TUGAS AKHIR – KS184822

TOPIC DISCOVERY PADA JURNAL-JURNAL DI *IEEE XPLORE* MENGGUNAKAN *ASSOCIATION RULE MINING* DENGAN PENDEKATAN *CLOSED FREQUENT ITEMSET*

REZA MUSTOFA
NRP 062117 4500 0024

Dosen Pembimbing
Irhamah, M.Si., Ph.D

PROGRAM STUDI SARJANA
DEPARTEMEN STATISTIKA
FAKULTAS MATEMATIKA, KOMPUTASI, DAN SAINS DATA
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2019

TUGAS AKHIR – KS184822

TOPIC DISCOVERY PADA JURNAL-JURNAL DI *IEEE XPLORE* MENGGUNAKAN *ASSOCIATION RULE MINING* DENGAN PENDEKATAN *CLOSED FREQUENT ITEMSET*

REZA MUSTOFA
NRP 062117 4500 0024

Dosen Pembimbing
Irhamah, M.Si., Ph.D

PROGRAM STUDI SARJANA
DEPARTEMEN STATISTIKA
FAKULTAS MATEMATIKA, KOMPUTASI, DAN SAINS DATA
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2019

FINAL PROJECT – KS184822

TOPIC DISCOVERY OF IEEE XPLORE JOURNALS USING ASSOCIATION RULES WITH CLOSED FREQUENT ITEMSET APPROACH

REZA MUSTOFA
NRP 062117 4500 0024

Supervisor
Irhamah, M.Si., Ph.D

UNDERGRADUATE PROGRAMME
DEPARTMENT OF STATISTICS
FACULTY OF MATHEMATICS, COMPUTING, AND DATA SCIENCE
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2019

LEMBAR PENGESAHAN

TOPIC DISCOVERY PADA JURNAL-JURNAL DI *IEEE XPLORE* MENGGUNAKAN *ASSOCIATION RULE MINING* DENGAN PENDEKATAN *CLOSED FREQUENT ITEMSET*

TUGAS AKHIR

Diajukan untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Statistika
pada
Program Studi Sarjana Departemen Statistika
Fakultas Matematika, Komputasi, dan Sains Data
Institut Teknologi Sepuluh Nopember

Oleh :
Reza Mustofa
NRP. 062117 4500 0024

Disetujui oleh Pembimbing:
Irhamah, M.Si., Ph.D ( )
NIP. 19780406 200112 2 002

Mengetahui,
Kepala Departemen Statistika

Dr. Suhartono
NIP. 19710929 199512 1 001

SURABAYA, JULI 2019

TOPIC DISCOVERY PADA JURNAL-JURNAL DI *IEEE XPLORE* MENGGUNAKAN *ASSOCIATION RULE MINING* DENGAN PENDEKATAN *CLOSED FREQUENT ITEMSET*

Nama Mahasiswa : Reza Mustofa
NRP : 062117 4500 0024
Departemen : Statistika
Dosen Pembimbing : Irhamah, M.Si., Ph.D

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 bila dibandingkan dengan 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.

(*Halaman ini sengaja dikosongkan*)

TOPIC DISCOVERY IN IEEE XPLORE JOURNALS USING ASSOCIATION RULES WITH CLOSED FREQUENT ITEMSET APPROACH

Name	**: Reza Mustofa**
NRP	**: 062117 4500 0024**
Departement	**: Statistics**
Supervisor	**: Irhamah, M.Si., Ph.D**

Abstract

Finding topics from a collection of documents such as scientific publications has many benefits. With the increasing number of text documents produced on the web and digital archives, Topic Discovery is a very important tool for browsing, summarizing, and grouping documents. One of the application in Association Rule Mining is to find topics in a document by looking for patterns that often appear on all documents. Data was taken from IEEE Xplore which is a collection of abstracts from journals at the International Conference on Data Mining (ICDM) and the International Conference on Data Engineers (ICDE) from 2009-2018. Each abstract is represented as a transaction while the keyword words contained in it are represented as items. Combination of keywords that appear most often, called frequent itemset, will be used as a candidate for a topic. The algorithms that can be used to generating frequent itemset is the Apriori and ECLAT algorithms. The execution time for generating frequent itemset of ECLAT is faster than Apriori. Closed frequent itemset is able to reduce the frequent itemset that is formed, so the topic formed is a unique topic.

Keywords : Apriori Algorithm, Association Rule, Closed Frequent Itemset, Eclat Algorithm, Network Analysis, Text Mining*.*

(*Halaman ini sengaja dikosongkan*)

KATA PENGANTAR

Puji syukur kehadirat Allah SWT yang senantiasa memberikan rahmat, hidayah serta karunia-Nya. Sholawat serta salam semoga selalu tercurahkan untuk Nabi Muhammad SAW. Atas berkat rahmat Allah, penulis dapat menyelesaikan Tugas Akhir dengan berjudul **"*Topic Discovery* pada Jurnal-jurnal Penelitian di *IEEE Explore* Menggunakan *Association Rule Mining* dengan *Pendekatan Closed Frequent Itemset*"**. Dalam penyelesaian Tugas Akhir ini, penulis mendapatkan banyak bantuan dan dukungan dari berbagai pihak. Maka dari itu penulis dengan penuh hormat ingin mengucapkan terimakasih kepada:

1. Ibu Irhamah, M.Si., Ph.D selaku dosen pembimbing yang telah banyak meluangkan waktu untuk memberikan arahan, bimbingan serta dukungan dalam penyusunan laporan Tugas Akhir ini.
2. Bapak Prof. Drs. Nur Irawan, M.Ikom, Ph.D. dan Ibu Dr. Kartika Fithriasari, M.Si selaku dosen penguji yang telah memberikan saran dan perbaikan Tugas Akhir ini.
3. Bapak Dr. Suhartono selaku Kepala Departemen Statistika ITS dan Ibu Santi Wulan Purnami, M.Si, Ph.D selaku Ketua Program Studi Sarjana Departemen Statistika FMIPA ITS.
4. Seluruh bapak/ibu dosen pengajar di Jurusan Statistika ITS atas segala ilmu yang telah diberikan serta seluruh staf dan karyawan Jurusan Statistika atas kerja keras dan bantuannya selama ini.
5. Kedua orang tua dan keluarga yang selalu memberikan doa, bimbingan, dukungan, kasih sayang serta kesabarannya dalam mendidik baik secara materiil, moril, maupun spiritual.
6. Kepada teman-teman Lintas Jalur Statistika ITS 2017 yang telah berjuang bersama dan saling mendukung satu sama lain.

7. Pihak-pihak lain yang sudah banyak membantu dalam proses pengerjaan laporan Tugas Akhir ini yang tidak dapat disebutkan satu-persatu.

Penulis menyadari bahwa laporan ini masih memiliki banyak kekurangan dan jauh dari kesempurnaan, maka segala kritik serta saran ataupun diskusi sangat dibutuhkan oleh penulis agar lebih baik kedepaanya. Besar harapan penulis agar informasi sekecil apapun dari laporan ini dapat memberikan kebermanfaatan bagi berbagai pihak.

Surabaya, Juni 2019

Penulis

DAFTAR ISI

DAFTAR GAMBAR

Halaman

(*Halaman ini sengaja dikosongkan*)

DAFTAR TABEL

(*Halaman ini sengaja dikosongkan*)

DAFTAR LAMPIRAN

(*Halaman ini sengaja dikosongkan*)

BAB I
PENDAHULUAN

1.1 Latar Belakang

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. Dengan semakin banyaknya dokumen teks yang dihasilkan di web dan arsip-arsip digital, *Topic Discovery* menjadi alat yang sangat penting untuk menelusuri, meringkas, dan pengelompokkan dokumen. Menemukan topik dalam kumpulan dokumen dapat digunakan dalam berbagai penerapan seperti peringkasan dokumen, *recommendation system*, analisis *trend*.

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 dianggap sebagai transaksi sedangkan kata-kata yang terkandung didalamnya dianggap sebagai item. Sehingga kombinasi antar kata yang paling sering muncul, yang disebut *frequent itemset*, diduga merupakan suatu topik. Salah satu algoritma pada *association rule mining* yang paling sering digunakan dan merupakan algoritma paling awal yaitu algoritma *Apriori* (Agrawal & Srikant, 1994). Algoritma tersebut digunakan untuk mendapatkan *frequent itemset*

berdasarkan *minimum support* maupun *minimum confidence*. Algoritma lain yang digunakan yaitu algoritma *Eclat*, yang menggunakan *vertical databese layout* (Zaki, et al., 1997). Algoritma *Eclat* lebih cepat bila dibandingkan dengan *apriori* karena pada algoritma *Eclat* hanya akan memeriksa (*scan*) dataset sebanyak satu kali, tidak melakukannya berulang-ulang karena menggunakan *vertikal tid-list*, sehingga *tid-list* sudah memberikan informasi tentang *support count* dari itemset.

Dalam rangka penentuan Topik yang bermakna, *frequent itemset* menghasilkan *itemset* yang terlalu banyak padahal sesungguhnya memiliki makna yang sama, misal *itemset {association, mining}* dan {*association, rule, mining}* mungkin mengandung makna Topik yang sama. Sehingga Topik *association mining* dapat dihilangkan karena informasi tersebut sudah terkandung dalam Topik *association rule mining*, maka dengan menggunakan pendekatan *closed frequent itemset* Topik tersebut bisa dihapus. *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 jurnal penelitian pernah dilakukan oleh Shubhankar, Singh, & Pudi (2011) 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, Agarwal, & Zhu (2016) 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 untuk menemukan Topik dari sekumpulan dokumen dengan menggunakan pendekatan *closed frequent itemset*. Algoritma yang digunakan untuk menghasilkan *frequent itemset* adalah *Apriori* dan *Eclat*. Selain itu dilakukan pula

analisis korelasi untuk mengetahui topik-topik mana saja yang mempunyai hubungan yang kuat. Data yang digunakan pada penelitian ini berupa *keywords* pada abstrak, sehingga pencarian topik pada *keywords* diharapkan dapat mewakili topik dari keseluruhan dokumen. Alasan digunakan *keywords* daripada isi abstrak dikarenakan *keywords* merupakan *short text* yang berisi tentang kata-kata penting dalam suatu dokumen, dan juga pada abstrak banyak mengandung kata *noise* dan apabila diterapkan pada *association rule* akan mengakibatkan tidak munculnya Topik-topik yang bermakna.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, permasalahan utama yang dibahas dalam penelitian ini adalah bagaimana 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*.

1.3 Tujuan

Berdasarkan rumusan masalah yang telah disebutkan, tujuan yang ingin dicapai dalam penelitian ini adalah sebagai berikut:

1. Mengetahui *text preprocessing* pada data *keywords* yang diambil dari jurnal-jurnal *IEEE Xplore*,
2. Mengetahui algoritma yang lebih cepat dalam membangkitkan *frequent itemset*,
3. Menemukan Topik pada jurnal-jurnal *IEEE Xplore* dengan menggunakan *closed frequent itemset*,

1.4 Manfaat

Manfaat yang dapat diambil pada penelitian ini adalah mengetahui penerapan *association rule mining* selain digunakan

pada *Market Basket Analysis*. Penerapan *association rule* pada *text* (kata/*keywords*) digunakan untuk mengetahui pasangan kata/*keywords* yang sering muncul secara bersamaan di suatu dokumen. Dimana pasangan kata/*keywords* yang sering muncul bersamaan tersebut akan digunakan sebagai kandidat topik.

1.5 Batasan Masalah

Dalam penelitian ini data yang digunakan berasal dari *ieeexplore.ieee.org*. Data yang diambil berupa *IEEE keywords* dari jurnal-jurnal di *International Conference of Data Mining (ICDM)* dan *International Conference of Data Engineering (ICDE)*. Rentang waktu penerbitan jurnal yang diambil adalah antara tahun 2009-2018.

BAB II
TINJAUAN PUSTAKA

Tinjauan pustaka dalam penelitian ini memuat *text preprocessing*, *Association Rule, Frequent itemset mining, Closed frequent itemset,* Analisis Korelasi dan *Topic Community*.

2.1 *Text Preprocessing*

Data prepocessing merupakan tahapan-tahapan yang dilakukan sebelum mengolah data yang telah diperoleh agar bisa dilakukan analisis lebih lanjut. *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*

Untuk alasan tata Bahasa atau dalam Bahasa Inggris disebut *grammar*, dalam satu dokumen akan menggunakan berbagai bentuk kata, seperti *organize, organizes, dan organizing.* Selain itu ada kelompok kata yang merupakan turunan dari kata dasar yang mempunyai makna yang sama, seperti *democracy*, *democratic,* dan *democratization*. Dalam banyak situasi, pencarian kata dasar untuk kata-kata yang mempunyai makna yang sama akan sangat berguna (Manning, Raghavan, & Schütze, 2008). Ada dua metode dalam pencarian kata dasar yaitu *Stemming* dan *Lemmatization*

Stemming merupakan suatu proses untuk menemukan kata dasar dari sebuah kata. Dengan menghilangkan semua imbuhan

(*affixes*) baik yang terdiri dari awalan (*prefixes*), sisipan (*infixes*), akhiran (*suffixes*) dan *confixes* (kombinasi dari awalan dan akhiran) pada kata turunan. *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 (Manning, Raghavan, & Schütze, 2008). 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.

