

BAB III

LANDASAN TEORI

A. Apotek dan Produk Kesehatan

Produk kesehatan merupakan suatu produk yang dapat menjaga daya tahan tubuh dan mengobati penyakit [14]. Produk kesehatan pada umumnya dijual di apotek. Apotek merupakan tempat yang menjual obat-obatan dan produk kesehatan lainnya. Obat dan produk kesehatan yang dijual beragam, ada yang berbentuk padat hingga bubuk, ada yang dapat diminum harus dengan resep dokter dan ada juga yang tidak. Menurut Peraturan Menteri Kesehatan Republik Indonesia Nomor 9 Tahun 2017 Tentang Apotek, apotek adalah tempat yang melayani kepentingan kefarmasian yang dilakukan oleh apoteker. Dalam mendirikan, diwajibkan mendapatkan surat izin dari pemerintah kabupaten/kota setempat.

Keberadaan apotek di suatu wilayah, baik wilayah perkotaan hingga pedesaan merupakan hal yang penting. Kebutuhan akan obat-obatan dan produk kesehatan lainnya, baik tradisional dan modern tidak bisa dilepaskan dari kehidupan manusia. Adapun pengertian dari obat tradisional merupakan obat atau produk kesehatan yang bahannya berasal dari hewan atau tumbuhan yang dapat dikonsumsi secara langsung atau diolah tanpa menggunakan teknologi di bidang kesehatan, sedangkan obat modern merupakan obat yang dalam pemakaiannya perlu diolah dan diuji terlebih dahulu baik secara klinis maupun farmakologis [15]. Berdasarkan hal tersebut, jika ketersediaan obat di suatu wilayah kurang atau bahkan tidak tersedia, maka akan dapat menimbulkan masalah kesehatan jika penyakit yang diderita seseorang memerlukan obat dengan cepat. Oleh karena itu, diperlukan penanganan persediaan obat yang baik dan tepat.

B. Data Mining

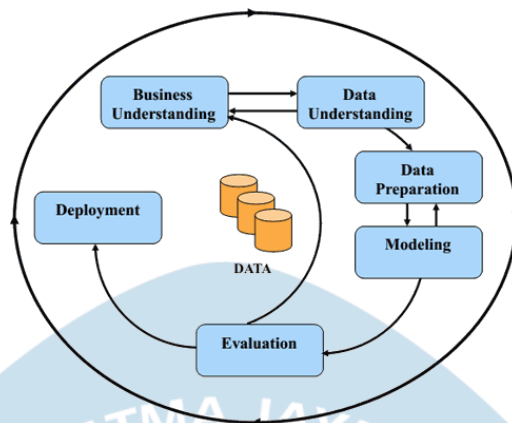
Data mining terdiri atas 2 kata yakni data dan *mining*. Adapun definisi kata tersebut, data merupakan kumpulan keterangan berupa fakta yang mengandung informasi, sedangkan *mining* adalah penambangan. *Data mining* atau

penambangan data merupakan sebuah proses di mana kumpulan data dari berbagai sumber dikumpulkan kemudian diolah sehingga dapat menghasilkan informasi [16]. Informasi yang didapatkan kemudian dapat digunakan untuk pengambilan keputusan. *Data mining* sendiri adalah gabungan dari berbagai ilmu seperti pembelajaran mesin, basis data, hingga visualisasi data [17]. Berdasarkan hal tersebut, *data mining* dapat diterapkan di berbagai sektor seperti kesehatan, perekonomian, dan politik.

Sebagai penerapan dalam berbagai sektor, penambangan data akan menganalisis data-data yang telah dikumpulkan. Dalam menganalisis data-data tersebut dibutuhkan tahapan yang terstruktur agar proses analisis dapat berjalan dengan lancar dan informasi yang didapatkan dalam digunakan secara maksimal untuk pengambilan keputusan baik bagi organisasi maupun perusahaan. Dalam pengambilan keputusan yang melalui berbagai tahapan yang terstruktur, dibutuhkan *tools* untuk pengolahan data tersebut, seperti *tool* untuk pelaporan dan *query*. Agar tahapan dapat terstruktur, maka digunakan pendekatan CRISP-DM (*Cross Industry Standard Process for Data Mining*).

