

PROPOSAL PROYEK

12S4054– DATA MINING

**Market Basket Analysis Using FP-Growth
for Product Recommendation**



Disusun oleh:

Melani Basaria Pakpahan	12S17064
Juanda Antonius Pakpahan	12S17058
Yeni Chintya Panjaitan	12S17040

**PROGRAM STUDI SARJANA SISTEM INFORMASI
FAKULTAS INFORMATIKA DAN TEKNIK ELEKTRO
INSTITUT TEKNOLOGI DEL
NOVEMBER 2020**

DAFTAR ISI

DAFTAR GAMBAR	ii
DAFTAR TABEL.....	iii
Bab 1. Business Understanding	1
1.1 Determine Business Objectives	1
1.2 Situation Assessment	2
1.3 Determine Data Mining Goal.....	2
1.4 Produce Project Plan.....	2
Bab 2. Data Understanding	4
2.1 Collect Initial Data.....	4
2.2 Describe Data.....	4
2.3 Explore Data	5
2.4 Verify Data Quality.....	6
Bab 3. Data Preparation	7
3.1 Data Set.....	7
3.2 Select Data	7
3.3 Clean Data.....	7
3.4 Transform Data	8
Bab 4. Modeling.....	9
4.1 Select Modeling Technique	9
4.2 Generate Test Design	11
4.3 Build Model	11
4.4 Asses Model.....	12
Bab 5. Evaluation.....	13
5.1 Evaluate Result	13
5.2 Review Process	14
Bab 6. Deployment	15
6.1 Plan Deployment.....	15
6.2 Plan Monitoring and Maintenance	15
Bab 7. Penutup	16
7.1 Kesimpulan	16
7.2 Saran	16
REFERENSI	17

DAFTAR GAMBAR

Gambar 1. Data Describe	5
Gambar 2. Korelasi pada data	5
Gambar 3. Sampel data yang digunakan.....	6
Gambar 4. Deskripsi jumlah baris dan kolom.....	7
Gambar 5. Deskripsi nilai null pada kolom	8
Gambar 6.Deskripsi missing value pada data	8
Gambar 7.Implementasi library re	8
Gambar 8. Inisiasi fp-growth	9
Gambar 9.Inisiasi pembentukan tree.....	10
Gambar 10. Update Tree pada Fp-growth.....	10
Gambar 11. Script UpdateHeader	11
Gambar 12.Output model.....	14

DAFTAR TABEL

Tabel 1. Deskripsi dataset	4
Tabel 2. Unique value pada data	6

Bab 1. Business Understanding

Pada fase ini akan berfokus pada pemahaman akan tujuan dan kebutuhan berdasarkan penilaian bisnis. Sehingga pemahaman ini nantinya akan diubah menjadi rencana awal untuk mencapai tujuan [1].

1.1 Determine Business Objectives

Mengelola persediaan produk dalam sebuah bisnis adalah hal yang sangat dibutuhkan. Permintaan pelanggan yang berubah secara dinamis adalah masalah operasional *supermarket XYZ* yang sering terjadi. Perubahan tersebut sering membuat *supermarket* mengalami kesulitan dalam pengambilan keputusan terkait penyediaan stok suatu produk. Ketika *supermarket* menyiapkan pengadaan semua produk dengan jumlah yang sama, maka *supermarket* akan mengalami kerugian. Kerugian karena tidak sesuaiannya permintaan akan sebuah produk dengan persediaannya. Akibat lain yang juga ditimbulkan yaitu produk yang kurang diminati atau produk dengan frekuensi permintaan yang rendah tidak akan habis terjual dan jika terlalu lama akan memasuki masa kadaluarsa. Tingginya pemintaan pelanggan terhadap persediaan yang sedikit akan menghasilkan kekecewaan dan tidak menutup kemungkinan pelanggan tidak kembali datang untuk membeli produk.

Salah satu cara yang dapat dilakukan untuk mengetahui kondisi pasar adalah dengan mengamati transaksi penjualan produk *supermarket*. Data transaksi penjualan yang disimpan dalam basis data dapat diolah lebih lanjut sehingga didapatkan informasi baru yang membantu perusahaan dalam mengambil keputusan terkait persediaan atau *stock* produk. *History* transaksi penjualan produk dapat digunakan untuk mengetahui frekuensi pembelian setiap produk.

