

Implementasi Himpunan, Algoritma Apriori, dan *Decision Tree* dalam Rekomendasi Produk *e-commerce*

Haziq Abiyyu Mahdy - 13521170¹
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13521170@mahasiswa.itb.ac.id

Abstrak—*E-commerce* merupakan salah satu metode jual beli melalui sistem elektronik berbasis aplikasi. Rekomendasi produk yang dipersonalisasi untuk pengguna merupakan hal yang penting bagi perusahaan *e-commerce* guna meningkatkan ketertarikan pengguna untuk berbelanja pada *e-commerce* tersebut. Untuk menciptakan rekomendasi produk, kita dapat mengolah data yang diperoleh dari *web* untuk mencari pelanggan sasaran serta mencari asosiasi antarproduk, sehingga kita dapat memberikan rekomendasi produk yang tepat kepada pengguna. Pada makalah ini, akan dibahas implementasi himpunan, algoritma Apriori, serta pohon keputusan dalam pembuatan rekomendasi produk *e-commerce* serta mekanisme umum pembuatan rekomendasi produk *e-commerce*.

Kata kunci—Himpunan, pohon, rekomendasi, *e-commerce*, *web usage mining*.

I. PENDAHULUAN

Industri *e-commerce* merupakan industri yang tengah berkembang pesat dengan adanya kemajuan teknologi. Dewasa ini, semakin banyak bermunculan *platform* untuk berbelanja online, sehingga persaingan antar-*platform* juga semakin meningkat. Dengan adanya persaingan tersebut, maka perusahaan *e-commerce* membutuhkan strategi untuk menarik pelanggan untuk berbelanja di *platform* milik perusahaan tersebut. Salah satu solusi yang dapat ditawarkan adalah dengan membuat rekomendasi yang dipersonalisasi untuk pengguna. Dengan sistem rekomendasi, aplikasi *e-commerce* dapat meletakkan produk-produk yang diminati oleh pengguna pada beranda aplikasi, sehingga menarik pengguna untuk melakukan pembelian.

Pemilihan strategi/algoritma dalam memberikan rekomendasi pada pengguna merupakan hal yang penting, karena rekomendasi yang buruk dapat berakibat pada dua kemungkinan, yaitu: [1]

- (1) *false negative*, yaitu ketika produk yang diminati pengguna tidak direkomendasikan oleh sistem, serta
- (2) *false positive*, yaitu ketika produk yang direkomendasikan tidak diminati oleh pengguna. *False positive* bisa berakibat fatal, karena dapat menyebabkan hilangnya minat pengguna untuk berbelanja di *e-commerce* tersebut.

Web usage mining dapat menjadi salah satu upaya untuk menemukan rekomendasi yang tepat untuk pengguna. Hal ini disebabkan banyaknya data terkait penggunaan *web* yang dimiliki oleh *e-commerce*, dibandingkan dengan toko *offline*. Salah satu data yang dimiliki oleh *e-commerce* adalah *clickstream*. Data *clickstream* adalah data mengenai halaman *website* yang dikunjungi pengguna serta urutan pengunjungan halaman *website* tersebut. Beberapa metrik yang digunakan pada data *clickstream* adalah sebagai berikut. [2]

- (1) Apakah pengguna tersebut sudah pernah mengunjungi halaman *website* tersebut sebelumnya.
- (2) Halaman mana yang dikunjungi pertama oleh pengguna.
- (3) Waktu yang dihabiskan pengguna saat mengunjungi halaman tertentu.
- (4) Produk yang pernah dibeli oleh pengguna.
- (5) Produk yang dimasukkan atau dikeluarkan oleh pengguna ke keranjang (*shopping cart*).

Setelah mendapatkan data dari *web usage mining*, kita dapat menemukan asosiasi antarproduk, misalnya produk A cenderung dibeli bersamaan dengan produk B, dengan algoritma Apriori serta pelanggan yang berpotensi untuk membeli produk tersebut dengan menggunakan *decision tree*.

II. TEORI DASAR

A. Himpunan

Himpunan (*set*) adalah sekumpulan objek yang berbeda. Objek di dalam himpunan disebut elemen, unsur, atau anggota. Keterurutan elemen di dalam himpunan tidak penting. Keanggotaan dinyatakan dalam notasi \in . $x \in A$ berarti x merupakan anggota himpunan A, dan $x \notin A$ berarti x bukan merupakan anggota himpunan A. Beberapa cara penyajian himpunan adalah dengan enumerasi, yaitu setiap anggota himpunan didaftarkan secara rinci, seperti $A = \{1,2,3,4\}$ dan dengan notasi pembentuk himpunan, seperti $B = \{x \mid x < 5, x \in \mathbb{Z}\}$. Beberapa terminologi mengenai himpunan yang akan digunakan dalam makalah ini adalah sebagai berikut.