2.2 *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 $W = \{w_1, w_2, \ldots, w_n\}$ yang terdiri dari *n* kata (item) dan *databese* $D = \{d_1, d_2, \ldots, d_m\}$ terdiri dari *m* dokumen (transaksi). Maka masing-masing dokumen terdiri dari kumpulan kata-kata yang disebut *itemset*, dinotasikan dengan I ($I \in W$). Sedangkan *itemset* yang mempunyai item sebanyak *k*, disebut *k-itemset* (Han, Kamber, & Pei, 2012). *Association Rule* mempunyai bentuk umum $I_1 \rightarrow I_2$ dimana I_1 merupakan *antecedent item* dan I_2 merupakan *consequent item*. Association rule ini nantinya akan menghasilkan *rule* yang menentukan seberapa besar hubungan antar *X* dan *Y*, dan diperlukan dua ukuran untuk *rule* ini, yakni *Support* dan *Confidence*. *Support* merupakan persentase dokumen yang mengandung *itemset* I_1 dan I_2, sedangkan *confidence* merupakan kemungkinan munculnya I_2 ketika I_1 juga muncul. Pada

penelitian ini yang digunakan hanya *support, r*umus untuk menghitung *Support* dinyatakan dalam persamaan (2.1)

$$Support(I_1 \rightarrow I_2) = \frac{\text{Banyak dokumen yang mengandung kata } I_1 \text{ dan } I_2}{\text{Total dokumen}} \quad (2.1)$$

2.2.1 *Frequent Itemset Mining*

Langkah pertama pada *association rule* adalah menghasilkan semua *itemset* yang memungkinkan, kemungkinan *itemset* yang muncul pada *m-item* adalah 2^m. 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*.

a. Algoritma *Apriori*

Algoritma *Apriori* pertama kali diperkenalkan oleh Agrawal & Shrikant (1994) yang berguna untuk menemukan *frequent itemset* pada sekumpulan data. Algoritma apriori digunakan untuk mencari *frequent itemset* yang memenuhi *minimum support* kemudian mendapatkan rule yang memenuhi *minimum confident* dari *frequent itemset* tadi (Zheng, Kohav, & Mason, 2001). Cara algoritma ini bekerja adalah algoritma akan menghasilkan kandidat baru dari *frequent k-itemset* pada langkah sebelumnya dan menghitung nilai *support k-itemset* tersebut. *Itemset* yang memiliki nilai support di bawah dari *minimum support* akan dihapus. Algoritma berhenti ketika tidak ada lagi *frequent itemset* baru yang dihasilkan. Adapun algoritmanya ditampilkan pada Tabel 2.1,

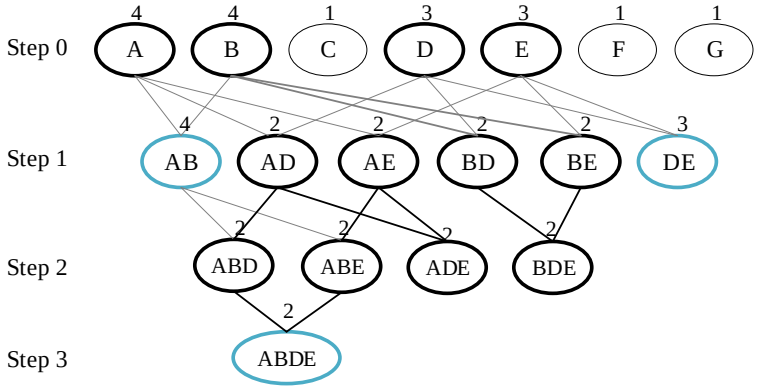
Tabel 2.1 *Pseudocode* Algoritma Apriori

Algorithm 1: Apriori	
Input:	*A transaction database D,* *A user specified threshold minimum support*
Output:	*Frequent itemsets F*
Procedure:	*1.* C_1*={all itemsets of len 1}* *2.* $i \leftarrow 1$ *3. while* $C_i \neq \emptyset$ *4.* $F_i \leftarrow \{c \in C_i \mid supp(c,D) \geq$ *minimum support}*

Tabel 2.1 Lanjutan

Procedure:	*5. $C_{i+1} \leftarrow$ {generate (i+1)-itemsets having i+1 frequent itemsets}* *6. $i \leftarrow i+1$* *return $F_0 \cup ... \cup F_i$*

Ilustrasi dari algoritma apriori disajikan pada Gambar 2.1 dengan menggunakan metode pencarian *breadth-first search* (*BFS*)



Gambar 2.1 Ilustrasi Algoritma Apriori (*BFS*)

b. Algoritma *Eclat*

Algoritma *Eclat* (Zaki, et al., 1997) menggunakan *vertical databese layout*, berbeda dengan algoritma apriori yang menggunakan *horizontal databese layout*. Setiap *item* dinyatakan dalam tabel *tid-list* secara vertikal (Gambar 2.2) 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.

Algoritme ECLAT membangkitkan kandidatnya dengan pencarian *Depth-First Search* dan menggunakan titik potong tid-list antar-item untuk menghitung nilai *support* (Borgelt, 2003). Ilustrasi *Depth-First Search* untuk membangkitkan kandidat

itemset dalam menemukan *frequent itemset* ditampilkan pada Gambar 2.3.

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 2.2 *Horizontal vs Vertical databese layout*

Algoritme ECLAT dalam membangkitkan *frequent itemset* terdapat pada Tabel 2.2,

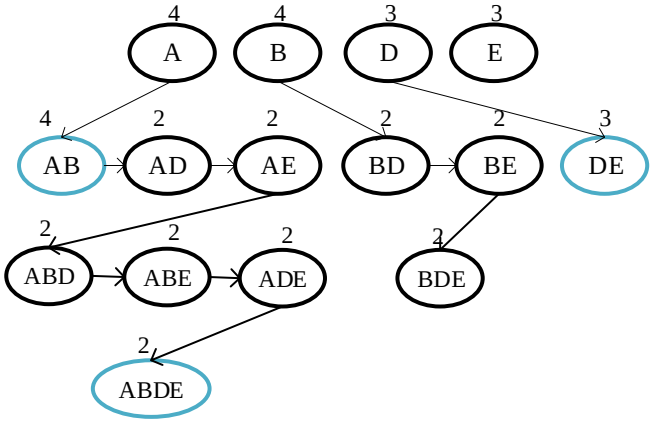
Tabel 2.2 *Pseudocode* Algoritma ECLAT (Xu, Zhang, & Li, 2011)

Algorithm 2: ECLAT	
Input:	*A transaction database D,* *A user specified threshold minimum support* *A set of frequent k-itemset* $F_k = \{I_1, \ldots, I_n\}$
Output:	*Frequent itemsets F*
Procedure: *Eclat(F_k)*	*1. for all* $I_i \in F_k$ *2.* $F_{k+1} = \emptyset$ *3. for all* $I_j \in F_k$, *with* $j > i$ *do* *4.* $N = I_i \cup I_j$; *// Both should be from same equivalence class* *5. If support(N)* $\geq$ *minsup then* *6.* $F_{k+1} = F_{k+1} \cup \{N\}$; *7.* $F_{\|R\|} = F_{\|R\|} \cup \{N\}$; *8. end* *9. end* *10. for all* $F_{k+1} \neq \emptyset$ *do Eclat*(F_{k+1}); *11. end*

2.2.2 *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 2.3 merupakan *closed frequent itemset*.



Gambar 2.3 Ilustrasi Algoritma ECLAT (*DFS*)

2.3 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. Koefisien korelasi merupakan ukuran keeratan hubungan antar 2 Topik. Salah satu metode untuk menghitung korelasi yang paling sering digunakan yaitu *Pearson correlation* (Walpole, et al., 2011). Misalka $X_1 = \langle x_{11}, x_{12}, \ldots, x_{1k} \rangle$ dan $X_2 = \langle x_{21}, x_{22}, \ldots, x_{2k} \rangle$ maka

rumus untuk menghitung koefisien korelasi *Pearson* antara Topik I_1 dan I_2 dinyatakan dalam persamaan (2.3)

$$\hat{\rho}(X_1, X_2) = \frac{\sum_{q=1}^{k}(x_{1q} - \bar{x}_1)(x_{2q} - \bar{x}_2)}{\sqrt{\sum_{q=1}^{k}(x_{1q} - \bar{x}_1)^2 \times \sum_{q=1}^{k}(x_{2q} - \bar{x}_2)^2}} \quad (2.3)$$

Nilai koefisien korelasi ini digunakan untuk membentuk *Network graph,* dengan masing-masing *node* merupakan topik dan *edge* yang menghubungkan dua *node* merupakan hubungan antar dua topik yang signifikan. Untuk mengetahui mana saja Topik yang mempunyai hubungan secara signifikan, dapat dilakukan pengujian korelasi dengan hipotesisnya adalah (Walpole, et al., 2011)

$H_0 : \rho = 0$

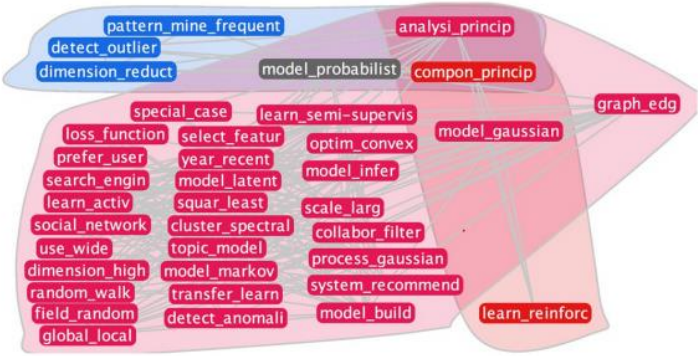
$H_1 : \rho \neq 0$

Statistik uji yang digunakan adalah

$$t_{hitung} = \frac{r_{x_1x_2}\sqrt{n-2}}{\sqrt{1-r_{x_1x_2}^2}}. \quad (2.4)$$

H_0 ditolak apabila nilai statistik uji $\left|t_{\text{hitung}}\right| > t_{\alpha/2,(n-2)}$ dengan *n* merupakan banyak pengamatan. Setelah semua topik 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*, dimana *node* dalam suatu *community* berkorelasi signifikan satu sama lain dibandingkan korelasi dengan *node* diluar *community*. 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 2.4 merupakan contoh dari hasil dari *Topic Community* yang dilakukan oleh Hurtado, Agarwal, & Zhu (2016).



Gambar 2.4 *Topic Community*

2.4 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 (Shumway & Stoffer, 2006). Jika data X_t memiliki trend, model regresi dapat ditulis sebagai berikut (Bowerman, O'Connell, & Koehler, 2005):

$$X_t = \delta_t + a_t \tag{2.5}$$

dimana,

X_t : data pengamatan pada periode t

δ_t : komponen *trend* pada periode t

a_t : komponen *error* pada periode t

Pada model regresi linier *trend*, menyatakan bahwa *time series* X_t dapat diwakili oleh level rata-rata (dinotasikan dengan μ) dimana selalu berubah dari waktu ke waktu berdasarkan persamaan $\mu_t = \delta_t$ dan dengan *error* (ε_t). *Error* ini mewakili fluktuasi acak yang menyebabkan nilai X_t menyimpang dari level rata-rata μ_t.

2.5 *IEEE Xplore Digital Library*

IEEE Xplore adalah database penelitian untuk menemukan dan mengakses artikel jurnal, *conference proceedings*, *technical standards*, dan materi terkait tentang ilmu komputer, teknik elektro dan elektronik, dan bidang-bidang yang terkait. *IEEE Xplore* berisi materi yang diterbitkan terutama oleh *Institute of Electrical and Electronics Engineers (IEEE)* dan penerbit mitra lainnya. *IEEE Xplore* menyediakan akses web ke lebih dari 4 juta dokumen dari publikasi di bidang ilmu komputer, teknik elektro, elektronik dan bidang-bidang yang terkait. Dokumen dan bahan-bahan lainnya terdiri dari lebih dari 195 jurnal, lebih dari 1.400 *conference proceedings*, lebih dari 5100 *technical standards*, sekitar 2.000 buku, dan lebih dari 400 kursus online. Sekitar 20.000 dokumen baru ditambahkan setiap bulan. Siapa pun dapat mencari di *IEEE Xplore* dan menemukan *bibliografi* dan abstrak untuk isinya, sementara akses ke dokumen teks lengkap memerlukan langganan individu atau institusi (IEEE, 2019a). Abstrak yang diambil pada penelitian ini berasal dari koleksi *International Conference on Data Mining* (ICDM) dan *International Conference on Data Engineers* (ICDE)

(*Halaman ini sengaja dikosongkan*)

BAB III
METODOLOGI PENELITIAN

Metodologi penelitian akan menjelaskan tentang data yang digunakan, variabel penelitian, langkah analisis beserta diagram alirnya.

3.1 Sumber Data

Penelitian ini menggunakan data sebanyak 3.374 asbtrak dari koleksi *International Conference on Data Mining* (ICDM) (IEEE, 2019b) dan *International Conference on Data Engineers* (ICDE) (IEEE, 2019c) di *IEEE Xplore Digital Library* dari tahun 2009 sampai 2018. Dokumen yang digunakan sebagai data adalah kata kunci atau *keywords* yang terdapat di Abstrak.