Pada proyek ini, *market basket analysis* dilakukan dengan algoritma *fp-growth* untuk mengetahui hubungan atau pola-pola yang dihasilkan berdasarkan produk yang dibeli oleh pelanggan. *Market Basket Analysis* adalah solusi bagi *supermarket* untuk mengambil keputusan dalam menentukan strategi penjualan serta menyiapkan *stock* produk dengan tepat berdasarkan pola pembelian pelanggan. Hal tersebut dapat meningkatkan pelayanan, kepercayaan pelanggan terhadap

supermarket dan meminimalisir kerugian. Dari hal tersebut *supermarket* mendapatkan informasi terkait produk apa saja yang memiliki frekuensi pembelian yang tinggi maupun yang rendah, sehingga *supermarket* dapat dengan mudah mempersiapkan *stock* produk dengan efisien tanpa harus takut mengalami kerugian yang besar.

1.2 Situation Assessment

Supermarket XYZ merupakan toko yang menjual kebutuhan baik berupa makanan, minuman, dan lain sebagainya. Dalam pengoperasiannya toko ini masih menggunakan cara tradisional dalam penempatan tata letak produk dan penyediaan produk sehingga strategi pemasaran yang dilakukan oleh supermarket XYZ belum maksimal. Untuk meningkatkan strategi pemasaran yang tepat dapat digunakan history data penjualan pada supermarket XYZ untuk mengetahui hubungan atau pola-pola yang dihasilkan berdasarkan transaksi yang dilakukan oleh konsumen. Atribut *history* data yang dapat digunakan yaitu *order_id*, *user_id*, *order_number*, *order_dow*, *order_hour_of_day*, *days_since_prior_order*, *products*. Dengan mengetahui pola-pola yang dihasilkan dari transaksi produk yang terjual maka supermarket XYZ dapat mengetahui manakah produk yang sering dibeli oleh konsumen agar supermarket XYZ bisa lebih fokus pada produk tersebut baik dari segi tata letak dan penyediaanya.

1.3 Determine Data Mining Goal

Tujuan penerapan Data mining dalam penggeraan proyek ini adalah sebagai berikut:

1. Menghasilkan model sistem rekomendasi sebuah produk berdasarkan transaksi data dan probabilitas dari produk.
2. Menerapkan algoritma *fp-growth* dalam melakukan pemodelan sistem rekomendasi produk yang akan dibangun.

1.4 Produce Project Plan

Pada penggeraan proyek ini, algoritma *fp-growth* digunakan untuk menghasilkan sistem rekomendasi produk. Sebelum penggunaan algoritma tersebut, pertama yang perlu dilakukan yaitu menganalisis data transaksi penjualan *Supermarket XYZ*. Analisis data yang akan dilakukan yaitu pengecekan atribut dan tipe atribut yang terdapat pada dataset, selanjutnya akan dilakukan *data preparation* dengan

menghapus atribut yang tidak dibutuhkan untuk penggerjaan proyek. Data yang telah dianalisis dan telah dipilih atribut yang diperlukan akan dilakukan proses *market basket analysis* dengan menggunakan algoritma *fp-growth* untuk menganalisa pola-pola yang dihasilkan dari transaksi. Algoritma *fp-growth* dapat digunakan untuk menentukan himpunan data yang paling sering muncul (*frequent itemset*) pada kumpulan data. Selain itu, algoritma *fp-growth* menggunakan konsep struktur *tree* dalam pencarian *frequent itemset*. Karakteristik algoritma *fp-growth* merupakan struktur data menggunakan *tree* yang disebut dengan *fp-tree*. Penggunaan *fp-tree* membuat algoritma *fp-growth* dapat langsung mengekstrak *frequent itemset* dari *fp-tree*.

Hasil dari proses *market basket analysis* menggunakan *fp-growth* diharapkan menghasilkan informasi dan pengetahuan berdasarkan rekomendasi yang dapat digunakan dalam pemenuhan stok produk dan tata letak produk. Selain itu, diharapkan juga dapat memberikan pengetahuan mengenai karakteristik dari *customer* berdasarkan transaksi yang telah dilakukan sebelumnya untuk membuat keputusan yang tepat.

Bab 2. Data Understanding

Pada fase ini akan dilakukan pengumpulan data yang akan digunakan kemudian akan dilanjutkan dengan pengenalan data serta mengidentifikasi masalah kualitas data yang digunakan [1].