1. Himpunan Bagian (*Subset*)

Himpunan A dikatakan himpunan bagian dari himpunan B jika dan hanya jika setiap elemen A merupakan elemen dari B. Himpunan bagian

dinyatakan dalam notasi \subseteq . $A \subseteq B$ berarti A merupakan himpunan bagian dari B.

2. Selisih (*Difference*)

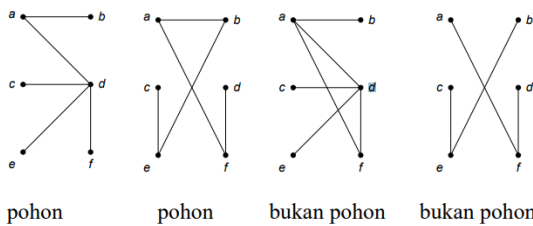
Selisih himpunan A dengan B adalah seluruh anggota himpunan A yang bukan merupakan anggota himpunan B. Selisih himpunan A dengan B dinyatakan dalam notasi $A - B = \{ x \mid x \in A \text{ dan } x \notin B \}$.

3. Himpunan kosong (*Null set*)

Himpunan kosong adalah himpunan dengan jumlah anggota = 0. Himpunan kosong dapat dilambangkan sebagai $\{ \}$ atau \emptyset [3].

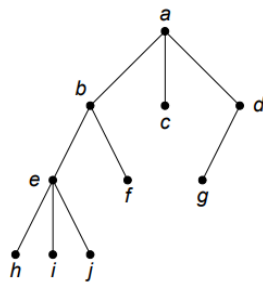
B. Pohon

Pohon adalah graf tak-berarah terhubung yang tidak mengandung sirkuit. Suatu graf G merupakan *tuple* (V, E), dengan V adalah himpunan tidak-kosong dari simpul-simpul (*vertices*) dan E adalah himpunan sisi (*edges*) yang menghubungkan sepasang simpul.



Gambar 2.1. Contoh pohon dan bukan pohon. Sumber: <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Pohon-2020-Bag1.pdf>

Pohon berakar adalah pohon yang satu buah simpulnya diperlakukan sebagai akar dan sisi-sisinya diberi arah sehingga menjadi graf berarah.



Gambar 2.2. Contoh pohon berakar. Sumber: <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2021-2022/Pohon-2021-Bag2.pdf>

Beberapa terminologi pohon berakar yang digunakan pada makalah ini ialah sebagai berikut.

1. Anak (*child/children*) dan orangtua (*parent*)

Pada gambar 2.2, simpul b, c, dan d merupakan anak dari simpul a, atau simpul a merupakan orangtua dari simpul b, c, dan d.

2. Daun (*leaf*)

Daun adalah simpul yang tidak memiliki anak. Contoh daun pada gambar 2.2 adalah simpul f, g, h, i, dan j.

3. Simpul dalam (*internal node*)

Simpul dalam adalah simpul yang memiliki anak. Contoh simpul dalam pada gambar 2.2 adalah simpul b, d, dan e.

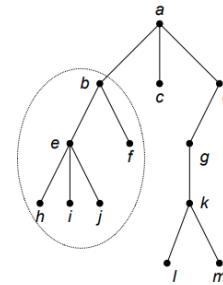
4. Tingkat (*level*)

Tingkatan dari suatu simpul dihitung dari simpul

akar. Pada gambar 2.2, simpul a memiliki tingkat 0, simpul b, c, dan d memiliki tingkat 1, simpul e, f, dan g memiliki tingkat 2, dan seterusnya.

5. Upapohon (*subtree*)

Upapohon adalah salah satu simpul dari suatu pohon dan semua keturunannya [4], [5].



Gambar 2.3. Contoh upapohon. Sumber: <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Pohon-2020-Bag1.pdf>

C. Web Usage Mining

Web usage mining adalah proses penerapan teknik data mining untuk menemukan pola perilaku pengguna berdasarkan data web. Proses Web usage mining secara umum terbagi atas dua bagian; persiapan data (data preparation) dan penemuan pola (pattern discovery). Pada makalah ini, digunakan data dari penggunaan web serta data lainnya yang berkaitan dengan pelanggan [1].

D. Association Rule Mining

Association rule mining merupakan metode untuk menentukan asosiasi (hubungan) antarproduk. Hubungan antarproduk dinyatakan dalam bentuk aturan (rule) berupa $A \Rightarrow B$, yang artinya jika produk A dibeli, maka produk B juga cenderung untuk dibeli. Terdapat tiga metrik yang biasa digunakan untuk menentukan asosiasi produk, yaitu [6]:

1. Support, yaitu popularitas rata-rata suatu produk yang dinyatakan dalam persamaan berikut.

$$support(A) = \frac{freq(A)}{total\ transactions} \quad (1)$$

2. Confidence, yaitu kecenderungan untuk membeli beberapa produk secara bersamaan. Confidence dinyatakan dalam persamaan berikut.

$$confidence(A, B) = \frac{freq(A, B)}{freq(A)} \quad (2)$$

Confidence(A,B) dapat dilihat seperti banyak jumlah total pembelian produk A dan B bersamaan dari seluruh total pembelian produk A.