C. CRISP-DM

CRISP-DM sebagai salah satu metodologi penelitian pada penambangan data digunakan untuk membantu dalam memecahkan masalah bisnis atau penelitian [18]. CRISP-DM telah dikembangkan sejak tahun 1996. Dalam pengembangan metodologi ini, ada beberapa perusahaan yang berpartisipasi yakni Integral Solutions Ltd. (ISL), OHRA, Daimler-Benz (Daimler Chrysler), dan NCR. CRISP-DM memberikan gambaran dalam siklus *data mining*. Siklus atau tahapan pada CRISP-DM terdiri atas 6 fase yang ditunjukkan pada Gambar 3.1. Berikut gambar siklus beserta penjelasannya:



Gambar 3. 1 Fase Pada CRISP-DM

1. *Business Understanding*

Tahap ini merupakan tahap pertama pada CRISP-DM. Tahap ini merupakan tahap vital atau tahap terpenting. Hal ini dikarenakan, perlu dilakukan pemahaman bisnis yang akan digunakan untuk penelitian. Hal-hal yang perlu dipahami berkenaan dengan bisnis apa yang akan diteliti dan bagaimana situasinya, tujuan bisnis, menentukan batasan, bagaimana mendapatkan data untuk penelitian, dan model yang akan digunakan untuk mencapai tujuan [19].

2. *Data Understanding*

Tahap ini dimulai dengan mengumpulkan data. Kemudian, setelah data dikumpulkan, data dipelajari, dipahami, dan identifikasi kualitasnya. *Data understanding* harus dilakukan dengan teliti dan tidak terburu-buru agar tahap selanjutnya dapat dilakukan dengan maksimal. Beberapa permasalahan dalam data yang dapat diidentifikasi seperti *missing value* dan *outlier*.

3. *Data Preparation*

Setelah mengetahui karakteristik dan permasalahan pada data, maka tahap selanjutnya yang dapat dilakukan yaitu *data preparation*. Beberapa hal yang dilakukan pada tahap ini yaitu pemilihan data (*table, field*), pembersihan data atau *data cleaning*, dan transformasi yang disesuaikan dengan *tools* yang dipakai.

4. *Modeling*

Tahap ini merupakan tahap di mana menentukan teknik penambangan data, algoritma, dan implementasinya. Dalam pengimplementasiannya, perlu digunakan alat atau *tools* tertentu.

5. *Evaluation*

Data yang telah melalui tahap proses pemodelan, selanjutnya dievaluasi dan dipastikan apakah hasilnya sesuai dengan tujuan awal. Jika tidak sesuai, maka akan kembali ke tahap awal (*business understanding*) agar hasil yang didapatkan dapat maksimal (sesuai tujuan awal).

6. *Deployment*

Hasil interpretasi pada tahap sebelumnya jika telah sesuai dengan tujuan awal, maka akan disajikan dalam bentuk laporan atau bentuk lainnya untuk pengambilan keputusan.

D. Metode Asosiasi

Metode asosiasi atau yang dikenal sebagai *association rule* merupakan suatu metode pada penambangan data yang mencari hubungan atau keterkaitan antar item pada suatu himpunan data [20]. Keterkaitan antar item tersebut dapat dicari dengan menggunakan aturan berupa himpunan item yang sering muncul (*frequent itemset*). Dalam penambangan data, metode ini memiliki peranan penting dalam membantu memecahkan permasalahan di organisasi atau perusahaan. Pada usaha jual-beli baik barang maupun jasa (ritel), metode asosiasi yang dikenal sebagai *market basket analysis* dapat melihat hubungan antara pelanggan dengan produk yang terjual [21]. Hubungan ini akan membentuk pola yang kemudian jika diolah akan memberikan informasi untuk kepentingan pengambilan keputusan.