2.1 Collect Initial Data

Data yang digunakan tersedia dalam format *csv* dan diambil dari BigML yang merupakan portal penyedia *dataset* yang dikembangkan oleh BigML.Inc. Dataset yang digunakan dapat diakses pada *link* berikut ini:

<https://bigml.com/user/czuriaga/gallery/dataset/5a7a2e4392fb563c2d000cef>

2.2 Describe Data

Data yang digunakan dalam membangun sistem rekomendasi produk menggunakan algoritma *fp-growth* merupakan kumpulan data transaksi pembelian suatu produk. Data transaksi pembelian ini terdiri dari 131.209 ribu baris data untuk observasi dan memiliki 7 *feature* atau atribut. Berikut adalah penjelasan tiap atribut yang terdapat di dalam data yang digunakan:

Tabel 1. Deskripsi dataset

No.	Name	Dtype	Description
1.	<i>Order_id</i>	int64	ID atas sebuah pesanan yang telah dilakukan
2.	<i>User_id</i>	int64	ID atas pengguna atau <i>user</i> yang telah melakukan pemesanan
3.	<i>Order_number</i>	int64	Jumlah dari produk yang dipesan di dalam transaksi
4.	<i>Order_dow</i>	int64	Hari dalam seminggu dimana pemesanan/transaksi terjadi
5.	<i>Order_hour_of_day</i>	int64	Jam saat pemesanan atau transaksi terjadi
6.	<i>Days_since_prior_order</i>	int64	Jumlah hari sejak pemesanan atau transaksi sebelumnya
7.	<i>Products</i>	object	Produk yang ada didalam transaksi atau pemesanan

Berikut akan kita lihat deskripsi dari data transaksi yang digunakan dalam melakukan analisis basket.

	order_id	user_id	order_number	order_dow	order_hour_of_day	days_since_prior_order
count	1.312090e+05	131209.000000	131209.000000	131209.000000	131209.000000	131209.000000
mean	1.704050e+06	103166.825317	16.603937	2.768431	13.581363	17.049859
std	9.889077e+05	59565.307804	16.661077	2.121701	4.221325	10.676832
min	1.000000e+00	1.000000	4.000000	0.000000	0.000000	0.000000
25%	8.451640e+05	51587.000000	6.000000	1.000000	10.000000	7.000000
50%	1.696847e+06	103150.000000	10.000000	3.000000	14.000000	15.000000
75%	2.561480e+06	154868.000000	20.000000	5.000000	17.000000	30.000000
max	3.421070e+06	206209.000000	100.000000	6.000000	23.000000	30.000000

Gambar 1. Data Describe

Dari tabel diatas dapat kita lihat distribusi dari setiap data yang ada pada tiap kolom. Hal ini akan membantu kita dalam mendeteksi masalah yang mungkin terjadi seperti adanya *outliers* atau penyimpangan yang besar pada data transaksi yang kita gunakan. Pada tabel diatas juga menampilkan nilai rata-rata dari setiap kolom, standar deviasi setiap kolom, nilai *max* dan *min*, serta menampilkan nilai dari masing- masing kuartil.

2.3 Explore Data

Data transaksi yang memiliki atribut yang terdiri dari ratusan transaksi produk dikumpulkan melalui metode penelitian data. Hubungan data juga akan dijelaskan pada tahap eksplorasi ini, sebelumnya pada deskripsi data sudah dijelaskan mengenai keterangan dari setiap atribut. Berikut adalah korelasi untuk mengetahui keterkaitan antara variabel dalam data transaksi yang digunakan.

	order_id	user_id	order_number	order_dow	order_hour_of_day	days_since_prior_order
order_id	1.000000	-0.002231	0.000409	0.002272	-0.001267	0.001000
user_id	-0.002231	1.000000	-0.004775	-0.000987	-0.000091	0.002088
order_number	0.000409	-0.004775	1.000000	0.010221	-0.021343	-0.275802
order_dow	0.002272	-0.000987	0.010221	1.000000	0.003368	-0.019117
order_hour_of_day	-0.001267	-0.000091	-0.021343	0.003368	1.000000	-0.002162
days_since_prior_order	0.001000	0.002088	-0.275802	-0.019117	-0.002162	1.000000

Gambar 2. Korelasi pada data

Berdasarkan korelasi yang ditunjukkan diatas apabila nilai korelasi yang didapat kurang dari 0,4 maka tidak ada hubungan atau korelasi antara variabel tersebut. Selanjutnya berikut adalah jumlah *unique value* pada tiap kolom data yang digunakan.