3. Lift, yaitu peningkatan rasio pembelian suatu produk ketika terdapat pembelian produk lainnya. Lift dapat dinyatakan dalam persamaan berikut.

$$lift = \frac{support(A, B)}{support(A) support(B)} \quad (3)$$

Pada association rule mining, didefinisikan pula frequent itemset (himpunan item yang sering dibeli) sebagai himpunan item yang nilai support nya lebih dari atau sama dengan ambang (threshold). Association rule dapat ditemukan dengan menggunakan algoritma Apriori [7]. Ide dari algoritma tersebut adalah, subset dari suatu frequent itemset haruslah merupakan frequent itemset juga. Dari ide tersebut,

kita dapat memulai membuat *frequent itemset* dengan K anggota, kemudian kita dapat mencari *frequent itemset* dengan K+1 anggota berdasarkan *frequent itemset* sebelumnya. Berikut adalah contoh penerapan algoritma Apriori dalam menentukan *association rule*. Misalkan diketahui *threshold* untuk *frequent itemset* adalah 2 dan terdapat lima transaksi (T1, T2, T3, T3, dan T5) sebagai berikut.

Transaction ID	Items
T1	1, 3, 4
T2	2, 3, 5
T3	1, 2, 3, 5
T4	2, 5
T5	1, 3, 5

Tabel 2.1. Contoh transaksi yang digunakan untuk mencari *association rule*.
Sumber: dokumen pribadi. Contoh kasus dan tabel diadaptasi dari video berikut.
<https://www.youtube.com/watch?v=guVvtZ7ZClw>

Pada iterasi pertama, kita dapat menentukan *frequent itemset* dengan 1 anggota.

Itemset	Support
{1}	3
{2}	3
{3}	4
{4}	1
{5}	4

Tabel 2.2. Iterasi pertama pada algoritma Apriori. Sumber: dokumen pribadi

Karena *itemset* {4} memiliki nilai *support* di bawah *threshold*, kita dapat mengeliminasi *itemset* tersebut dari *frequent itemset*. Pada iterasi kedua, kita dapat menentukan *frequent itemset* dengan 2 anggota berdasarkan *frequent itemset* sebelumnya.

Itemset	Support
{1,2}	1
{1,3}	3
{1,5}	2
{2,3}	2
{2,5}	3
{3,5}	3

Tabel 2.3. Iterasi kedua pada algoritma Apriori. Sumber: dokumen pribadi

Karena *itemset* {1,2} memiliki nilai *support* di bawah *threshold*, kita dapat mengeliminasi *itemset* tersebut dari *frequent itemset*. Pada iterasi ketiga, kita dapat menentukan *frequent itemset* dengan 3 anggota berdasarkan *frequent itemset* sebelumnya. *Itemset* yang memungkinkan adalah {1,2,3}, {1,2,5}, {1,3,5}, dan {2,3,5}. Namun, karena *itemset* {1,2,3} dan {1,2,5} memiliki subset yang bukan merupakan *frequent itemset*, yaitu {1,2}, maka *itemset* {1,2,3} dan {1,2,5} dapat dieliminasi dari *frequent itemset* sebelum melanjutkan iterasi ketiga.

Itemset	Support
{1,3,5}	2
{2,3,5}	2

Tabel 2.3. Iterasi ketiga pada algoritma Apriori. Sumber: dokumen pribadi

Karena kedua *itemset* memiliki nilai *support* sama dengan *threshold*, kita dapat melanjutkan ke iterasi keempat.

Itemset	Support
{1,2,3,5}	1

Tabel 2.4. Iterasi keempat pada algoritma Apriori. Sumber: dokumen pribadi

Karena seluruh *itemset* pada iterasi keempat memiliki nilai

support di bawah *threshold*, maka kita dapat menghentikan iterasi dan kembali ke *frequent itemset* pada iterasi sebelumnya, yaitu {1,3,5} dan {2,3,5}. Untuk $I = \{1,3,5\}$, terdapat subset berupa {1,3}, {1,5}, {3,5}, {1}, {3}, dan {5}. Untuk $I = \{2,3,5\}$, terdapat subset berupa {2,3}, {2,5}, {3,5}, {2}, {3}, dan {5}. Berdasarkan algoritma Apriori, untuk setiap *subset* S dari I, dapat ditentukan *rule* berupa $S \Rightarrow I - S$ jika

$$\frac{\text{support}(I)}{\text{support}(S)} \leq \text{minimum confidence value} \quad (4)$$

Misalkan minimum confidence value = 65%. Maka, dapat diperoleh *association rule* sebagai berikut.

