

Menggunakan SPADE Algorithm Untuk Sistem Rekomendasi Film

Aldy Noah¹, Rolly Intan², Alvin Nathaniel Tjondrowiguno³

Program Studi Informatika Fakultas Teknologi Industri Universitas Kristen Petra

Jl. Siwalankerto 121 – 131 Surabaya 60236

Telp. (031) – 2983455, Fax. (031) - 8417658

Email: aldynoah98@gmail.com¹, rintan@petra.ac.id², alvin.nathaniel@petra.ac.id³

ABSTRAK

Di masa kini, internet telah menjadi satu hal yang utama dalam kehidupan manusia. Hal ini menyebabkan konsumsi media hiburan dapat dinikmati di mana saja dan kapan saja. Salah satu dari media hiburan ini adalah film. Dengan adanya situs penyedia jasa menonton film secara *online*, penggemar film juga meningkat.

Karena meningkatnya penggemar film maka dibutuhkan sistem rekomendasi yang dapat merekomendasikan film. Sistem rekomendasi film merupakan suatu hal yang rumit. Hal ini disebabkan oleh banyaknya film yang ada dan banyaknya penggemar film.

Sequence pattern mining merupakan salah satu metode *data mining* yang dapat digunakan untuk mendapatkan *frequent pattern* dari sekelompok data. *Frequent pattern* merupakan serangkaian *item* yang membentuk pola yang sering muncul dalam sekelompok data. SPADE merupakan salah satu metode untuk menemukan *frequent sequence* dalam sekelompok data. Keunggulan metode SPADE adalah kecepatan dimana SPADE dapat menemukan *frequent sequence*. *Frequent sequence* yang dihasilkan dapat dijadikan sebagai data rekomendasi terhadap pengguna.

Kata Kunci: SPADE, *Frequent Sequence*, *data mining*, Sistem Rekomendasi Film

ABSTRACT

Nowadays, internet has become main thing in everyday life. This cause the consumption of entertainment media can be done anywhere and everywhere. One of the entertainments we enjoy is movie. With the increasing popularity of streaming media, movie fans also increase.

Because of the increase of movie fans, the need of recommendation system that can recommend movie to its user also rise. System recommendation for movie is a complicated thing because of considerable amount of movie and movie fans.

Sequence pattern mining is one of data mining method that can be used to gain frequent pattern from a set of data. Frequent pattern is a series of items that forms a pattern in a set of data. SPADE is one of the methods to find frequent sequence. The advantage of using SPADE is that speed in which SPADE can find frequent sequence in a data set. The benefit of using SPADE algorithm is in the speed of the algorithm to find frequent sequence. The resulting frequent sequence then can be used as a basis for recommendation to the user.

Keywords: SPADE, *Frequent Sequence*, *data mining*, *Movie Recommendation System*

1. PENDAHULUAN

Internet telah menjadi satu hal yang utama dalam kehidupan manusia pada jaman modern sekarang ini. Internet telah diintegrasikan ke dalam banyak aspek dalam aktivitas sehari-harinya. Dalam melakukan aktivitas sehari-harinya kita menggunakan alat yang memanfaatkan internet, termasuk saat mengonsumsi media hiburan. Banyak media hiburan menggunakan fitur *online* untuk mempermudah penyajian konten kepada konsumen.

Media hiburan yang dikonsumsi terutama secara digital telah mengadopsi penggunaan internet sebagai salah satu fitur utama dari produk mereka. Sebagai contoh Netflix merupakan salah satu aplikasi yang menyediakan jasa *streaming* film secara *online* menggunakan bantuan internet. Dengan adanya situs penyedia jasa *streaming* film seperti Netflix, mengonsumsi film menjadi semakin mudah. Hal ini menjadi salah satu faktor yang menyebabkan meningkatnya penggemar film.

Karena banyaknya jumlah penggemar film maka dibutuhkan sistem rekomendasi film yang dapat menyesuaikan dengan preferensi setiap orang. Hal ini menyebabkan rekomendasi film menjadi suatu masalah yang dapat diselesaikan menggunakan bantuan komputer. *Sequence pattern mining* merupakan salah satu metode *data mining* yang dapat digunakan untuk mendapatkan *frequent pattern* dari sekelompok data. *Frequent pattern* merupakan pola yang muncul dalam sekelompok data yang memiliki *sequence*. Dengan menggunakan *sequence pattern mining*, data *rating* pengguna dapat diubah menjadi sebuah data *sequence* yang kemudian dapat olah lebih lanjut.

