarian *frequent pattern*.

Berikut contoh penerapan Algoritma FP-Growth untuk mencari aturan asosiasi (*association rule*):

1. Dataset transaksi penjualan.

Tabel 2.2 Dataset transaksi penjualan

Transaksi ID	Item
1	apel, bir, nasi, ayam
2	apel, bir, nasi
3	apel, bir
4	apel, bir
5	bir, nasi, ayam, susu
6	bir, nasi, susu
7	bir, susu
8	pir, susu

2. Menentukan nilai minimum *support* dan *confidence*.

Dari tabel 2.2 data transaksi diatas akan dicari pola asosiasi dengan minimal *support* 25% dan *confidence* 75%.

3. Menyusun tabel *Generated Frequent Itemset*.

Tabel 2.3 Generated Frequent Itemset

No.	Item	Frekuensi	Support
1.	Bir	7	87.5%
2.	Apel	4	50%
3.	Nasi	4	50%
4.	Susu	4	50%
5.	ayam	2	25%

Tabel 2.3 *generated frequent itemset* diurutkan secara menurun (*descending*) berdasarkan frekuensi *item* yang memenuhi nilai minimum *support* sebanyak 25%. *Item* pir disingkirkan karena tidak memenuhi nilai minimum *support*.

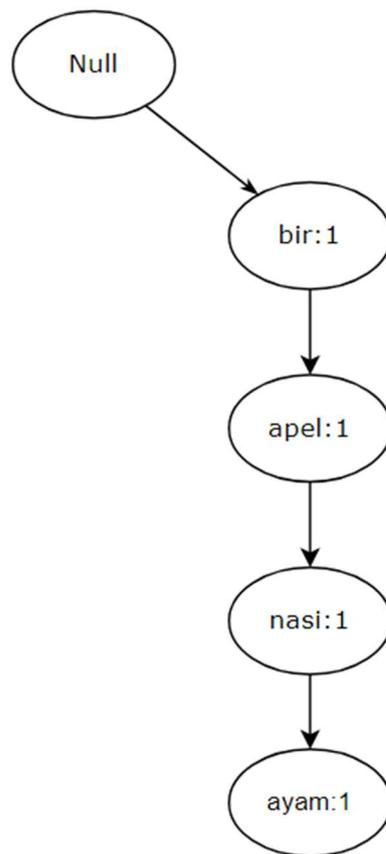
- Menyusun tabel *FP-List*.

Tabel 2.4 FP-List

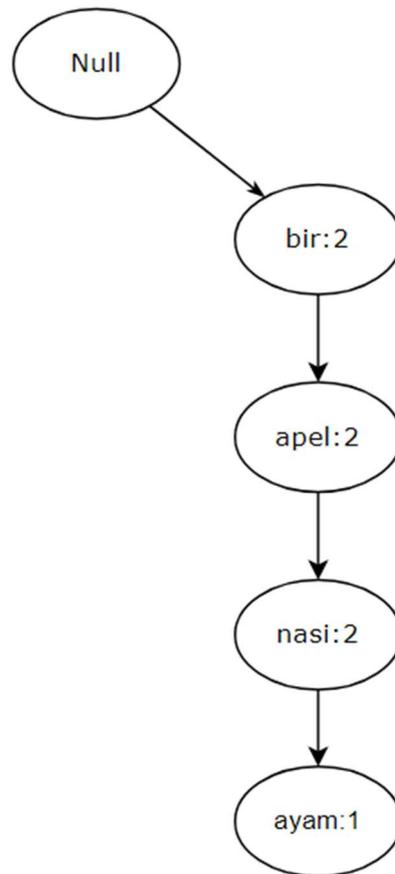
Transaksi ID	Item
1	bir, apel, nasi, ayam
2	bir, apel, nasi
3	bir, apel
4	bir, apel
5	bir, nasi, susu, ayam
6	bir, nasi, susu
7	bir, susu

Transaksi ID	Item
8	Susu

Pada tabel 2.4 diatas dapat dijelaskan bahwa *item* pada setiap transaksi diurutkan secara *descending* berdasarkan frekuensi yang didapat pada langkah sebelumnya.