1. $I = \{1,3,5\}$

S	I - S	$\text{support}(I)/\text{support}(S)$	$\geq 65\%$
{1,3}	{5}	66.6%	T
{1,5}	{3}	100%	T
{3,5}	{1}	66.6%	T
{1}	{3,5}	66.6%	T
{3}	{1,5}	50%	F
{5}	{1,3}	50%	F

Tabel 2.5. Penentuan *association rule* pada algoritma Apriori. Sumber: dokumen pribadi

2. $I = \{2,3,5\}$

S	I - S	$\text{support}(I)/\text{support}(S)$	$\geq 65\%$
{2,3}	{5}	100%	T
{2,5}	{3}	66.6%	T
{3,5}	{2}	66.6%	T
{2}	{3,5}	66.6%	T
{3}	{2,5}	66.6%	F
{5}	{3,2}	50%	F

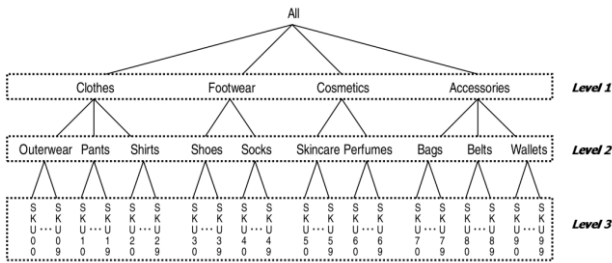
Tabel 2.6. Penentuan *association rule* pada algoritma Apriori. Sumber: dokumen pribadi

Berdasarkan tabel di atas, dapat diperoleh *associative rule* sebagai berikut.

- $1 \ \& \ 3 \Rightarrow 5$
- $1 \ \& \ 5 \Rightarrow 3$
- $3 \ \& \ 5 \Rightarrow 1$
- $1 \Rightarrow 3 \ \& \ 5$
- $2 \ \& \ 3 \Rightarrow 5$
- $2 \ \& \ 5 \Rightarrow 3$
- $3 \ \& \ 5 \Rightarrow 2$
- $2 \Rightarrow 3 \ \& \ 5$
- $3 \Rightarrow 2 \ \& \ 5$

E. Taksonomi Produk (*Product Taxonomy*)

Pengelompokan produk dapat direpresentasikan dengan struktur data pohon (tree) yang mengklasifikasikan produk dari tingkat tinggi ke tingkat rendah. Tiap-tiap simpul daun merupakan contoh (instance) dan kode dari produk dan tiap simpul dalam merepresentasikan kelas dari simpul-simpul keturunannya. Simpul akar ditandai sebagai 'semua', yang merupakan kelas paling umum. Tiap simpul dapat diberi tingkatan sesuai pemberian tingkatan pada pohon berakar standar, yaitu akar sebagai tingkat 0, kemudian anaknya sebagai tingkat 1, dan seterusnya.



Gambar 2.4. Contoh *product taxonomy*. Sumber: <https://www.sciencedirect.com/science/article/pii/S0957417402000520>

Pada penerapannya, kebanyakan *association rules* yang kuat ditemukan pada tingkat yang lebih tinggi pada taksonomi produk, namun asosiasi tersebut biasanya juga merupakan hal yang umum diketahui, misalnya ‘70% orang yang membeli sepatu juga membeli kaus kaki’. Asosiasi pada tingkat yang lebih rendah, seperti ‘40% orang yang membeli kemeja juga membeli sepatu’, merupakan asosiasi yang ‘menarik’, namun sulit ditemukan. Oleh karena itu, penting untuk mencari *association rule* pada level yang tepat pada taksonomi produk [1].

F. Decision Tree Induction

Klasifikasi pada *data mining* adalah melabeli atau mengkategorikan sejumlah kasus pada basis data menjadi beberapa kelas berdasarkan model klasifikasi. Dalam hal ini, suatu *model set*, yaitu sekumpulan kasus yang label kelasnya sudah diketahui, dianalisis dan suatu model klasifikasi dapat dibangun berdasarkan data-data yang ada pada *model set*. Kemudian, model klasifikasi tersebut digunakan untuk mengelompokkan suatu *score set*, yaitu sekumpulan kasus yang belum diketahui label kelasnya.

Metode klasifikasi yang paling populer adalah pohon keputusan (*decision tree*). Pohon keputusan adalah pohon di mana tiap simpul dalam / simpul non-daun merupakan suatu pengujian pada atribut kasus, dan tiap cabang menunjukkan hasil pengujian, dan tiap daun menunjukkan prediksi kelas.

Sebuah *model set* terdiri atas berbagai atribut sebagai variabel independen dan suatu label kelas yang diketahui yang berkaitan dengan atribut tersebut, yang disebut sebagai variabel dependen. Variabel independen direpresentasikan sebagai vektor \mathbf{x} yang berisi nilai atribut $\mathbf{x} = (x_1, x_2, \dots, x_i)$ dan variabel independen direpresentasikan sebagai y . Sebuah *model set* dapat ditunjukkan oleh $\mathbf{M} = \{(\mathbf{x}_m, y_m)\}$, dengan $\mathbf{x}_m \in \mathbf{X}$ (seluruh atribut), $y_m \in C$ (seluruh kasus yang mungkin), dan $m = 1, 2, \dots, M$ (ukuran *model set*).

