Topic Discovery pada Jurnal-jurnal di *IEEE Explore* menggunakan *Association Rule Mining* dengan Pendekatan *Closed Frequent Itemset*

Reza Mustofa dan Irhamah
Departemen Statistika, Fakultas Matematika, Komputasi & Sains Data,
Institut Teknologi Sepuluh Nopember (ITS)
e-mail: irhamah@statistika.its.ac.id

***Abstrak*—Menemukan topik dari koleksi dokumen seperti publikasi ilmiah mempunyai banyak manfaat. Dengan semakin banyaknya dokumen teks yang dihasilkan di web dan arsip-arsip digital, *Topic Discovery* menjadi alat yang sangat penting untuk menelusuri, meringkas, dan mengelompokkan dokumen. Salah satu penerapan *Association Rule Mining* adalah digunakan untuk menemukan topik dalam suatu dokumen dengan cara mencari pola yang sering muncul pada semua dokumen. Data diambil dari *IEEE Xplore* yang merupakan kumpulan abstrak dari jurnal-jurnal di *International Conference on Data Mining (ICDM)* dan *International Conference on Data Engineers (ICDE)* dari tahun 2009-2018. Masing-masing abstrak direpresentasikan sebagai transaksi sedangkan kata *keywords* yang terkandung didalamnya direpresentasikan sebagai item. Kombinasi antar kata *keywords* yang paling sering muncul, yang disebut *frequent itemset*, akan digunakan sebagai kandidat dari suatu topik. Algoritma yang dapat digunakan untuk membangkitkan itemset adalah algoritma *Apriori* dan *ECLAT*. Waktu eksekusi perolehan *frequent itemset* dari *ECLAT* lebih cepat dari *Apriori*. *Closed frequent itemset* juga mampu mengurangi frequent itemset yang terbentuk, sehingga Topik yang terbentuk merupakan Topik yang unik.**

Kata Kunci—Apriori Algorithm, Association Rule, Closed Frequent Itemset, Eclat Algorithm, Network Analysis, Text Mining.

I. PENDAHULUAN

BANYAKNYA publikasi ilmiah atau jurnal dari tahun ke tahun selalu mengalami kenaikan. Jurnal-jurnal tersebut menjadi sumber informasi yang berharga bagi mahasiswa maupun peneliti yang akan membuat penelitian. Akan tetapi banyaknya koleksi-koleksi jurnal di internet menjadikan tantangan tersendiri dalam pencarian dokumen. Bagaimana mencari dan menemukan dokumen yang layak untuk dibaca adalah pertanyaan bagi setiap peneliti yang ingin mencari dokumen. Salah satu cara menemukan dokumen yang tepat yaitu dengan mencari dokumen yang mempunyai topik yang sama. *Topic Discovery* bertujuan untuk mengekstraksi pola-pola yang bermakna dari dokumen teks.

Association Rule dapat digunakan untuk menemukan Topik-topik yang bermakna dalam suatu dokumen. *Association rule* merupakan salah satu metode yang bertujuan mencari pola atau *pattern* pada ukuran *databese* berskala besar. *Association rule* identik dengan *Market Basket Analysis* yang bertujuan untuk mencari item/barang apa saja yang sering dibeli bersamaan dalam satu kali transaksi. Dalam penerapan *Topic discovery*, dokumen direpresentasikan sebagai transaksi sedangkan kata-kata yang terkandung didalamnya direpresentasikan sebagai item. Sehingga kombinasi antar kata yang paling sering muncul, yang disebut *frequent itemset*, digunakan sebagai kandidat suatu topik.

Salah satu algoritma pada *association rule mining* yang berguna untuk membangkitkan *frequent itemset* adalah algoritma *Apriori*. Algoritma *Apriori* paling sering digunakan dan merupakan algoritma paling awal [1]. Algoritma yang lain yaitu algoritma *ECLAT* merupakan pengembangan dari *apriori* yang menggunakan *vertical databese layout* [2]. Dalam rangka penentuan Topik yang bermakna, *frequent itemset* menghasilkan *itemset* yang terlalu banyak padahal sesungguhnya memiliki makna yang sama untuk itu digunakanlah pendekatan *closed frequent itemset*. *Closed frequent itemset* adalah *frequent itemset*, *I*, dimana tidak terdapat superset *I* yang memiliki nilai support yang sama dengan *I*.

Penelitian sebelumnya mengenai pencarian topik yang bermakna pada suatu dokumen pernah dilakukan oleh Shubhankar [3] yang menggunakan pendekatan *closed frequent itemset* untuk mendeteksi topik. *Closed frequent itemset* digunakan untuk mengurangi jumlah *frequent itemset* yang dihasilkan dan jumlah *association rule* yang dibangkitkan. Data yang digunakan pada penelitian tersebut berupa judul-judul penelitian sedangkan Hurtado, et al. [4] menggunakan judul dan isi abstrak dalam mendeteksi topik. Akan tetapi menggunakan *all frequent itemset* dengan algoritma *apriori* untuk menentukan kandidat topik yang bermakna.

Penelitian ini dilakukan dengan mengambil data *keywords* dari suatu abstrak. penerapan *Association Rule* dalam pencarian Topik dengan menggunakan dua pendekatan. Pendekatan yang pertama yaitu *Association Rule* yang diterapkan pada kata yang bertujuan untuk mendeteksi Topik, dan juga menggunakan c*losed frequent itemset* untuk mengeliminasi Topik yang mempunyai makna yang sama. Pendekatan yang kedua yaitu *Association Rule* yang diterapkan pada *keywords*. Pendekatan kedua bertujuan untuk menelusuri keterkaitan masing-masing *keywords* dengan *keywords* yang lain.