Tabel 2. Unique value pada data

Column	Total Unique Value
<i>Order_id</i>	131.209
<i>User_id</i>	131.209
<i>Order_number</i>	97
<i>Order_dow</i>	7
<i>Order_hour_of_day</i>	24
<i>Days_since_prior_order</i>	31
<i>Products</i>	126.954

2.4 Verify Data Quality

Data transaksi yang digunakan untuk melakukan analisis *market basket* ini memiliki kualitas yang baik, yang mana pada data yang digunakan tidak berisi nilai *null*. Sehingga akan memudahkan dalam melakukan analisis terhadap tiap transaksi yang dilakukan.

	order_id	user_id	order_number	order_dow	order_hour_of_day	days_since_prior_order	products
0	1	112108	4	4	10	9	Bulgarian Yogurt Organic 4% Milk Fat Whole Mil...
1	36	79431	23	6	18	30	Grated Pecorino Romano Cheese Spring Water Org...
2	38	42756	6	6	16	24	Shelled Pistachios Organic Biologique Limes Or...
3	96	17227	7	6	20	30	Roasted Turkey Organic Cucumber Organic Grape ...
4	98	56463	41	3	8	14	Natural Spring Water Organic Orange Juice With...
5	112	125030	5	5	14	26	Fresh Cauliflower Heart Baby Kale Sea Salt B...
6	170	182389	7	0	13	14	Asian Chopped Salad with Dressing XL Emerald W...
7	218	98711	12	0	21	17	Natural Artisan Water Okra Organic Yellow Peac...
8	226	51011	4	0	12	30	Clementines, Bag Banana Kids! Chewables ProBio...
9	349	156353	9	3	16	30	Pure Irish Butter Pure Almond Unsweetened Orig...

Gambar 3. Sampel data yang digunakan

Jika dilihat dari data sampel yang ditunjukan, format dari isi data pada setiap kolom memiliki format data yang sama. Dalam melakukan analisis *market basket* pada penggerjaan proyek membutuhkan data *products*. Sehingga akan digunakan algoritma *fp-growth* untuk menemukan kumpulan item yang sering muncul pada transaksi produk.

Bab 3. Data Preparation

Pada fase ini akan dilakukan persiapan data untuk keseluruhan fase berikutnya. Ada beberapa hal yang akan dilakukan antara lain, deskripsi data set, memilih data, membangun data, dan membersihkan data [1].

3.1 Data Set

Data yang digunakan diunduh dari BigML yang menyediakan berbagai dataset untuk *machine learning*. Terdapat 131.209 ribu baris data transaksi dan 7 atribut/*feauter* yang terdapat pada dataset yang akan digunakan dalam penggerjaan proyek. Sehingga data ini akan diproses lebih lanjut untuk mengidentifikasi masalah atau kendala apa yang terdapat dalam data yang akan dipakai.

```
#Melihat jumlah baris  
data.shape  
  
(131209, 7)
```

Gambar 4. Deskripsi jumlah baris dan kolom

3.2 Select Data

Dalam pemilihan data yang digunakan untuk melakukan analisis *market basket* dengan metode *association* tidak menggunakan keseluruhan data yang ada untuk menghindari waktu komputasi yang lama. Untuk dapat merekomendasikan produk berdasarkan *history* transaksi yang telah dilakukan pelanggan maka data yang akan digunakan adalah data *product* yang ada pada setiap transaksi yang dilakukan.