Pengolahan data tersebut dapat dilakukan dengan metode SPADE. SPADE (*Sequential pattern discovery using equivalent class*) merupakan suatu algoritma *pattern mining* yang sering digunakan. Keuntungan menggunakan metode SPADE dibandingkan dengan metode lain ialah SPADE dapat menemukan *frequent pattern* dengan cepat.

2. TINJAUAN PUSTAKA

2.1 Sistem Rekomendasi

Sistem rekomendasi merupakan teknik dan perangkat lunak untuk memberikan anjuran kepada pengguna. Anjuran yang diberikan berhubungan atau berdasarkan dari proses pengambilan keputusan, seperti barang yang dibeli, lagu yang didengar, berita yang dibaca [5].

Sistem rekomendasi diperlukan agar kita sebagai pembuat sistem rekomendasi mendapat gambaran besar tentang produk yang di jual. Sistem dapat menunjukkan barang mana yang perlu di rekomendasikan kepada kelompok tertentu. Selain itu *user* akan merasa lebih nyaman untuk menggunakan produk kita karena apa yang dibutuhkan dapat ditemukan dengan mudah [4].

Pada pendekatan sistem rekomendasi personal, setiap *user* akan mendapatkan rekomendasi yang berbeda berdasarkan selera user. Sistem ini akan memerlukan data dari setiap *user* untuk membentuk *user model* atau *user profile*. dalam contoh rekomendasi sistem pada toko buku, sistem dapat mengingat buku apa saja yang telah dibaca oleh *user* dan memprediksi buku yang akan menarik perhatian *user* [2].

Content-based recommendation merupakan teknik rekomendasi yang bertujuan menyamakan atribut dari *user* dengan atribut benda yang direkomendasikan. Selain itu sistem rekomendasi harus dapat mencari kesamaan objek dalam melakukan rekomendasi [6].

2.2 Sequential Pattern Mining

Sequential pattern mining merupakan salah satu metode dalam *data mining*. *Sequential pattern mining* biasa digunakan untuk mengekstrak *pattern* atau pola yang unik, menarik, dan berguna dalam *database*, seperti kelompok objek yang sering muncul, asosiasi, dan *sequential rules*. Pada umumnya *Sequential pattern mining* digunakan untuk menemukan *subsequence* yang sering muncul pada *database*. *Subsequence* inilah yang disebut pola berurutan yang sering muncul. Penerapan utama metode ini ada dalam industri *retail* dimana *pattern mining* digunakan untuk memprediksi dalam kurun waktu tertentu, seorang pelanggan yang membeli buku akan membeli buku lanjutannya [7].

DATABASE			FREQUENT SEQUENCES	
SID	Time (EID)	Items		
1	10	C D	A	4
1	15	A B C	B	4
1	20	A B F	D	2
1	25	A C D F	F	4
2	15	A B F	Frequent 2-Sequences	
2	20	E	AB	3
			AF	3
			B->A	2
			BF	4
			D->A	2
			D->B	2
			D->F	2
			F->A	2
3	10	A B F	Frequent 3-Sequences	
			ABF	3
			BF->A	2
			D->BF	2
			D->B->A	2
			D->F->A	2
4	10	D G H	Frequent 4-Sequences	
4	20	B F	D->BF->A	2
4	25	A G H		

Gambar 1. *Input-sequence* Dari Database [8].

Gambar 1 merupakan contoh data yang digunakan dalam penulisan *paper*. Dari Gambar 1 digambarkan bahwa *database* D memiliki delapan *items* (A sampai H), empat *input sequence*, dan sepuluh *event*. Gambar 1 juga menunjukkan minimum *support* sebesar 50% (muncul dalam 2 dari empat *sequence*). Dalam contoh yang dikemukakan dapat dilihat bahwa contoh tersebut memiliki dia *frequent sequence* maksimal, A B F dan $D \rightarrow BF \rightarrow A$ [8].

