

TOPIK 2: Aturan Asosiasi Item

2.1. Pendahuluan

Di era Society 5.0, yang dikenal sebagai Era Rekomendasi, bisnis dan organisasi dihadapkan pada arus deras informasi yang terus bertambah. Di tengah lautan data ini, tersimpan pola dan hubungan tersembunyi yang, jika berhasil diungkap, dapat memberikan wawasan berharga untuk pengambilan keputusan strategis dan menciptakan rekomendasi yang dipersonalisasi bagi pengguna. *Association Rule Mining (ARM)*, atau Penambangan Aturan Asosiasi, hadir sebagai solusi efektif untuk mengungkap hubungan-hubungan tersebut.

ARM adalah teknik data mining yang bertujuan menemukan pola menarik antara item-item dalam dataset besar. Pola ini dinyatakan dalam bentuk aturan asosiasi, seperti "jika pelanggan membeli roti, maka mereka juga cenderung membeli mentega." Aturan semacam itu dapat membantu bisnis memahami perilaku pelanggan, mengoptimalkan strategi pemasaran, meningkatkan tata letak toko, dan memberikan rekomendasi produk yang relevan kepada pelanggan, sehingga meningkatkan pengalaman pengguna dan mendorong penjualan.

2.2. Konsep Dasar Association Rule Mining

Frequent Itemsets (FI)

Frequent Itemset (FI) atau himpunan item (itemset) disebut *frequent* jika itemset dimaksud muncul bersamaan dalam dataset dengan frekuensi melebihi ambang batas tertentu yang disebut *minimum support* (minsup). Misalnya, pemilik supermarket menetapkan minsup = 20% sebagai batas bawah sebuah itemset dikatakan frequent. Jika ternyata kombinasi item [kopi, roti] memiliki Support 50% (\geq minsup) maka [kopi, roti] disebut frequent itemset. Nilai minsup fleksibel tergantung kebutuhan pemilik toko.

Association Rules (AR)

Mengikuti definisi asli oleh (Agrawal & Srikant, 1994) dan (Agrawal et al., 1993), masalah *association rule mining* didefinisikan sebagai berikut:

- Misalkan $I = \{i_1, i_2, \dots, i_n\}$ adalah himpunan dari n atribut biner yang disebut *item*.
- Misalkan $D = \{t_1, t_2, \dots, t_N\}$ adalah himpunan transaksi yang disebut *database* atau dataset. N adalah ukuran D , yaitu jumlah record di D .

- Setiap *transaksi* dalam D memiliki ID transaksi unik dan berisi subset dari item-item dalam I .

Atribut seringkali disebut juga fitur, *field* atau variabel. Sebuah *aturan* didefinisikan sebagai implikasi dalam bentuk:

$$X \rightarrow Y, \text{ di mana } X, Y \subseteq I.$$

Dalam Agrawal, Imieliński, Swami [2], sebuah *aturan* didefinisikan hanya antara sebuah himpunan dan sebuah item tunggal, $X \Rightarrow i_j$ untuk $i_j \in I$.

Setiap aturan terdiri dari dua himpunan item (itemset) yang berbeda, juga dikenal sebagai *itemset*, X dan Y , di mana X disebut *antecedent* atau sisi kiri atau left-hand side (LHS) dan Y disebut *consequent* atau sisi kanan atau right-hand side (RHS). Antecedent adalah item yang dapat ditemukan dalam data, sedangkan consequent adalah item yang ditemukan ketika dikombinasikan dengan antecedent.

Pernyataan $X \rightarrow Y$ sering dibaca sebagai "jika X muncul maka Y juga muncul", di mana antecedent (X) adalah "jika" dan consequent (Y) adalah "maka". Ini berarti bahwa, secara teori, setiap kali X muncul dalam dataset, maka Y juga akan muncul.

Setiap aturan asosiasi memiliki dua metrik penting:

- 1. Support:** Mengukur seberapa sering kumpulan item atau itemset XY muncul bersamaan dalam transaksi-transaksi di dataset D . Support yang tinggi menunjukkan bahwa aturan tersebut berlaku untuk sebagian besar transaksi. Rumus Support relatif diberikan dalam persamaan (1), namun jika tidak dibagi N maka $Sup(XY)$ menjadi nilai Support absolut.

$$Sup(XY) = \frac{\text{Jumlah } t_i \in D, XY \text{ } t_i}{N}$$

- 2. Confidence:** Mengukur peluang aturan $X \rightarrow Y$ terjadi. Confidence yang tinggi menunjukkan bahwa jika kondisi X terpenuhi, maka kondisi Y juga cenderung terpenuhi. Rumus Confidence diberikan dalam Persamaan (2)

$$Conf(X \rightarrow Y) = \frac{Sup(XY)}{Sup(X)}$$

Selain itu, ada metrik lain yang berguna dalam mengevaluasi aturan asosiasi:

- 3. Lift:** Mengukur seberapa kuat hubungan antara X dan Y dibandingkan jika X dan Y muncul secara independen. Lift lebih dari 1 menunjukkan adanya hubungan positif antara X dan Y . Lift sama dengan 1, menunjukkan bahwa X dan Y saling bebas atau tidak memengaruhi satu dan lainnya.

$$Lift(X \rightarrow Y) = \frac{Conf(XY)}{Sup(Y)} = \frac{Sup(XY)}{Sup(X).Sup(Y)}$$

Algoritma Apriori

Algoritma Apriori dikenal sebagai algoritma pertama, dan merupakan salah satu algoritma yang paling populer dan cukup efisien untuk menemukan frequent itemset dan association rule dalam data transaksi. Algoritma ini bekerja berdasarkan dua prinsip utama:

- **Prinsip Apriori property:** Jika suatu itemset dianggap *frequent*, maka semua subset dari itemset tersebut juga harus *frequent*. Sebaliknya, jika suatu itemset dianggap *tidak frequent*, maka semua superset dari itemset tersebut juga *tidak frequent*. Misalnya, jika itemset [kopi, gula] adalah frequent
- **Pendekatan bottom-up:** Algoritma dimulai dengan mencari frequent 1-itemset (item tunggal yang sering muncul), kemudian secara iteratif membangun frequent itemset yang lebih besar dengan menggabungkan frequent itemset yang lebih kecil.

Dengan memanfaatkan kedua prinsip ini, algoritma Apriori dapat memangkas jumlah kandidat itemset yang perlu dihitung, sehingga meningkatkan efisiensi proses pencarian frequent itemset, terutama pada dataset yang besar.

Langkah-langkah Algoritma Apriori:

1. Inisialisasi:

1. Tentukan nilai parameter minimum support (*minsup*), minimum confidence (*minconf*), dan parameter opsional: maksimum panjang itemset (*maxlen*)
2. Hitung support untuk setiap item individual dalam dataset.
3. Buang item yang support-nya kurang dari minimum support. Item-item yang tersisa disebut sebagai frequent 1-itemset.

2. Generate k-itemset dari (k-1)-itemset:

1. Gabungkan frequent (k-1)-itemset untuk menghasilkan kandidat k-itemset.
2. Hitung support untuk setiap kandidat k-itemset.
3. Buang kandidat k-itemset yang support-nya kurang dari *minsup*. Kandidat k-itemset yang tersisa disebut sebagai frequent k-itemset.

3. Ulangi langkah 2 hingga tidak ada lagi frequent itemset yang dapat dihasilkan atau semua frequent *maxlen*-itemset terbentuk.

4. Generate association rules dari frequent itemset:

1. Untuk setiap frequent itemset, buat semua permutasi aturan asosiasi dengan membagi itemset menjadi dua bagian: antecedent (*X*) dan consequent (*Y*).
2. Hitung confidence untuk setiap aturan asosiasi. Panjang *X* dan *Y* masing-masing dibentuk mulai dari panjang 1 item sampai *maxlen-1* item
3. Buang aturan asosiasi yang confidence-nya kurang dari minimum confidence yang ditentukan.

Efisiensi Algoritma Apriori

Meskipun algoritma Apriori merupakan salah satu algoritma yang paling awal dan banyak digunakan untuk *AR mining*, namun bukanlah yang paling efisien, terutama untuk dataset yang sangat besar atau memiliki minimum support yang rendah.

Pencarian *frequent itemset* (FI) dan *association rule* (AR) memang merupakan salah satu topik riset yang paling aktif dan terus berkembang di bidang data mining. Banyak peneliti dan praktisi yang telah mengembangkan berbagai algoritma dan teknik baru untuk mengatasi keterbatasan Apriori dan meningkatkan efisiensi serta efektivitas proses penambangan aturan asosiasi.