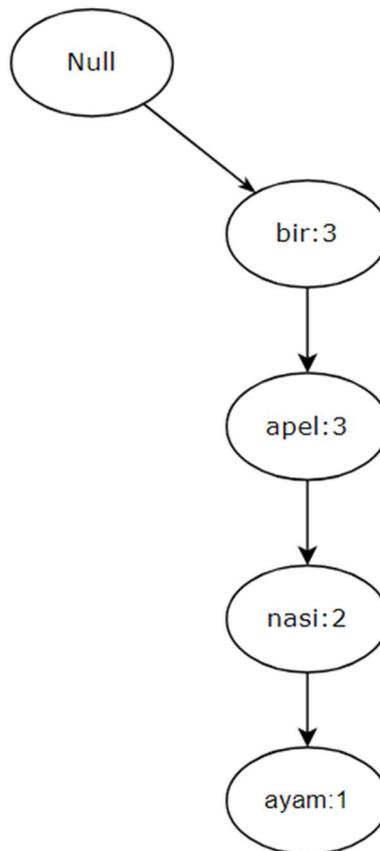
5. Pembentukan *FP-Tree***Gambar 2.2 Transaksi ID 1**

Gambar 2.1 Transaksi ID 1 *item* yang terjual adalah {bir, apel, nasi, ayam} yang kemudian akan membuat simpul bir, apel, nasi, dan ayam, sehingga terbentuk lintasan Null→bir→apel→nasi→ayam dengan *support count* dari setiap simpul bernilai 1.



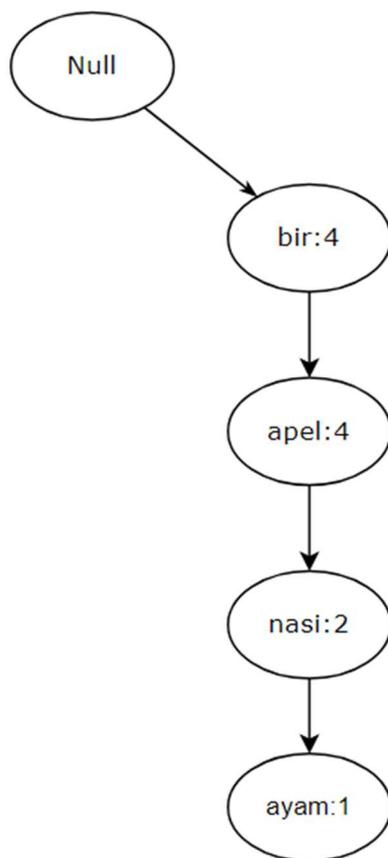
Gambar 2.3 Transaksi ID 2

Gambar 2.3 Transaksi ID 2 *item* yang terjual adalah {bir, apel, nasi} yang kemudian akan membuat simpul bir, apel, dan nasi. Karena simpul bir, apel, dan nasi sudah ada dalam lintasan maka lintasan Null→bir→apel→nasi akan ditimpa dan *support count* dari setiap simpul yang ditimpa bertambah 1.



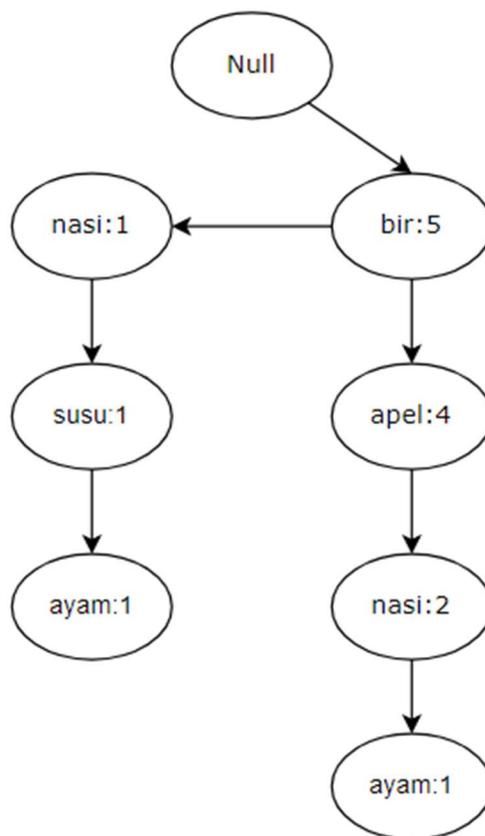
Gambar 2.4 Transaksi ID 3

Gambar 2.4 Transaksi ID 3 *item* yang terjual adalah {bir, apel} yang kemudian akan membuat simpul bir dan apel. Karena simpul bir dan apel sudah ada dalam lintasan maka lintasan Null→bir→apel akan ditimpa dan *support count* dari setiap simpul yang ditimpa bertambah 1.