II. TINJAUAN PUSTAKA

A. Text Preprocessing

Text preprocessing bertujuan untuk mengubah data textual yang tidak berstruktur ke dalam data yang terstruktur dan disimpan dalam basis data. Adapun tahapan-tahapan praproses data adalah sebagai berikut

a. *Case Folding* dan *Remove punctuation*
   Case folding dilakukan untuk mengubah seluruh huruf menjadi *lowercase*. *Remove punctuation* digunakan untuk menghilangkan tanda baca seperti tanda koma (,).
b. *Removing stopword*

Stopword merupakan kata yang sering muncul dalam dokumen seperti "between", "and", "this", "on", "an", "a", "the", dll. Kata-kata yang masuk dalam stopword seringkali dianggap tidak memiliki makna, sehingga kata yang tercantum dalam daftar ini dibuang dan tidak ikut diproses pada tahap selanjutnya.

c. *Lemmatization*
Lemmatization hampir sama seperti *stemming*, yang membedakan adalah pada *stemming* lebih banyak memotong akhir kata, dan sering juga membuang imbuhan tetapi pada *lemmatization* menghasilkan kata dasar dengan memperhatikan kamus. Sebagai contoh kata "studies" pada *stemming* menghasilkan output "studi" sedangkan pada *lemmatization* menghasilkan output "study".

d. Tokenisasi
Tokenisasi adalah proses untuk membagi teks input menjadi unit-unit kecil yang disebut token [5]. Token atau biasa disebut juga term bisa berupa suatu kata, angka atau tanda baca. Pada penelitian ini tanda baca dihilangkan sehingga tidak dianggap sebagai token.

B. *Association Rule*

Association rule merupakan salah satu metode yang bertujuan mencari pola yang sering muncul di antara banyak transaksi, dimana setiap transaksi terdiri dari beberapa item. Misalkan $\omega = \{w_1, w_2, \ldots, w_n\}$ yang terdiri dari *n* kata (item) dan *databese* $\delta = \{d_1, d_2, \ldots, d_m\}$ terdiri dari *m* dokumen (transaksi). Maka masing-masing dokumen terdiri dari kumpulan kata-kata yang disebut *itemset*, dinotasikan dengan I ($I \in \omega$). Sedangkan i*temset* yang mempunyai item sebanyak *k*, disebut *k-itemset* [6].

Langkah pertama pada *association rule* adalah menghasilkan semua *itemset* yang memungkinkan, kemungkinan *itemset* yang muncul pada *m-item* adalah 2^m. Parameter yang digunakan untuk membangkitkan itemset yaitu nilai *minimum support*. *Support* merupakan persentase dokumen yang mengandung *itemset* I_1 dan I_2,

$$Support(I_1 \rightarrow I_2) = \frac{\text{Banyak dokumen yang mengandung kata } I_1 \text{ dan } I_2}{\text{Total dokumen}}$$

Karena besarnya komputasi untuk menghitung *frequent itemset*, yang membandingkan setiap kandidat *itemset* dengan setiap transaksi, maka ada beberapa pendekatan untuk mengurangi komputasi tersebut, yaitu dengan menggunakan algoritma *Apriori* dan *Eclat*.

1. *Apriori algorithm*

Algoritma *Apriori* pertama kali diperkenalkan oleh Agrawal & Shrikant [1] yang berguna untuk menemukan *frequent itemset* pada sekumpulan data. Algoritma ini merupakan algoritma yang paling awal dan paling sering digunakan untuk *Association Rule*. Cara algoritma ini bekerja adalah algoritma akan menghasilkan kandidat baru dari *k-itemset* dari *frequent itemset* pada langkah sebelumnya dan menghitung nilai *support k-itemset* tersebut. *Itemset* yang memiliki nilai support di bawah dari *minimal support* akan dihapus. Algoritma berhenti ketika tidak ada lagi *frequent itemset* baru yang dihasilkan. Algoritma *Apriori* menggunakan metode pencarian *breadth-first search* (*BFS*) dalam membangkitkan kandidat *itemset*.

2. *ECLAT algorithm*

Algoritma *ECLAT* merupakan pengembangan dari *Apriori*. Algoritma *ECLAT* menggunakan *vertical databese layout*, berbeda dengan algoritma apriori yang menggunakan *horizontal databese layout*. Setiap *item* dinyatakan dalam tabel *tid-list* secara vertikal (Gambar 1) dan menggunakan titik potong *tid-list* antar-item untuk menghitung *support*. *Eclat* hanya akan memeriksa (*scan*) dataset sebanyak satu kali, tidak melakukannya berulang-ulang karena menggunakan *vertical layout*, sehingga *tid-list* sudah memberikan informasi tentang *support count* dari itemset. Algoritma *ECLAT* menggunakan metode pencarian *Depth-First Search* (DFS) dalam membangkitkan kandidat *itemset*.

TID	*Item*
1	A, B, C
2	A, B, D, E
3	D, E, F
4	A, B
5	A, B, D, E, G

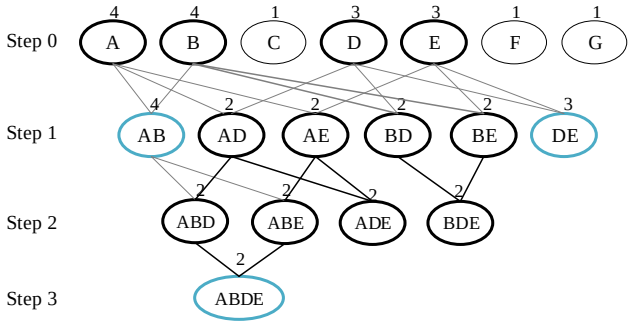
→

A	B	C	D	E	F	G
1	1	1	2	2	3	5
2	2		3	3		
4	4		5	5		
5	5					

Gambar 1. *Horizontal vs Vertical databese layout*

C. *Closed Frequent Itemset*

Closed frequent itemset digunakan untuk mengurangi jumlah *frequent itemset* yang dihasilkan dan jumlah *association rule* yang dibangkitkan. *Closed frequent itemset* adalah *frequent itemset*, *I*, dimana tidak terdapat superset *I* yang memiliki nilai support yang sama dengan *I*. Pada penelitian kali ini, Topik didapat dari rangkaian set kata kunci yang sering muncul (*frequent itemset*). Dalam *frequent itemset* seringkali didapatkan set kata kunci Topik yang *non*-closed, sebagai contoh *association mining* dan *association rule mining* mungkin memandung makna Topik yang sama. Dalam hal ini kita dapat menghapus Topik *association mining* dan *association rule mining* merupakan *closed frequent itemset*. Topik harus terdiri dari jumlah maksimum kata kunci umum yang ada di semua dokumen, sehingga *closed frequent itemset* digunakan sebagai kandidat topik. Sebagai ilustrasi, itemset dengan garis oval biru pada Gambar 2 merupakan closed frequent itemset.