Ketika *frequent sequences* telah diketahui, *sequence* ini dapat digunakan untuk membuat *rules* yang menghubungkan antara *items*. Sebagai contoh (B F) muncul pada empat *input sequence*, sedangkan (A B F) muncul dalam tiga *input sequence* maka bisa dikatakan bahwa jika B F muncul akan ada 75% kemungkinan A akan muncul. Dengan kata lain $(B F) \Rightarrow (B F A)$ memiliki 75% *confidence*. Dengan *minimum confidence* yang ditetapkan, kita dapat mencari semua *rules*

2.3 Sequential Pattern Discovery Using Equivalence Classes (SPADE)

SPADE merupakan algoritma *Sequential pattern mining* yang diajukan oleh Zaki. SPADE merupakan algoritma yang di buat untuk mempercepat proses dari *sequence pattern mining*. SPADE

tidak hanya meminimalkan penggunaan I/O dengan mengurangi *database scan*, namun juga meminimalkan proses komputasi dengan menggunakan cara pencarian yang efisien. Pendekatan berbasis *id-list* vertikal juga tidak terpengaruh oleh *data-skew*. [8]

Serangkaian eksperimen yang luas menunjukkan bahwa SPADE memiliki performa yang lebih baik dari teknik-teknik sebelumnya. Selain itu, SPADE *scale* secara linear dengan ukuran *database*, dan jumlah parameter lain dalam *database*.

A		B		D		F	
SID	EID	SID	EID	SID	EID	SID	EID
1	15	1	15	1	10	1	20
1	20	1	20	1	25	1	25
1	25	2	15	4	10	2	15
2	15	3	10			3	10
3	10	4	20			4	20
4	25						

Gambar 2. Contoh *Id-list* [8].

Seperti pada Gambar 1, format *database* horizontal terdiri dari sekelompok *input sequence*. Setiap *input sequence* memiliki kelompok *event*. Namun pada algoritma SPADE, format *database* yang digunakan adalah format data vertikal seperti pada Gambar 2 Setiap entri dari setiap *item* merupakan pasangan dari (*sid*,*eid*).

SPADE (min_sup, D):

$\mathcal{F}_1 = \{ \text{frequent items or 1-sequences} \};$

$\mathcal{F}_2 = \{ \text{frequent 2-sequences} \};$

$\mathcal{E} = \{ \text{equivalence classes } [X]_{\theta_1} \};$

for all $[X] \in \mathcal{E}$ do Enumerate-Frequent-Seq($[X]$);

Gambar 3. *Input-sequence* Dari Database [8].

Pada Gambar 3, dituliskan *pseudo-code* pada algoritma SPADE. Pada penggambaran tersebut merupakan fungsi yang dipanggil untuk menjalankan algoritma SPADE. Terdapat tiga tahap utama dalam pembuatan pola menggunakan algoritma SPADE. Tahap pertama dilakukan untuk mengubah data sekaligus menemukan *frequent items* yang berisi 1 *element*. Tahap kedua dilakukan pengubahan format data untuk mempermudah proses pada tahap ketiga dan sekaligus menemukan *frequent items* yang berisi 2 *element*. Tahap ketiga dilakukan secara rekursif untuk menemukan pola dengan n *element* dengan batasan minimum *support* yang diberikan pada algoritma.

D		D->B		D->BF		D->BF->A							
SID	EID(D)	SID	EID(D)	EID(B)	SID	EID(D)	EID(B)	EID(F)	SID	EID(D)	EID(B)	EID(F)	EID(A)
1	10	1	10	15	1	10	20	20	1	10	20	20	25
1	25	1	10	20	4	10	20	20	4	10	20	20	25
4	10	4	10	20									

Gambar 4. Contoh Pembentukan Satu *Sequence* [8].

Pada Gambar 4 digambarkan relasi antara data yang ada dengan pembuatan pola pada algoritma SPADE. *Item* yang dipilih akan masuk ke dalam *element* pertama. *Item* yang dipilih harus memiliki *eid* (*event id*) yang terkecil. Setelah itu *item* yang selanjutnya dipilih merupakan *item* yang memiliki *eid* lebih besar daripada *item* dalam *element* sebelumnya. Jika terdapat dua *item* dengan *eid* yang sama

makan kedua *item* tersebut akan masuk ke dalam *element* yang sama. proses penambahan *element* ini akan dilakukan sampai jumlah maksimal dari *element* yang dapat dihasilkan.

proses komputasi pada F1 algoritma akan membaca data dengan format data vertikal. Untuk setiap data akan di baca dari *disk* ke *memory* dan *support* akan ditambah secara inkremental setiap *sid* baru. Data yang dihasilkan akan berupa data seperti pada 0. Pada komputasi F2 akan dilakukan perubahan *vertical-to-horizontal* data. Hal ini akan menghasilkan *recovered horizontal database* dimana *2-sequences* bisa didapat dengan *join* [8].