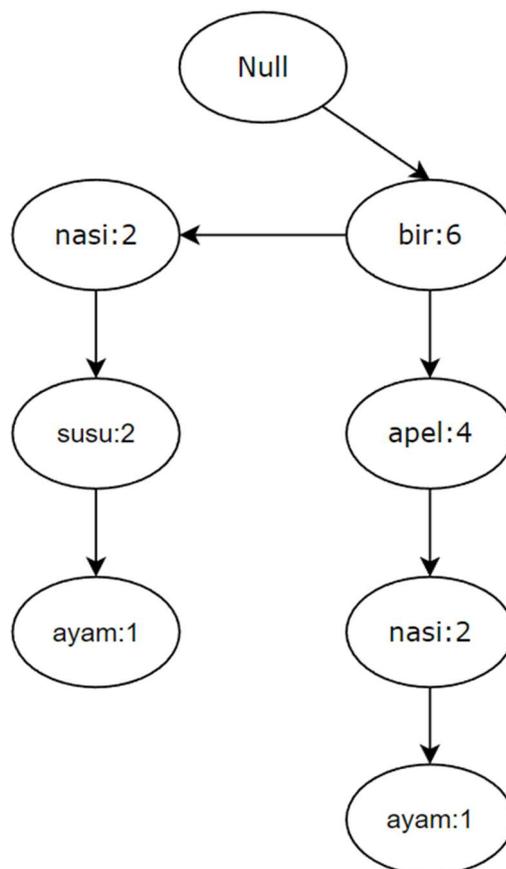
Gambar 2.5 Transaksi ID 4

Gambar 2.5 Transaksi ID 4 *item* yang terjual adalah {bir, apel} yang kemudian akan membuat simpul bir dan apel. Karena simpul bir dan apel sudah ada dalam lintasan maka lintasan Null→bir→apel akan ditimpa dan *support count* dari setiap simpul yang ditimpa bertambah 1.



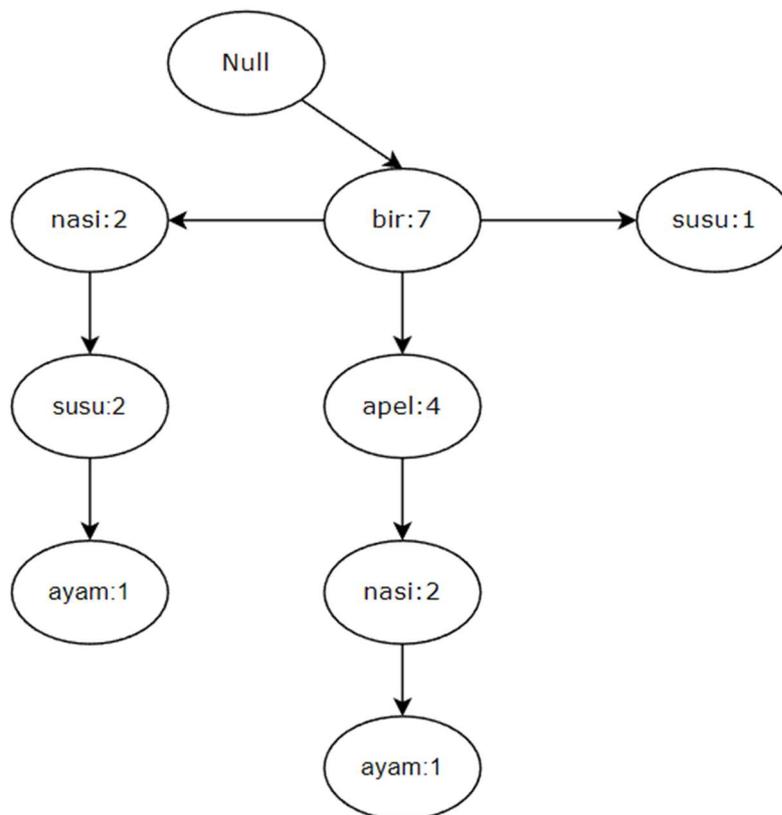
Gambar 2.6 Transaksi ID 5

Gambar 2.6 Transaksi ID 5 *item* yang terjual adalah {bir, nasi, susu, ayam} yang kemudian akan membuat simpul bir, nasi, susu, dan ayam, sehingga terbentuk lintasan Null→bir→apel→nasi→ayam dengan *support count* dari setiap simpul yang baru bernilai 1 dan yang ditimpa akan bertambah 1.