Gambar 2. *Closed Frequent itemset.*

D. *Analisis Korelasi dan Topic Community*

Topik-topik yang dihasilkan dengan *association rule* dapat dihitung koefisien korelasi untuk menghitung sejauh mana hubungan antara topik satu dengan topik lainnya. Salah satu metode untuk menghitung korelasi yang paling sering digunakan yaitu *Pearson correlation* [7]. Misalkan $I_1 = \langle x_{11}, x_{12}, \ldots, x_{1k} \rangle$ dan $I_2 = \langle x_{21}, x_{22}, \ldots, x_{2k} \rangle$ maka rumus untuk menghitung koefisien korelasi *Pearson* antara Topik I_1 dan I_2 dinyatakan dalam persamaan (2.3)

$$\hat{\rho}(I_1, I_2) = \frac{\sum_{q=1}^{k}(x_{1q} - \bar{x}_1)(x_{2q} - \bar{x}_2)}{\sqrt{\sum_{q=1}^{k}(x_{1q} - \bar{x}_1)^2 \times \sum_{q=1}^{k}(x_{2q} - \bar{x}_2)^2}} \quad (1)$$

Nilai koefisien korelasi ini digunakan untuk membentuk *Network graph*, dengan masing-masing *node* merupakan topik dan *edge* yang menghubungkan dua *node* merupakan korelasi antar dua topik yang signifikan. Untuk mengetahui mana saja Topik yang mempunyai hubungan secara signifikan, dapat dilakukan pengujian korelasi dengan hipotesisnya adalah [7]

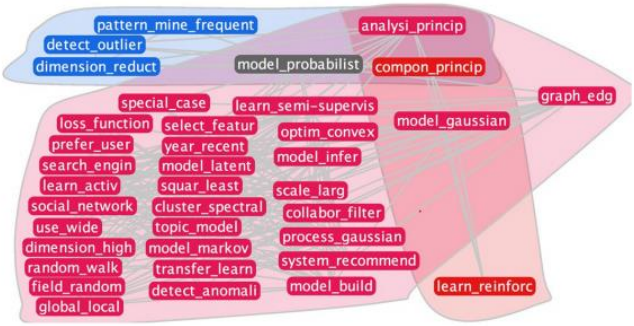
$H_0 : \rho = 0$

$H_1 : \rho \neq 0$

Statistik uji yang digunakan adalah

$$t_{hitung} = \frac{r\sqrt{n-2}}{\sqrt{1-r^2}}. \quad (2)$$

H_0 ditolak apabila nilai statistik uji $|t_{\text{hitung}}| > t_{\alpha/2,(n-2)}$ dengan n merupakan banyaknya pengamatan. Setelah semua topik (*node*) dihubungkan dengan masing-masing *edge*, maka langkah selanjutnya yaitu mencari *Community* dengan menggunakan algoritma *Community detection* berbasis CPM (*Clique Percolation Method)*. CPM adalah metode untuk menemukan *community* / klaster yang saling *overlapping*. CPM pertama-tama mengidentifikasi semua *clique* dengan ukuran *k* (ditentukan oleh user) dari *network*. Dengan menggunakan definisi *clique adjacency*, CPM menganggap dua *clique* sebagai *adjacent* jika *clique* tersebut berbagi k-1 *node*. Suatu community didefinisikan sebagai gabungan dari semua *k-clique* yang dapat dicapai satu sama lain melalui *adjacent k-clique*. Gambar 3 merupakan contoh dari hasil dari Topic Community yang dilakukan oleh Hurtado, et al [4].



Gambar 3. *Topic Community*.

E. *Regresi Time Series*

Regresi dalam konteks time series memiliki bentuk yang sama dengan regresi linier umum. Dengan mengasumsikan output atau bentuk dependen X_t, untuk t = 2009, 2010,…,2018, yang dipengaruhi oleh kemungkinan data input atau independen, dimana input pertama diketahui, hubungan ini dapat ditunjukkan dengan model regresi linier [8]. Jika data X_t memiliki trend, model regresi dapat ditulis sebagai berikut [9]:

$$X_t = \delta_t + a_t \quad (3)$$

dimana,

X_t : data pengamatan pada periode t

δ_t : komponen *trend* pada periode t

a_t : komponen *error* pada periode t

III. METODOLOGI PENELITIAN

Penelitian ini menggunakan data sebanyak 3.374 asbtrak dari koleksi *International Conference on Data Mining* (ICDM) [10] dan *International Conference on Data Engineers* (ICDE) [11] di *IEEE Xplore Digital Library* dari tahun 2009 sampai 2018. Dokumen yang digunakan sebagai data adalah kata kunci atau *keywords* yang terdapat di Abstrak.

A. *Struktur Data*

Data yang digunakan pada pencarian topik menggunakan *association rule* yaitu kata kunci atau *keywords* yang terdapat di abstrak. *Keywords* yang sudah melalui tahap *preprocessing* akan ditransformasi ke dalam bentuk *document term matrix* dengan struktur seperti Tabel 1.