3.3 Clean Data

Pada proses ini akan dipastikan apakah data yang dipilih telah layak untuk ditambang. Kegiatan yang akan dilakukan yaitu membersihkan data, menghapus data yang tidak diperlukan serta menyeragamkan isi dari tiap baris data, sehingga data yang akan digunakan siap untuk dimodelkan. Setelah melakukan eksplorasi pada data tidak ditemukan nilai *null* pada kolom data yang akan digunakan, isi data tiap barisnya juga sesuai, dan tidak ada *missing value* pada tiap kolom data. Sehingga data dapat digunakan untuk fase pemodelan.

```

#   Column           Non-Null Count   Dtype  
---  --  
0   order_id        131209 non-null   int64  
1   user_id         131209 non-null   int64  
2   order_number    131209 non-null   int64  
3   order_dow       131209 non-null   int64  
4   order_hour_of_day 131209 non-null   int64  
5   days_since_prior_order 131209 non-null   int64  
6   products        131209 non-null   object  
dtypes: int64(6), object(1)

```

Gambar 5. Deskripsi nilai *null* pada kolom

```

order_id 0
user_id 0
order_number 0
order_dow 0
order_hour_of_day 0
days_since_prior_order 0
products 0

```

Gambar 6. Deskripsi *missing value* pada data

3.4 Transform Data

Transformasi pada data bertujuan untuk mengubah format data sesuai kebutuhan. Dalam mempersiapkan data yang akan digunakan, maka pengembang menggunakan *library re* (*library regex*), untuk menghapus adanya *whitespace* berlebih yang ada pada data.

```

for arr in index:
    for row in range(len(arr)):
        idx = re.sub(r"\s+|\s+$", "", arr[row])
        arr[row] = idx
    items.append(arr)

```

Gambar 7. Implementasi *library re*

Bab 4. Modeling

Pada fase ini akan melibatkan teknik *data mining* yang akan digunakan, menentukan parameter, dan algoritma yang akan digunakan [1].

4.1 Select Modeling Technique

Pada pengerjaan projek *market basket analysis* ini menggunakan algoritma *fp growth*, dimana algoritma yang digunakan diambil dari modul *fp growth* yang telah tersedia pada bahasa pemogramman *python*. Algoritma *fp growth* berfungsi untuk menemukan kumpulan item yang sering muncul tanpa menggunakan generasi kandidat. Pada proses algrotima *fp growth* menggunakan struktur data yaitu *frequent pattern tree* yang menyimpan informasi asosiasi itemset. Berikut adalah potongan kode *fp-growth* untuk menginisiasi pembentukan pohon pada algoritma *fp-growth*

```
class treeNode:
    def __init__(self, nameValue, numOccur, parentNode):
        self.name = nameValue
        self.count = numOccur
        self.nodeLink = None
        self.parent = parentNode
        self.children= {}

    def inc(self, numOccur):
        self.count += numOccur

    def disp(self, ind=1):
        print (' '*ind, self.name, ' ', self.count)
        for child in self.children.values():
            child.disp(ind+1)
```

Gambar 8. Inisiasi fp-growth

```

def createTree(dataSet, minSup):
    headerTable = {}
    for trans in dataSet:
        for item in trans:
            headerTable[item] = headerTable.get(item, 0) + dataSet[trans]
    for k in list(headerTable):
        if headerTable[k] < minSup:
            del (headerTable[k])
    freqItemSet = set(list(headerTable))
    if len(freqItemSet) == 0:
        return None, None
    for k in headerTable:
        headerTable[k] = [headerTable[k], None]
    retTree = treeNode('Null Set', 1, None)
    for tranSet, count in dataSet.items():
        localD = {}
        for item in tranSet:
            if item in freqItemSet:
                localD[item] = headerTable[item][0]
        if len(localD) > 0:
            orderedItems = [v[0] for v in sorted(localD.items(), key=lambda p:p[1], reverse=True)]
            updateTree(orderedItems, retTree, headerTable, count)
    return retTree, headerTable

```

Gambar 9. Inisiasi pembentukan tree

Pada potongan *script* kode *fp-growth* membantu dalam membangun pohon atau *tree* yang akan dihasilkan nantinya. Kita memerlukan *header* → *headerTable* = () untuk mengarahkan ke *instance* pertama. *Header* ini juga akan memungkinkan kita mengakses dengan cepat semua elemen dari tipe tertentu di *FP Tree*.