Pada metode asosiasi, terdapat beberapa istilah ukuran nilai atau parameter. Berikut penjelasannya:

a) *Support*

Nilai ini merupakan nilai kemungkinan suatu produk atau item dibeli oleh konsumen secara bersamaan.

b) *Confidence*

Nilai kepastian suatu produk akan dibeli secara bersamaan yang dipengaruhi oleh nilai *support*. Sebagai contoh, jika pelanggan membeli roti, maka akan membeli selai.

c) *Lift ratio*

Nilai ini digunakan untuk mengetahui tingkat kekuatan *rule* atau aturan yang terbentuk [22]. Biasanya *lift ratio* digunakan untuk menentukan suatu *rule* valid atau tidak valid. Agar dapat mengetahui nilai *lift ratio*, perlu dicari terlebih dahulu nilai *benchmark confidence*:

$$\text{Benchmark confidence} = \frac{Nc}{N} \quad (3.1)$$

Keterangan:

- Nc = Jumlah transaksi yang mengandung *item* yang menjadi *consequent*
- N = Jumlah transaksi

$$\text{Lift ratio} = \frac{\text{Confidence}}{\text{Benchmark Confidence}} \quad (3.2)$$

Dalam melakukan analisis asosiasi, terdapat 2 tahapan dasar metodologi yang digunakan untuk mencari nilai parameter, yakni sebagai berikut [23]:

a) Analisis pola frekuensi tinggi

Tahapan ini merupakan tahap di mana menemukan kombinasi item untuk *minimum support* dalam basis data. Adapun nilai ini dapat diperoleh dari perhitungan:

$$\text{Support A} = \frac{\sum \text{Transaksi A}}{\sum \text{Transaksi}} \quad \text{untuk 1 item} \quad (3.3)$$

$$\text{Support (A, B)} = \frac{\sum \text{Transaksi A} \cap \text{B}}{\sum \text{Transaksi}} \quad \text{untuk 2 item} \quad (3.4)$$

b) Pembentukan aturan asosiatif

Pada tahap ini, dilakukan pencarian nilai *confidence* berdasarkan nilai *minimum support* dengan aturan “If A then B” yang dihitung berdasarkan

rumus (1) dan (2).

$$Confidence = P(B|A) = \frac{\sum Transaksi A \cap B}{\sum Transaksi A} \quad (3.5)$$

Selain parameter dan tahapan dasar dalam menggunakan metodologi ini, terdapat istilah lain yang akan digunakan, yakni [24]:

a) *Itemset*

Kumpulan dari suatu *item* yang jumlahnya bisa satu maupun lebih, contohnya seperti {Ramen}, {Ramen, Bulgogi}.

b) *k-item*

Menunjukkan banyaknya *item* pada suatu itemset, misalnya {Sabun, Sikat gigi}, itemset ini disebut 2-itemset di mana 2 merupakan nilai dari *k*.

c) *Support count* (σ)

Nilai ini mewakili banyaknya itemset pada suatu himpunan data. Adapun contohnya yaitu jika itemset {buku, pulpen} muncul sebanyak 4 kali, maka nilai dari *support count* adalah 4.

E. Algoritma FP-Growth

FP-Growth adalah algoritma pengembangan dari algoritma apriori yang diperkenalkan oleh Han, Pei, dan Yin pada tahun 2000. Algoritma ini dapat digunakan untuk mencari item atau nilai yang sering muncul atau biasa disebut *frequent itemset* pada himpunan data. Sebagai pengembangan dari algoritma Apriori, ada beberapa hal yang membedakannya. Jika pada algoritma Apriori dibangkitkan semua kandidat yang ada untuk menemukan nilai yang paling sering muncul dan kemudian diuji apakah memenuhi nilai minimum *support*, namun pada algoritma FP-Growth pencarian nilai yang sering muncul dilakukan dengan cara membangun pohon(*tree*) agar dapat menghilangkan calon kandidat dari algoritma apriori [25]. Sehingga dengan menggunakan algoritma FP-Growth maka waktu yang dibutuhkan relatif lebih cepat dan memori yang dibutuhkan lebih sedikit [26].