Tabel 1.
Document term matrix

Dokumen (*d*) ke-	Kata (*w*) ke-					
	1	2	⋯	*j*	⋯	*n*
1	f_{11}	f_{12}	⋯	f_{1j}	⋯	f_{1n}
2	f_{21}	f_{22}	⋯	f_{2j}	⋯	f_{2n}
⋮	⋮	⋮	⋱	⋮	⋱	⋮
i	f_{i1}	f_{i2}	⋯	f_{ij}	⋯	f_{in}
⋮	⋮	⋮	⋱	⋮	⋱	⋮
m	f_{m1}	f_{m2}	⋯	f_{mn}	⋯	f_{mn}

keterangan :

m = banyaknya dokumen abstrak,

n = banyaknya term/kata pada semua dokumen abstrak,

$f_{ij} = \begin{cases} 0, & \text{jika kata ke-}j\text{ tidak muncul pada dokumen ke-}i \\ 1, & \text{jika kata ke-}j\text{ muncul pada dokumen ke-}i \end{cases}$

Setelah melalui tahap pencarian topik menggunakan *association rule* maka Topik-topik yang terbentuk disusun kedalam matriks dimana setiap Topik direpresentasikan sebagai vektor dengan dimensinya yaitu frekuensi per tahun, Matriks tersebut digunakan untuk mencari korelasi antar Topik-topik. Struktur dari *Topic per years matrix* diberikan pada Tabel 2

Tabel 2.
Topic per year Matrix

Topik (*I*)	Tahun ke-					
	2009	2010	⋯	*q*	⋯	2018
1	$x_{1,2009}$	$x_{1,2010}$	⋯	$x_{1,q}$	⋯	$x_{1,2018}$
2	$x_{2,2009}$	$x_{2,2010}$	⋯	$x_{2,q}$	⋯	$x_{2,2018}$
⋮	⋮	⋮	⋱	⋮	⋱	⋮
p	$x_{p,2009}$	$x_{p,2010}$	⋯	$x_{p,q}$	⋯	$x_{p,2018}$
⋮	⋮	⋮	⋱	⋮	⋱	⋮
b	$x_{b,2009}$	$x_{b,2010}$	⋯	$x_{b,q}$	⋯	$x_{b,2018}$

keterangan :

b = banyaknya Topik yang terbentuk

$x_{p,q}$ = frekuensi dokumen yang mempunyai topik ke-*p* pada tahun ke-*q*

B. *Langkah Analisis*

Langkah analisis yang dilakukan untuk mencapai tujuan yang diharapkan dalam penelitian ini sesuai permasalahan yang telah dirumuskan.

1. Menyiapkan data *keywords*
2. Pencarian Topik dengan *Association Rule* menggunakan *item* per-kata
   a. Melakukan *preprocessing data*, dalam tahap *preprocessing data* terdapat tahap-tahap yang dilakukan yaitu :
      i. Melakukan *case folding*, yaitu mengubah semua teks dengan huruf kecil,
      ii. Menghapus tanda baca koma (,),
      iii. Menghapus kata yang mengandung *stopwords* "and",
      iv. Melakukan *lemmatization* untuk mendapatkan kata dasar pada kata benda (*noun*),
      v. Melakukan *tokenizing* untuk memecah *keywords* menjadi kata per kata,
   b. Membentuk *document term matrix*, yaitu merepresentasikan masing-masing kata ke dalam bentuk vektor dimana setiap dimensi sesuai dengan *term* atau kata dan nilainya merupakan bilangan biner yang menunjukkan apakah kata tersebut muncul atau tidak,
   c. Mencari Topik dari *document term matrix* menggunakan *frequent itemset mining*,
      i. Membangkitkan *frequent itemset* menggunakan algoritma *Apriori* dan *ECLAT*
      ii. Membandingkan performa dari algoritma *Apriori* dan *ECLAT* dalam membangkitkan *frequent itemset* menggunakan kriteria *minimum support* 10%, 7,5%, 5%, 2,5%, 1%, 0,75%, 0,5%, 0,25%, dan 0,1%.
      iii. Mengevaluasi topik menggunakan *closed frequent itemset* sehingga topik yang dihasilkan merupakan topik yang unik dan *single*,
      iv. Membentuk *Topic per years Matrix*, dimana setiap Topik direpresentasikan sebagai vektor dengan dimensinya yaitu frekuensi per tahun,
   d. Membentuk *Topic Community Graph*
      i. Menghitung korelasi antar Topik dengan menggunakan data pada *Topic per years Matrix*. Topik direpresenta-sikan sebagai *node* dan garis antar *node* atau *edge* merupakan nilai korelasi antar Topik,
      ii. Menguji korelasi antar Topik, apabila nilai *p-value* > 0,05 maka garis *edge* akan dihapus,
      iii. Membentuk *network graph* dengan *node* dan *edge* yang didapatkan dari proses sebelumnya,
      iv. Menemukan sekumpulan Topik yang membentuk *Community* menggunakan algoritma *Community detection* berbasis CPM (*Clique Percolation Method)*
3. Pencarian Topik dengan *Association Rule* menggunakan *item per-keywords*
   a. Melakukan *tokenizing* untuk memecah *keywords* berdasarkan tanda koma,
   b. Membentuk *document term matrix*
   c. Mencari Topik dari *document term matrix* menggunakan *frequent itemset mining*,
      i. Membangkitkan *frequent itemset* menggunakan algoritma *Apriori* dan *ECLAT*
      ii. Membandingkan performa dari algoritma *Apriori* dan *ECLAT* dalam membangkitkan *frequent itemset* menggunakan kriteria *minimum support* 2,5%, 1%, 0,75%, 0,5%, 0,25%, dan 0,1%.
      iii. Mengevaluasi topik menggunakan *closed frequent itemset* sehingga topik yang dihasilkan merupakan topik yang unik dan *single*,
      iv. Membentuk *Topic per years Matrix*, dimana setiap Topik direpresentasikan sebagai vektor dengan dimensinya yaitu frekuensi per tahun,
4. Menarik kesimpulan berdasarkan hasil analisis dan pembahasan serta memberikan saran yang bersesuaian.