Pada **createTree (dataset, minsup)**, akan mengambil set data yang digunakan dengan *minimum support* untuk membangun *FP Tree*.

```

def updateTree(items, inTree, headerTable, count):
    if items[0] in inTree.children:
        inTree.children[items[0]].inc(count)
    else:
        inTree.children[items[0]] = treeNode(items[0], count, inTree)
        if headerTable[items[0]][1] == None:
            headerTable[items[0]][1] = inTree.children[items[0]]
        else:
            updateHeader(headerTable[items[0]][1], inTree.children[items[0]])

    if len(items) > 1:
        updateTree(items[1::], inTree.children[items[0]], headerTable, count)

```

Gambar 10. Update Tree pada Fp-growth

Pada potongan *script* kode *fp-growth* diatas akan membantu dalam pembaruan tiap *node* yang ada di dalam tree yang sudah dibangun. **UpdateHeader ()** memastikan setiap *node* yang bertautan menunjuk atau mengarah pada *item* yang ada pada *Tree*

```
def updateHeader(nodeToTest, targetNode):
    while (nodeToTest.nodeLink != None):
        nodeToTest = nodeToTest.nodeLink
    nodeToTest.nodeLink = targetNode
```

Gambar 11. *Script UpdateHeader*

4.2 Generate Test Design

Pada bab ini menjelaskan prosedur atau mekanisme untuk menguji kualitas dan validasi model. Model yang telah dibangun akan dilakukan pengujian atau dilakukan *testing* dengan menggunakan data yang telah ditentukan. Dimana data yang digunakan adalah data yang berisi itemset atau data transaksi pembelian produk oleh pelanggan. Sebelum diolah oleh model *fp growth*, data terlebih dahulu dibersikan dan ditransformasikan ke bentuk yang dapat diolah oleh model yang digunakan.

Untuk data yang digunakan pada tes sebesar 40 persen dan 60 persen untuk train. Pada model *fp growth* ini tidak dilakukan evaluasi sehingga tidak ada pembagian data untuk validasi. Pembagian data tes dan train menggunakan *train test split* pada *sklearn* python dengan parameter *test_size = 0.4*. Penentuan ukuran data tes tersebut bertujuan untuk memproleh data produk yang lebih banyak berdasarkan jumlah kemunculan produk pada suatu transaksi.

4.3 Build Model

a. Parameter Setting

Pada model *fp growth* yang dibangun terdapat parameter atau batasan untuk menghasilkan urutan produk yang sering dibeli oleh konsumen. Parameter tersebut adalah *minimum support*, dimana untuk nilai *minimum support* yang digunakan sebesar 0,01. Penentuan nilai tersebut bertujuan agar daftar produk yang dihasilkan berdasarkan kemunculan tidak terlalu banyak dan tidak terlalu sedikit, sehingga dapat menghasilkan keluaran yang diinginkan.

b. Model

Model yang dibangun untuk memprediksi produk yang sering dibeli oleh konsumen adalah *fp growth*.

c. Model description

Model *fp growth* yang dihasilkan berfungsi untuk menemukan kumpulan item yang sering muncul dengan menghitung kemunculan pada data transaksi. Pada proses algoritma *fp growth* menggunakan struktur data yaitu *frequent pattern tree* yang menyimpan informasi asosiasi itemset. Setiap produk yang tidak memenuhi *minimum support* tidak akan dimasukkan ke dalam daftar produk yang sering dibeli oleh pengguna. Namun, untuk produk memenuhi *minimum support* akan masuk dalam daftar barang yang sering dibeli oleh pengguna.

```
def FpGrowth(dataset, min_support):
    import pandas as pd
    from mlxtend.preprocessing import TransactionEncoder
    from mlxtend.frequent_patterns import fpgrowth

    te = TransactionEncoder()
    te_ary = te.fit(dataset).transform(dataset)
    df = pd.DataFrame(te_ary, columns=te.columns_)
    #fpgrowth(df, min_support)
    data = fpgrowth(df, min_support, use_colnames=True)

    return data
```

Pada *script* di atas digunakan untuk mengimport *library fp-growth* untuk mengekstrak *frequent item set* untuk pengembangan algoritma.

4.4 Asses Model

Hasil dari model *fp growth* yang digunakan adalah daftar produk yang memiliki kemunculan yang tinggi antara satu atau beberapa produk dengan produk lainnya berdasarkan data transaksi. Hasil dari model tersebut tidak dilakukan pertitungan akurasi untuk mengetahui ketepatan perhitungan setiap produk atau antara beberapa produk. Namun, keluaran dari model yang digunakan mampu memberikan daftar produk yang sering dibeli oleh konsumen berdasarkan *minimum support* yang dibuat. Daftar produk atau barang yang dihasilkan dapat digunakan sebagai keputusan dalam pemenuhan stok produk maupun susunan atau tata letak produk dalam toko xyz. Sehingga dapat meningkatkan pembelian barang atau produk oleh konsumen.