Beberapa contoh algoritma yang lebih efisien daripada Apriori antara lain:

- FP-Growth: Algoritma ini menggunakan struktur data yang disebut *FP-tree* untuk menyimpan informasi tentang frequent itemset secara ringkas. FP-Growth tidak memerlukan pemindaian (scan) berulang pada dataset seperti Apriori, sehingga lebih efisien untuk dataset yang besar.
- Eclat: Algoritma ini menggunakan representasi vertikal dari dataset dan teknik *intersection* untuk menemukan frequent itemset. Eclat juga dapat lebih efisien daripada Apriori, terutama untuk dataset yang sparse (memiliki banyak item yang jarang muncul).
- Algoritma berbasis hashing: Beberapa algoritma menggunakan teknik hashing untuk mengurangi jumlah kandidat itemset yang perlu dihitung, sehingga meningkatkan efisiensi.

Selain pengembangan algoritma baru, penelitian di bidang ARM juga mencakup topik-topik seperti:

- Penambangan AR yang menarik (*interesting rules*): Mengembangkan metrik dan teknik baru untuk mengevaluasi dan memilih aturan asosiasi yang paling menarik dan berguna bagi pengguna, dengan kriteria menarik dan berguna yang bervariasi (Ju et al., 2015),
- Penambangan AR dan FI dengan kondisi tertentu: seperti batasan waktu, lokasi, atau atribut lainnya (Husni et al., 2022).
- Penambangan FI dan AR dalam data streaming: Menemukan aturan asosiasi secara real-time dari aliran data yang terus berubah; misalnya menggunakan *Time-windows* seperti yang dikembangkan dalam (Akhriza et al., 2015, 2018; Akhriza & Mumpuni, 2022; Akhriza & Utsalina, 2023).
- Penambangan FI dan AR dalam data yang tidak pasti: Menemukan aturan asosiasi dari data yang mengandung ketidakpastian atau ketidaklengkapan.

Dengan demikian, meskipun Apriori merupakan algoritma penting dalam sejarah perkembangan ARM, namun bidang ini terus berkembang dengan pesat. Banyak algoritma dan teknik baru telah dikembangkan untuk mengatasi keterbatasan Apriori dan meningkatkan efisiensi serta efektivitas penambangan aturan asosiasi. Data Scientist dapat memilih algoritma dan metrik yang paling sesuai untuk kebutuhan analisis data perusahaan.

Menambang Aturan Asosiasi

1. Data Transaksi

Beberapa contoh penggunaan *Frequent Itemset* (FI) dan *Association Rule* (AR) pada data transaksi dalam berbagai kasus diberikan sebagai berikut:

- Penjualan di Toko Online dan Offline
 - Rekomendasi Produk: Dengan menganalisis pola pembelian pelanggan, toko dapat menemukan produk yang sering dibeli bersamaan. Misalnya, aturan asosiasi "Jika pelanggan membeli laptop, maka mereka juga cenderung membeli tas laptop" dapat digunakan untuk memberikan rekomendasi produk yang relevan kepada pelanggan, meningkatkan potensi *cross-selling*.
 - Penataan Produk: Dengan mengetahui produk mana yang sering dibeli bersamaan, toko dapat mengatur tata letak produk secara strategis. Misalnya, menempatkan roti dan selai di rak yang berdekatan dapat meningkatkan penjualan kedua produk tersebut.
 - Promosi dan Diskon: Aturan asosiasi dapat membantu toko merancang promosi atau bundling produk yang menarik. Misalnya, menawarkan diskon untuk pembelian kopi dan gula bersamaan dapat meningkatkan penjualan kedua produk tersebut.
 - Analisis Keranjang Belanja: Dengan menganalisis isi keranjang belanja pelanggan, toko dapat memahami pola pembelian dan mengidentifikasi peluang *cross-selling* dan *up-selling*. Misalnya, jika pelanggan sering membeli produk perawatan rambut tertentu, toko dapat merekomendasikan produk pelengkap lainnya dari merek yang sama.
- Perpustakaan
 - Rekomendasi Buku: Dengan menganalisis riwayat peminjaman buku, perpustakaan dapat menemukan buku-buku yang sering dipinjam bersamaan. Informasi ini dapat digunakan untuk memberikan rekomendasi buku yang relevan kepada anggota perpustakaan, mendorong mereka untuk membaca lebih banyak buku.
 - Penataan Koleksi: Dengan mengetahui buku-buku mana yang sering dipinjam bersamaan, perpustakaan dapat mengatur tata letak koleksi buku secara lebih efektif. Misalnya, menempatkan buku-buku dengan topik yang serupa di rak yang berdekatan dapat memudahkan anggota perpustakaan menemukan buku yang mereka cari.
 - Pengadaan Buku: Analisis frequent itemset dapat membantu perpustakaan mengidentifikasi buku-buku populer yang perlu ditambah stoknya atau buku-buku baru yang relevan dengan minat anggota perpustakaan.

– Job Skillset

- Analisis Keterampilan yang Dibutuhkan: Dengan menganalisis data lowongan pekerjaan, perusahaan dapat menemukan keterampilan yang sering dibutuhkan bersamaan untuk posisi tertentu. Informasi ini dapat membantu perusahaan dalam proses rekrutmen dan pengembangan karyawan.
- Rekomendasi Pelatihan: Dengan mengetahui keterampilan mana yang sering dibutuhkan bersamaan, perusahaan atau lembaga pelatihan dapat merancang program pelatihan yang lebih relevan dan efektif bagi para pencari kerja atau karyawan.
- Perencanaan Karir: Individu atau pengelola Kurikulum dapat menggunakan analisis association rule untuk mengidentifikasi keterampilan yang saling melengkapi dan mengembangkan jalur karir mereka secara strategis.

Frequent itemset dan association rule memiliki aplikasi yang luas dalam berbagai bidang, termasuk penjualan, perpustakaan, dan analisis keterampilan kerja. Dengan mengidentifikasi pola dan hubungan tersembunyi dalam data transaksi, kita dapat memperoleh wawasan berharga yang dapat digunakan untuk meningkatkan pengambilan keputusan, memberikan rekomendasi yang dipersonalisasi, dan mengoptimalkan berbagai proses bisnis.

Penerapan FI dan AR perlu disesuaikan dengan konteks dan tujuan spesifik dari setiap kasus. Dengan pemahaman yang baik tentang konsep dan teknik ini, serta kemampuan untuk menginterpretasikan hasil analisis secara tepat, potensi penuh dari association rule mining dapat dimanfaatkan untuk mencapai tujuan bisnis.

2. Data Tekstual

Data tekstual, seperti dokumen, artikel, ulasan online, atau hasil mesin pencari juga dapat menjadi sumber dalam ARM.

Frequent itemset dan association rule dapat digunakan untuk menemukan topik kunci dari hasil pencarian Google dengan cara mengidentifikasi kata-kata atau frasa yang sering muncul bersamaan dalam hasil pencarian tersebut. Dengan menganalisis frequent itemset, kita dapat mengetahui kata-kata atau frasa yang sering dicari oleh pengguna, yang dapat memberikan gambaran tentang topik-topik yang sedang populer atau diminati.

Sementara itu, association rule dapat membantu kita menemukan hubungan antara kata-kata atau frasa yang berbeda dalam hasil pencarian. Misalnya, jika kita menemukan aturan asosiasi seperti "jika pengguna mencari 'data mining', maka mereka juga cenderung mencari 'machine learning'", maka kita dapat menyimpulkan bahwa kedua topik tersebut terkait erat dan mungkin merupakan topik kunci dalam hasil pencarian.

Dengan demikian, frequent itemset dan association rule dapat memberikan wawasan berharga tentang topik-topik kunci yang muncul dari hasil pencarian Google, yang dapat digunakan untuk berbagai keperluan, seperti:

- Optimasi mesin pencari (SEO): Dengan mengetahui topik kunci yang dicari pengguna, pemilik situs web dapat mengoptimalkan konten mereka agar lebih relevan dengan pencarian tersebut dan meningkatkan peringkat situs mereka di hasil pencarian.
- Analisis tren: Dengan memantau perubahan dalam frequent itemset dan association rule dari waktu ke waktu, kita dapat mengidentifikasi tren topik yang sedang berkembang atau menurun popularitasnya.
- Personalisasi konten: Dengan memahami topik kunci yang diminati oleh pengguna tertentu, penyedia konten dapat memberikan rekomendasi atau konten yang lebih dipersonalisasi kepada pengguna tersebut.
- Pengembangan produk atau layanan baru: Dengan menganalisis frequent itemset dan association rule, perusahaan dapat mengidentifikasi kebutuhan atau minat pelanggan yang belum terpenuhi, yang dapat menjadi peluang untuk mengembangkan produk atau layanan baru.