Suatu *score set* juga memiliki berbagai atribut sebagai variabel independen (\mathbf{x}_s) dan suatu label kelas yang belum diketahui yang berkaitan dengan atribut tersebut (y_s), yang disebut sebagai variabel independen. Sebuah *score set* dapat ditunjukkan oleh $\mathbf{S} = \{(\mathbf{x}_s, y_s)\}$, dengan $\mathbf{x}_s \in \mathbf{X}$, $y_s \in \emptyset$ karena nilai tersebut belum diketahui, dan $s = 1, 2, \dots, S$ (ukuran *score set*) [1].

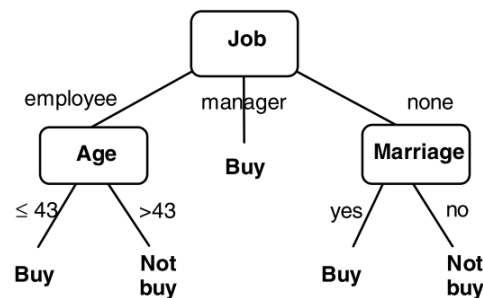
Pada intinya, kita dapat menentukan suatu *model set*, kemudian membuat pohon keputusan dari *model set* tersebut, kemudian kita dapat memprediksi nilai variabel dependen yang ada di *score set*. Kualitas dari pohon keputusan akan bergantung pada akurasi klasifikasi dan ukuran dari pohon tersebut. Proses menghasilkan pohon keputusan dari suatu *model set* disebut

dengan *decision tree induction*.

Berikut adalah contoh *model set* serta pohon keputusan yang dihasilkan.

Customer ID	Job	Age	Marriage	Buy
1	none	>43	N	N
2	employee	<=43	Y	Y
3	manager	<=43	Y	Y
4	manager	>43	N	Y
5	employee	>43	N	N
6	none	>43	Y	Y
7	employee	<=43	N	Y
8	none	<=43	Y	Y
9	employee	>43	Y	N
10	manager	>43	Y	Y
11	manager	<=43	N	Y
12	none	<=43	N	N
13	none	<=44	N	N
14	none	<=45	N	N
Atribut data sebagai \mathbf{x}			variabel dependen (y)	

Tabel 2.7. Contoh *model set*. Sumber: dokumen pribadi



Gambar 2.5. Pohon keputusan yang dihasilkan dari *model set*. Sumber: <https://www.sciencedirect.com/science/article/pii/S0957417402000520>

Setelah didapatkan pohon keputusan dari *model set* yang ada, kita dapat memprediksi apakah pengguna yang ada di *score set* dapat direkomendasikan produk tertentu berdasarkan pohon keputusan tersebut.

Customer ID	Job	Age	Marriage	Buy
210	Manager	>= 43	N	...
211	None	>=43	Y	...
...

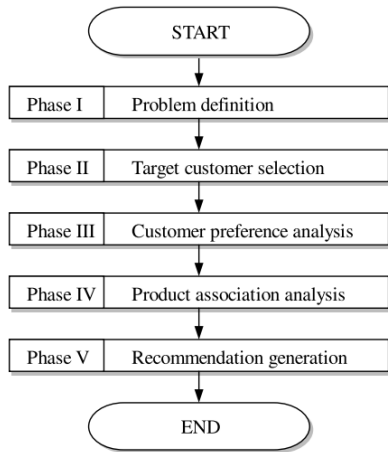
Tabel 2.7. Contoh ilustrasi *score set* yang akan diprediksi variabel dependennya. Sumber: dokumen pribadi

Terdapat beberapa algoritma *decision tree induction* terkenal yang dapat digunakan, di antaranya CHAID (Kass, 1980), CART (Beiman, Friedman, Olshen, & Stone, 1984), C4.5 (Quinlan, 1993) dan QUEST (Loh & Shih, 1997), dan sebagainya. Namun, algoritma yang digunakan tidak akan dibahas lebih lanjut pada makalah ini.

III. MEKANISME REKOMENDASI PRODUK *E-COMMERCE*

Pada makalah ini, mekanisme rekomendasi produk *e-commerce* akan terbagi menjadi lima bagian, yaitu identifikasi masalah, pemilihan pelanggan sasaran (target customer), analisis preferensi pelanggan, analisis asosiasi produk, dan

pembuatan rekomendasi produk.



Tabel 3.1. Diagram alur mekanisme rekomendasi produk *e-commerce*. Sumber: <https://www.sciencedirect.com/science/article/pii/S0957417402000520>

A. Identifikasi Masalah

Untuk memberikan gambaran yang jelas mengenai cakupan yang akan dibahas pada makalah ini, kita perlu mengidentifikasi masalah terlebih dahulu. Secara umum, pada permasalahan rekomendasi produk *e-commerce*, terdapat tiga pertanyaan yang harus dipertimbangkan.