IV. HASIL DAN PEMBAHASAN

A. Association Rule dengan item per-kata

Analisis *Association Rule* dengan menggunakan kata pada *keywords* sebagai item bertujuan untuk mencari Topik dengan menggunakan *closed frequent itemset* sebagai kandidat topik. Sebagai contoh apabila kata "*support*", "*vector",* dan "*machine"* mempunyai nilai *support* yang cukup tinggi maka ketiga tersebut dapat diduga merupakan sebuah topik yang merepresentasikan metode SVM untuk klasifikasi.

1. Preprocessing Data

Data *keywords* yang terambil kemudian dilakukan praproses data sebelum analisis *association rule*. Berikut merupakan contoh tahap praproses data pada dokumen abstrak dari salah satu dokumen yang ditunjukan pada Tabel 3.

Tabel 3.
Praproses Data

Keywords	Upper bound, Partitioning algorithms, Algorithm design and analysis, Estimation, Approximation algorithms, Memory management, Clustering algorithms
Remove Punctuation (,)	Upper bound Partitioning algorithms Algorithm design and analysis Estimation Approximation algorithms Memory management Clustering algorithms
Case Folding	upper bound partitioning algorithms algorithm design and analysis estimation approximation algorithms memory management clustering algorithms
Remove Stopwords ('and')	upper bound partitioning algorithms algorithm design analysis estimation approximation algorithms memory management clustering algorithms
Lemmatization	upper bound partitioning algorithm algorithm design analysis estimation approximation algorithm memory management clustering algorithm

Setelah dilakukan praproses data maka proses selanjutnya yaitu tokenisasi dan pembuatan *document term matrix* (DTM). Struktur dari DTM yang terbentuk disajikan pada Tabel 4 berikut ini

Tabel 4.
Document Term Matrix

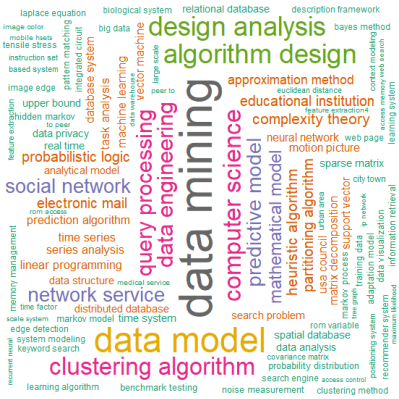
No	abstract	access	⋯	wide	xml	youtube
1	0	0	⋯	0	0	0
2	0	0	⋯	0	0	0
⋮	⋮	⋮	⋱	⋮	⋮	⋮
522	0	0	⋯	0	0	0
523	1	0	⋯	0	0	0
524	0	1	⋯	0	0	0
⋮	⋮	⋮	⋱	⋮	⋮	⋮
3373	0	0	⋯	0	0	0
3374	0	0	⋯	0	0	0
Jumlah	6	47	⋯	5	67	9

Setelah proses pembuatan *document term matrix*, kemudian dilakukan visualisasi word cloud. *Wordcloud*

merupakan salah satu metode untuk menampilkan data teks secara visual dengan informasi yang mudah untuk dianalisis yang ditampilkan pada Gambar 4.

Gambar 4. *Wordcloud Unigram.*

Dari Gambar 4 dapat diketahui bahwa 1-kata yang paling sering muncul sebagai *keywords* adalah *data*. Sedangkan apabila menggunakan *Bigram* (Gambar 5), maka 2-kata berurutan yang paling muncul di *keywords* adalah *data mining*.



Gambar 5. *Wordcloud Bigram.*

Apabila menggunakan pemenggalan 3-kata atau *trigram* maka 3-kata yang paling sering muncul di *keywords* adalah *algorithm design analysis.*

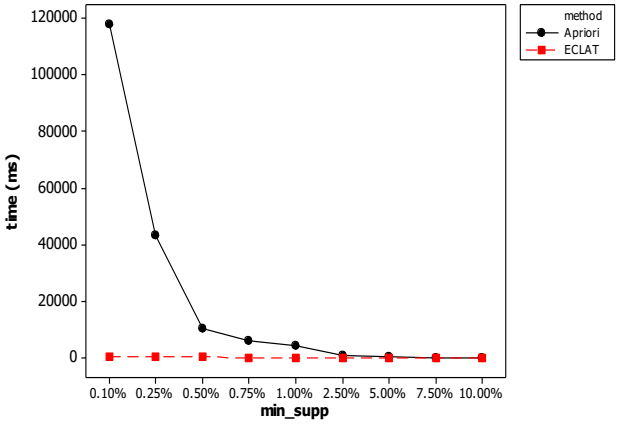
Gambar 6. *Wordcloud Trigram.*

2. *Frequent Itemset*

Dalam membangkitkan *Frequent itemset* dilakukan dengan menggunakan 2 algoritma yaitu *Apriori* dan *ECLAT*. Penggunaan dua algoritma digunakan untuk membandingkan algoritma mana yang lebih cepat dalam membangkitkan *frequent itemset*. Perbandingan waktu pembangkitan *itemset* pada masing-masing algoritma dapat dilihat pada Gambar 7

Dengan *minimum support* yang kecil maka waktu yang dibutuhkan untuk membangkitkan *frequent itemset* akan lebih lama, karena dengan *support* yang kecil tentu *itemset* yang masuk kriteria akan semakin banyak. Begitupun sebaliknya dengan menggunakan *support* yang tinggi maka waktunya akan semakin cepat karena menghasilkan *frequent itemset* yang sedikit.

Dari penggunaan beberapa nilai minimum support yang berbeda, terlihat bahwa di semua nilai minimum support algoritma ECLAT mempunyai waktu yang lebih cepat bila dibandingkan dengan menggunakan Apriori. Hal ini dikarenakan ECLAT menggunakan *vertical layout databese* sehingga untuk menghitung nilai support tidak perlu melakukan *scan databese* secara berulang.