3.2 Struktur Data

Tabel 3.1 merupakan contoh dokumen publikasi dari ICDE yang diunduh dari halaman web *IEEE Xplore* pada laman https://ieeexplore.ieee.org

Tabel 3.1 Struktur Data Awal

label	Cheng_2011_20IE27InCoonDaEn
type	CONF
author	Cheng, J and Ke, Y and Chu, S and zsu, M T
year	2011
title	Efficient core decomposition in massive networks
journal	2011 IEEE 27th International Conference on Data Engineering
pages	51-62
abstract	The k-core of a graph is the largest subgraph in which every vertex is connected to at least k other vertices within the subgraph. Core decomposition finds the k-core of the graph for every possible k. Past studies have shown important applications of core decomposition such as in the study of the properties of large networks (e.g., sustainability, connectivity, centrality, etc.), for solving NP-hard problems efficiently in real networks …

Tabel 3.1 Lanjutan

keywords	Algorithm design and analysis and Approximation algorithms and Clustering algorithms and Estimation and Memory management and NP-hard problem and Partitioning algorithms and Upper bound and approximation theory and computational complexity and core decomposition and external memory algorithm and graph theory and inmemory algorithm and large-scale network fingerprinting and large-scale network visualization and massive graph and network theory (graphs) and online social networks and optimisation and real-world network and subgraph approximation
doi	10.1109/ICDE.2011.5767911
issn	2375-026X VO -

Data yang digunakan pada pencarian topik menggunakan *association rule* yaitu kata kunci atau *keywords* yang terdapat di abstrak. Kemudian setelah melalui tahap *preprocessing* maka abstrak akan ditransformasi ke dalam bentuk *document term matrix*. *Document term matrix* digunakan pada tahap pencarian topik menggunakan *association rule.* Struktur data setelah ditransformasi kedalam bentuk *document term matrix* disajikan pada Tabel 3.2

Tabel 3.2 *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 3.3

Tabel 3.3 *Topic per year Matrix*

Topik	**Tahun ke-**					
(*X*)	**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

3.3 Langkah Analisis

Berikut adalah 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

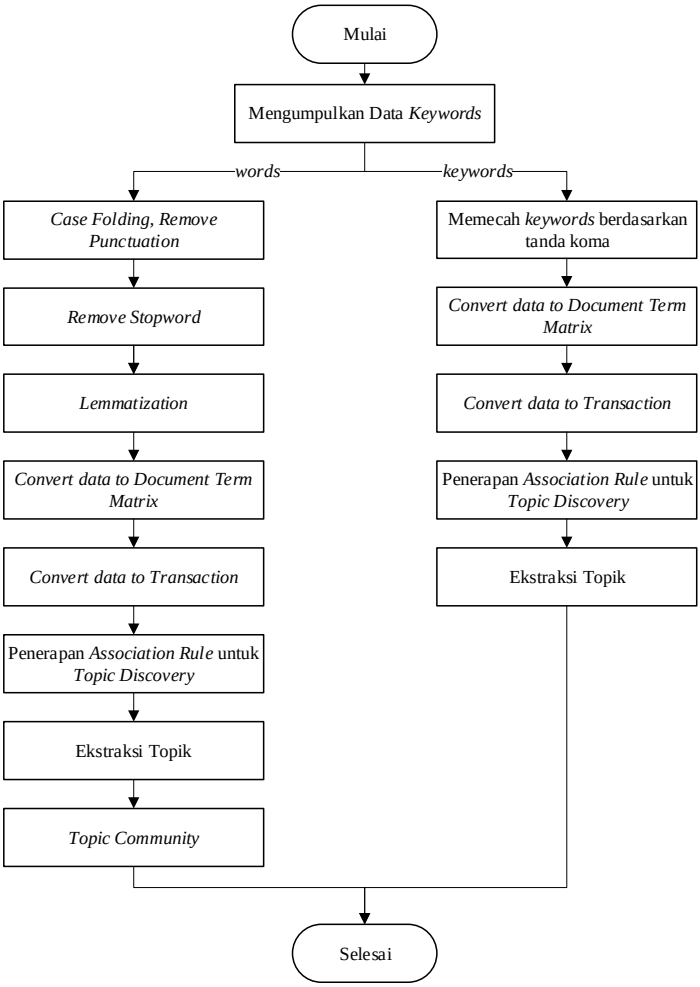
a. Melakukan *preprocessing data*, dalam tahap *preprocessing data* terdapat tahap-tahap yang dilakukan yaitu :
   i. Melakukan *case folding*, yaitu mengubah semua teks dengan huruf kecil,
   ii. Menghapus tanda baca koma (,),
   iii. Menghapus kata yang mengandung *stopwords* "and",
   iv. Melakukan *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 direpresentasikan 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.

3.4 Diagram Alir

Langkah analisis tersebut digambarkan dalam diagram alir Gambar 3.1



Gambar 3.1 Diagram Alir Penelitian

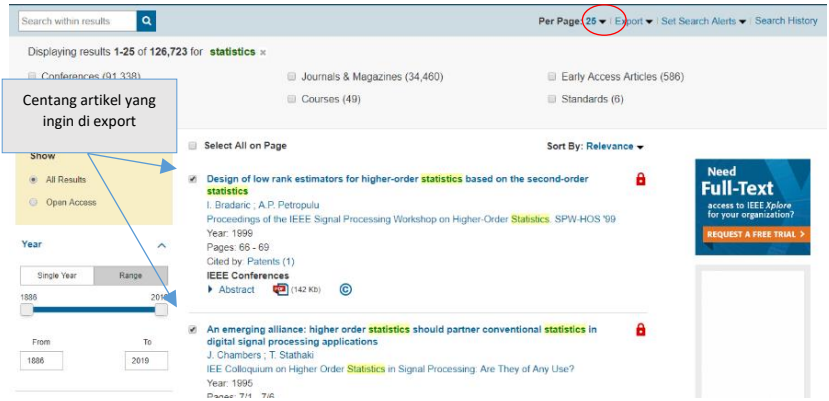
BAB IV
ANALISIS DAN PEMBAHASAN

Pada bab ini membahas mengenai hasil analisis data untuk menjawab rumusan masalah yang diambil. Sebelum melakukan analisis Association Rule untuk mendeteksi topik maka data abstrak yang diambil perlu dilakukan *preprocessing*, agar data tersebut siap untuk dianalisis. Pada bab ini dijelaskan pula cara untuk *crawling* atau mengambil abstrak jurnal di website https://ieeexplore.ieee.org.

4.1 Tata Cara Pengambilan Abstrak

Data yang digunakan pada penelitian ini terdiri dari 3374 abstrak yang diambil dari jurnal-jurnal di *IEEE Xplore*. Sebelum melakukan analisis, perlu dijelaskan terlebih dahulu bagaimana cara melakukan pengambilan data abstrak karena banyak orang yang masih belum tahu. Berikut merupakan tata cara pengambilan abstrak di *IEEE Xplore*,

1. Buka alamat website di https://ieeexplore.ieee.org kemudian cari artikel yang diinginkan,
2. Pada hasil pencarian centang artikel pada kotak di samping judul artikel (lihat Gambar 4.1),



Gambar 4.1 Halaman Pencarian di *IEEE Xplore*

3. Kemudian pada menu di atas klik *Export*, maka akan muncul *Dialog box* seperti pada Gambar 4.2
4. Pada *Dialog box* pilih Tab *Citations*, untuk *Format* pilih RIS dan untuk *Include* pilih *Citation & Abstract*. Kemudian klik *Export*

Gambar 4.2 *Dialog Box* pada menu *Export*

5. Maka *file* yang berisi abstrak akan berhasil terunduh, struktur data dari file abstrak yang telah diunduh bisa dilihat pada Tabel 3.1,
6. Data yang diambil adalah keywords namun jika dilihat pada Tabel 3.1, terlihat bahwa data *keywords* merupakan gabungan dari 4 macam tipe keywords yaitu *IEEE keywords, INSPEC Controlled Indexing, INSPEC Non Controlled Indexing, Author keywords*,
7. Karena itu diperlukan *crawling data* menggunakan *R* (Lampiran 1) untuk mendapatkan masing-masing *keywords* secara terpisah, dengan menggunakan alamat DOI sebagai *unique-key* yang telah didapat pada langkah sebelumnya. *Keywords* yang dipakai pada penelitian ini adalah *IEEE keywords* karena memiliki data *missing* yang paling sedikit diantara keempat *keywords*.

4.2 *Association Rule* dengan *item* per-kata

4.2.1 *Preprocessing Data*

Data yang diambil berupa kata *keywords* dari tiap abstrak yang telah dikumpulkan. Sebelum dilakukan analisis maka dilakukan praroses data dengan tahap *case folding, remove punctuation, remove stopwords, lemmatization* dan *tokenizing*. Berikut dilakukan praproses data pada *keywords* dari salah satu dokumen. Data hasil simulasi praproses ditunjukan sebagaimana pada Tabel 4.1 berikut.

Tabel 4.1 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 des ign and analysis Estimation Approximation algorith ms Memory management Clustering algorithms
Case Folding	upper bound partitioning algorithms algorithm desi gn and analysis estimation approximation algorithm s 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). DTM yaitu menyimpan *term*/kata dalam bentuk matriks kemudian term dihitung kemunculannya pada setiap dokumen. Kemudian setelah DTM terbentuk, *term* yang muncul kurang dari 3 dokumen akan dihapus untuk mengurangi dimensi matriks agar tidak terlalu membebani pada proses pembangkitan *itemset*. Struktur dari DTM yang terbentuk disajikan pada Tabel 4.2 berikut ini

Tabel 4.2 *Document Term Matrix*

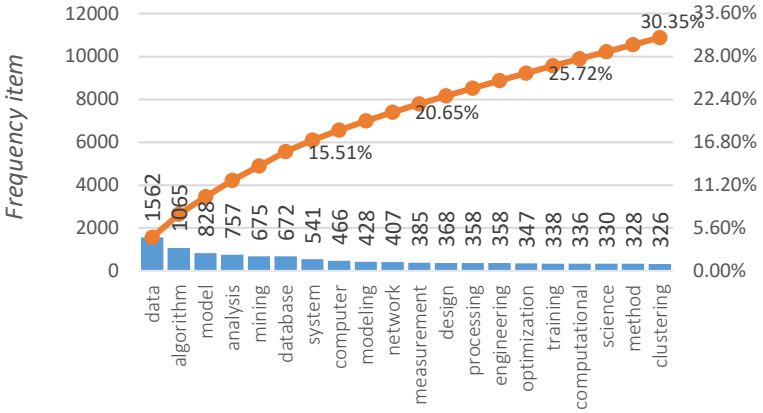
No	abstract	access	accuracy	⋯	wide	xml	youtube
1	0	0	0	⋯	0	0	0
2	0	0	0	⋯	0	0	0
⋮	⋮	⋮	⋮	⋱	⋮	⋮	⋮
522	0	0	1	⋯	0	0	0
523	1	0	0	⋯	0	0	0
524	0	1	0	⋯	0	0	0
⋮	⋮	⋮	⋮	⋱	⋮	⋮	⋮
3373	0	0	0	⋯	0	0	0
3374	0	0	0	⋯	0	0	0
Jumlah	6	47	163	⋯	5	67	9

Setelah proses pembuatan *document term matrix*, kemudian dilakukan visualisasi *word cloud*. *Word cloud* merupakan salah satu metode untuk menampilkan data teks secara visual dengan informasi yang mudah untuk dianalisis. Visualisasi *1-word cloud* dapat dilihat pada Gambar 4.3

Gambar 4.3 1-*word cloud*

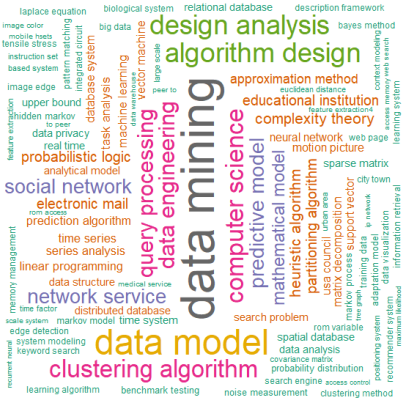
Dengan menggunakan diagram pareto pada Gambar 4.4, jika diambil 15,51% dari total frekuensi kemunculan kata, maka satu kata yang paling sering digunakan pada *keywords* jurnal ICDM dan

ICDE adalah *data*, *algorithm*, *model*, *analysis*, *mining*, *databese*, dan *system*.