Bab 5. Evaluation

Pada tahapan ini akan dilakukan peninjauan terhadap tahapan dan hasil pemodelan yang telah dibangun untuk memastikan bahwa model yang telah dibangun sesuai dengan tujuan yang telah didefinisikan sebelumnya pada *business understanding* [1].

5.1 Evaluate Result

Pada tahap ini akan dilakukan penilaian terhadap sejauh apa hasil dari pemodelan yang telah dilakukan, apakah sudah memenuhi tujuan penggerjaan proyek yang ditentukan pada tahap *business understanding* atau tidak. Tahap evaluasi sebaiknya melibatkan pihak pengguna yang berkaitan dengan pembuatan model, seperti orang-orang pemilik supermarket atau toko. Pada tahap ini, biasanya dilakukan dengan menguji coba model yang telah dibentuk terhadap bisnis untuk mengetahui pemodelan yang telah dibuat dapat membantu para pebisnis atau tidak. Dengan ada pola yang telah terbentuk melalui pemodelan, maka pebisnis akan mengetahui produk apa yang paling banyak diminati dan dapat mengambil keputusan untuk mempersiapkan jumlah suatu barang pada *inventory*. Uji coba yang dilakukan sebagai bentuk evaluasi membutuhkan waktu, karena dibutuhkan data pembelian barang dalam beberapa jangka waktu tertentu untuk dapat menyimpulkan model yang telah dibuat berguna atau tidak. Nilai tertinggi rekomendasi kombinasi produk adalah 1.0 dan yang lainnya berada di bawah 1.0. Melalui persentasi rekomendasi tersebut, kombinasi produk dengan nilai tertinggi akan menjadi pertimbangan. Meskipun akan sulit untuk mengamati hasil kombinasi produk dikarenakan jumlahnya yang sangat banyak. Hal ini juga dapat memakan waktu dalam mengambil keputusan, hanya saja dengan melalui hasil ini orang yang berwenang mengambil keputusan sudah memiliki dasar untuk memutuskan sesuatu berdasarkan data. Tidak menutup kemungkinan juga, permintaan konsumen yang dinamis terus terjadi, sehingga hasil pemodelan ini dapat terus digunakan dengan data terbaru dari Supermarket XYZ.

```
In [30]: model_test_2
Out[30]: {('Honey Gluten Free Granola',): (('Oats',), 1.0),
 ('Oats',): (('Honey Gluten Free Granola',), 0.5494505494505495),
 ('Bean',): (('Cheese Burrito',), 0.6219512195121951),
 ('Cheese Burrito',): (('Bean',), 0.9714285714285714),
 ('Roasted',): (('Salted Almonds',), 0.6077348066298343),
 ('Salted Almonds',): (('Roasted',), 1.0),
 ('Aged White Cheddar Baked Rice',): (('Corn Puffs Gluten Free Lunch Packs',),
 1.0),
 ('Corn Puffs Gluten Free Lunch Packs',): (('Aged White Cheddar Baked Rice',),
 1.0),
 ('Broccoli',): (('Cheddar Bake Meal Bowl',), 0.345454545454546),
 ('Cheddar Bake Meal Bowl',): (('Broccoli',), 1.0),
 ('Orange Calcium',): (('Vitamin D Pulp Free',), 1.0),
 ('Vitamin D Pulp Free',): (('Orange Calcium',), 1.0),
 ('Cucumber',): (('Garlic Tzatziki',), 0.7388535031847133),
 ('Garlic Tzatziki',): (('Cucumber',), 1.0),
 ('No Pulp Calcium',): (('Vitamin D Pure Orange Juice',), 0.556390977443609),
```

Gambar 12.*Output model*

Dari gambar di atas dapat kita lihat bahwa, pasangan produk dengan nilai 1.0 dapat menjadi pertimbangan untuk pemilik bisnis dalam mengambil keputusan terkait penyediaan barang. Sebaliknya dengan nilai yang lebih kecil dari 0.5, barang tersebut akan disediakan secukupnya karena permintaan terhadap barang tersebut tidak sebanyak yang sebelumnya. Terkait keputusan yang akan diambil pemilik bisnis, model akan memberikan rekomendasi produk apa yang harus diperbanyak jumlah persedian nya pada *inventory* melalui nilai yang dihasilkan oleh model.