Gambar 7. Perbandingan algoritma Apriori vs ECLAT.

Frequent itemset yang dihasilkan dengan algoritma *Apriori* dan *ECLAT* ditampilkan seperti Tabel 5, dengan menggunakan *minimum support* sebesar 7.5% maka didapatkanlah 15 *frequent itemset*

Tabel 5.
Top 15 *Frequent Itemset* data dari jurnal *IEEE Xplore*

No	items	support	count
1	{data,mining}	0.199	671
2	{data,model}	0.167	563
3	{algorithm,analysis}	0.131	443
4	{algorithm,data}	0.123	414
5	{analysis,data}	0.107	362
6	{algorithm,design}	0.105	353
7	{design,analysis}	0.104	352
8	{algorithm,design,analysis}	0.104	351
9	{database,data}	0.102	344
10	{computer,science}	0.093	315
11	{algorithm,clustering}	0.093	314
12	{data,engineering}	0.092	309
13	{computational,modeling}	0.086	290
14	{processing,query}	0.080	269
15	{computer,data}	0.079	265

Dua kata yang paling muncul bersamaan di *keywords* adalah *data* dan *mining*. Dengan *support* sebesar 19,9% maka dapat diinterpretasikan bahwa kedua kata digunakan secara bersaamaan di *keywords* di 671 dokumen dari 3374 dokumen.

3. *Closed Frequent Itemset*

Karena masih besarnya itemset yang dihasilkan dan masih adanya itemset yang merupakan subset itemset yang lain, maka diperlukan penghapusan itemset dengan pendekatan *Closed frequent itemset*.

Tabel 6.
Perbandingan jumlah itemset pada masing-masing kriteria

min_support	*Frequent itemset*	*Closed Frequent Itemset*	*Remove Subset*
10,00 %	9	9	6
7,50 %	15	15	12
5,00 %	36	34	22
2,50 %	188	162	85
1,00 %	1100	843	448
0,75 %	1798	1293	629
0,50 %	4019	2613	1204
0,25 %	12227	6952	1466
0,1%	67379	23868	10150

Dari hasil Tabel 6, *Closed frequent itemset* dan *remove subset* mampu mengurangi jumlah *frequent itemset* yang dihasilkan, sehingga dua pendekatan tersebut digunakan sebagai kandidat untuk menentukan Topik. Namun yang digunakan sebagai kriteria dalam menentukan Topik adalah *Closed frequent itemset* karena apabila menggunakan *remove subset* maka akan terlalu banyak itemset yang dipotong sehingga ada informasi yang ikut hilang.

Dengan menggunakan *closed frequent itemset* dan *minimum support* sebesar 7,5% didapatkan itemset yang sama dengan *frequent itemset* yaitu sebanyak 15 itemset. Kemudian dari 15 itemset tersebut masing-masing dihitung frekuensi kemunculannya tiap tahun dari tahun 2009 sampai 2018. Data frekuensi per tahun digunakan untuk mencari *trend* pada 10 tahun terakhir dan juga digunakan untuk menghitung korelasi yang akan digunakan untuk membentuk *network graph*.

Dari Tabel 7 dapat dilihat perkembangan masing-masing topik dari tahun 2009 sampai 2018, sebagai contoh Topik *Data model* dan *computational modeling* mengalami kenaikan *trend* pada 10 tahun terakhir. Sedangkan *algorithm clustering* dan *processing query* mengalami penurunan *trend* sepanjang 10 tahun terakhir.

4. Topic Community

Dalam menemukan satu set topik yang berkorelasi tinggi satu sama lain (atau berkorelasi terbalik), maka digunakanlah koefisien korelasi antara dua *itemset* untuk membangun grafik. Setiap *node* dari grafik menunjukkan sebuah *itemset*, dan garis *edge* yang menghubungkan dua *node* menunjukkan korelasi antara dua *itemset*.

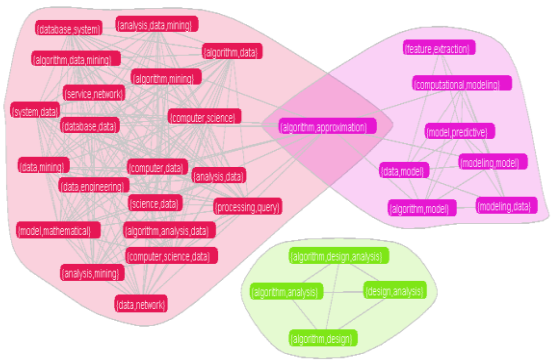
Untuk melihat tingkat korelasi antar dua *itemset* maka dilakukanlah pengujian korelasi. Sehingga apabila ada 2 *itemset* yang tidak mempunyai hubungan yang signifikan, maka garis *edge* antar 2 topik tersebut akan dihapus. Setelah itu, dengan menggunakan algoritma *Community detection* berbasis CPM (*Clique Percolation Method)* didapatkanlah grup topik dengan korelasi yang signifikan (Gambar 8).

Dengan menggunakan *minimum support* sebesar 5% yaitu dengan 34 itemset maka terbentuk 3 kelompok Topik yang saling *overlapping*. Masing-masing itemset di dalam *community* merupakan itemset yang mempunyai korelasi yang signifikan. Sebagai contoh kelompok yang berwarna hijau yaitu {algorithm, design, analysis}, {algorithm, analysis}, {design, analysis}, dan {algorithm, analysis} mempunyai korelasi yang signifikan sehingga keempat topik tersebut membentuk 1 kelompok tersendiri yang merepresentasikan suatu Topik tertentu yaitu tentang *algorithm design & analysis*. Kelompok 2 yang bewarna merah merupakan topik-topik tentang *Data mining*,

Tabel 7.
Perkembangan topik dari tahun 2009-2018

No	items	*line plot*
1	{data,mining}	
2	{data,model}	
3	{algorithm,analysis}	
4	{algorithm,data}	
5	{analysis,data}	
6	{algorithm,design}	
7	{design,analysis}	
8	{algorithm,design,analysis}	
9	{database,data}	
10	{computer,science}	
11	{algorithm,clustering}	
12	{data,engineering}	
13	{computational,modeling}	
14	{processing,query}	
15	{computer,data}	

sedangkan kelompok 3 yang berwarna ungu merupakan topik-topik tentang *Modeling data & feature extraction.*



Gambar 8. *Topic Community* dari 34 Topik.