Gambar 4.4 Diagram Pareto *unigram*

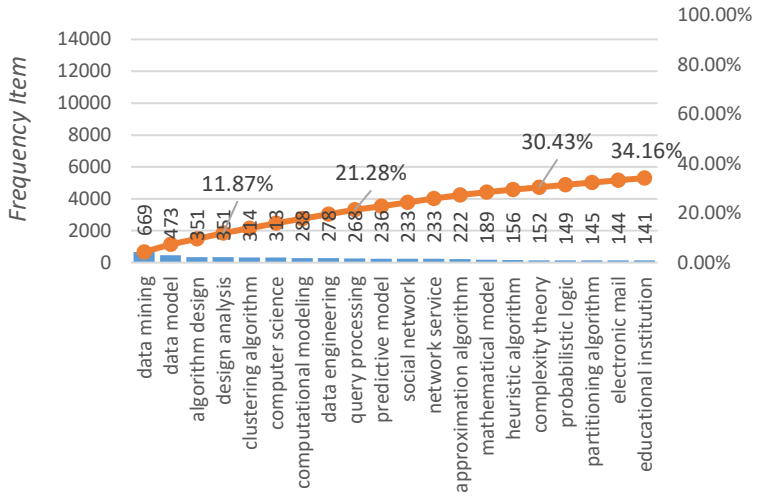
Apabila menggunakan *bigram* maka visualisasi *wordcloud* dapat dilihat pada Gambar 4.5



Gambar 4.5 *2-word cloud*

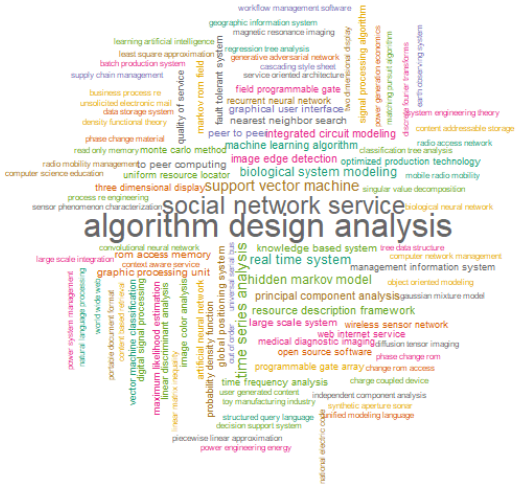
Apabila diambil 11,87% dari total frekuensi kemunculan kata, maka dua kata yang paling sering muncul atau paling sering digunakan pada *keywords* adalah *data mining*, *data model*,

algorithm design, design analysis. Untuk lebih jelasnya dapat dilihat pada Gambar 4.6



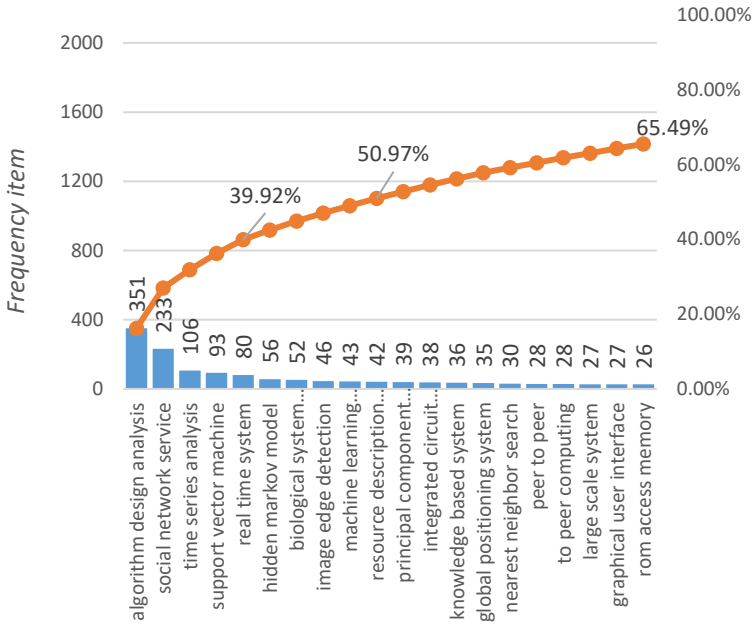
Gambar 4.6 Diagram Pareto *bigram*

Apabila menggunakan *trigram* maka visualisasi *wordcloud* dapat dilihat pada Gambar 4.7



Gambar 4.7 *3-word cloud*

Apabila menggunakan tiga kata atau *trigram* maka tiga kata yang paling sering digunakan untuk *keywords* di jurnal ICDM dan ICDE adalah *algorithm design analysis, social network service, time series analysis, support vector machine, real time system*. lima *keywords* tersebut diambil dari 40% total frekuensi kemunculan kata. Untuk lebih jelasnya dapat dilihat di Gambar 4.8.



Gambar 4.8 Diagram Pareto *trigram*

4.2.2 *Frequent Itemset Mining*

Frequent itemset merupakan langkah awal yang digunakan untuk mencari pola kata atau gabungan dari beberapa kata yang sering muncul yang nantinya akan membentuk topik di antara banyak dokumen. Dari hasil *document term matriks* yang sudah terbentuk kemudian diproses menggunakan dua algoritma yang berbeda yaitu *Apriori* dan *ECLAT*. Penggunaan dua algoritma digunakan untuk membandingkan algoritma mana yang lebih cepat dalam membangkitkan itemset.

a. Algoritma *Apriori*

Algoritma *Apriori* merupakan algoritma yang paling awal dan sering digunakan untuk membangkitkan *frequent itemset*. Berikut merupakan contoh dari penerapan algoritma *Apriori* dengan menggunakan data pada Tabel 4.3

Tabel 4.3 Data contoh

id	Keywords
1	Database systems, Computational
2	Database systems, Awards activities
3	Awards activities, Scalability
4	Database systems
5	Games, Database systems, Awards activities

Sedangkan data pada Tabel 4.4 merupakan data yang telah melalui tahap *preprocessing*

Tabel 4.4 *Post-processing*

id	Keywords
1	database system computational
2	database system award activity
3	award activity scalability
4	database system
5	game database system award activity

Kemudian setelah melalui tahap preprocessing maka data akan dirubah ke dalam *document term matrix* yang akan digunakan sebagai *input* pada algoritma *Apriori*. Tabel 4.5 merupakan struktur dari *document term matrix*

Tabel 4.5 *Document Term Matrix*

id	database (*A*)	system (*B*)	computational (*C*)	award (*D*)	activity (*E*)	scalability (*F*)	game (*G*)
1	1	1	1	0	0	0	0
2	1	1	0	1	1	0	0
3	0	0	0	1	1	1	0
4	1	1	0	0	0	0	0
5	1	1	0	1	1	0	1

Berikut merupakan langkah-langkah pembangkitan itemset menggunakan algoritma *Apriori* sampai didapatkan *frequent itemset* dengan minimum support = 2.

Notasi :

C_k = kandidat *itemset* dengan panjang *k-item*

F_k = *frequent itemset* dengan panjang *k-item*

F = Output frequent itemset

Step 0:

$$C_1 \leftarrow \{A \rightarrow 4, B \rightarrow 4, C \rightarrow 1, D \rightarrow 3, E \rightarrow 3, F \rightarrow 1, G \rightarrow 1\}$$

Step 1:

$$F_1 \leftarrow \{A \rightarrow 4, B \rightarrow 4, D \rightarrow 3, E \rightarrow 3\}$$

$$C_2 \leftarrow \{AB \rightarrow 4, AD \rightarrow 2, AE \rightarrow 2, BD \rightarrow 2, BE \rightarrow 2, DE \rightarrow 3\}$$

Step 2:

$$F_2 \leftarrow \{AB \rightarrow 4, AD \rightarrow 2, AE \rightarrow 2, BD \rightarrow 2, BE \rightarrow 2, DE \rightarrow 3\}$$

$$C_3 \leftarrow \{ABD \rightarrow 2, ABE \rightarrow 2, ADE \rightarrow 2, BDE \rightarrow 2\}$$

Step 3:

$$F_3 \leftarrow \{ABD \rightarrow 2, ABE \rightarrow 2, ADE \rightarrow 2, BDE \rightarrow 2\}$$

$$C_4 \leftarrow \{ABDE \rightarrow 2\}$$

Step 4:

$$F_4 \leftarrow \{ABDE \rightarrow 2\}$$

$$C_5 \leftarrow \{\ \}$$

Final Result

$$F \leftarrow \{F_1 \cup F_2 \cup F_3 \cup F_3\}$$

$$\left\{\begin{matrix} A, B, D, E, AB, AD, AE, BD, BE, DE, ABD, ABE, ADE, \\ BDE, ABDE \end{matrix}\right\}$$

Frequent itemset yang didapatkan dengan menggunakan *Apriori* ditampilkan pada Tabel 4.6, sehingga kandidat Topik yang didapatkan sebanyak 15 Topik. Dengan menggunakan pendekatan *Closed Frequent Itemset*, maka dari 15 topik tersebut yang

merupakan Topik yang unik dan *single* adalah {databese, system}, {award, activity}, dan {databese, system, award, activity}.

Tabel 4.6 *Frequent Itemset (Apriori)* data contoh

Itemset	*Support Count*	*Support*
{database}	4	80%
{system}	4	80%
{award}	3	60%
{activity}	3	60%
{database,system}*	4	80%
{database,award}	2	40%
{database,acticity}	2	40%
{system,award}	2	40%
{system,acticity}	2	40%
{award,activity}*	3	60%
{database,system,award}	2	40%
{database,system,activity}	2	40%
{databese,award,acticity}	2	40%
{system,award,acticity}	2	40%
{databese, system, award, activity}*	2	40%

(*) *Closed Frequent Itemset*

b. Algoritma *ECLAT*

Berbeda dengan algoritma *Apriori* yang menggunakan *horizontal databese layout* (Tabel 4.5), Algoritma *ECLAT* menggunakan *vertical databese layout* (Tabel 4.7). Setiap *item* dinyatakan dalam tabel *tid-list* secara vertikal dan menggunakan titik potong *tid-list* antar item untuk menghitung *support*.

Tabel 4.7 *Vertical Databese layout*

Item	TID set
database (*A*)	1,2,4,5
system (*B*)	1,2,4,5
computational (*C*)	1
award (*D*)	2,3,5
activity (*E*)	2,3,5

Tabel 4.7 Lanjutan

Item	TID set
scalability (*F*)	3
game (*G*)	5

Berikut merupakan langkah-langkah dalam algoritma *ECLAT* sampai didapatkan *frequent itemset* dengan menggunakan minimum support = 2

Input :

$F_1 = F = \{A, B, D, E\}$

eclat$(F_1 = \{A, B, D, E\})$

Step 1

- *Candidates* : $\{AB \to 4, AD \to 2, AE \to 2\}$
- *Frequent* (F_2): $\{AB \to 4, AD \to 2, AE \to 2\}$
- *adding {AB, AD, AE}* to *F*
- eclat$(F_2 = \{AB, AD, AE\})$
  - *Candidates* : $\{ABD \to 2, ABE \to 2, ADE \to 2\}$
  - *Frequent* (F_3): $\{ABD \to 2, ABE \to 2, ADE \to 2\}$
  - *adding {ABD, ABE, ADE}* to *F*
  - eclat$(F_3 = \{ABD, ABE, ADE\})$
    - *Candidates* : $\{ABDE \to 2\}$
    - *Frequent* (F_3): $\{ABDE \to 2\}$
    - *adding {ABDE}* to *F*
    - eclat$(F_4 = \{ABDE\})$
      - *Candidates* : {}
      - *Frequent* (F_4): {}

Step 2

- *Candidates*: $\{BD \to 2, BE \to 2\}$
- *Frequent* (F_2) : $\{BD \to 2, BE \to 2\}$
- *adding {BD, BE}* to *F*

- eclat*(F_2={BD, BE})*
  - *Candidates* : $\{BDE \rightarrow 2\}$
  - *Frequent (F_3)*: $\{BDE \rightarrow 2\}$
  - *adding {BDE}* to *F*
  - eclat*(F_3={BDE})*
    - *Candidates* : {}
    - *Frequent (F_3)*: {}

Step 3
- *Candidates* : $\{DE \rightarrow 3\}$
- *Frequent (F_2)* : $\{DE \rightarrow 3\}$
- *adding DE* to *F*
- eclat*(F_2=DE)*
  - *Candidates* : {}
  - *Frequent (F_3)*: {}

Final Result

$$F \leftarrow \begin{Bmatrix} A, B, D, E, AB, AD, AE, ABD, ABE, ADE, ABDE, BD, BE, \\ BDE, DE \end{Bmatrix}$$

Hasil frequent itemset dengan menggunakan *ECLAT* sama dengan *Apriori* yang ditampilkan pada Tabel 4.8. Dari 15 itemset tersebut dihasilkan 3 Topik dengan menggunakan pendekatan *Closed Frequent Itemset*

Tabel 4.8 *Frequent Itemset (ECLAT)* data contoh

Itemset	*Support Count*	*Support*
{database}	4	80%
{system}	4	80%
{award}	3	60%
{activity}	3	60%
{database,system}*	4	80%
{database,award}	2	40%
{database,acticity}	2	40%
{database,system,award}	2	40%

Tabel 4.8 Lanjutan

Itemset	*Support Count*	*Support*
{database,system,activity}	2	40%
{databese,award,acticity}	2	60%
{databese, system, award, activity}*	2	40%
{system,award}	2	40%
{system,acticity}	2	40%
{system,award,acticity}	2	40%
{award,activity}*	3	40%

(*) *Closed Frequent Itemset*

Kemudian apabila menggunakan data kata *keywords* dari jurnal-jurnal *IEEE Xplore* maka *frequent itemset* yang dihasilkan dengan algoritma *Apriori* dan *ECLAT* ditampilkan seperti Tabel 4.9, dengan menggunakan *minimum support* sebesar 5%

Tabel 4.9 *Frequent Itemset* data dari jurnal *IEEE Xplore*

No	itemset	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
16	{algorithm,approximation}	0.073	247
17	{system,data}	0.073	246

Tabel 4.9 Lanjutan

No	itemset	support	count
18	{service,network}	0.071	238
19	{model,predictive}	0.070	236
20	{service,network,social}	0.069	233
21	{network,social}	0.069	233
22	{service,social}	0.069	233
23	{modeling,model}	0.069	232
24	{algorithm,mining}	0.065	220
25	{algorithm,data,mining}	0.065	219
26	{science,data}	0.062	209
27	{algorithm,model}	0.060	201
28	{computer,science,data}	0.058	197
29	{modeling,data}	0.058	197
30	{feature,extraction}	0.057	192
31	{model,mathematical}	0.056	189
32	{analysis,mining}	0.056	188
33	{analysis,data,mining}	0.055	187
34	{data,network}	0.053	179
35	{algorithm,analysis,data}	0.052	176
36	{database,system}	0.051	173

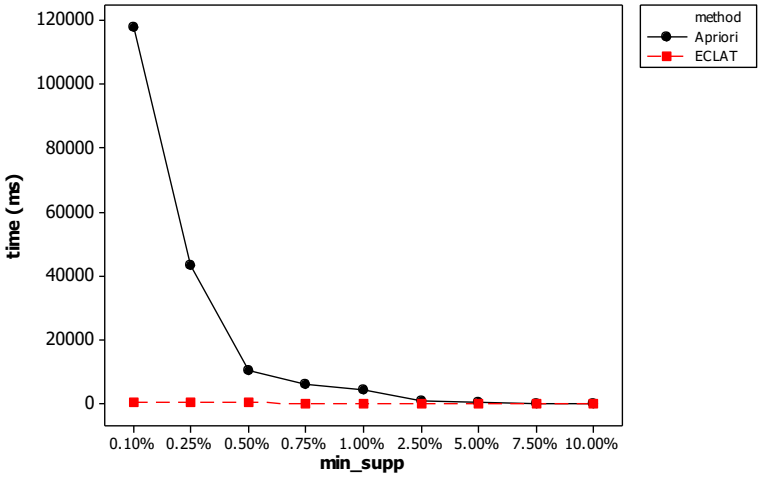
Dua kata yang paling sering muncul bersamaan di *keywords* adalah *data* dan *mining*. Dengan *support* sebesar 19,9% maka dapat diinterpretasikan bahwa kedua kata tersebut muncul bersaamaan di *keywords* sebanyak 671 dokumen dari 3374 dokumen.

c. Perbandingan Algoritma *Apriori* dan *ECLAT*

Untuk membandingkan performa pada masing-masing algoritma digunakan waktu pada masing-masing algoritma pada saat membangkitkan *frequent itemset* dengan *minimum support* yang berbeda-beda. Perbandingan waktu pembangkitan itemset dapat dilihat pada Gambar 4.9

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 databese* sehingga untuk menghitung nilai *support* tidak perlu melakukan *scan databese* secara berulang.