5.2 Review Process

Pada tahap ini akan dilakukan pemeriksaan kembali pada keseluruhan tahapan, mulai dari awal sampai tahapan akhir untuk memastikan tidak ada hal penting dalam proses tersebut yang terlewati.

Bab 6. Deployment

Pada fase ini akan dijelaskan mengenai rencana *deploy* yang akan dilakukan serta proses monitor dan maintenance sebagai Langkah selanjutnya dalam pemeliharaan sistem yang sudah dibangun [1].

6.1 Plan Deployment

Dalam pengembangan *fp growth* dilakukan beberapa tahapan atau proses untuk memperoleh hasil terbaik. Strategi pertama yang dilakukan adalah mengelola data yang akan digunakan seperti *load data*, *data cleaning*, *exploratory data* dan *data transformation*. *Load data* dilakukan untuk mengambil data yang terdapat pada file CSV untuk digunakan dalam pengembangan sistem *market basket analisis*. *Data cleaning* dilakukan untuk membersikan data yang memiliki nilai yang kosong atau *null* pada salah satu sell dengan manghapus baris sel tersebut. *Exploratory* merupakan metode eksplorasi data dengan menggunakan teknik aritmatika sederhana dan teknik grafis dalam meringkas data pengamatan. Tujuana dari dilakukannya *exploratory data* adalah mengecek tipe atribut, hubungan atau korelasi antara atribut, deskripsi data dan nilai yang unik pada data. *Data transformation* pada strategi ini digunakan untuk membentuk data ke format yang dapat diterima oleh model nantinya. Proses pada transformasi data seperti, menghapus garis lurus yang membedakan produk yang dibeli oleh konsumen pada atribut produk. Terakhir pada tahap transformasi adalah data pada setiap barisnya akan dijadikan *array* dua dimensi.

Setelah tahapan transformasi, data yang sudah dibentuk dalam format yang dibutuhkan akan digunakan pada model untuk menghasilkan daftar produk yang paling banyak dibeli oleh konsumen. Ini merupakan strategi kedua dalam pengembangan *market basket analisis* pada supermarket xyz.

6.2 Plan Monitoring and Maintenance

Pada tahap ini, hal atau strategi yang dilakukan adalah memastikan bahwa setiap proses atau fungus pada sistem dapat berjalan sesuai kebutuhan. Dan melakukan perbaikan pada setiap fungsi yang tidak sesuai dengan bentuk data yang digunakan.

Bab 7. Penutup

Pada bab ini akan berisi kesimpulan dan saran dari penggerjaan proyek “Analisis Market Basket untuk Sistem Rekomendasi Produk dengan Algoritma *FP-Growth*”.

7.1 Kesimpulan

Beberapa kesimpulan yang didapatkan dalam penggerjaan proyek “Analisis Market Basket untuk Sistem Rekomendasi Produk dengan Algoritma *FP-Growth*”, yaitu semakin banyak *frequent itemset* yang ada di dalam data, maka semakin kompleks *Tree* yang akan dibangun, Penelitian *market basket analysis* dengan menggunakan algoritma *fp-growth* terhadap atribut *products* pada transaksi yang ada menghasilkan sejumlah aturan asosiasi yang berbeda- beda.

7.2 Saran

Beberapa saran yang dapat yang dapat diberikan antara lain:

1. Perlu dilakukan penelitian lebih lanjut untuk mengkaji kelemahan pada algoritma *fp-growth* yang mengakibatkan lemahnya aturan asosiasi yang dihasilkan melalui penerapan *market basket analysis* terhadap atribut *products* pada dataset yang digunakan.
2. Penelitian yang berjalan dilakukan dengan menggunakan algoritma *fp-growth*. Hasil penelitian berikutnya dapat diperkaya dengan menambahkan penggunaan algoritma lain. Kombinasi pemakaian algoritma akan menghasilkan informasi yang lebih detail dan bermanfaat dalam pembentukan strategi penjualan produk dalam perusahaan XYZ.

REFERENSI

- [1] P. Chapman and J. Clinton, "CRISP-DM 1.0," 2000. [Online]. Available: <https://the-modeling-agency.com/crisp-dm.pdf>. [Accessed December 2020].