Secara keseluruhan, frequent itemset dan association rule merupakan alat yang berguna untuk mengekstrak informasi berharga dari hasil pencarian Google dan memahami topik-topik kunci yang diminati oleh pengguna.

Algoritma pencari FI dan AR seringkali menerima input dalam bentuk Bag of Words, dimana 'Words' bisa disesuaikan dengan kasus yang ditangani.

- Dalam pencarian di data abstrak publikasi, Words bisa berupa kata-kata kunci;
- Di data transaksi penjualan Words bisa berupa nama produk, atau hanya kode produk saja (Akhriza & Mumpuni, 2022; Akhriza & Utsalina, 2023);
- Di data job skillset, Words bisa berupa nama-nama keterampilan (skill) yang diwakili misalnya dengan nama software (Akhriza et al., 2017; Latifah et al., 2020);
- Words di data transaksi peminjaman perpustakaan bisa berupa kode buku (Adistia et al., 2019), dan
- Words di data penjualan restoran bisa berupa kode menu.

Langkah-langkah dalam menambang AR dari data tekstual:

1. **Persiapan data transaksi:** tujuan akhir dari penyiapan ini adalah mendapatkan format "Bag of Words", dimana Words dapat berupa nama produk atau ID produk. Data transaksi penjualan mentah yang direkam bisa dalam bentuk relasi antara beberapa entitas, sehingga dapat 'ditarik' menggunakan query. Sedangkan data transaksi dari penjualan online, satu nomer invoice penjualan merekam ID produk satu per satu, sehingga kita harus melakukan GroupBy berdasarkan nomer invoice. Contoh:

No.Invoice	Tanggal	ID Produk
A001	1 Agustus 2024	X001
A001	1 Agustus 2024	Y001
A002	1 Agustus 2024	X001
A002	1 Agustus 2024	Z001

A002	1 Agustus 2024	W002
------	----------------	------

Ditransformasi menjadi Bag of ID Produk sehingga berbentuk sebagai berikut:

No.Invoice	Tanggal	Bag of ID Produk
A001	1 Agustus 2024	{X001, Y001}
A002	1 Agustus 2024	{X001, Z001, W002}

2. **Persiapan data teks:** Membersihkan teks dari noise (misalnya, tanda baca, angka, URL), melakukan tokenisasi (memecah teks menjadi kata-kata), menghilangkan *stop words* (kata-kata umum yang tidak membawa banyak makna, seperti "dan", "atau", "itu"), dan mungkin melakukan stemming atau lemmatization (mengubah kata-kata ke bentuk dasarnya) untuk mengurangi dimensi data dan meningkatkan akurasi analisis.
3. **Transformasi data ke bentuk 'Bag of Words':** Mengubah tipe data ke tipe String atau Document, atau bentuk lain yang diminta oleh algoritma ARM yang digunakan. Setiap data atau dokumen direpresentasikan sebagai kumpulan kata-kata yang muncul di dalamnya. Kadangkala, frekuensi kemunculan setiap kata juga disertakan sebagai bobotnya.
4. **Menemukan FI dan AR:** Menerapkan algoritma ARM pada representasi 'Bag of Words' untuk menemukan frequent itemsets dan menghasilkan aturan asosiasi.
5. **Evaluasi dan interpretasi hasil:** Memilih dan menginterpretasikan aturan asosiasi yang paling menarik untuk mendapatkan wawasan tentang sentimen, topik, atau tren yang muncul dalam data tekstual.

Contoh: Diberikan data transaksi pembelian dalam bentuk "Bag of Product Name", atau "Tas nama produk", dimana tiap baris adalah string dari nama produk yang dibeli di sebuah transaksi:

1. kopi gula beras
2. kopi gula buku sabun
3. sabun sampo sendal kaos_kaki
4. sendal kaos_kaki gunting_kuku sapu
5. kopi beras sapu sabun
6. buku pensil pulpen penggaris
7. buku pensil penggaris pulpen penghapus
8. ember sapu sabun sampo buku
9. sampo sabun sendal kopi gula
10. sisir kaos sendal pulpen
11. kopi susu gula teh
12. susu roti gula snack_1
13. snack_1 kopi roti sapu
14. teh gula snack_1 snack_2

15. snack_1 snack_2 kopi teh

Kita akan menghitung *support* dan *confidence* untuk beberapa itemset dan aturan asosiasi, dengan minimum support (minsup) = 0.2 (20%) dan minimum confidence (minconf) = 0.5 (50%).

1. Perhitungan Support

Support dari suatu itemset dihitung dengan membagi jumlah transaksi yang mengandung itemset tersebut dengan total jumlah transaksi. Di sini terdapat 15 transaksi, sehingga Support beberapa itemset yang memenuhi minsup adalah sebagai berikut:

- Support [kopi], atau $\text{Sup}(\{\text{kopi}\}) = 7$, atau support relatif = $7/15 = 46,67\%$
- $\text{Sup}(\{\text{gula}\}) = 6$, atau $6/15 = 40\%$
- $\text{Sup}(\{\text{kopi, gula}\}) = \text{Sup}(\{\text{gula, kopi}\}) = 4$, atau $26,67\%$

2. Perhitungan Confidence

Confidence dari suatu aturan asosiasi $x \rightarrow y$ dihitung dengan membagi support dari itemset $\{A, B\}$ dengan support dari itemset $\{A\}$. Beberapa Rules yang memenuhi minconf yaitu:

- $\text{Conf}(kopi \rightarrow gula) = \text{Sup}(\{\text{kopi, gula}\}) / \text{Sup}(\{\text{kopi}\}) = 26,67\% / 46,67\% = 0.571428$
- $\text{Conf}(gula \rightarrow kopi) = \text{Sup}(\{\text{kopi, gula}\}) / \text{Sup}(\{\text{gula}\}) = 26,67\% / 40\% = 0.667$

Di sini didemonstrasikan bahwa Confidence tidak komutatif, artinya $\text{Conf}(kopi \rightarrow gula)$ tidak sama dengan $\text{Conf}(gula \rightarrow kopi)$

3. Perhitungan Lift

Lift dari suatu aturan asosiasi $x \rightarrow y$ dihitung dengan membagi $\text{conf}(x \rightarrow y)$ dengan $\text{sup}(x)$. Nilai Lift bersifat komutatif.

- $\text{Lift}(kopi \rightarrow gula) = \text{Conf}(kopi \rightarrow gula) / \text{Sup}(gula) = 0.571428 / 0.4 \approx 1.429$
- $\text{Lift}(gula \rightarrow kopi) = \text{Conf}(gula \rightarrow kopi) / \text{Sup}(kopi) = 0.667 / 0.4667 \approx 1.429$

Interpretasi Confidence dan Lift:

- Aturan "*kopi* \rightarrow *gula*" memiliki confidence 0.571428, artinya ada 57,1% transaksi yang membeli kopi juga membeli gula.
- Lift "*kopi* \rightarrow *gula*" sebesar 1.429 menunjukkan adanya hubungan positif antara kopi dan gula, meskipun tidak terlalu kuat. Kemunculan kopi sedikit meningkatkan kemungkinan kemunculan gula dalam transaksi.
- Aturan lain, seperti "*kaos_kaki* \rightarrow *sendal*" Lift lebih tinggi yaitu 3,75. Ini menunjukkan bahwa item-item dalam aturan ini memiliki hubungan yang sangat kuat, jauh lebih kuat daripada jika mereka muncul secara independen.

- Jika ada aturan " $p \rightarrow q$ " memiliki lift kurang dari 1, misalnya 0.8. Ini menunjukkan bahwa kemunculan " p " justru sedikit menurunkan kemungkinan kemunculan " q ", meskipun aturan ini tetap memenuhi ambang batas minimum confidence.

Dengan mempertimbangkan nilai lift bersama dengan support dan confidence, kita dapat lebih memahami kekuatan dan relevansi dari setiap aturan asosiasi, sehingga membantu pembuatan keputusan yang lebih tepat berdasarkan analisis data.