1. Apakah pelanggan sasaran merupakan pelanggan yang selektif atau terbuka.
2. Apakah proses rekomendasi bertujuan untuk memprediksi seberapa tertarik seorang pelanggan terhadap suatu produk (*prediction problem*) atau untuk memprediksi list produk yang mungkin disukai oleh pelanggan (*top-N recommendation problem*).
3. Apakah rekomendasi dibuat pada waktu tertentu atau secara terus-menerus.

Pada makalah ini, yang menjadi cakupan masalah yang akan dibahas adalah pelanggan selektif, top-N recommendation problem, serta rekomendasi pada waktu tertentu. Pada permasalahan ini, kita dapat mendefinisikan $Rec(l, n, p, t)$ sebagai rekomendasi n produk dengan kelas *level-l* pada taksonomi, kepada pelanggan yang telah membeli p atau lebih produk dengan kelas *level-l*, dan rekomendasi diberikan pada waktu t [1].

B. Pemilihan pelanggan sasaran

Pemilihan pelanggan sasaran yang tepat merupakan hal yang penting untuk menghindari terjadinya *false positive* pada rekomendasi produk. Oleh karena itu, kita sebaiknya memberikan rekomendasi produk hanya kepada pelanggan yang berkemungkinan tertarik pada produk tersebut. Untuk menyelesaikan permasalahan $Rec(l, n, p, t)$, dibutuhkan suatu *model set* dan *score set*. Dari *model set*, kita dapat melakukan *decision tree induction*, kemudian memperoleh nilai variabel dependen pada *score set*.

Kita dapat mendefinisikan *model set* \mathbf{M} sebagai himpunan berikut:

$$\mathbf{M} = \{(\mathbf{x}_m, y_m)\},$$

$$m = 1, 2, \dots, M \text{ (ukuran model set),}$$

$$y_m = 1 \text{ jika pelanggan telah membeli produk dengan kelas level-}l \text{ pada range waktu yang ditentukan,}$$

$y_m = 0$ jika tidak.

CID	Age	Gender	Job	Purchase amount	Purchase frequency	Last visit	...	Y
101	22	M	Student	64	4	0827	...	0
103	36	F	Employee	57	6	1018	...	1
104	23	F	None	128	10	1104	...	1
...

Tabel 3.1. Contoh ilustrasi *model set*. Sumber:

<https://www.sciencedirect.com/science/article/pii/S0957417402000520>

Setelah menentukan *model set*, kita dapat melakukan *decision tree induction*. Kemudian, kita dapat memilih *score set* dan menentukan nilai y (variabel dependen) pada *score set* tersebut berdasarkan pohon keputusan yang sudah dibuat. Pelanggan yang memiliki nilai $y = 1$ akan menjadi pelanggan sasaran dan fase selanjutnya hanya akan dilakukan pada pelanggan tersebut [1].

C. Analisis preferensi pelanggan

Setelah mendapatkan pelanggan sasaran, kita dapat melakukan analisis terhadap preferensi pelanggan menggunakan data *clickstream*. Terdapat tiga metrik yang dapat menjadi indikasi preferensi pelanggan terhadap suatu produk, berdasarkan tiga langkah berbelanja online [8]:

1. *click-through*, yaitu seberapa sering pengguna mengklik tautan dari suatu produk pada *e-commerce*,
2. *basket placement*, yaitu apakah pengguna menaruh produk ke keranjang (*shopping cart*), dan
3. *purchase*, yaitu apakah pengguna menyelesaikan transaksi.

Kita dapat menyimpan data *click-through*, *basket placement*, dan *purchase* pengguna pada suatu tabel/matriks P , yaitu tabel preferensi pelanggan (*customer preference model*). Pada matriks ini, baris melambangkan ID pengguna dan kolom melambangkan produk pada kelas *level-l*. Kita dapat menetapkan preferensi melalui *click-through* sebagai p_{ij}^c , preferensi melalui *basket placement* sebagai p_{ij}^b , dan preferensi melalui *purchase* sebagai p_{ij}^p dengan $i = 1, 2, \dots, M$ (total jumlah pelanggan) dan $j = 1, 2, \dots, N$ (total jumlah produk dengan kelas *level-l*). Kemudian, kita dapat menghitung preferensi secara keseluruhan (p_{ij}) dengan persamaan berikut.

$$p_{ij} = \frac{p_{ij}^c - \min_{1 \leq j \leq N} (p_{ij}^c)}{\max_{1 \leq j \leq N} (p_{ij}^c) - \min_{1 \leq j \leq N} (p_{ij}^c)} + \frac{p_{ij}^b - \min_{1 \leq j \leq N} (p_{ij}^b)}{\max_{1 \leq j \leq N} (p_{ij}^b) - \min_{1 \leq j \leq N} (p_{ij}^b)} + \frac{p_{ij}^p - \min_{1 \leq j \leq N} (p_{ij}^p)}{\max_{1 \leq j \leq N} (p_{ij}^p) - \min_{1 \leq j \leq N} (p_{ij}^p)} \quad (5)$$