Penerapan algoritma ini dapat dilakukan dalam beberapa tahapan, berikut tahapan-tahapan tersebut [27]:

a) Pengumpulan data.

Data-data yang dibutuhkan dikumpulkan terlebih dahulu.

- b) Tentukan frekuensi dari setiap item.

Pada masing-masing item, tentukan frekuensinya dan urutkan dari frekuensi tertinggi hingga ke terkecil. Setelah itu, agar penulisannya lebih mudah, maka pada masing-masing item cukupditulis kode label yang sudah ditentukan, misalnya untuk obat amoxilin diwakili dengan huruf 'A'.

- c) Urutkan berdasarkan prioritas.

Urutkan frekuensi pada masing-masing *item* berdasarkan prioritas atau mulai dari nilai tertinggi ke nilai terendah untuk memudahkan pemetaan untuk *minimum support* dan *minimum confidence*.

- d) *Dataset* diurutkan prioritas.

- e) Buatlah FP-Tree berdasarkan urutan transaksi.

- f) Bangkitkan *conditional pattern base*.

Pada tahap ini dilakukan *tracking* terhadap seluruh rute *item*. *Tracking* pada tahap ini didasarkan pada FP-Tree.

- g) Bangkitkan *conditional* FP-Tree.

Kemudian jumlahkan masing-masing nilai pada *item* (nilai *support count*).

- h) Tahap *frequent pattern generated*.

Pada tahap ini, cek terlebih dahulu apakah nilai pada FP-Tree bernilai *singlepath* atau tidak. Jika *single path*, lakukan kombinasi *item* pada setiap kondisi. Namun, jika tidak maka lakukan pembangkitan FP-Growth (secara rekursif).

- i) Hitung nilai *confidence* dengan menggabungkan nilai *itemset* pada tahap (g). Nilai *confidence* yang didapatkan pada tahap ini berdasarkan *single path* yang dikombinasikan dengan nilai *itemset* pada tahap (f).

Berikut contoh perhitungan dengan menggunakan algoritma FP-Growth pada sebuah toko yang memiliki:

- a) Pengumpulan data

Berikut adalah tabular *dataset* pada toko alat tulis yang ditunjukkan pada Tabel 3.1. *Dataset* di bawah ini terdiri atas 10 *item* dan 12 transaksi.

Tabel 3. 1 Tabular *Dataset*

penggaris	buku tulis	buku gambar	pensil	penghapus	pulpen	tip-ex	nota	rautan	spidol
1	1	0	0	0	1	1	0	0	0
1	1	0	1	1	0	0	1	1	1
0	0	0	1	1	0	0	0	0	1
1	0	1	0	0	0	0	0	0	0
0	0	1	1	0	0	1	0	0	0
1	1	0	0	0	1	0	0	0	0
0	0	0	0	1	0	0	0	1	1
0	0	1	1	1	1	1	1	0	0
1	1	0	0	0	0	0	0	1	0
0	0	1	0	0	0	1	0	0	0
1	1	1	0	0	0	0	0	0	0
0	0	0	0	1	1	1	0	0	0

b) Tentukan frekuensi / *support count* setiap *item*

Setelah mengumpulkan data, hitung frekuensi atau *support count* setiap *item* pada *dataset* dan tentukan minimum *support count*. Minimum *support count* yang digunakan yaitu 2. Berikut tabel frekuensi setiap *item* yang ditunjukkan pada Tabel 3.2.

Tabel 3. 2 Tabel Frekuensi Setiap *Item*

ID	Item	Frekuensi
A	penggaris	6
B	buku tulis	5
C	buku gambar	5
D	pensil	4
E	penghapus	5
F	pulpen	4
G	tip-ex	5
H	nota	2
I	rautan	3

J	spidol	3
---	--------	---

c) Urutkan berdasarkan prioritas

Setelah frekuensi setiap *item* dihitung, kemudian *item* tersebut diurutkan berdasarkan *frequent* tertinggi. Berikut Tabel 3.3 yang berisikan *item* barang yang telah diurutkan berdasarkan *frequent* tertinggi.