Jenis-jenis Frequent Itemset

Selain frequent itemset biasa, ada beberapa jenis frequent itemset lain yang berguna dalam association rule mining, antara lain:

- **Frequent Closed Itemset (FCI):** Suatu itemset dikatakan *closed* jika tidak ada superset-nya yang memiliki support yang sama. Frequent closed itemset dapat membantu meringkas informasi dari frequent itemset biasa karena mereka mewakili kelompok item yang selalu muncul bersamaan.
- **Frequent Maximal Itemset (FMI):** Suatu itemset dikatakan *maximal* jika tidak ada superset-nya yang frequent. Frequent maximal itemset memberikan informasi tentang itemset terpanjang yang sering muncul bersamaan, yang dapat berguna untuk mengidentifikasi pola yang lebih spesifik.

Untuk contoh data transaksi sebelumnya, diberikan daftar FI yang mengandung gula, kopi dan sabun dalam tabel berikut ini:

Support	FI
0.1333	[gula, kopi, sabun]
0.2666	[gula, kopi]
0.1333	[gula, sabun]
0.2	[kopi, sabun]
0,467	[kopi]
0,4	[gula]
0,3333	[sabun]

Maka FCI adalah FI yang menutupi (covering) subset yang memiliki support sama dengannya. Dengan demikian FCI dari FI di atas adalah sebagai berikut, di mana tampak bahwa FI [gula, sabun] tidak muncul karena memiliki Support sama dengan [gula, kopi, sabun] = 0,1333

Support	FCI
0.1333	[gula, kopi, sabun]
0.2666	[gula, kopi]
0.2	[kopi, sabun]
0,467	[kopi]
0,4	[gula]
0,3333	[sabun]

FMI dari FI di atas adalah hanya [gula, kopi, sabun] saja karena FI ini menutupi **semua subsetnya** tanpa memandang support-nya. Di sini terlihat bahwa jika FCI masih menyimpan informasi mengenai Support dari FI yang merupakan sub-itemsetnya, maka FMI tidak menyimpan informasi dimaksud.

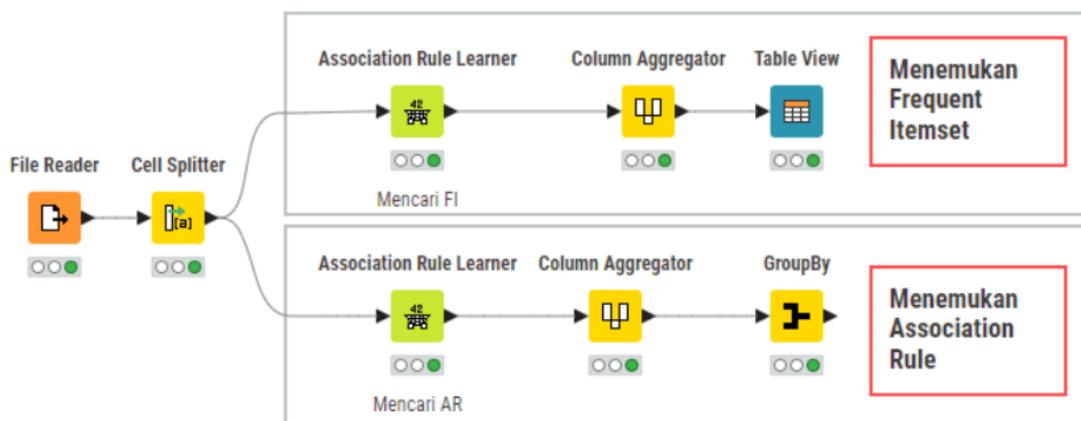
Tujuan menemukan FCI dan FMI adalah untuk mengurangi ukuran FI yang dihasilkan, khususnya pada saat menggunakan minsup yang cukup kecil, yang menyebabkan ukuran FI dapat melebihi ukuran dataset berkali-kali lipat. Di contoh ini, terdapat tujuh FI untuk mengamati kemunculan gula, kopi dan sabun bersama-sama, menggunakan FCI membutuhkan enam itemset, sedangkan FMI hanya 1 itemset.

2.3. Praktikum dengan KNIME

Menemukan Rekomendasi Produk dari Data transaksi

Masalah bisnis yang diangkat adalah ‘bagaimana merekomendasi produk-produk yang pernah dibeli bersama-sama’. Untuk menemukan rekomendasi ini, diperlukan data transaksi penjualan.

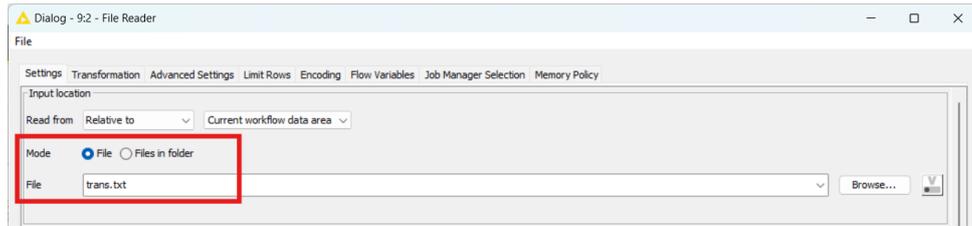
Workflow dirancang untuk dua tugas: 1) menemukan FI dan 2) menemukan AR, sebagaimana yang dijelaskan dalam Gambar 2.1. Data yang diolah adalah dataset transaksi penjualan sederhana yang tersimpan dalam file trans.txt



Gambar 2.1 Workflow Penemuan FI dan AR

Untuk tugas ke-1, menemukan FI, node yang digunakan adalah:

1. **File Reader**, berfungsi untuk membaca isi file transaksi penjualan sederhana yang disimpan dalam file *trans.txt*. Konfigurasi untuk File reader untuk tugas ini diberikan pada Gambar 2,2, dengan beberapa record di dalam trans.txt diberikan dalam Gambar 2.3. Isi dari file ini ditunjukkan dalam Column0, di mana tiap transaksi direkam dalam bentuk untaian atau string dari nama 'produk 1, produk 2, produk 3, dst.'



Gambar 2.2 Form Dialog File Reader

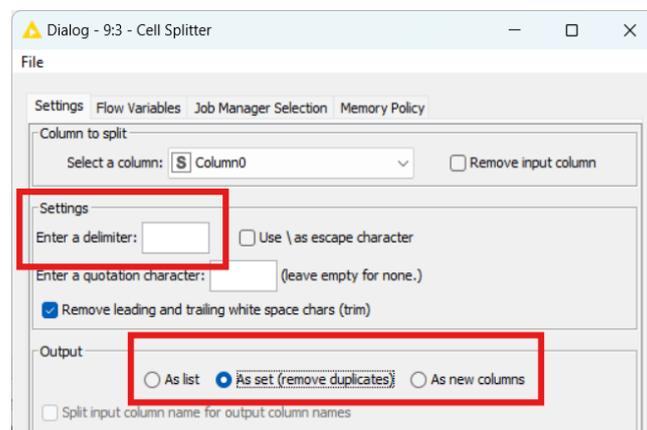
▶ 1: File Table 📄 Flow Variables

Rows: 15 | Columns: 1

#	RowID	Column0
1	Row0	kopi gula beras
2	Row1	kopi gula buku sabun
3	Row2	sabun sampo sendal kaos_kaki
4	Row3	sendal kaos_kaki gunting_kuku sapu
5	Row4	kopi beras sapu sabun
6	Row5	buku pensil pulpen penggaris
7	Row6	buku pensil penggaris pulpen penghapus
8	Row7	ember sapu sabun sampo buku

Gambar 2.3 Hasil File Reader

2. **Cell Splitter**, berfungsi untuk memisahkan string nama produk di Column0 menjadi himpunan (Set). Konfigurasi yang dibutuhkan diberikan di Gambar 2.4. Delimiter atau pemisah elemen diisi dengan satu spasi kosong, sedangkan Output diatur ke "As set (remove duplicates)" agar string nama produk dipisahkan dalam bentuk himpunan di mana nama produk muncul secara unik. Hasil dari Cell Splitter diberikan dalam Gambar 2.5, dimana tampak nama produk di dalam string "kopi gula beras", misalnya, dipisah dan disimpan dalam Set [kopi, gula, beras]. Dalam tahap ini, bentuk Set[produk 1, produk 2, ...] dapat dikatakan sebagai "Bag of Product", atau tas yang berisi nama-nama produk, dan diletakkan di kolom **Column0_SplitResultSet**.