CID	Outerwear	Pants	Shirts	Shoes	Socks	Skincare	Perfumes	Bags	Belts	Wallets
p_{ij}^c : preference in click-through step										
203	100	120								
205	1	1	0	0	0	200	250	0	0	3
212	0	0	0	0	0	0	0	30	45	46
217	75	75	80	60	65	80	70	70	60	60
218	0	0	4	4	4	3	4	3	4	4
p_{ij}^b : preference in basket placement step										
203	3	3	1	0	0	0	0	0	0	0
205	0	0	0	0	0	5	4	0	0	0
212	0	0	0	0	0	0	0	1	6	6
217	0	0	0	2	2	1	0	1	1	2
218	0	0	0	2	2	2	0	2	0	2
p_{ij}^p : preference in purchase step										
203	1	1	1	0	0	0	0	0	0	0
205	0	0	0	0	0	2	1	0	0	0
212	0	0	0	0	0	0	0	1	0	0
217	0	0	0	0	0	0	0	0	1	0
218	0	0	0	0	0	2	0	0	0	0
p_{ij} : customer preference matrix										
203	2.833	3	2	0.042	0.042	0	0	0	0	0
205	0.004	0.004	0	0	0	2.8	2.3	0	0	0.012
212	0	0	0	0	0	0	0	1.82	1.98	2
217	0.75	0.75	1	1	1.25	1.5	0.5	1	1.5	1
218	0	0	0	2	3	2	0.75	2	0.75	2

Tabel 3.2. Tabel *customer preference model*. Sumber:

<https://www.sciencedirect.com/science/article/pii/S0957417402000520>

D. Analisis asosiasi produk

Kita dapat melakukan analisis asosiasi produk dengan metode associative rule mining yang telah dijelaskan sebelumnya. Rule berupa $A \Rightarrow B$ dapat kita modifikasi menjadi tiga bagian:

1. $A \Rightarrow^c B$, yang berarti 'jika produk A diklik, maka produk B juga cenderung diklik'
2. $A \Rightarrow^b B$, yang berarti 'jika produk A dimasukkan ke keranjang, maka produk B juga cenderung dimasukkan ke keranjang'
3. $A \Rightarrow^p B$, yang berarti 'jika produk A dibeli, maka produk B juga cenderung dibeli'

Setelah kita mendapatkan tiga hubungan tersebut dengan algoritma Apriori, kita dapat mengisi matriks asosiasi produk $A = (a_{ij})$, $i = 1, 2, \dots, M$ (total jumlah pelanggan), $j = 1, 2, \dots, N$ (total jumlah produk dengan kelas *level-l*). Nilai a_{ij} dapat didefinisikan sebagai berikut.

$$a_{ij} = \begin{cases} 1 & \text{if } i = j \\ 1 & \text{if } i \xrightarrow{p} j \\ 0.5 & \text{if } i \xrightarrow{b} j \\ 0.25 & \text{if } i \xrightarrow{c} j \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

	Outerwear	Pants	Shirts	Shoes	Socks	Skincare	Perfumes	Bags	Belts	Wallets
Outerwear	1		1	0.5	0.5			0.25	0.25	0.25
Pants		1		0.5	0.5			0.25	0.5	0.25
Shirts	1		1	0.5	0.5			0.25	0.25	0.25
Shoes				1						
Socks					1					
Skincare						1	0.25			
Perfumes						0.25	1			
Bags	0.25	0.25	0.25					1		
Belts	0.25	0.5	0.25						1	
Wallets	0.25	0.25	0.25							1

Tabel 3.3. Tabel association matrix. Sumber:

<https://www.sciencedirect.com/science/article/pii/S0957417402000520>

E. Pembuatan rekomendasi produk

Setelah mendapatkan tabel preferensi pelanggan (p_{ij}) dan tabel asosiasi produk (a_{ij}), kita dapat mengisi tabel kecocokan skor (*matching scores*) $S = (s_{ij})$ dengan rumus cosine similarity sebagai berikut.

$$s_{ij} = \frac{\mathbf{P}_i \cdot \mathbf{A}_j}{\|\mathbf{P}_i\| \|\mathbf{A}_j\|} = \frac{\sum_{k=1}^N p_{ik} a_{jk}}{\sqrt{\sum_{k=1}^N p_{ik}^2} \sqrt{\sum_{k=1}^N a_{jk}^2}} \quad (7)$$

Jika kita menggunakan tabel 3.2 sebagai (p_{ij}) dan tabel 3.3 sebagai (a_{ij}), maka kita akan memperoleh S sebagai berikut.