Tabel 3. 3 Tabel *Item* yang Telah diurutkan Berdasarkan *Frequent* Tertinggi

ID	Item	Frekuensi
A	penggaris	6
B	buku tulis	5
C	buku gambar	5
E	penghapus	5
G	tip-ex	5
D	pensil	4
F	pulpen	4
I	rautan	3
J	spidol	3
H	nota	2

d) *Dataset* diurutkan berdasarkan prioritas

Dataset transaksi penjualan yang ditunjukkan pada Tabel 3.4 merupakan data sampel yang diurutkan berdasarkan prioritas sebelum membangun FP-Tree.

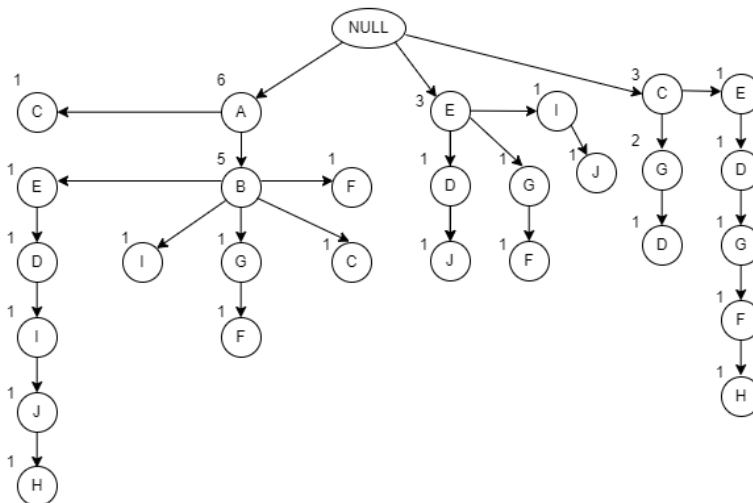
Tabel 3. 4 *Dataset* Transaksi yang Telah diurutkan berdasarkan *Frequent* per-*Item*

TID	ITEM
1	{A,B,G,F}
2	{A,B,E,D,I,J,H}
3	{E,D,J}
4	{A,C,}

5	{C,G,D}
6	{A,B,F}
7	{E,I,J}
8	{C,E,D,G,F,H}
9	{A,B,I}
10	{C,G}
11	{A,B,C}
12	{E,G,F}

e) Membangun *FP-Tree*

Berikut adalah gambar *frequent pattern tree* (FP-Tree) dataset transaksi yang ditunjukkan pada Gambar 3.3.



Gambar 3. 2 *FP-Tree*

f) Bangkitkan *conditional pattern base*

Pembangkitan atau pembuatan *conditional pattern base* dibuat berdasarkan FP-Tree yang berisi *path* (lintasan). Berikut pembuatan *conditional pattern base* yang ditunjukkan pada Tabel 3.5.

Tabel 3. 5 Tabel *Conditional Pattern Base*

Item	Conditional Pattern Base
H	{{A,B,E,D,I,J : 1}, {C,E,D,G,F : 1}}
J	{{A,B,E,D,I : 1}, {E,D : 1}, {E,I : 1}}
I	{{A, B, E, D : 1}, {A, B : 1}, {E, I : 1}}

F	{{A, B, G : 1}, {A, B : 1}, {E, G : 1}, {C, E, D, G : 1}}
D	{{A, B, E : 1}, {E : 1}, {C, G : 1}, {C, E : 1}}
G	{{A, B : 1}, {C : 2}, {C, E, D : 1}, {E : 1}}
E	{{A, B : 1}, {C : 1}}
C	{{A : 1}, {A, B : 1}}
B	{A : 5}
A	-

g) Bangkitkan *Conditional FP-Tree*

Sebelum membangkitkan *conditional FP-Tree*, jumlahkan *support count* setiap *item*. Apabila *support count* lebih besar atau sama dengan *minimum support count*, maka bangkitkan dengan *conditional FP-Tree*. Hasil pembangkitan *conditional FP-Tree* ditunjukkan pada Tabel 3.6.