Gambar 2.4 Form Dialog Cell Splitter

► 1: Output Table Flow Variables

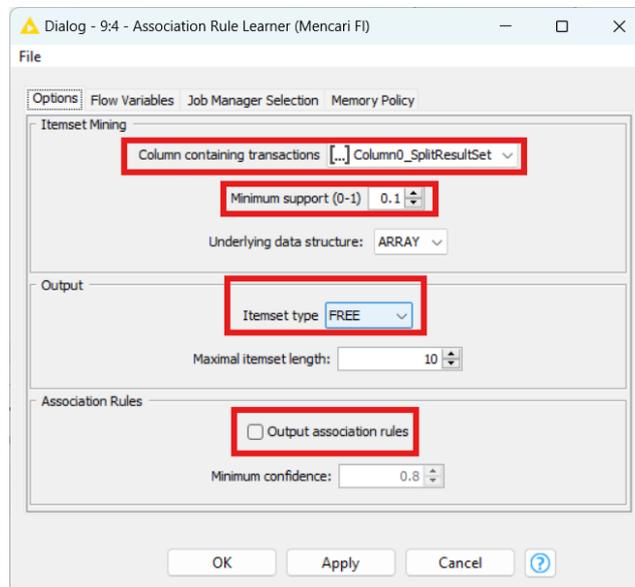
Rows: 15 | Columns: 2

Table Statistics

#	RowID	Column0 <i>String</i>	Column0_SplitResultSet <i>Set</i>
1	Row0	kopi gula beras	[kopi,gula,beras]
2	Row1	kopi gula buku sabun	[kopi,gula,buku,...]
3	Row2	sabun sampo sendal kaos_kaki	[sabun,sampo,sendal,...]
4	Row3	sendal kaos_kaki gunting_kuku sapu	[sendal,kaos_kaki,gunting_kuku,...]
5	Row4	kopi beras sapu sabun	[kopi,beras,sapu,...]
6	Row5	buku pensil pulpen penggaris	[buku,pensil,pulpen,...]
7	Row6	buku pensil penggaris pulpen penghapus	[buku,pensil,penggaris,...]

Gambar 2.5 Hasil Cell Splitter

3. **Association Rule Learner**, node utama yang berfungsi untuk menemukan FI dan AR dari dalam dataset yang telah ditransformasikan ke dalam **Set[produk 1, produk 2, ...]**. Konfigurasi node ini diberikan dalam Gambar 2.6, di mana kolom yang diproses adalah **Column0_SplitResultSet**, minimum support (minsup) diatur ke 0.1 (10%), Output itemset type diatur ke FREE, yaitu frequent itemset biasa, dan Opsi Output association rules tidak dicentang. Maximal itemset length dibiarkan di nilai 10, yang artinya, jumlah item di dalam itemset yang dihasilkan tidak lebih dari 10 item.



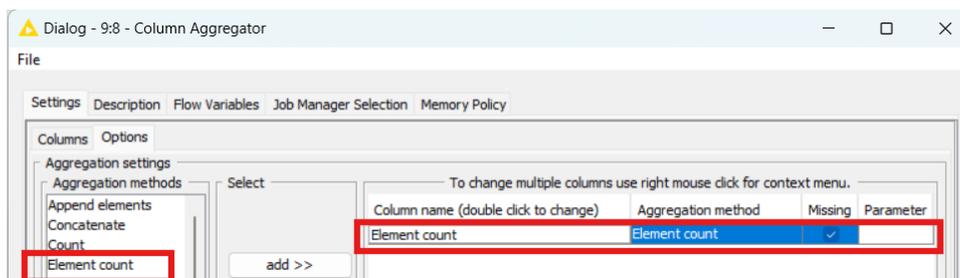
Gambar 2.6

Gambar 2.7 menayangkan beberapa Frequent Itemsets yang dicari dengan pengaturan minsup sebagaimana dijelaskan.

RowID	Support(0-1):	Items
item set 0	0.467	[kopi]
item set 1	0.267	[gula,kopi]
item set 2	0.133	[gula,kopi,sabun]
item set 3	0.133	[beras,kopi]
item set 4	0.2	[kopi,sabun]
item set 5	0.133	[sapu,kopi]

Gambar 2.7 FI yang Ditemukan

4. **Column Aggregator.** Node ini bersifat opsional yang berfungsi untuk menghitung jumlah elemen di dalam tiap Items. Di dalam form konfigurasinya, hanya Elemen count yang dipilih untuk fungsi dimaksud, sebagaimana ditunjukkan dalam Gambar 2.8.



Gambar 2.8 Form Dialog Column Aggregator

5. **Table View** juga bersifat opsional dan digunakan di sini untuk menampilkan tabel yang dihasilkan oleh Column Aggregator, sebagaimana ditunjukkan dalam Gambar 2.9.

RowID	Support(0-1):	Items	Element count
item set 0	0.467	[kopi]	1
item set 1	0.267	[gula,kopi]	2
item set 2	0.133	[gula,kopi,sabun]	3
item set 3	0.133	[beras,kopi]	2
item set 4	0.2	[kopi,sabun]	2
item set 5	0.133	[sapu,kopi]	2
item set 6	0.133	[teh,kopi]	2
item set 7	0.133	[snack_1,kopi]	2
item set 8	0.4	[gula]	1
item set 9	0.133	[gula,sabun]	2
item set 10	0.133	[susu,gula]	2
item set 11	0.133	[gula,teh]	2

Gambar 2.9 Hasil Column Aggregator

Untuk tugas ke-2, menemukan AR, nodes yang digunakan adalah:

1. File reader, dan Cell Splitter, dengan konfigurasi yang sama dengan yang diatur di dalam tugas ke-1
2. Association Rule Learner, diatur agak berbeda sedikit dengan tugas ke-1, yaitu dengan mencentang Opsi Output association rules dengan Minimum Confidence untuk contoh ini yaitu 0.5 (50%) (Gambar 2.10).



Gambar 2.10 Setting Minimum Confidence

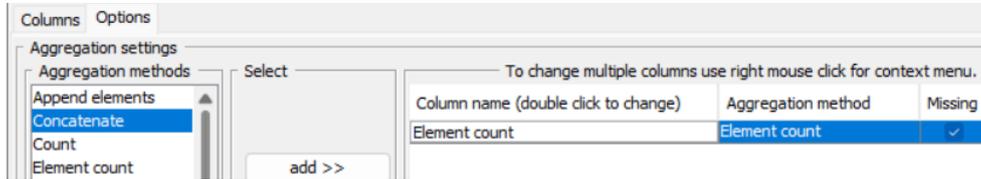
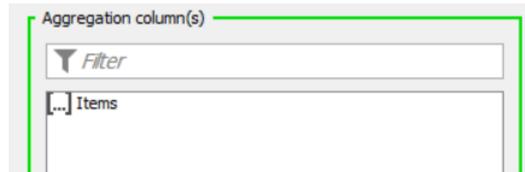
AR yang ditemukan ditunjukkan di Gambar 2.11, dimana Rules dibaca dari kolom Consequent, Implies, Items (anteseden); misalnya $\text{snack_2} \leftarrow [\text{teh, snack_1}]$, atau secara umum berbentuk $Y \leftarrow X$. Aturan ini dibaca: jika teh dan snack_1 dibeli, maka snack_2 juga dibeli dengan peluang (confidence) 100%, dan aturan ini menjadi dasar pengembangan metode rekomendasi berbasis AR, seperti yang ditemukan di Amazon.com, misalnya.

Di tabel ini, terlihat adanya nilai Lift dari Aturan meskipun di dalam form konfigurasi tidak terdapat opsi untuk mengatur Lift. Sebagaimana dijelaskan, Lift (X, Y) yang bernilai lebih dari 1, menunjukkan bahwa X dan Y saling terikat dan berarti akan selalu muncul bersamaan di masa mendatang, meskipun Support XY mungkin tidak terlalu tinggi, atau XY tidak terlalu populer. Untuk mengembangkan sistem rekomendasi, metrik Lift biasanya menjadi pertimbangan, selain Confidence dan Support dari Aturannya.

#	RowID	Support Number (double)	Confidence Number (double)	Lift ↓ Number (double)	Consequent String	Implies String	Items Ser
31	rule30	0.133	1	7.5	snack_2	<--	[teh,snack_1]
33	rule32	0.133	1	7.5	pinsil	<--	[pulpen,buku,penggaris]
35	rule34	0.133	1	7.5	penggaris	<--	[pulpen,buku,pinsil]
27	rule26	0.133	1	5	sampo	<--	[sendal,sabun]
29	rule28	0.133	1	5	teh	<--	[snack_2,snack_1]
34	rule33	0.133	1	5	pulpen	<--	[buku,penggaris,pinsil]
19	rule18	0.133	1	3.75	sendal	<--	[kaos,kaki]
20	rule19	0.133	0.5	3.75	kaos_kaki	<--	[sendal]

Gambar 2.11 AR yang Ditemukan

3. **Column Aggregator**, bersifat opsional namun digunakan di sini untuk mengagregasi Kolom **Items** sebagai Anteseden., dengan metode Agregasi "Elemen Count" dengan tujuan agar dapat diketahui Jumlah elemen di dalam Items. Parameter yang digunakan diatur seperti dalam Gambar 2.12, dan hasil dari pengaturan ini diberikan dalam Gambar 2.13.