B. Association Rule dengan item per-keywords

Analisis *Association Rule* dengan menggunakan *keywords* sebagai item bertujuan untuk mencari *keywords* mana saja yang muncul bersamaan, seperti misal *keywords* "*Data mining"* akan muncul bersamaan dengan "*Data models*".

1. Preprocessing Data

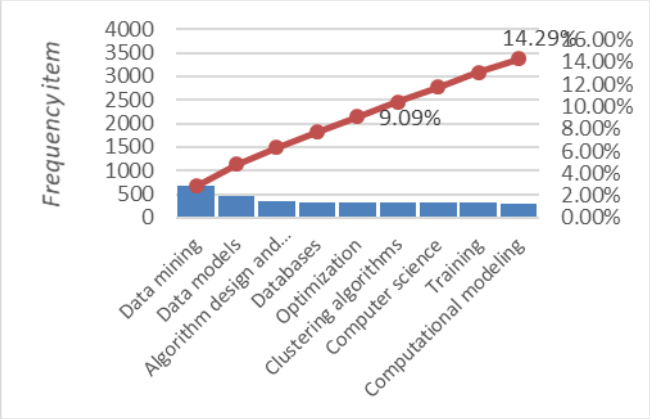
Dengan menggunakan *keywords* sebagai item maka untuk preprocessing data tidak banyak yang dilakukan, hanya melakukan pemisahan masing-masing *keywords* yang dipisahkan dengan tanda baca koma (,).

Tabel 8.
Document Term Matrix per keywords

No	Access control	Access protocols	⋯	Yield estimation	YouTube
1	0	0	⋯	0	0
2	0	0	⋯	0	0
⋮	⋮	⋮	⋱	⋮	⋮
1094	1	1	⋯	0	0
⋮	⋮	⋮	⋱	⋮	⋮
3373	0	0	⋯	0	0
3374	0	0	⋯	0	0
Jumlah	16	3	⋯	1	9

Setelah itu data akan diubah ke dalam *document term matrix*. Struktur *document term matrix* yang terbentuk dapat dilihat pada Tabel 8

Sebelum dilakukan *Association Rule* maka terlebih dahulu dilihat karakteristik data dari *keywords* yang diambil. Gambar 9 merupakan plot 10 besar frekuensi *keywords*, dimana *keywords* yang paling sering muncul di 9% dari keseluruhan *keywords* adalah *Data mining*, *Data models*, *Algorithm design and analysis*, *Databese*, *Optimization*, *Clustering algorithm*. Apabila dibandingkan dengan hasil *Association rule* dengan item per-kata maka dari 10 keywords yang paling sering muncul tersebut mampu dideteksi menggunakan *closed frequent itemset* dengan menggunakan kata sebagai item.



Gambar 9. *Frequency keywords*.

Adapun untuk melihat perkembangan masing-masing *keywords* dari tahun 2009 sampai 2018 dapatn menggunakan regresi *time series*, yaitu dengan menjadikan tahun sebagai variabel prediktor dan frekuensi kemunculan *keywords* per tahun sebagai variabel respon maka *keywords* yang mempunyai *trend* secara signifikan ditampilkan pada Tabel 9

Tabel 9.
Model regresi *time series*

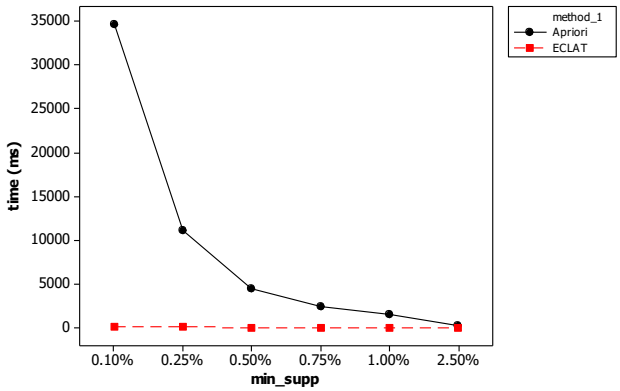
Keywords (X)	Model	*p-value*	R^2
Data models	$X_{tahun} = 18.5 + 5.23 * tahun$	0.001	79.7%
Optimization	$X_{tahun} = 11.9 + 3.68 * tahun$	0.006	62.6%
Clustering algorithms	$X_{tahun} = 41.6 - 1.85 * tahun$	0.019	51.7%
Computational modeling	$X_{tahun} = 6.73 + 4.01 * tahun$	0.001	73.6%

Keywords Clustering algorithm jika dilihat tandanya maka mengalami *trend* penurunan dari tahun 2009 sampai tahun 2018. Sedangkan 3 *keywords* lainnya mengalami peningkatan dari tahun ke tahun antara tahun 2009 sampai tahun 2018.

2. *Frequent Itemset*

Hampir sama pada *Association rule* dengan menggunakan item per-kata, pada item per-*keywords* juga menggunakan dua algoritma yang berbeda untuk melihat mana yang lebih cepat dalam membangkitkan *frequent itemset*. Sehingga dengan menggunakan 2 dataset berbeda dapat ditarik kesimpulan algoritma mana yang lebih cepat.