Gambar 4.9 Perbandingan algoritma Apriori vs ECLAT

4.2.3 Ekstraksi Topik menggunakan *Closed Frequent Itemset*

Closed frequent itemset digunakan untuk mengurangi jumlah *frequent itemset* yang dihasilkan. Sebagai contoh itemset {association, mining} dan {association, rule, mining} memiliki nilai *support* yang sama, sehingga itemset yang digunakan hanya

satu karena kedua itemset tersebut mengandung makna Topik yang sama.

Tabel 4.10 merupakan perbandingan jumlah itemset antara *frequent itemset*, *closed frequent itemset* dan *remove subset*.

Tabel 4.10 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 4.10, *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 juga *minimum support* sebesar 5% maka Topik yang terbentuk dapat dilihat pada Lampiran 6. Lampiran 6 menampilkan informasi tentang frekuensi kemunculan Topik per tahun. Dengan menggunakan data pada Lampiran 6 maka dapat diketahui *trend* dari masing-masing Topik. Topik-topik seperti seperti *data model, computational modeling, algorithm approximation, model predictive, algorithm model, modeling data, feature extraction, database system* mengalami kenaikan *trend* dari tahun 2009 ke

tahun 2018. Sedangkan *algorithm clustering*, *processing query*, *database system* mengalami penurunan *trend* dari tahun 2009 ke tahun 2018.

4.2.4 *Topic Community*

Dalam menemukan satu set topik yang berkorelasi tinggi satu sama lain (atau berkorelasi terbalik), maka digunakanlah koefisien korelasi antara dua topik untuk membangun grafik. Setiap *node* dari grafik menunjukkan sebuah topik, dan garis *edge* yang menghubungkan dua *node* menunjukkan tingkat korelasi antara dua topik. *Network graph* ini memungkinkan untuk memodelkan interaksi antar *itemset* secara eksplisit. Sebagai contoh, diambil 2 topik dari Lampiran 6

{data, mining} → $X_1 = [156, 71, 53, 45, 47, 36, 66, 75, 49, 73]$

{algorithm, data} → $X_4 = [88, 42, 32, 29, 32, 13, 32, 56, 49, 41]$

rata-rata dihitung sebagai berikut,

$$\bar{X}_1 = \frac{156+71+53+45+47+36+66+75+49+73}{10} = 67,1$$

$$\bar{X}_4 = \frac{88+42+32+29+32+13+32+56+49+41}{10} = 41,4$$

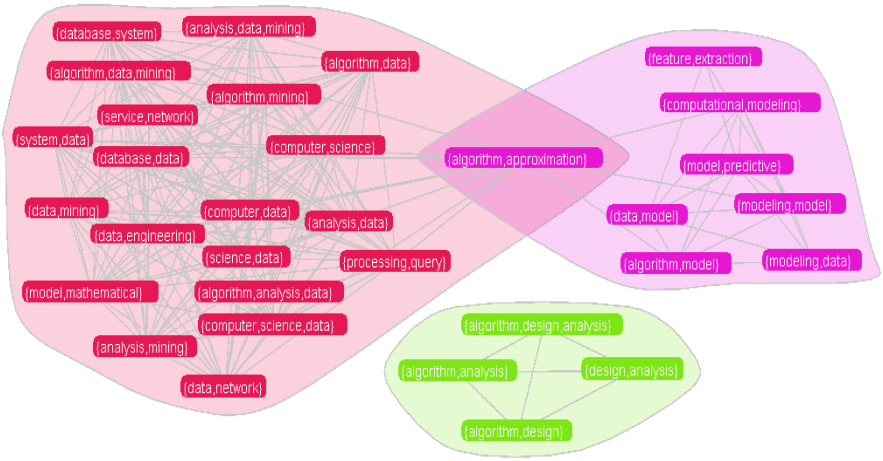
dengan menggunakan persamaan (2.3), maka korelasi antara X_1 dan X_4 sebesar 0.9081. Untuk menemukan sekumpulan topik yang berkorelasi tinggi maka dilakukanlah pengujian korelasi (Lampiran 7). Sehingga apabila ada 2 topik yang tidak mempunyai hubungan yang signifikan, maka garis *edge* antar 2 topik tersebut akan dihapus. Dengan menggunakan persamaan (2.4) maka statistik uji yang digunakan adalah,

$$t = \frac{0,9081\sqrt{10-2}}{\sqrt{1-0,9081^2}} = 6,1337$$

Taraf signifikansi yang digunakan sebesar 5% dengan daerah penolakannya adalah $t < -2,306$ atau $t > 2,306$. Nilai statistik uji yang digunakan (6,1337) lebih besar dari 2,306, sehingga keputusan yang dapat diambil adalah Tolak H_0, yang artinya *itemset* {data, mining} dengan {algorithm, data} mempunyai

hubungan yang signifikan. Sehingga pada *network graph* kedua *itemset* tersebut mempunyai garis *edge*.

Setelah semua *itemset* dihubungkan dengan masing-masing *edge*, maka langkah selanjutnya yaitu mencari *Community* dengan menggunakan algoritma *Community detection* berbasis CPM (*Clique Percolation Method)*. Dengan menggunakan bantuan *software CFinder* didapatkanlah grup topik dengan korelasi kuat. Dari Gambar 4.10 terlihat bahwa dari 34 topik terbentuk 3 kelompok yang saling *overlapping*. Dimana topik-topik yang berada dalam satu kelompok mempunyai korelasi yang signifikan. Sebagai contoh diambil dua topik dari kelompok bewarna ungu yaitu *modeling data* dan *model predictive*, maka dapat diinterpretasikan apabila banyak jurnal dengan topik *modeling data* meningkat maka jurnal dengan topik *model predictive* juga akan turut meningkat.



Gambar 4.10 *Topic Community* dari 34 Topik

4.3 *Association Rule* dengan *item* per-*keywords*

Setelah pada pembahasan sebelummnya membahas tentang *Topic discovery* dengan menggunakan *closed frequent itemset*. Pada sub bab ini akan dilakukan analisis *Association Rule* jika item

yang dipakai adalah per-*keywords*, sehingga 1 item diwakili oleh 1 *keywords*.

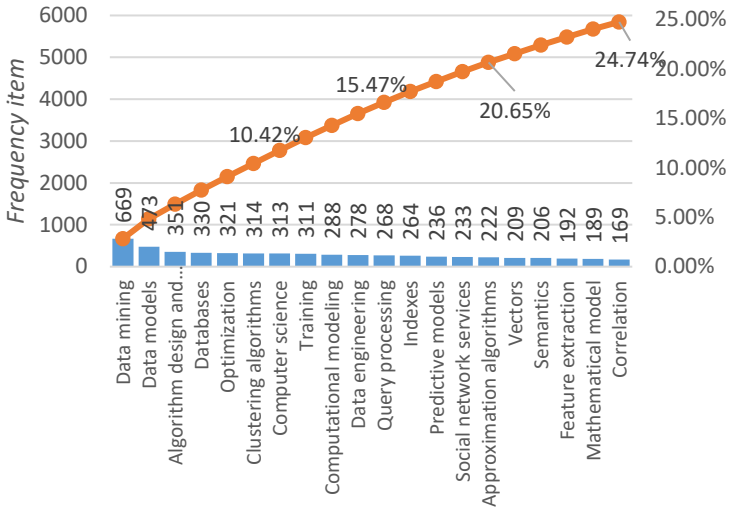
4.3.1 *Preprocessing Data*

Dengan menggunakan 1-*keywords* sebagai item maka untuk *preprocessing data* tidak banyak yang dilakukan, hanya melakukan pemisahan masing-masing *keywords* yang dipisahkan dengan tanda baca koma (,). Setelah itu data akan diubah ke dalam bentuk *document term matrix* dan siap untuk dilakukan analisis pada tahap selanjutnya. Struktur *document term matrix* yang terbentuk dapat dilihat pada Tabel 4.11.

Tabel 4.11 *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

Sebelum dilakukan *Association Rule* maka terlebih dahulu dilihat karakteristik data dari *keywords* yang diambil. Gambar 4.11 merupakan plot 20 dari *keywords* yang sering digunakan di jurnal ICDE dan ICDM. *Keywords* yang paling sering muncul di 10% dari keseluruhan *keywords* adalah *Data mining, Data models, Algorithm design and analysis, Databese, Optimization, Clustering algorithm. Keywords Data mining* muncul di 669 dokumen dari 3374 dokumen. Sedangkan *Data models* muncul di 473 dokumen dari 3374 dokumen. Apabila dibandingkan dengan hasil *Association rule* pada Lampiran 6 maka dari 20 keywords yang paling sering muncul tersebut mampu dideteksi menggunakan *closed frequent itemset* dengan menggunakan 1-kata sebagai item.



Gambar 4.11 *Frequency keywords*

Adapun untuk melihat perkembangan masing-masing *keywords* dari tahun 2009 sampai 2018 dapat dilihat pada Lampiran 8. Dengan 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 4.12

Tabel 4.12 Model regresi *time series* dari 11 *keywords*

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%
Query processing	$X_{tahun} = 40.2 - 2.44 * tahun$	0.047	40.6%

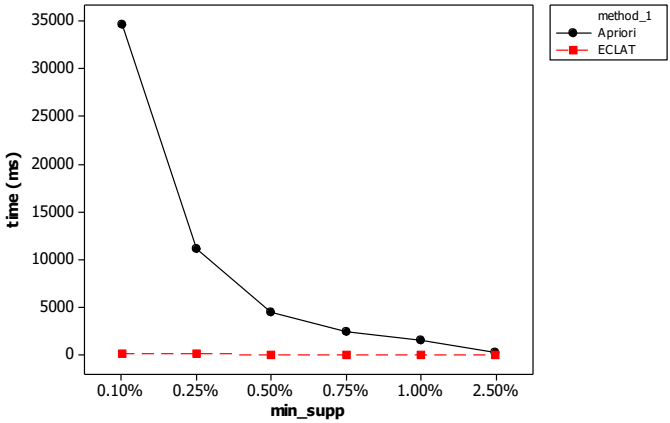
Tabel 4.12 Lanjutan

Keywords (X)	Model	*P-value*	R^2
Indexes	$X_{tahun} = 4.87 + 3.92 * tahun$	0.001	76.5%
Predictive models	$X_{tahun} = 8.8 + 2.69 * tahun$	0.002	72.1%
Approximation algorithms	$X_{tahun} = 8.47 + 2.5 * tahun$	0.009	59.8%
Semantics	$X_{tahun} = 5.73 + 2.7 * tahun$	0.007	62.0%
Feature extraction	$X_{tahun} = 3.13 + 2.92 * tahun$	0.008	61.0%
Correlation	$X_{tahun} = 3.87 + 2.37 * tahun$	0.006	63.6%

Keywords Clustering algorithm dan *Query processing* mengalami *trend* penurunan dari tahun 2009 sampai tahun 2018. Sedangkan 9 *keywords* sisanya mengalami peningkatan dari tahun ke tahun antara tahun 2009 sampai tahun 2018.