Gambar 2.12 Setting Parameter di Column Aggregator

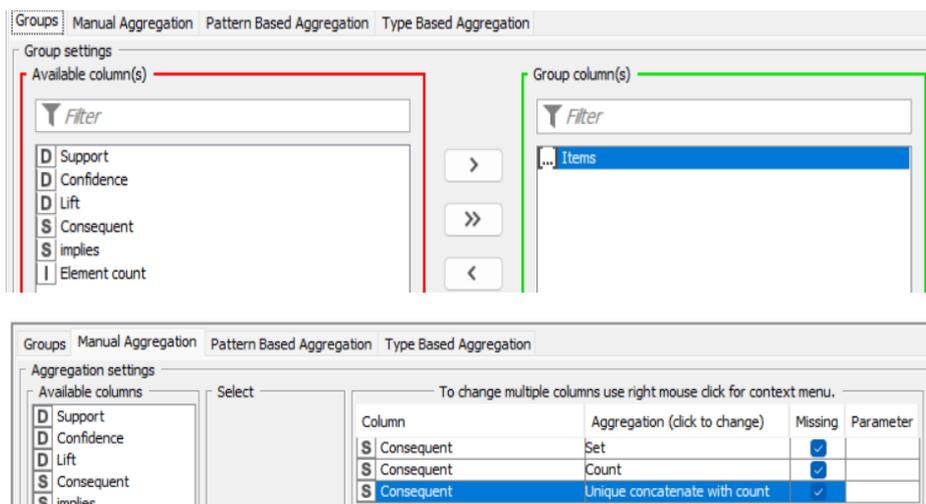
► 1: Table with aggregation columns Flow Variables

Rows: 41 | Columns: 7

#	RowID	Support Number (double)	Confidence Number (double)	Lift Number (double)	Consequent String	implies String	Items Set	Element co... Number (integer)
32	rule31	0.133	1	3.75	buku	<--	[pulpen,penggaris,p...	3
33	rule32	0.133	1	7.5	pinsil	<--	[pulpen,buku,pengg...	3
34	rule33	0.133	1	5	pulpen	<--	[buku,penggaris,pin...	3
35	rule34	0.133	1	7.5	penggaris	<--	[pulpen,buku,pinsil]	3
23	rule22	0.133	1	2.143	kopi	<--	[gula,sabun]	2
24	rule23	0.133	0.667	1.667	gula	<--	[kopi,sabun]	2

Gambar 2.13 Hasil dari Agregasi Kolom

4. **GroupBy**, juga merupakan node opsional dan digunakan di sini setelah Column Aggregator untuk mengelompokkan Konsekuen yang berbeda berdasarkan Item yang sama, seperti dalam Gambar 2.14. Konsekuen diagregasi dalam tiga metode: *Set*, untuk membentuk himpunan konsekuen dari tiap Item; *Count*, untuk menghitung jumlah konsekuen dalam himpunan; dan *Unique Concatenate with Count*, untuk menyambungkan (sebagai string) tiap anggota dalam Konsekuen dengan jumlahnya.



Gambar 2.14 Seting Form Dialog GroupBy

Hasilnya diberikan dalam Gambar 2.15. Di sini dapat diketahui bahwa Items Kopi, memiliki enam item di dalam Konsekuennya, yaitu beras, sapu, teh, snack_1, sabun, dan gula. Dalam sistem rekomendasi, ketika seseorang pernah membeli Item Kopi, maka item lain yang dapat ditawarkan kepadanya adalah enam item Konsekuensi tersebut.

► 1: Group table Flow Variables

Rows: 26 | Columns: 4

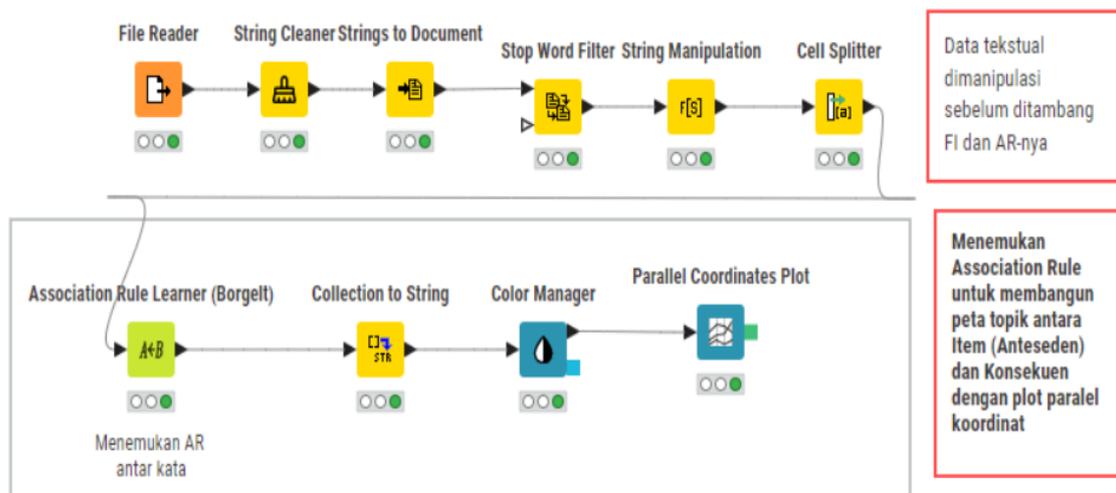
#	RowID	Items Set	Consequent (Set) Set	Conse... Number (integ...)	Consequent (Unique concatenate with count) String
9	Row8	[kopi]	[beras,sapu,teh,...]	6	beras(1), sapu(1), teh(1), snack_1(1), sabun(1), gula(1)
6	Row5	[gula]	[susu,teh,snack_1,...]	4	susu(1), teh(1), snack_1(1), kopi(1)
14	Row13	[sabun]	[buku,sapu,kopi,...]	4	buku(1), sapu(1), kopi(1), sampu(1)
21	Row20	[snack_1]	[kopi,gula,roti]	3	kopi(1), gula(1), roti(1)
17	Row16	[sapu]	[kopi,sabun]	2	kopi(1), sabun(1)
26	Row25	[teh]	[kopi,gula]	2	kopi(1), gula(1)

Gambar 2.15 Hasil GroupBy

Topic Mapping dari Hasil Mesin Pencari

Masalah bisnis yang diangkat adalah "Bagaimana membangun peta topik yang terkandung di dalam hasil pencarian mesin pencari online seperti Google?". Untuk masalah ini, data yang dibutuhkan adalah data teks hasil Google, misalnya untuk kata kunci "Data Science dan Mining".

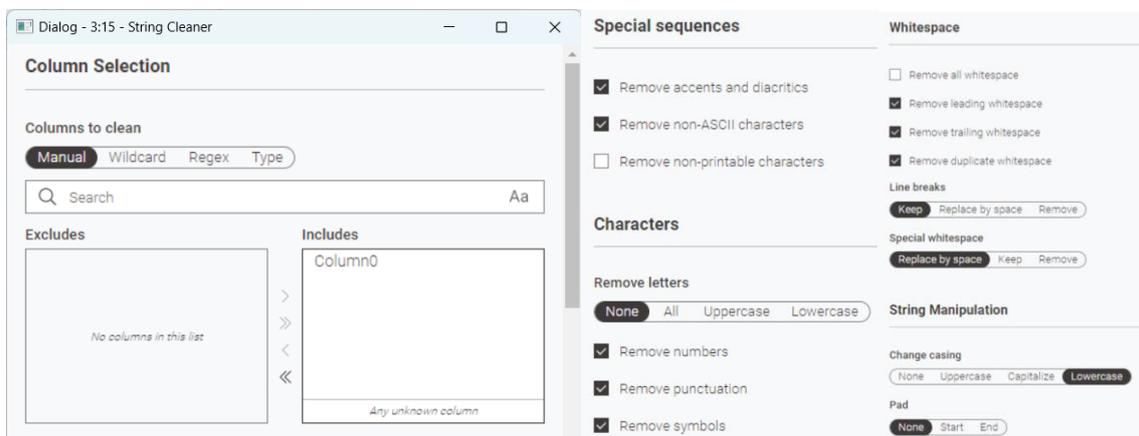
Sebagai solusinya, dibangun sebuah workflow yang menggunakan nodes untuk menemukan AR, dan memetakan topik dengan plot koordinat paralel. Topik dimaksud adalah hubungan antara keyword yang diekstrak dari hasil pencarian Google, dan ditambah menggunakan nodes untuk menemukan AR. Workflow dimaksud diberikan dalam Gambar 2.16.