Tabel 3. 6 Tabel *Conditional FP-Tree*

Item	Conditional FP-Tree
H	{{E:2, D:2}}
J	{{E:3, D:2, I:2}}
I	{{A:2, B:2, E:2}}
F	{{A:2, B:2, G:3, E:2}}
D	{{E:3, C:2}}
G	{{C:3, E:2}}
E	-
C	{{A:2}}
B	{{A:5}}
A	-

h) *Frequent Pattern Generated*

Kombinasikan *single path* dengan *item* yang sudah diketahui. Berikut hasil kombinasi *single path* yang ditunjukkan pada Tabel 3.7.

Tabel 3. 7 Tabel *Frequent Pattern Generated*

Item	Frequent Pattern Generated
H	{{E, H : 2}, {D, H : 2}, {E, D, H : 2}}

J	{{E, J : 3}, {D, J : 2}, {I, J : 2}, {E, D, I, J : 2}}
I	{{A, I : 2}, {B, I : 2}, {E, I : 2}, {A, B, E, I : 2}}
F	{{A, F : 2}, {B, F : 2}, {G, F : 3}, {E, F : 2}, {A, B, G, E, F : 2}}
D	{{E, D : 3}, {C, D : 2}, {E, C, D : 2}}
G	{{C, G : 3}, {E, G : 2}, {C, E, G : 2}}
E	-
C	{{A, C : 2}}
B	{{A, B : 5}}
A	-

i) Menghitung *confidence*

Dalam menghitung *confidence*, sampel yang digunakan yaitu *item* {E, D, H}. Apabila nilai *confidence* tidak memenuhi minimum *confidence*, maka dieliminasi. Nilai minimum *confidence* yang ditetapkan yaitu 85%. Berikut hasil perhitungannya yang ditunjukkan pada Tabel 3.8:

Tabel 3. 8 Tabel Perhitungan *Confidence*

<i>Item</i>	<i>Kemungkinan</i>	<i>Confidence</i>	<i>Lift Ratio</i>
H	$E \wedge D$	$2/2 = 100\%$	$100\% / (3/12 \times 100\%) = 4$
$E \wedge H$	$D \wedge H$	$2/2 = 100\%$	$100\% / (2/12 \times 100\%) = 6$
$D \wedge H$	E	$2/2 = 100\%$	$100\% / (5/12 \times 100\%) = 2.4$

F. Aplikasi RapidMiner

RapidMiner merupakan aplikasi yang digunakan untuk keperluan *data science* yang bersifat *open source*. Aplikasi ini dikembangkan pada tahun 2001 oleh Ralf Klinkenberg, Simon Fischer, dan Ingo Mierswa di Unit Kecerdasan Buatan, Universitas Teknik Dortmund. RapidMiner dimanfaatkan dalam pendidikan, penelitian, pembuatan prototipe cepat, penelitian, aplikasi industri, dan pengembangan penelitian. Sebagai aplikasi yang dapat digunakan untuk *data science*, RapidMiner mampu membuat model yang cepat dan prediktif. Aplikasi ini dapat menambang data, teks, hingga analitik prediktif [28].

Data yang digunakan untuk keperluan *data science* dapat diolah pada aplikasi ini, seperti pengolahan data mentah. Pada pengolahan data mentah, hal-hal yang dapat dilakukan seperti mengisi nilai yang hilang, menghasilkan data *noise*, dan lain-lain. Kemudian, RapidMiner akan menganalisisnya secara otomatis walaupun data yang dimasukkan dalam skala yang besar. RapidMiner telah digunakan dalam berbagai penelitian, termasuk penentuan jumlah *cluster* optimal. Dalam melakukan penelitian tersebut, perlu diatur parameter-parameter yang dibutuhkan agar ketika membandingkan hasil uji coba secara manual dan menggunakan RapidMiner, hasil yang didapatkan dapat sama.