4.3.2 *Frequent Itemset Mining*

Dalam proses membangkitkan *frequent itemset* digunakan dua algoritma yang berbeda yaitu *Apriori* dan *ECLAT*. Perbandingan waktu yang dibutuhkan untuk membangkitkan *frequent itemset* ditampilkan pada Gambar 4.12



Gambar 4.12 Perbandingan algoritma Apriori vs ECLAT

Dari penggunaan beberapa nilai *minimum support* yang berbeda, terlihat bahwa algoritma *ECLAT* mempunyai waktu yang lebih cepat dalam membangkitkan *frequent itemset* bila dibandingkan *Apriori*. Dengan menggunakan *minimum support* sebanyak 1% maka hasil *frequent itemset* ditampilkan pada Tabel 4.14

Tabel 4.13 55 *Frequent Itemset* dengan *item per-keywords*

No	items	support	count
1	{Computational modeling,Data models}	0.0302	102
2	{Computer science,Data engineering}	0.0299	101
3	{Data models,Predictive models}	0.0293	99
4	{Computer science,Data mining}	0.0285	96
5	{Clustering algorithms,Data mining}	0.0246	83
6	{Data mining,Data models}	0.0234	79
7	{Algorithm design and analysis, Data mining}	0.0228	77
8	{Data mining,Feature extraction}	0.0205	69
9	{Data models,Training}	0.0202	68
10	{Data engineering,USA Councils}	0.0196	66
11	{Algorithm design and analysis, Approximation algorithms}	0.0184	62
12	{Algorithm design and analysis, Clustering algorithms}	0.0172	58
13	{Approximation algorithms, Approximation methods}	0.0169	57
14	{Data models,Mathematical model}	0.0169	57
15	{Equations,Mathematical model}	0.0163	55
16	{Conferences,Data mining}	0.0163	55
17	{Data mining,Databases}	0.0157	53
18	{Data engineering,Data mining}	0.0154	52
19	{Conferences,Data engineering}	0.0151	51
20	{Accuracy,Training}	0.0151	51
21	{Data mining,Social network services}	0.0151	51
22	{Data mining,Predictive models}	0.0151	51
23	{Clustering algorithms, Partitioning algorithms}	0.0148	50

Tabel 4.13 Lanjutan

No	items	support	count
24	{Computational modeling,Mathematical model}	0.0148	50
25	{Data mining,Itemsets}	0.0145	49
26	{Data engineering,Query processing}	0.0145	49
27	{Predictive models,Training}	0.0142	48
28	{Data mining,Training}	0.0139	47
29	{Analytical models,Data models}	0.0136	46
30	{Indexes,Query processing}	0.0136	46
31	{Algorithm design and analysis,Optimization}	0.0136	46
32	{Algorithm design and analysis, Heuristic algorithms}	0.0133	45
33	{Data mining,Optimization}	0.0133	45
34	{Data models,Optimization}	0.0130	44
35	{Computer science,Databases}	0.0130	44
36	{Computer science,USA Councils}	0.0124	42
37	{Data models,Probabilistic logic}	0.0122	41
38	{Computational modeling,Predictive models}	0.0122	41
39	{Optimization,Training}	0.0122	41
40	{Data engineering,Databases}	0.0122	41
41	{Computer science,Query processing}	0.0119	40
42	{Data privacy,Privacy}	0.0116	39
43	{Prediction algorithms,Predictive models}	0.0116	39
44	{Kernel,Training}	0.0116	39
45	{Clustering algorithms,Optimization}	0.0116	39
46	{Training,Training data}	0.0110	37
47	{Computational modeling,Data mining}	0.0110	37
48	{Data mining,Vectors}	0.0107	36
49	{Data mining,Machine learning}	0.0104	35
50	{Algorithm design and analysis, Partitioning algorithms}	0.0104	35
51	{Computational modeling, Social network services}	0.0104	35
52	{Computational modeling,Training}	0.0104	35
53	{Data models,Databases}	0.0104	35

Tabel 4.13 Lanjutan

No	items	support	count
54	{Feature extraction,Training}	0.0101	34
55	{Training,Vectors}	0.0101	34

Dari Tabel 4.13 dapat diketahui bahwa *2 keyword* yang paling sering muncul bersamaan adalah *Computational modelling* dan *Data* model. Dengan mengambil salah satu *itemset* dari Tabel 4.14 yaitu itemset {Data mining, feature extraction}, maka dapat diinterpretasikan bahwa *keywords Data mining* dengan *Feature extraction* muncul bersamaan di 69 dokumen dari 3374 dokumen.

4.3.3 Ekstraksi Topik menggunakan *Closed Frequent Itemset*

Tabel 4.14 merupakan perbandingan jumlah itemset antara *frequent itemset*, *closed frequent itemset* dan *remove subset*.

Tabel 4.14 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

Dari Tabel 4.14 terlihat bahwa untuk minimum support lebih dari 0,25% tidak ada perbedaan antara *frequent itemset* dengan *closed frequent itemset*. Sehingga dengan menggunakan *keywords* sebagai item, *closed frequent itemset* tidak perlu digunakan.

Hasil dari 55 Topik dengan masing-masing frekuensi kemunculan tiap tahun disajikan pada Lampiran 9. Topik-topik seperti {Computational modeling, Data models}, {Data models, Predictive models}, {Data mining, Data models}, {Conferences, Data mining}, {Conferences, Data engineering}, {Computational modeling, Mathematical model}, {Analytical models, Data models}, {Computational modeling, Predictive models},

{Computational modeling, Social network services} mengalami kenaikan dari tahun ke tahun. Sedangkan {Data mining, Databases} mengalami penurunan dari tahun ke tahun antara tahun 2009 sampai 2018.

(*Halaman ini sengaja dikosongkan*)

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Analisis dan pembahasan yang dilakukan menunjukan hasil yang mendukung beberapa kesimpulan sebagai berikut :

1. Setelah dilakukan *preprocessing* pada *keywords* kemudian divisualisasikan dengan menggunakan *wordcloud* dan diagram pareto, maka kata yang paling sering muncul di *keywords* adalah *data, algorithm, model, analysis, mining, databese,* dan *system*. Sedangkan apabila menggunakan *bigram* maka 2 kata berurutan yang paling sering muncul di *keywords* adalah *data mining, data model, algorithm design, design analysis*. Apabila menggunakan *trigram* maka 3 kata berurutan yang paling sering muncul di *keywords* adalah *algorithm design analysis, social network service, time series analysis, support vector machine, real time system*.
2. 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*.
3. Pada *association rule* dengan *item* per kata, dengan menggunakan *closed frequent itemset* dan juga *minimum support* sebesar 5%, maka Topik yang terbentuk adalah sebanyak 34 topik. 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. Sedangkan apabila menggunakan *item* per *keywords* dengan *minimum support* sebesar 1% didapatkan 55 topik. 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.

5.2 Saran

Saran yang dapat diberikan untuk pembaca yang ingin membuat penelitian tentang *Data Mining* atau *Data Engineers* yaitu topik-topik yang sering digunakan untuk penelitian antara lain *Data modeling, Mathematical model, Predictive model, Algorithm design & analysis, Clustering algorithm, Query processing, Approximation methods, Feature extraction, Social network services*. Sehingga pembaca yang ingin membuat penelitian tentang *Data Mining* atau *Data Engineers* dapat mencari bahan dengan *keywords* seperti yang disebut diatas.

Saran yang dapat diberikan untuk penelitian selanjutnya adalah menggunakan data *text* pada isi abstrak maupun judul penelitian yang terdapat di abstrak sehingga topik yang didapatkan diharapkan akan lebih representatif.

DAFTAR PUSTAKA

Agrawal, R., & Srikant, R. (1994). Fast Algorithms for Mining Association Rules in Large Databases. *Proceedings of the 20th International Conference on Very Large Data Bases*, 487-499.

Borgelt, C. (2003). Efficient Implementations of Apriori and Eclat. *Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations*.

Bowerman, B. I., O'Connell, R. T., & Koehler, A. B. (2005). *Forecasting, Time Series and Regression in Applied Approach* (4rd ed.). California: Duxbury Press.

Han, J., Kamber, M., & Pei, J. (2012). *Data Mining Concepts and Techniques* (3rd ed.). Waltham (US): Morgan Kaufmann Publisher.

Hurtado, J. L., Agarwal, A., & Zhu, X. (2016). Topic discovery and future trend forecasting for texts. *Journal of Big Data, III*. doi:10.1186/s40537-016-0039-2

IEEE. (2019a, June 12). *About IEEE Xplore® Digital Library*. Retrieved from https://ieeexplore.ieee.org/xpl/aboutUs.jsp

IEEE. (2019b, March 24). *IEEE International Conference on Data Mining (ICDM)*. Retrieved from https://ieeexplore.ieee.org/xpl/conhome/1000179/all-proceedings

IEEE. (2019c, May 1). *International Conference on Data Engineering (ICDE)*. Retrieved from https://ieeexplore.ieee.org/servlet/opac?punumber=1000178

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*. New York: Cambridge University Press.

Shubankar, K., Singh, A., & Pudi, V. (2011). A frequent keyword-set based algorithm for topic modeling and clustering of

research papers. *3rd Conference on Data Mining and Optimization (DMO)*, 96-102.

Shumway, R. H., & Stoffer, D. S. (2006). *Time Series Analysis and Its Application with R* (3rd ed.). New York: Springer.

Walpole, R. E., Myers, R. H., Myers, S. L., & Ye, K. (2011). *Probability & Statistics for Engineers & Scientists* (9th ed.). Boston: Prentice Hall.

Xu, G., Zhang, Y., & Li, L. (2011). *Web Mining and Social Networking : Techniques and Applications.* New York: Springer.

Zaki, M. J., Parthasarathy, S., Ogihara, M., & Li, W. (1997). New algorithms for fast discovery of Association Rules. *In 3rd International Conference on Knowledge and Data Engineering*, 283-286.

Zheng, Z., Kohav, R., & Mason, L. (2001). Real world performance of association rule algorithms. *In ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 401-406.

LAMPIRAN

Lampiran 1. Syntax R *crawling keywords*

```
library(rvest)
library(textclean)
library(qdapRegex)
library(revtools)

get.keywords= function(url){
  # read html
  download.file(url, destfile = 'C:/Users/Mustofa/Documents/sample.html')
  page <- read_html('C:/Users/Mustofa/Documents/sample.html')
  node <- html_nodes(page,xpath = '//*[@id="LayoutWrapper"]/div/div/div/script[3]/text()')
  output <- html_text(node)

  # remove symbol & punctuation
  data=replace_white(output)  #remove /t /n
  data=gsub("[^A-Za-z,{}]", " ", data) #remove all character except number & alphabet
  data=gsub(' ,',',',data) #remove space
  data=gsub(' +',' ',data) #remove double space

  # extraxt keywords
  keyword.ieee = data.frame(IEEE_Keywords=rm_between(data, 'type IEEE Keywords, kwd', '}', extract=TRUE)[[1]])
  keyword.controlled = data.frame(controlled_index=rm_between(data, 'type INSPEC Controlled Indexing, kwd', '}', extract=TRUE)[[1]])
  keyword.noncontrolled = data.frame(noncontrolled_index=rm_between(data, 'type INSPEC Non Controlled Indexing, kwd', '}', extract=TRUE)[[1]])
  keyword.author = data.frame(author_keyword=rm_between(data, 'type Author Keywords , kwd', '}', extract=TRUE)[[1]])
  keywords = cbind(keyword.ieee, keyword.controlled, keyword.noncontrolled, keyword.author)
  return(keywords)
```

```
}

df = read_bibliography("D:/My Document/LJ/SEMESTER
4/Topic Discovery ~ TA/Data/ALL.ris", return_df = TRUE)
write.csv2(df,"C:/Users/Mustofa/Documents/Py-
thon/Data/ALL.csv")
DOI = df$doi
url <- paste0("http://doi.org/", DOI)

n = length(df$url)
all_keywords = NULL
for (i in 1:n) {
  new_keywords <- get.keywords(url[i])
  all_keywords <- rbind(all_keywords, new_keywords)
}

write.csv2(all_keywords,"C:/Users/Mustofa/Documents/Py-
thon/Data/all_keywords.csv")
```

Lampiran 2. Syntax Python untuk Preprocessing Data

```
import pandas as pd

## Import data
data = pd.read_csv(r'C:\Users\Mustofa\Documents\Py-
thon\Data\all_keywords.csv', sep=';')
keywords = data['IEEE_Keywords']
keywords.size

## lower case, remove koma, remove and
import re

data_lower = []
remove_koma =[]
remove_and = []

for line in keywords:
    result1 = str(line).lower()
    result2 = re.sub(',', '', result1)
    result3 = re.sub('and', '', result2)
    data_lower.append(result1)
    remove_koma.append(result2)
```

```
    remove_and.append(result3)
remove_and

## Lemmatization
import nltk
from nltk.stem import WordNetLemmatizer
from nltk.tokenize import word_tokenize
wnl = WordNetLemmatizer()

def lemmaSentence(sentence):
    token_words=word_tokenize(sentence)
    token_words
    lemma_sentence=[]
    for word in token_words:
        lemma_sentence.append(wnl.lemmatize(word,
        pos='n'))
        lemma_sentence.append(" ")
    return "".join(lemma_sentence)

data_lemma = []
for line in remove_and:
    result=lemmaSentence(line)
    data_lemma.append(result)

data_lemma

## export post-processing
pd.DataFrame(data_lemma).to_csv(r'C:\Us-
ers\Mustofa\Documents\Python\Data\post_key-
words.csv')
```

Lampiran 3. Syntax R untuk *Wordcloud*

```
library("quanteda")
library("wordcloud")
library("RColorBrewer")

input <- read.csv("C:/Users/Mustofa/Documents/Py-
thon/Data/post_keywords.csv")
keyword <- as.character(input$X0)

# WordCloud 1-word
doc.freq <- sort(docfreq(df_trim),decreasing=TRUE)
```

```
term <- data.frame(word =
names(doc.freq),freq=doc.freq, row.names = NULL)
set.seed(1234)
wordcloud(words = term$word, freq = term$freq,
min.freq = 1,max.words=200, random.order=FALSE,
rot.per=0.35,colors=brewer.pal(8, "Dark2"))

# WordCloud 2-word
toks2 <- tokens_ngrams(tokens(keyword), n = 2, con-
catenator = " ")
df_bigram <- dfm(toks2)
doc.freq2 <- sort(docfreq(df_bigram),decreas-
ing=TRUE)
term2 <- data.frame(word =
names(doc.freq2),freq=doc.freq2, row.names = NULL)
set.seed(1234)
wordcloud(words = term2$word, freq = term2$freq,
min.freq = 1,max.words=200, random.order=FALSE,
rot.per=0.35,colors=brewer.pal(8, "Dark2"))

# WordCloud 3-word
toks3 <- tokens_ngrams(tokens(keyword), n = 3, con-
catenator = " ")
df_bigram <- dfm(toks3)
doc.freq3 <- sort(docfreq(df_bigram),decreas-
ing=TRUE)
term3 <- data.frame(word =
names(doc.freq3),freq=doc.freq3, row.names = NULL)
set.seed(1234)
wordcloud(words = term3$word, freq = term3$freq,
min.freq = 1,max.words=200, random.order=FALSE,
rot.per=0.35, colors=brewer.pal(8, "Dark2"))
```