Dari penggunaan beberapa nilai *minimum support* yang berbeda, terlihat bahwa algoritma *ECLAT* lebih cepat dalam membangkitkan *frequent itemset* bila dibandingkan *Apriori*. Untuk lebih jelasnya, perbandingan algoritma *ECLAT* dengan *Apriori* disajikan pada Gambar 10



Gambar 10. Perbandingan algoritma Apriori vs ECLAT.

Tabel 10 merupakan *9 frequent itemset* dengan menggunakan *minimum support* sebesar 2%. Dua *keyword* yang paling sering muncul bersamaan adalah *Computational modelling* dan *Data models*. Dengan mengambil salah satu *itemset* dari Tabel 10 yaitu itemset {Data mining, feature extraction}, maka dapat diinterpretasikan bahwa *keywords Data mining* dengan *Feature extraction* muncul bersamaan di 69 dokumen dari 3374 dokumen.

Tabel 10.
Top 9 dari *Frequent Itemset* dengan item per-*keywords*

No	items	support	count
1	{Computational modeling, Data models}	0.030	102
2	{Computer science, Data engineering}	0.030	101
3	{Data models, Predictive models}	0.029	99
4	{Computer science, Data mining}	0.028	96
5	{Clustering algorithms, Data mining}	0.025	83
6	{Data mining,Data models}	0.023	79
7	{Algorithm design and analysis,Data mining}	0.023	77
8	{Data mining,Feature extraction}	0.020	69
9	{Data models,Training}	0.020	68

3. *Closed Frequent Itemset*

Dari Tabel 11 terlihat bahwa untuk minimum support lebih dari 0,25% tidak ada perbedaan antara *frequent itemset* dengan *closed frequent itemset*. Sehingga dengan menggunakan 1-*keywords* sebagai item, *closed frequent itemset* tidak perlu digunakan.

Tabel 11.
Perbandingan jumlah itemset pada masing-masing kriteria

min_support	*Frequent itemset*	*Closed Frequent Itemset*	*Remove Subset*
2,50 %	4	4	0
1,00 %	55	55	0
0,75 %	109	109	101
0,50 %	304	304	280
0,25 %	1003	1001	876
0,1%	6052	4472	3298

Kemudian langkah selanjutnya sama seperti pada *Association rule* dengan item per-kata yaitu *itemset* yang didapatkan dari *closed frequent itemset* dihitung frekuensi kemunculan per tahun yang digunakan untuk mencari *trend* dari masing-masing *keywords*.

Jika dilihat pada Tabel 12 maka salah satu topik yaitu {Computational modeling, Data models}, {Data models,

Predictive models}, {Data mining, Data models} mengalami kenaikan *trend* antara tahun 2009 sampai 2018.

Tabel 12.
Perkembangan Topik dari tahun 2009-2018

No	items	*line plot*
1	{Computational modeling, Data models}	
2	{Computer science, Data engineering}	
3	{Data models, Predictive models}	
4	{Computer science, Data mining}	
5	{Clustering algorithms, Data mining}	
6	{Data mining,Data models}	
7	{Algorithm design and analysis,Data mining}	
8	{Data mining,Feature extraction}	
9	{Data models,Training}	

V. KESIMPULAN DAN SARAN

Beberapa kesimpulan yang diperoleh adalah dengan menggunakan dua dataset yang berbeda yaitu dataset dengan item per-kata dan dataset dengan item per-*keywords*, dapat disimpulkan bahwa algoritma *ECLAT* lebih cepat dalam membangkitkan *frequent itemset* bila dibandingkan algoritma *Apriori*. Pada *association rule* dengan *item* per-kata, *Itemset* yang paling tinggi yaitu dengan nilai *support* sebesar 0,199 adalah *data mining*, artinya bahwa kata *data* dan *mining* muncul bersamaan di *keywords* di 671 dokumen dari 3374 dokumen. Dengan menggunakan *community detection* maka kelompok yang terbentuk sebanyak 3, dengan *node-node* didalam *community* memiliki korelasi yang signifikan. Sedangkan apabila menggunakan *item* per-*keywords*, *Itemset* dengan nilai *support* terbesar yaitu dengan nilai 0,0302 adalah {Computational modeling,Data models}, yang artinya *keywords Computational modelling* dan *Data Models* muncul bersamaan di 102 dokumen dari 3374 dokumen.

DAFTAR PUSTAKA

[1] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules in Large Databases," in *Proceedings of the 20th International Conference on Very Large Data Bases*, 1994, pp. 487–499.

[2] M. Zaki and et al, "New algorithms for fast discovery of Association Rules," in *In 3rd International Conference on Knowledge and Data Engineering*, 1997, pp. 283–286.

[3] K. Shubankar and et al, "A frequent keyword-set based algorithm for topic modeling and clustering of research papers," in *3rd Conference on Data Mining and Optimization (DMO)*, 2011, pp. 96–102.

[4] J. Hurtado and et al, "Topic discovery and future trend forecasting for texts," *J. Big Data*, vol. 3, 2016.

[5] C. Manning and et al, *Introduction to Information Retrieval*. New York: Cambridge University Prees, 2008.

[6] J. Han, M. Kamber, and J. Pei, *Data Mining Concepts and Techniques*. USA: Elsevier, 2012.

[7] R. E. Walpole, R. H. Myers, S. L. Myers, and K. Ye, *Probability & Statistics for Engineers & Scientists*, 9th ed. USA: Pearson Education Inc, 2012.

[8] R. Shumway and D. Stoffer, *Time Series Analysis and Its Application with R*. New York: Springer, 2006.

[9] B. Bowerman, R. O'Connell, and A. Koehler, *Forecasting, Time Series and Regression in Applied Approach*. California: Duxbury Press, 2005.

[10] IEEE, "IEEE International Conference on Data Mining (ICDM)," 2019. [Online]. Available: https://ieeexplore.ieee.org/xpl/conhome/1000179/all-proceedings.

[11] IEEE, "International Conference on Data Engineering (ICDE)," 2019. [Online]. Available: https://ieeexplore.ieee.org/servlet/opac?punumber=1000178.