Gambar 2.16 Workflow Pembuatan Topic Mapping

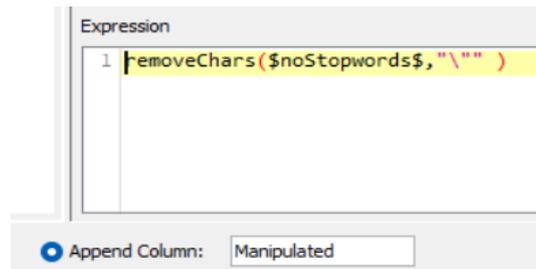
Node yang digunakan adalah:

1. **File Reader**, membaca hasil pencarian dalam bahasa Inggris yang disimpan dalam `searching-en.txt`. Hasil pembacaan file biasanya diletakkan di **Column0**
2. **String Cleaner**, berguna untuk membersihkan string di kolom **Column0** dengan beberapa opsi seperti *Remove accent and diacritics*, yaitu tanda aksan di alfabet nama orang Eropa (mis. *Á, â*), *Remove non-Ascii Character*; kemudian *Remove numbers, punctuation* dan *symbols* untuk membuang angka, tanda baca dan simbol seperti tagar, dolar dan lainnya; kemudian perlakuan terhadap spasi kosong di depan dan belakang kata; lalu ada pengaturan terhadap line breaks (baris baru), dan Case pada kata seperti Uppercase, Lowercase dan Capitalize.



Gambar 2.17 Form Dialog String Cleaner

3. **Strings to Document**, mengubah tipe data pada **Column0** dari String menjadi Document. Opsi yang perlu diatur adalah nama kolom hasil, misalnya **Docs**, dan jenis tokenisasi yang diinginkan. Di sini dipilih *English Word Tokenizer*. Dari hasil pemrosesan Node ini, terlihat bahwa `Column0` yang sebelumnya bertipe String, sekarang menjadi Text Document di dalam kolom **Docs**
4. **Stop Word Filter**, membuang kata-kata yang tidak diperlukan dalam analisis berdasarkan kamus data tertentu. Di sini digunakan built-in list untuk bahasa "English", dan diterapkan di kolom yang diinginkan yaitu **Docs**
5. **String Manipulation**, memanipulasi String di dalam Docs sehingga tidak memiliki tanda kutip quotes di depan dan di belakang tiap barisnya. Masukkan Ekspresi sebagaimana dalam Gambar 2.17, di form konfigurasi Node ini. Hasil dari penerapan node String Manipulation, Stop Word Filter dan String to Document diberikan di Gambar 2.18 yaitu di kolom `Manipulated`, dimana tanda quotes sudah hilang, namun tipe data kembali menjadi String.



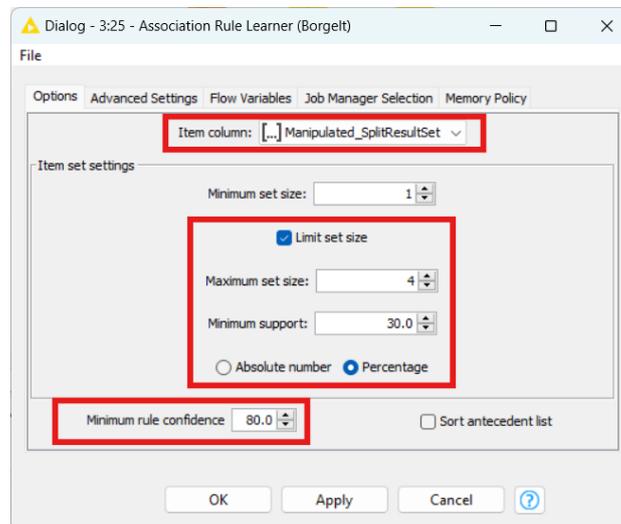
Gambar 2.17 Ekspresi yang Digunakan untuk removeChars

The image shows a table with three columns: 'Docs' (Text document), 'noStopwords' (Text document), and 'Manipulated' (String). The first four rows of data are visible, showing text snippets from documents and their corresponding manipulated string representations.

Docs	noStopwords	Manipulated
"data science data science is a field o...	"data science data science field dom...	data science data science field domai...
"data science an interdisciplinary field...	"data science interdisciplinary field da...	data science interdisciplinary field da...
"data science is an interdisciplinary fi...	"data science interdisciplinary field co...	data science interdisciplinary field co...
"data mining and data science have b...	"data mining data science ubiquitous ...	data mining data science ubiquitous t...

Gambar 2.18 Hasil penerapan Beberapa Nodes pengolah Document dan String

6. Cell Splitter, memisahkan string di kolom Manipulated menjadi Set kata-kata menggunakan pemisah (delimiter) sebuah spasi kosong; spasi kosong di depan kata-kata yang sudah terpisah dibuang (trim), dan pilih 'As Set' sebagai array penampung hasilnya, karena tidak diinginkan adanya duplikasi elemen. Kolom terakhir dapat disebut sebagai "Bag of Words", berisi kata-kata unik di tiap barisnya.
7. Association Rule Learner (Borgelt), berfungsi untuk menghasilkan AR dari Set kata-kata. Di sini digunakan parameter minsup 30% dan minconf 80%, serta panjang itemset 1 sampai 4 item. Perlu dicatat bahwa itemset di posisi konsekuen selalu 1 item, sedangkan di posisi anteseden dapat diatur berbeda. Form Dialog node ini diberikan di Gambar 2.19, sedangkan beberapa hasil penemuan AR diberikan di Gambar 2.20. Parameter tersebut menghasilkan 921 Rules.



Gambar 2.19 Form Dialog Association Rule Learner (Borgelt)

Rows: 921 | Columns: 11 Table Statistics

<input type="checkbox"/>	#	RowID	Consequ... String	Antec... Set	ItemSets... Number (inte...)	Relative... Number (dou...)	RuleConfi... Number (dou...)	Absolute... Number (dou...)	RelativeB... Number (dou...)	RuleLift Number (dou...)	RuleLift% Number (dou...)
<input type="checkbox"/>	50	Row49	knowledge	[unstructured]	3	30	100	3	30	2	200
<input type="checkbox"/>	66	Row65	extract	[unstructured]	3	30	100	3	30	2	200
<input type="checkbox"/>	74	Row73	insights	[unstructured]	3	30	100	3	30	2	200

Gambar 2.20 Hasil penemuan AR

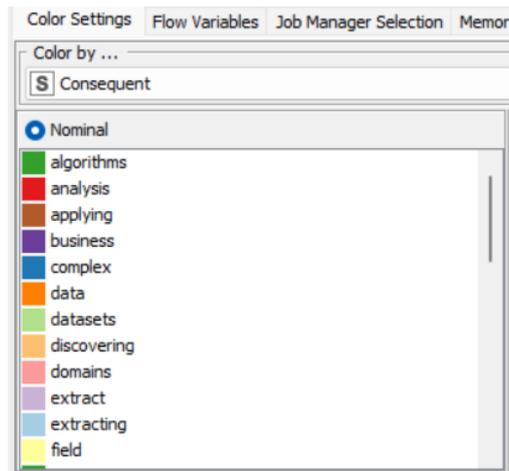
- Collection to String, digunakan untuk mengubah Anteseden yang berbentuk Set, menjadi String. Tujuannya adalah agar dapat diproses di Node Paralel Coordinates Plot. Hasil yang diharapkan adalah sebuah string, tanpa tanda pembuka (Prefix kosong) dan penutup (Suffix kosong), serta antar kata dipisahkan (Separator) dengan koma. Form dialog diberikan di Gambar 2.21, sedangkan hasilnya di Gambar 2.22.