Lampiran 4. Syntax R Association Rule (1 kata = 1 item)

```
library("arules")
library("arulesViz")
library("quanteda")
library("Rgraphviz")

input <- read.csv("C:/Users/Mustofa/Documents/Py-
thon/Data/post_keywords.csv")
```

```
keyword <- as.character(input$X0)
# convert data to dtm & remove sparse term
df <- dfm(keyword, verbose = FALSE)
df_trim <- dfm_trim(df, min_docfreq = 3) #item/kata
yang muncul <3 dihapus

# Export
write.csv2(data.frame(docfreq(df_trim)), file =
"C:/Users/Mustofa/Documents/Python/Data/sum_ko-
lom_keywords.csv")
write.csv2(as.matrix(df_trim), file = "C:/Us-
ers/Mustofa/Documents/Python/Data/dtm_trim_key-
words.csv")

# Convert dtm into transaction
dtm <- apply(df_trim,2,as.logical)
basket <- as(dtm, "transactions")

# ECLAT
eclat_set1 <- eclat(basket, parameter = list(sup-
port = 0.1, minlen=2, target='frequent item-
sets',tidLists=TRUE))

result.eclat <- inspect(eclat_set4)
tid <- inspect(tidLists(eclat_set3))
output <- data.frame(result.eclat, tid=tid$transac-
tionIDs)
write.csv2(output,"C:/Users/Mustofa/Documents/Py-
thon/Data/output_eclat.csv")

# Apriori
app_set1 <- apriori(basket, parameter = list(sup-
port = 0.1,  minlen=2, target='frequent itemsets'))

result2 <- inspect(sort(app_set1, by = 'support'))
write.csv2(result2,"C:/Users/Mustofa/Documents/Py-
thon/Data/output_apriori.csv.csv")

# Closed itemset
closed1 <- apriori(basket, parameter = list(support
= 0.1,  minlen=2, target='closed frequent item-
sets'))

result3 <- inspect(sort(closed1, by = 'support'))
```

```
#remove subset rules
subset.item <- which(rowSums(is.subset(app_set9,
app_set9)) > 1)
subset.remove <- app_set9[-subset.item]
inspect(sort(subset.remove, by = 'support'))
summary(subset.remove)
```

Lampiran 5. Syntax R untuk Correlation

```
flattenCorrMatrix <- function(cormat, pmat) {
  ut <- upper.tri(cormat)
  data.frame(
    row = rownames(cormat)[row(cormat)[ut]],
    column = rownames(cormat)[col(cormat)[ut]],
    cor  =(cormat)[ut],
    p = pmat[ut]
  )
}

library(Hmisc)
frequency <- read.csv("~/Python/Data/freq per
year.csv", sep=";", header = FALSE)
res2 <- rcorr(as.matrix(frequency))
corr.list <- flattenCorrMatrix(res2$r, res2$P)

# delete row with value >0.05
corr.list = corr.list[corr.list$p < 0.05,]
write.csv2(corr.list, file = "C:/Users/Mustofa/Doc-
uments/Python/Data/correlation_matrix.csv")
```

Lampiran 6. *Closed Frequent Itemset* dengan *minimum support* 5%

No	Itemset	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018
1	{data,mining}	156	71	53	45	47	36	66	75	49	73
2	{data,model}	39	50	42	50	40	52	61	77	68	84
3	{algorithm,analysis}	32	26	46	42	45	38	50	76	63	25
4	{algorithm,data}	88	42	32	29	32	13	32	56	49	41
5	{analysis,data}	72	37	21	25	23	14	27	41	40	62
6	{algorithm,design}	20	22	40	38	35	30	48	66	54	0
7	{design,analysis}	18	22	41	38	35	30	48	66	54	0
8	{algorithm,design,analysis}	18	22	40	38	35	30	48	66	54	0
9	{database,data}	125	41	17	19	19	16	15	28	27	37
10	{computer,science}	117	61	5	6	3	2	14	31	31	45
11	{algorithm,clustering}	43	38	32	41	30	22	24	31	33	20
12	{data,engineering}	213	11	3	9	0	4	4	5	28	32
13	{computational,modeling}	5	15	26	28	31	19	32	51	39	44
14	{processing,query}	52	39	20	27	30	12	18	27	19	25
15	{computer,data}	143	34	4	3	4	5	16	14	16	26
16	{algorithm,approximation}	5	18	29	29	23	15	36	26	33	33
17	{system,data}	99	27	5	8	5	12	21	24	23	22
18	{service,network}	32	15	16	18	23	14	18	40	30	32
19	{model,predictive}	17	17	11	18	25	14	29	33	32	40
20	{service,network,social}	28	14	16	18	23	14	18	40	30	32
21	{modeling,model}	2	19	14	18	24	17	32	39	29	38

No	Itemset	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018
22	{algorithm,mining}	52	21	22	18	14	8	19	31	17	18
23	{algorithm,data,mining}	52	20	22	18	14	8	19	31	17	18
24	{science,data}	117	29	2	1	1	1	8	11	14	25
25	{algorithm,model}	7	22	15	16	22	13	27	32	26	21
26	{computer,science,data}	111	28	2	1	1	1	8	10	13	22
27	{modeling,data}	9	21	12	15	14	15	24	36	24	27
28	{feature,extraction}	5	15	14	18	16	16	15	16	31	46
29	{model,mathematical}	2	14	22	22	26	22	17	24	21	19
30	{analysis,mining}	45	18	14	13	12	5	15	24	13	29
31	{analysis,data,mining}	45	18	14	13	12	5	15	24	13	28
32	{data,network}	63	10	8	7	7	8	11	24	11	30
33	{algorithm,analysis,data}	30	12	16	18	17	6	15	29	24	9
34	{database,system}	45	44	12	6	9	9	9	9	15	15

Lampiran 7. Pengujian Korelasi 34 Topik

X_1	X_2	*r*	*p-value*
{data,mining}	{algorithm,data}	0.9081	0
{data,mining}	{analysis,data}	0.8346	0.003
{algorithm,data}	{analysis,data}	0.8550	0.002
{algorithm,analysis}	{algorithm,design}	0.9433	0
{algorithm,analysis}	{design,analysis}	0.9419	0
{algorithm,design}	{design,analysis}	0.9993	0
{algorithm,analysis}	{algorithm,design,analysis}	0.9428	0
{algorithm,design}	{algorithm,design,analysis}	0.9995	0
{design,analysis}	{algorithm,design,analysis}	0.9999	0
{data,mining}	{database,data}	0.9552	0
{algorithm,data}	{database,data}	0.8784	0.001
{analysis,data}	{database,data}	0.8125	0.004
{data,mining}	{computer,science}	0.9287	0
{algorithm,data}	{computer,science}	0.8883	0.001
{analysis,data}	{computer,science}	0.8823	0.001
{database,data}	{computer,science}	0.9462	0
{data,mining}	{data,engineering}	0.9232	0
{algorithm,data}	{data,engineering}	0.8409	0.002
{analysis,data}	{data,engineering}	0.7682	0.009
{database,data}	{data,engineering}	0.9770	0
{computer,science}	{data,engineering}	0.8762	0.001
{data,model}	{computational,modeling}	0.8043	0.005
{data,mining}	{processing,query}	0.8253	0.003
{algorithm,data}	{processing,query}	0.7911	0.006
{analysis,data}	{processing,query}	0.6772	0.031
{database,data}	{processing,query}	0.8504	0.002
{computer,science}	{processing,query}	0.8451	0.002
{algorithm,clustering}	{processing,query}	0.7127	0.021
{data,engineering}	{processing,query}	0.7535	0.012
{data,mining}	{computer,data}	0.9635	0
{algorithm,data}	{computer,data}	0.8600	0.001
{analysis,data}	{computer,data}	0.7866	0.007
{database,data}	{computer,data}	0.9917	0
{computer,science}	{computer,data}	0.9374	0

X_1	X_2	*r*	*p-value*
{data,engineering}	{computer,data}	0.9784	0
{processing,query}	{computer,data}	0.8192	0.004
{database,data}	{algorithm,approximation}	-0.7093	0.022
{data,engineering}	{algorithm,approximation}	-0.6594	0.038
{computational,modeling}	{algorithm,approximation}	0.7658	0.01
{processing,query}	{algorithm,approximation}	-0.6468	0.043
{computer,data}	{algorithm,approximation}	-0.6898	0.027
{data,mining}	{system,data}	0.9613	0
{algorithm,data}	{system,data}	0.8861	0.001
{analysis,data}	{system,data}	0.7966	0.006
{database,data}	{system,data}	0.9759	0
{computer,science}	{system,data}	0.9365	0
{data,engineering}	{system,data}	0.9660	0
{processing,query}	{system,data}	0.7741	0.009
{computer,data}	{system,data}	0.9881	0
{algorithm,approximation}	{system,data}	-0.6559	0.039
{algorithm,data}	{service,network}	0.6947	0.026
{analysis,data}	{service,network}	0.7046	0.023
{data,model}	{model,predictive}	0.8566	0.002
{computational,modeling}	{model,predictive}	0.7962	0.006
{service,network}	{model,predictive}	0.6847	0.029
{data,model}	{service,network,social}	0.6397	0.046
{service,network}	{service,network,social}	0.9902	0
{model,predictive}	{service,network,social}	0.7440	0.014
{data,model}	{modeling,model}	0.8689	0.001
{computational,modeling}	{modeling,model}	0.9226	0
{algorithm,approximation}	{modeling,model}	0.7340	0.016
{model,predictive}	{modeling,model}	0.8642	0.001
{data,mining}	{algorithm,mining}	0.9458	0
{algorithm,data}	{algorithm,mining}	0.9327	0
{analysis,data}	{algorithm,mining}	0.7372	0.015
{database,data}	{algorithm,mining}	0.8864	0.001
{computer,science}	{algorithm,mining}	0.8370	0.003
{data,engineering}	{algorithm,mining}	0.8596	0.001
{processing,query}	{algorithm,mining}	0.7971	0.006
{computer,data}	{algorithm,mining}	0.8813	0.001

X_1	X_2	r	*p-value*
{system,data}	{algorithm,mining}	0.8918	0.001
{data,mining}	{algorithm,data,mining}	0.9437	0
{algorithm,data}	{algorithm,data,mining}	0.9314	0
{analysis,data}	{algorithm,data,mining}	0.7360	0.015
{database,data}	{algorithm,data,mining}	0.8836	0.001
{computer,science}	{algorithm,data,mining}	0.8286	0.003
{data,engineering}	{algorithm,data,mining}	0.8615	0.001
{processing,query}	{algorithm,data,mining}	0.7866	0.007
{computer,data}	{algorithm,data,mining}	0.8787	0.001
{system,data}	{algorithm,data,mining}	0.8900	0.001
{algorithm,mining}	{algorithm,data,mining}	0.9997	0
{data,mining}	{science,data}	0.9632	0
{algorithm,data}	{science,data}	0.8719	0.001
{analysis,data}	{science,data}	0.8184	0.004
{database,data}	{science,data}	0.9956	0
{computer,science}	{science,data}	0.9537	0
{data,engineering}	{science,data}	0.9757	0
{processing,query}	{science,data}	0.8282	0.003
{computer,data}	{science,data}	0.9977	0
{algorithm,approximation}	{science,data}	-0.6778	0.031
{system,data}	{science,data}	0.9848	0
{algorithm,mining}	{science,data}	0.8788	0.001
{algorithm,data,mining}	{science,data}	0.8758	0.001
{data,model}	{algorithm,model}	0.6526	0.041
{algorithm,analysis}	{algorithm,model}	0.6394	0.047
{computational,modeling}	{algorithm,model}	0.8009	0.005
{algorithm,approximation}	{algorithm,model}	0.6544	0.04
{model,predictive}	{algorithm,model}	0.6942	0.026
{modeling,model}	{algorithm,model}	0.8812	0.001
{data,mining}	{computer,science,data}	0.9625	0
{algorithm,data}	{computer,science,data}	0.8699	0.001
{analysis,data}	{computer,science,data}	0.8090	0.005
{database,data}	{computer,science,data}	0.9952	0
{computer,science}	{computer,science,data}	0.9521	0
{data,engineering}	{computer,science,data}	0.9753	0
{processing,query}	{computer,science,data}	0.8304	0.003