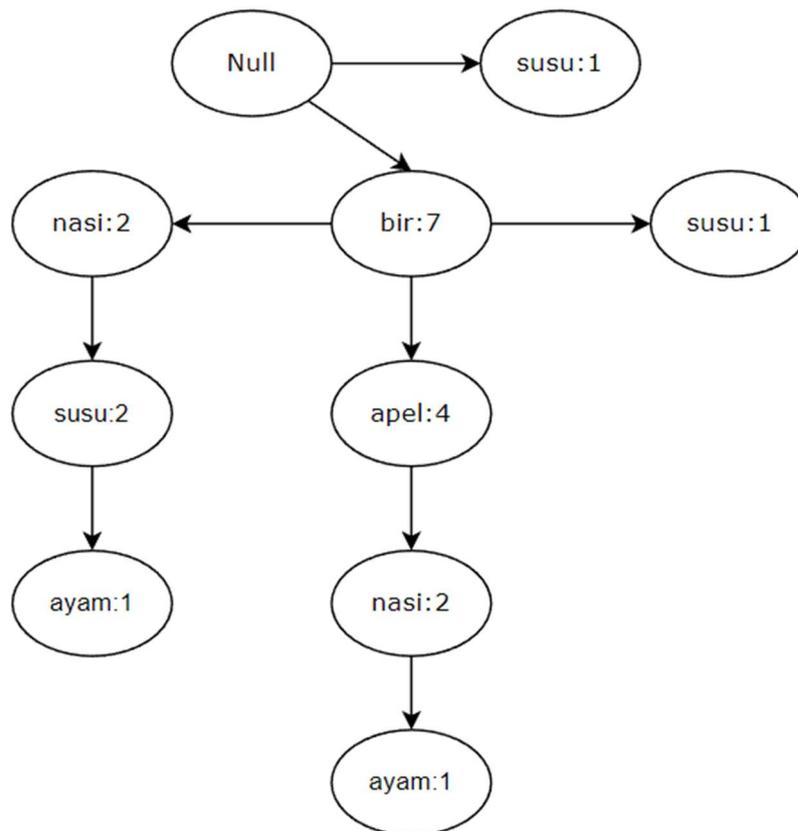
Gambar 2.7 Transaksi ID 6

Gambar 2.7 Transaksi ID 6 *item* yang terjual adalah {bir, nasi, susu} yang kemudian akan membuat simpul bir, nasi, dan susu. Karena simpul bir, nasi, dan susu sudah ada dalam lintasan maka lintasan Null→bir→nasi→susu akan ditimpa dan *support count* dari setiap simpul yang ditimpa bertambah 1.



Gambar 2.8 Transaksi ID 7

Gambar 2.8 Transaksi ID 7 *item* yang terjual adalah {bir, susu} yang kemudian akan membuat simpul bir dan susu, sehingga terbentuk lintasan Null→bir→susu dengan *support count* dari setiap simpul yang baru bernilai 1 dan yang ditimpa akan bertambah 1.



Gambar 2.9 Transaksi ID 8

Gambar 2.9 Transaksi ID 8 *item* yang terjual adalah {susu} yang kemudian akan membuat simpul susu, sehingga terbentuk lintasan Null→susu dengan *support count* dari simpul bernilai 1.

6. Pembangkitan *Conditional Pattern Base*

Conditional Pattern Base merupakan *sub database* yang berisi *prefix path* (lintasan prefix) dan *suffix pattern* (pola akhiran). Pembangkitan *conditional pattern base* didapatkan melalui FP-tree yang telah dibangun sebelumnya. Bentuk *conditional pattern base* dimulai dari *item* dengan *support count* terendah ke *item* dengan *support count* tertinggi.

Tabel 2.5 Conditional Pattern Base

No.	Item	Conditional Pattern base
1.	Ayam	{bir,apel,nasi:1}, {bir,nasi,susu:1}
2.	Susu	{bir,nasi:2}, {bir:1}
3.	Nasi	{bir,apel:2}, {bir:2}
4.	Apel	{bir:4}

Catatan :

Item Bir tidak diikutsertakan karena prefixnya adalah *Null* (root).

7. Pembangkitan *Conditional FP-Tree*

Pada tahap ini *support count* dari setiap *item* pada setiap *Conditional Pattern Base* dijumlahkan, lalu setiap *item* yang memiliki jumlah *support count* lebih besar atau sama dengan minimum *support count* akan dibangkitkan dengan *Conditional FP-Tree*.