CID	Outerwear	Pants	Shirts	Shoes	Socks	Skincare	Perfumes	Bags	Belts	Wallets
203	0.648	0.484	0.648	0.009	0.009	0.000	0.000	0.392	0.504	0.392
205	0.001	0.001	0.001	0.000	0.000	0.904	0.803	0.001	0.001	0.004
212	0.264	0.424	0.264	0.000	0.000	0.000	0.000	0.498	0.504	0.548
217	0.676	0.675	0.676	0.296	0.370	0.466	0.251	0.441	0.583	0.441
218	0.440	0.554	0.440	0.391	0.587	0.415	0.237	0.359	0.125	0.359

Tabel 3.4. Tabel matching scores. Sumber:

<https://www.sciencedirect.com/science/article/pii/S0957417402000520>

Nilai s_{ij} akan berada pada rentang 0-1. Semakin besar nilai s_{ij} maka semakin tinggi kecocokan preferensi pelanggan dengan produk tersebut. Kita dapat memilih produk yang memiliki nilai

s_{ij} terbesar, kemudian merekomendasikan produk tersebut kepada pelanggan yang bersangkutan seperti pada tabel berikut.

CID	Purchased products	Recommended product classes	Recommended products
203	SKU00, SKU15, SKU25	Outerwear, Shirts	SKU02, SKU20
205	SKU31, SKU55, SKU69	Skincare, Perfumes	SKU52, SKU63
212	SKU72	Wallets, Belts	SKU99, SKU80
217	SKU83	Outerwear, Shirts	SKU02, SKU25
218	SKU44, SKU48	Socks, Pants	SKU43, SKU13

Tabel 3.5. List rekomendasi produk. Sumber:

<https://www.sciencedirect.com/science/article/pii/S0957417402000520>

V. KESIMPULAN

Himpunan dan pohon memiliki beragam manfaat pada kehidupan sehari-hari, salah satunya pada permasalahan rekomendasi produk *e-commerce*. Terdapat lima langkah yang perlu dilakukan untuk menciptakan rekomendasi yang tepat bagi pengguna *e-commerce*, yaitu identifikasi masalah, pemilihan pelanggan sasaran (*target customer*), analisis preferensi pelanggan, analisis asosiasi produk, dan pembuatan rekomendasi produk. Pada permasalahan ini, konsep himpunan digunakan pada langkah analisis asosiasi produk menggunakan algoritma Apriori, sedangkan pohon digunakan pada pemilihan pelanggan sasaran.

VI. UCAPAN TERIMA KASIH

Penulis mengucapkan puji syukur ke hadirat Allah Swt. karena atas rahmat dan karunia-Nya, makalah ini dapat diselesaikan dengan baik. Penulis juga berterima kasih kepada Ibu Dr. Fariska Zakhralativa Ruskanda, S.T., M.T. selaku dosen mata kuliah Matematika Diskrit (IF2130) kelas K2 serta Bapak Dr. Rinaldi Munir, S.T., M.T. yang telah menyediakan materi pembelajaran yang digunakan pada makalah ini. Semoga makalah ini dapat memberi manfaat bagi penulis serta bagi pihak-pihak lainnya.

REFERENSI

- [1] Cho, Yoon Ho, Jae Kyeong Kim, and Soung Hie Kim. "A personalized recommender system based on web usage mining and decision tree induction." *Expert systems with Applications* 23.3 (2002): 329-342.
- [2] Gillis, A. S. (2022, May 5). *What is clickstream data (Clickstream Analytics)?* Customer Experience. Diakses pada 9 Desember 2022, melalui tautan berikut. <https://www.techtarget.com/searchcustomerexperience/definition/clickstream-analysis-clickstream-analytics>
- [3] Rinaldi Munir, *Himpunan (bagian 1 – update 2022)*. Diakses pada 9 Desember 2022 melalui tautan berikut. [https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2022-2023/Himpunan\(2022\)-1.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2022-2023/Himpunan(2022)-1.pdf)
- [4] Rinaldi Munir, *Pohon (bagian 1)*. Diakses pada 9 Desember 2022 melalui tautan berikut. <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Pohon-2020-Bag1.pdf>
- [5] Rinaldi Munir, *Pohon (bagian 2)*. Diakses pada 9 Desember 2022 melalui tautan berikut. <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Pohon-2020-Bag2.pdf>
- [6] Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules. *Proceedings of the International Conference on very large Data Bases* (pp. 407-419).
- [7] *Algoritma Apriori: Pengertian, Cara Kerja, Kelebihan, Dan Kekurangannya*. Trivusi. Diakses pada 9 Desember 2022 melalui tautan berikut. <https://www.trivusi.web.id/2022/08/algoritma-apriori.html>
- [8] Lee, J., Podlaseck, M., Schonberg, E., & Hoch, R. (2001). "Visualization and analysis of clickstream data of online stores for understanding

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 3 Desember 2020

Ttd

Handwritten signature in black ink that reads "Haziq AM". The signature is written in a cursive style with a small loop at the end of the first name.

Haziq Abiyyu Mahdy 13521170