X_1	X_2	*r*	*p-value*
{computer,data}	{computer,science,data}	0.9984	0
{algorithm,approximation}	{computer,science,data}	-0.6836	0.029
{system,data}	{computer,science,data}	0.9854	0
{algorithm,mining}	{computer,science,data}	0.8796	0.001
{algorithm,data,mining}	{computer,science,data}	0.8764	0.001
{science,data}	{computer,science,data}	0.9998	0
{data,model}	{modeling,data}	0.8946	0
{computational,modeling}	{modeling,data}	0.8114	0.004
{model,predictive}	{modeling,data}	0.7817	0.008
{modeling,model}	{modeling,data}	0.9018	0
{algorithm,model}	{modeling,data}	0.8719	0.001
{data,model}	{feature,extraction}	0.7585	0.011
{model,predictive}	{feature,extraction}	0.7220	0.018
{modeling,model}	{feature,extraction}	0.6531	0.041
{data,mining}	{model,mathematical}	-0.8905	0.001
{algorithm,data}	{model,mathematical}	-0.7263	0.017
{analysis,data}	{model,mathematical}	-0.7001	0.024
{database,data}	{model,mathematical}	-0.9030	0
{computer,science}	{model,mathematical}	-0.9050	0
{data,engineering}	{model,mathematical}	-0.8803	0.001
{computational,modeling}	{model,mathematical}	0.6762	0.032
{processing,query}	{model,mathematical}	-0.7410	0.014
{computer,data}	{model,mathematical}	-0.9373	0
{system,data}	{model,mathematical}	-0.9218	0
{algorithm,mining}	{model,mathematical}	-0.7718	0.009
{algorithm,data,mining}	{model,mathematical}	-0.7643	0.01
{science,data}	{model,mathematical}	-0.9325	0
{computer,science,data}	{model,mathematical}	-0.9354	0
{data,mining}	{analysis,mining}	0.9496	0
{algorithm,data}	{analysis,mining}	0.9003	0
{analysis,data}	{analysis,mining}	0.9358	0
{database,data}	{analysis,mining}	0.8907	0.001
{computer,science}	{analysis,mining}	0.8949	0
{data,engineering}	{analysis,mining}	0.8485	0.002
{processing,query}	{analysis,mining}	0.7845	0.007
{computer,data}	{analysis,mining}	0.8746	0.001

X_1	X_2	r	*p-value*
{system,data}	{analysis,mining}	0.8704	0.001
{service,network}	{analysis,mining}	0.6354	0.048
{algorithm,mining}	{analysis,mining}	0.8964	0
{algorithm,data,mining}	{analysis,mining}	0.8961	0
{science,data}	{analysis,mining}	0.8908	0.001
{computer,science,data}	{analysis,mining}	0.8848	0.001
{model,mathematical}	{analysis,mining}	-0.7714	0.009
{data,mining}	{analysis,data,mining}	0.9560	0
{algorithm,data}	{analysis,data,mining}	0.9081	0
{analysis,data}	{analysis,data,mining}	0.9300	0
{database,data}	{analysis,data,mining}	0.8975	0
{computer,science}	{analysis,data,mining}	0.8989	0
{data,engineering}	{analysis,data,mining}	0.8556	0.002
{processing,query}	{analysis,data,mining}	0.7928	0.006
{computer,data}	{analysis,data,mining}	0.8821	0.001
{system,data}	{analysis,data,mining}	0.8788	0.001
{algorithm,mining}	{analysis,data,mining}	0.9073	0
{algorithm,data,mining}	{analysis,data,mining}	0.9069	0
{science,data}	{analysis,data,mining}	0.8973	0
{computer,science,data}	{analysis,data,mining}	0.8917	0.001
{model,mathematical}	{analysis,data,mining}	-0.7781	0.008
{analysis,mining}	{analysis,data,mining}	0.9996	0
{data,mining}	{data,network}	0.9505	0
{algorithm,data}	{data,network}	0.8740	0.001
{analysis,data}	{data,network}	0.8899	0.001
{database,data}	{data,network}	0.9320	0
{computer,science}	{data,network}	0.8808	0.001
{data,engineering}	{data,network}	0.9255	0
{processing,query}	{data,network}	0.7138	0.02
{computer,data}	{data,network}	0.9231	0
{system,data}	{data,network}	0.9291	0
{algorithm,mining}	{data,network}	0.8827	0.001
{algorithm,data,mining}	{data,network}	0.8859	0.001
{science,data}	{data,network}	0.9306	0
{computer,science,data}	{data,network}	0.9253	0
{model,mathematical}	{data,network}	-0.7937	0.006

X_1	X_2	*r*	*p-value*
{analysis,mining}	{data,network}	0.9590	0
{analysis,data,mining}	{data,network}	0.9604	0
{algorithm,data}	{algorithm,analysis,data}	0.8009	0.005
{service,network}	{algorithm,analysis,data}	0.6824	0.03
{algorithm,mining}	{algorithm,analysis,data}	0.7568	0.011
{algorithm,data,mining}	{algorithm,analysis,data}	0.7624	0.01
{data,mining}	{database,system}	0.7352	0.015
{algorithm,data}	{database,system}	0.6593	0.038
{database,data}	{database,system}	0.7948	0.006
{computer,science}	{database,system}	0.8824	0.001
{data,engineering}	{database,system}	0.6831	0.029
{computational,modeling}	{database,system}	-0.6686	0.035
{processing,query}	{database,system}	0.8210	0.004
{computer,data}	{database,system}	0.7914	0.006
{algorithm,approximation}	{database,system}	-0.6666	0.035
{system,data}	{database,system}	0.7565	0.011
{algorithm,mining}	{database,system}	0.6326	0.05
{science,data}	{database,system}	0.8047	0.005
{computer,science,data}	{database,system}	0.8087	0.005
{model,mathematical}	{database,system}	-0.8504	0.002
{analysis,mining}	{database,system}	0.6327	0.05
{analysis,data,mining}	{database,system}	0.6397	0.046

Lampiran 8. Frekuensi per tahun Top 20 *Keywords*

No	*Keywords*	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018
1	{Data mining}	156	71	53	45	47	36	65	75	49	72
2	{Data models}	13	42	35	47	35	49	49	69	61	73
3	{Algorithm design and analysis}	18	22	40	38	35	30	48	66	54	0
4	{Databases}	52	39	29	31	29	28	21	33	28	40
5	{Optimization}	1	24	28	43	24	31	34	46	48	42
6	{Clustering algorithms}	43	38	32	41	30	22	24	31	33	20
7	{Computer science}	117	59	5	6	3	2	14	31	31	45
8	{Training}	0	41	35	33	31	33	26	30	33	49
9	{Computational modeling}	3	15	26	28	31	19	32	51	39	44
10	{Data engineering}	212	8	0	6	0	0	0	0	24	28
11	{Query processing}	52	39	20	27	30	12	18	27	19	24
12	{Indexes}	3	13	18	27	29	20	42	32	31	49
13	{Predictive models}	17	17	11	18	25	14	29	33	32	40
14	{Social network services}	28	14	16	18	23	14	18	40	30	32
15	{Approximation algorithms}	5	12	24	25	22	11	32	26	32	33
16	{Vectors}	2	1	44	57	51	54	0	0	0	0
17	{Semantics}	0	5	28	19	23	24	22	26	25	34
18	{Feature extraction}	5	15	14	18	16	16	15	16	31	46
19	{Mathematical model}	2	14	22	22	26	22	17	24	21	19
20	{Correlation}	0	15	12	13	18	19	20	21	15	36

Lampiran 9. *Frequent Itemset* dengan *minimum support* 1%

No	itemset	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018
1	{Computational modeling, Data models}	0	9	6	9	7	8	12	20	15	16
2	{Computer science,Data engineering}	81	6	0	1	0	0	0	0	7	6
3	{Data models,Predictive models}	0	9	5	9	9	8	16	17	14	12
4	{Computer science,Data mining}	54	13	2	0	1	1	5	7	3	10
5	{Clustering algorithms,Data mining}	31	10	6	5	7	3	2	7	4	8
6	{Data mining,Data models}	4	8	5	8	6	6	9	13	7	13
7	{Algorithm design and analysis, Data mining}	8	5	9	11	8	4	10	16	6	0
8	{Data mining,Feature extraction}	4	5	5	9	5	7	5	8	6	15
9	{Data models,Training}	0	12	5	7	5	10	5	8	10	6
10	{Data engineering,USA Councils}	66	0	0	0	0	0	0	0	0	0
11	{Algorithm design and analysis, Approximation algorithms}	0	1	6	4	7	5	14	11	14	0
12	{Algorithm design and analysis, Clustering algorithms}	3	4	8	7	8	4	5	8	11	0
13	{Approximation algorithms, Approximation methods}	0	4	10	7	15	5	16	0	0	0
14	{Data models,Mathematical model}	0	7	2	11	4	7	6	8	7	5
15	{Equations,Mathematical model}	0	4	15	10	14	12	0	0	0	0
16	{Conferences,Data mining}	0	6	3	3	2	2	5	11	9	14

No	itemset	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018
17	{Data mining,Databases}	15	7	6	4	3	4	2	6	0	6
18	{Data engineering,Data mining}	47	2	0	0	0	0	0	0	2	1
19	{Conferences,Data engineering}	0	0	0	6	0	0	0	0	20	25
20	{Accuracy,Training}	0	15	10	9	11	6	0	0	0	0
21	{Data mining,Social network services}	19	2	3	1	3	2	3	12	1	5
22	{Data mining,Predictive models}	14	3	3	2	3	1	6	7	4	8
23	{Clustering algorithms, Partitioning algorithms}	16	4	4	10	2	2	4	2	3	3
24	{Computational modeling, Mathematical model}	0	3	4	7	6	2	7	6	7	8
25	{Data mining,Itemsets}	6	5	9	3	9	4	1	3	4	5
26	{Data engineering,Query processing}	45	3	0	0	0	0	0	0	1	0
27	{Predictive models,Training}	0	7	5	5	4	3	6	3	8	7
28	{Data mining,Training}	0	4	5	5	1	4	10	5	4	9
29	{Analytical models,Data models}	0	3	2	5	3	5	4	10	7	7
30	{Indexes,Query processing}	0	2	4	8	2	4	8	7	4	7
31	{Algorithm design and analysis, Optimization}	0	3	6	7	4	0	6	13	7	0
32	{Algorithm design and analysis, Heuristic algorithms}	1	3	4	5	5	1	8	8	10	0
33	{Data mining,Optimization}	0	7	2	5	3	3	6	8	6	5
34	{Data models,Optimization}	0	5	3	4	1	5	2	13	7	4
35	{Computer science,Databases}	17	10	0	1	0	0	1	5	2	8

No	itemset	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018
36	{Computer science,USA Councils}	40	1	1	0	0	0	0	0	0	0
37	{Data models,Probabilistic logic}	0	2	5	4	6	3	5	7	5	4
38	{Computational modeling, Predictive models}	1	5	1	1	4	2	5	8	9	5
39	{Optimization,Training}	0	3	4	10	3	4	1	2	7	7
40	{Data engineering,Databases}	38	0	0	0	0	0	0	0	2	1
41	{Computer science,Query processing}	17	14	1	0	0	0	0	3	2	3
42	{Data privacy,Privacy}	0	1	2	6	2	3	5	11	7	2
43	{Prediction algorithms,Predictive models}	1	4	4	2	7	3	3	6	3	6
44	{Kernel,Training}	0	7	5	8	2	6	3	2	3	3
45	{Clustering algorithms,Optimization}	0	9	8	4	3	4	4	4	2	1
46	{Training,Training data}	0	4	7	5	6	2	4	3	1	5
47	{Computational modeling, Data mining}	2	4	4	3	5	2	4	7	2	4
48	{Data mining,Vectors}	2	0	9	11	6	8	0	0	0	0
49	{Data mining,Machine learning}	22	4	2	2	0	0	0	0	1	4
50	{Algorithm design and analysis, Partitioning algorithms}	2	1	6	4	2	3	4	4	9	0
51	{Computational modeling, Social network services}	0	3	2	5	1	3	4	8	3	6
52	{Computational modeling,Training}	0	4	5	2	3	4	1	4	5	7
53	{Data models,Databases}	3	1	5	5	3	1	2	2	6	7

No	itemset	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018
54	{Feature extraction,Training}	0	5	4	4	2	2	1	3	3	10
55	{Training,Vectors}	0	0	10	9	7	8	0	0	0	0

Lampiran 10. Surat Pernyataan Data Sekunder

SURAT PERNYATAAN

Saya yang bertanda tangan di bawah ini, mahasiswa Departemen Statistika FMKSD ITS:

Nama : Reza Mustofa

NRP : 06211745000024

menyatakan bahwa data yang digunakan dalam Tugas Akhir/ Thesis ini merupakan data sekunder yang diambil dari ~~penelitian / buku/ Tugas Akhir/ Thesis/~~ publikasi lainnya yaitu:

Sumber : *IEEE Xplore Digital Library*

Keterangan : data dari website *https://ieeexplore.ieee.org*

Surat Pernyataan ini dibuat dengan sebenarnya. Apabila terdapat pemalsuan data maka saya siap menerima sanksi sesuai aturan yang berlaku.

Mengetahui
Pembimbing Tugas Akhir

(Irhamah, M.Si, Ph.D)
NIP. 19780406 200112 2 002

Surabaya, 15 Juni 2019

(Reza Mustofa)
NRP. 06211745000024

*(coret yang tidak perlu)

BIODATA PENULIS

Penulis terlahir dengan nama Reza Mustofa, biasa dipanggil Reza atau Mustofa atau Tofa. Penulis dilahirkan di Karanganyar pada tanggal 6 Agustus 1995 dan merupakan anak kedua dari pasangan Bapak MukAlim dan Ibu Muryanti. Pendidikan formal yang ditempuh penulis adalah TK Pertiwi II, SDN 02 Banjarharjo, SMPN 1 Kebakkramat, dan SMAN 1 Karanganyar. Setelah lulus dari SMA, penulis melanjutkan studi di Diploma 3 ITS Jurusan Statistika. Setelah lulus dari D3 tahun 2016, penulis sempat bekerja di salah satu perusahaan Start Up di Yogyakarta yaitu PT. Gongsin Internasional Trasindo dengan aplikasinya yang bernama JAKPAT. Tahun 2017 penulis memutuskan untuk keluar dan melanjutkan studi Lintas Jalur S1 Statistika ITS. Segala kritik, saran dan pertanyaan untuk penulis dapat dikirimkan melalui alamat email reza.mustofa2@gmail.com atau jika kurang jelas bisa juga menghubungi di No. Hp 089673867435. Terimakasih.