Tabel 2.6 Conditional FP-Tree

No.	Item	Conditional FP-Tree
1.	ayam	{bir,nasi:2}, {bir:2}, {nasi:2}
2.	Susu	{bir,nasi:2}, {bir:3}, {nasi:2}
3.	Nasi	{bir,apel:2}, {bir:4}, {apel:2}
4.	Apel	{bir:4}

8. Pencarian *Frequent Pattern*

Tahap pencarian *frequent pattern* ini mengacu pada *conditional fp-tree* apabila *conditional fp-tree* merupakan lintasan tunggal (*single path*), maka didapatkan *frequent pattern* dengan melakukan kombinasi *item* pada setiap *conditional fp-tree*. Dan apabila bukan *single path* maka pencarian *frequent pattern* dilakukan secara rekursif.

Tabel 2.7 Frequent Pattern

No.	Item	Frequent Pattern
1.	ayam	bir, nasi, ayam (2)
2.	ayam	bir, ayam (2)
3.	ayam	nasi, ayam (2)
4.	susu	bir, nasi, susu (2)
5.	susu	bir, susu (3)
6.	susu	nasi, susu (2)
7.	nasi	bir, apel, nasi (2)
8.	nasi	bir, nasi (4)
9.	nasi	apel, nasi (2)
10.	apel	bir, apel (4)

9. Hasil

Aturan asosiasi yang dihasilkan adalah aturan yang telah memenuhi nilai minimum *support* dan *confidence*. Aturan yang tidak memenuhi nilai

minimum *support* dan *confidence* akan disingkirkan dan tidak ditampilkan dalam tabel.

Tabel 2.8 Aturan Asosiasi (Association Rule)

No.	Aturan Asosiasi	Support	Confidence	Lift Ratio
1.	Jika bir, ayam maka nasi	$2/8 = 25\%$	$2/2 = 100\%$	2
2.	Jika nasi, ayam maka bir	$2/8 = 25\%$	$2/2 = 100\%$	1.14
3.	Jika ayam maka bir, nasi	$2/8 = 25\%$	$2/2 = 100\%$	2
4.	Jika ayam maka bir	$2/8 = 25\%$	$2/2 = 100\%$	1.14
5.	Jika ayam maka nasi	$2/8 = 25\%$	$2/2 = 100\%$	2
6.	Jika nasi, susu maka bir	$2/8 = 25\%$	$2/2 = 100\%$	1.14
7.	Jika susu maka bir	$3/8 = 37.5\%$	$3/4 = 75\%$	0.86
8.	Jika apel, nasi maka bir	$2/8 = 25\%$	$2/2 = 100\%$	1.14
9.	Jika nasi maka bir	$4/8 = 50\%$	$4/4 = 100\%$	1.14
10.	Jika apel maka bir	$4/8 = 50\%$	$4/4 = 100\%$	1.14

Pada tabel 2.8 nilai *support* aturan nomor 1 diperoleh dari rumus sebagai berikut:

$$Support(bir, ayam \rightarrow nasi) = \frac{\sum \text{Transaksi mengandung bir, ayam, nasi}}{\sum \text{Transaksi}}$$

diketahui pada tabel 2.2 total transaksi yang mengandung *itemset* {bir, ayam, nasi} adalah berjumlah 2 transaksi. Sedangkan total seluruh transaksi

berjumlah 8 transaksi. Maka nilai *support itemset* {bir, ayam, nasi} adalah $2/8 = 0.25 = 25\%$.

Nilai *confidence* aturan asosiasi nomor 1 diperoleh dari rumus sebagai berikut:

$$\text{Confidence}(bir, ayam \rightarrow nasi) = \frac{\sum \text{Transaksi mengandung } bir, ayam, nasi}{\sum \text{Transaksi mengandung } bir, ayam}$$

Diketahui pada tabel 2.2 total transaksi yang mengandung *itemset* {bir, ayam, nasi} adalah berjumlah 2 transaksi. Sedangkan total transaksi yang mengandung *itemset* {bir, ayam} adalah berjumlah 2 transaksi. Maka nilai *confidence* aturan asosiasi nomor 1 adalah $2/2 = 1 = 100\%$.

Nilai *lift ratio* aturan asosiasi nomor 1 diperoleh dengan rumus sebagai berikut:

$$\text{Lift Ratio}(bir, ayam \rightarrow nasi) = \frac{\text{Support}(bir, ayam, nasi)}{\text{Support}(bir, ayam) * \text{Support}(nasi)}$$

Diketahui pada tabel 2.8 nilai *support* aturan asosiasi nomor 1 adalah 25% atau 0.25. pada tabel 2.2 total transaksi berjumlah 8 transaksi, sedangkan jumlah transaksi yang mengandung *itemset* {bir, ayam} berjumlah 2 transaksi dan jumlah transaksi yang mengandung *item* nasi berjumlah 4 transaksi.