Gambar 2.21 Form Dialog Collection to String

<input type="checkbox"/>	#	RowID	Consequ... String	Antecedent Set	Antecedent (String) String
<input type="checkbox"/>	1	Row0	business	[analytics,science,data]	analytics, science, data
<input type="checkbox"/>	2	Row1	business	[analytics,science]	analytics, science

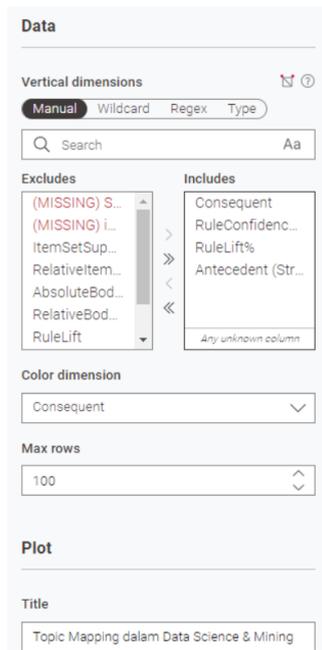
Gambar 2.22 Hasil Collection to String

- Color Manager, digunakan untuk memberikan warna pada tiap kata di dalam Consequent string. Di form dialog node ini sebagaimana Gambar 2.23, juga tersedia palet warna lainnya.



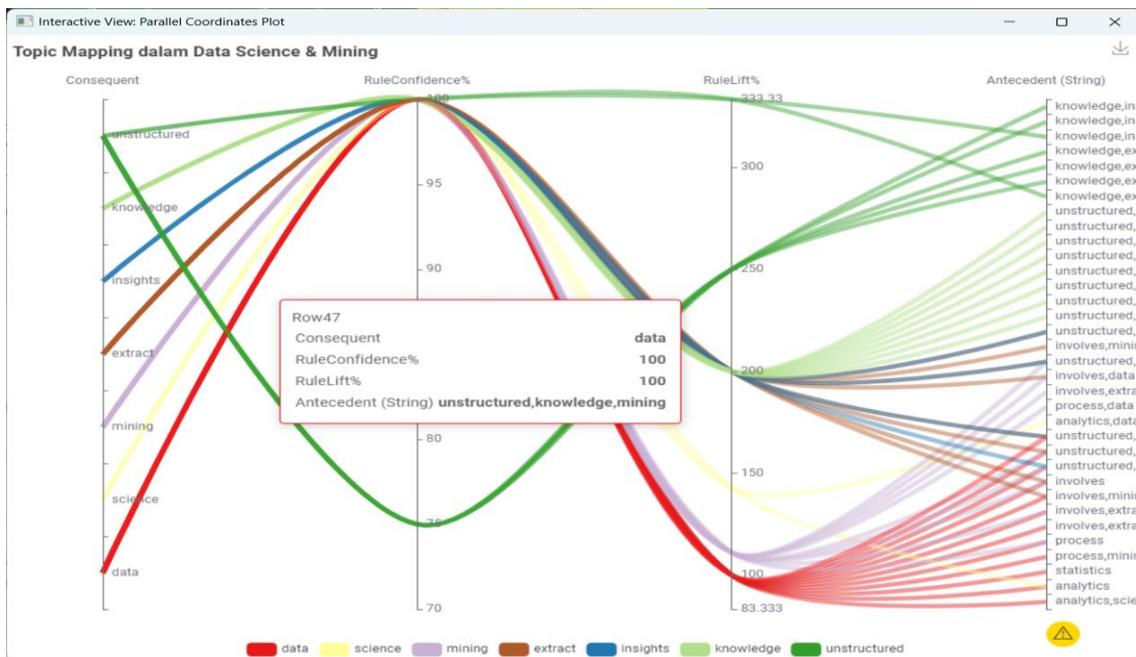
Gambar 2.23 Form Dialog Color Manager

10. Paralel Coordinates Plot, yaitu plot kordinat paralel yang menghubungkan tiap variabel atau fitur dengan garis dengan suatu nilai tertentu. Di sini didemonstrasikan hubungan antara kata-kata di Consequent dengan yang di Anteseden, berserta nilai confidence dan liftnya. Pengaturan lain yang disediakan adalah lebar garis, maksimum baris yang divisualisasi, dan bentuk garis. Form dialog node ini diberikan dalam Gambar 2.24, sedangkan hasil plot-nya diberikan dalam Gambar 2.25



Hasil plot koordinat paralel antara anteseden dan konsekuen ini sekaligus memvisualisasikan Peta Topik yang terkandung di dalam hasil pencarian Google “Data Science dan Mining”. Satu catatan untuk pengembangan ke depan, adalah bahwa topic mapping dengan AR mampu menunjukkan hubungan tersembunyi antara kata-kata kunci yang diekstrak, yaitu kemungkinan adanya Lift positif dan negatif antara anteseden dan konsekuen. Sedangkan metode Topic mapping lain

yang sudah umum seperti Latent Dirichlet Allocation (LDA) hanya memetakan topik berdasarkan kumpulan kata saja.



Gambar 2.25 Topic Mapping menggunakan Plot Kordinat Paralel

Kesimpulan

Association Rule Mining adalah teknik data mining yang powerful untuk menemukan hubungan tersembunyi antara item-item atau kata-kata dalam dataset besar, baik data transaksi maupun data tekstual. Dengan memahami konsep dasar, langkah-langkah, dan penerapan ARM, kita dapat mengungkap wawasan berharga yang dapat digunakan untuk meningkatkan pengambilan keputusan dan mencapai tujuan bisnis atau penelitian Anda di era Society 5.0, dimana semua transaksi digital yang kita ciptakan sudah waktunya dikembalikan kepada kita lagi dalam bentuk rekomendasi yang berharga.

Meskipun ARM memiliki banyak manfaat, penting untuk diingat bahwa teknik ini juga memiliki keterbatasan. Misalnya, ARM mungkin menghasilkan banyak aturan yang tidak relevan atau sulit diinterpretasikan. Oleh karena itu, penting untuk melakukan pemilihan dan pengaturan metrik (Support, Confidence, Lift dan lainnya) secara seksama, serta mempertimbangkan konteks masalah saat menginterpretasikan hasil analisis.

Referensi:

Adistia, L. D., Akhriza, T. M., & Jatmiko, S. (2019). Sistem Rekomendasi Buku untuk Perpustakaan Perguruan Tinggi Berbasis Association Rule. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 3(2). <https://doi.org/10.29207/resti.v3i2.971>

Agrawal, R., Imieliński, T., & Swami, A. (1993). Mining association rules between sets of items in large databases. *ACM SIGMOD Record*. <https://doi.org/10.1145/170036.170072>

- Agrawal, R., & Srikant, R. (1994). Fast Algorithms for Mining Association Rules. *The 20th International Conference on Very Large Data Bases*.
- Akhriza, T. M., Ma, Y., & Li, J. (2015). A novel Fibonacci windows model for finding emerging patterns over online data stream. *2015 International Conference on Cyber Security of Smart Cities, Industrial Control System and Communications, SSIC 2015 - Proceedings*. <https://doi.org/10.1109/SSIC.2015.7245323>
- Akhriza, T. M., Ma, Y., & Li, J. (2017). Revealing the Gap Between Skills of Students and the Evolving Skills Required by the Industry of Information and Communication Technology. *International Journal of Software Engineering and Knowledge Engineering*, 27(05), 675-698. <https://doi.org/10.1142/s0218194017500255>
- Akhriza, T. M., Ma, Y., & Li, J. (2018). Novel Push-Front Fibonacci Windows Model for Finding Emerging Patterns with Better Completeness and Accuracy: *ETRI Journal*, 40(1). <https://doi.org/10.4218/etrij.18.0117.0175>
- Akhriza, T. M., & Mumpuni, I. D. (2022). A Time-Window Approach to Recommending Emerging and On-the-rise Items. *2022 Seventh International Conference on Informatics and Computing (ICIC)*.
- Akhriza, T. M., & Utsalina, D. S. (2023). STIMATA Rule Adviser: Sistem Rekomendasi Produk e-Commerce. *IAII SISFOTEK VII*.
- Husni, M., Akhriza, T. M., Madenda, S., & Wibowo, E. P. (2022). Improving Recentness of the ICT Book Recommendation using an Adaptive Rules-based Recommender System. *International Journal of Computer Applications in Technology*, (In Press).
- Ju, C., Bao, F., Xu, C., & Fu, X. (2015). A Novel Method of Interestingness Measures for Association Rules Mining Based on Profit. *Discrete Dynamics in Nature and Society*. <https://doi.org/10.1155/2015/868634>
- Latifah, Akhriza, T. M., & Adistia, L. D. (2020). Constructing Recommendation about Skills Combinations Frequently Sought in IT Industries Based on Apriori Algorithm. *Advances in Computer Science Research*, 95. <https://doi.org/10.2991/miseic-19.2019.12>