Dari penjelasan diatas dapat dihitung nilai *support itemset* {bir, ayam} adalah $2/8 = 0.25$ dan nilai *support item* nasi adalah $4/8 = 0.5$. Maka nilai *lift ratio* aturan asosiasi nomor 1 adalah $0.25/(0.25*0.5) = 2$. Aturan ini dinyatakan *valid* karena memiliki nilai *lift ratio* lebih besar dari 1. Aturan asosiasi yang

digunakan rekomendasi hanyalah aturan yang memiliki nilai *lift ratio* lebih besar dari 1. Aturan dengan nilai *lift ratio* lebih kecil atau sama dengan 1 tidak digunakan untuk membuat rekomendasi.

Contoh pada tabel 2.8 terdapat aturan “Jika susu maka bir” dengan nilai *lift ratio* 0.86 aturan ini dianggap tidak *valid* karena memiliki nilai *lift ratio* lebih kecil atau sama dengan 1 sehingga aturan ini tidak dijadikan rekomendasi.

2.2.3 PHP (PHP: Hypertext Preprocessor)

PHP (PHP: *Hypertext Preprocessor*) adalah bahasa pemrograman script-script yang membuat dokumen HTML secara *on the fly* yang dieksekusi di server web, dokumen HTML yang dihasilkan dari suatu aplikasi bukan dokumen HTML yang dibuat dengan menggunakan editor teks atau editor HTML, dikenal juga sebagai bahasa pemrograman server side. (Sidik, 2012)

2.2.4 Framework Laravel

Laravel adalah sebuah framework PHP yang dirilis di bawah lisensi MIT, dibangun dengan konsep MVC (*model view controller*). Laravel adalah pengembangan website berbasis MVC (*model view controller*) yang ditulis dalam PHP yang dirancang untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan awal dan biaya pemeliharaan, dan untuk meningkatkan pengalaman bekerja dengan aplikasi dengan menyediakan sintaks yang ekspresif, jelas dan menghemat waktu. (Hermanto dkk., 2019)

Menurut Supono dan Putratama dalam penelitian Desy dkk. (2021) MVC (*model view controller*) adalah sebuah konsep yang diperkenalkan oleh penemu

Smalltalk (Trygve Reenskaug) untuk meng-enkapsulasi data bersama dengan pemrosesan (*model*), mengisolasi dari proses manipulasi (*controller*) dan tampilan (*view*) untuk direpresentasikan pada sebuah user interface (Desy dkk., 2021).

MVC membagi aplikasi menjadi tiga komponen utama yang saling terpisah namun saling berinteraksi (Supono dan Putratama, 2016):

1. Model, biasanya berhubungan langsung dengan database untuk memanipulasi data (insert, update, delete, search), menangani validasi dari bagian controller, namun tidak dapat berhubungan langsung dengan view.
2. View, merupakan bagian yang menangani presentation logic. Pada suatu aplikasi web bagian ini biasanya berupa file template HTML, yang diatur oleh controller. View berfungsi untuk menerima dan mempresentasikan data kepada user. Bagian ini tidak memiliki akses langsung terhadap bagian model.
3. Controller, merupakan bagian yang mengatur hubungan antara bagian model dan bagian view, controller berfungsi untuk menerima request dan data dari user kemudian menentukan apa yang akan diproses oleh aplikasi.

2.2.5 Basis Data (Database)

Menurut Jubilee (2017) dalam buku Otodidak MySQL untuk Pemula mengatakan bahwa basis data adalah sebuah aplikasi yang menyimpan sekumpulan data. Dan setiap basis data tersebut memiliki perintah tertentu untuk dapat menciptakan, mengakses, mengatur, mencari, serta menyalin data yang ada di dalamnya.

2.2.6 MySQL

Menurut Mundzir MF (2018) dalam buku Pemrograman Web Seri PHP mengatakan bahwa MySQL merupakan sistem manajemen database SQL yang bersifat *open source* (terbuka) dan paling banyak digunakan saat ini. Sistem database MySQL mampu mendukung beberapa fitur seperti *multithreaded*, *multiuser*, dan *SQL database management system (DBMS)*.