2.4 Tinjauan Studi

Dari penelitian-penelitian sebelumnya terdapat penelitian rekomendasi makanan menggunakan algoritma SPADE. Dalam jurnal tersebut hasil yang didapat oleh penulis cukup akurat dalam memprediksi makanan apa yang akan dibeli oleh pelanggan. Rekomendasi didapat dari menganalisis sejarah pembelian *user* menggunakan *sequential pattern mining*. Dalam kasus tersebut penulis mengatakan bahwa jumlah *sequence* yang ditemukan akan menurun bila parameter *support* dinaikkan. Namun pada jurnal ini penulis merekomendasikan kategori dari kemungkinan pembelian *user* selanjutnya [3].

Dengan menggunakan data sejarah rating *user* terhadap suatu film kita dapat membuat suatu prediksi jika ada *user* baru yang melakukan *rating* atau menonton pola film yang sama. Data ini didapat dari *movielens* [1], dimana di dalam data tersebut terdapat urutan sejarah film apa saja yang telah dirating oleh *user* tersebut.

SPADE merupakan *sequential pattern mining* yang dibuat oleh Zaki. Pada jurnal yang telah dibuat oleh zaki, SPADE mampu mencari *frequent pattern* dengan cepat dengan menggunakan sedikit *data read* [8].

3. ANALISA DAN DESAIN

3.1 Analisa

Algoritma SPADE merupakan algoritma *sequence pattern mining* yang menerima *input* berupa pola urutan *item* yang disajikan. Rekomendasi dapat dilakukan dengan menggunakan SPADE untuk mengolah data *user* yang telah diperoleh melalui urutan *rating* film. Data urutan *rating* film diperoleh dari *movielens dataset* dapat diolah menjadi data pola urutan film. Kumpulan pola ini kemudian dapat diolah lebih lanjut menggunakan *sequence pattern mining* untuk mendapatkan pola yang sering muncul dalam *rating* tersebut. Pola yang sering muncul dalam data *rating* ini dapat dijadikan data dasar untuk rekomendasi. Tabel 1 merupakan contoh *input* data dari *movielens dataset*.

Tabel 1. Contoh Data Rating *movielens*.

	UserId	movieId	rating	timestamp
0	1	1	4.0	964982703
1	1	3	4.0	964981247
2	1	6	4.0	964982224
3	1	47	5.0	964983815
4	1	50	5.0	964982931

3.2 Desain Sistem

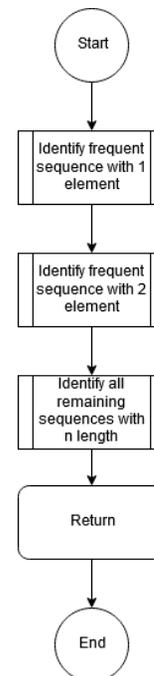
Sistem didesain dengan empat tahap yaitu *preprocessing*, SPADE, *result postprocessing*, sistem rekomendasi.

3.2.1 Preprocessing

Pada tahap *preprocessing* data yang diterima akan mengalami beberapa konversi. Data yang berada pada kolom *timestamp* merupakan data jumlah detik yang dihitung sejak tanggal 1 januari tahun 1970 *Coordinated Universal Time* (UTC). Data tersebut harus di ubah menjadi *eventId* yang dapat dioleh oleh algoritma SPADE. Data akan dikelompokkan menggunakan bahasa R. Data akan dikelompokkan berdasarkan *userId* dan *eventId*. setelah dikelompokkan data kemudian akan di *summarize*. Semua *movieId* akan dikelompokkan ke dalam satu kolom. Dataset juga diberi tambahan kolom *size* dimana *size* merupakan jumlah dari *movieId* yang terdapat pada satu *sequence*. Dataset baru akan di simpan dalam bentuk *text file* untuk dioleh lebih lanjut.

3.2.2 SPADE

Data yang sudah melalui tahap *preprocessing* akan diproses lebih lanjut untuk menemukan *frequent pattern*. Pengolahan ini akan dilakukan dengan metode SPADE. Algoritma ini, seperti pada Gambar 5, terdiri dari tiga bagian. Bagian – bagian itu antara lain menemukan *frequent sequence* dengan panjang 1 elemen, menemukan *frequent sequence* dengan panjang 2 elemen, menghitung semua *frequent sequence*. Tahap pertama dari algoritma SPADE yaitu menemukan semua *frequent sequences* yang terdiri dari satu elemen. *support* minimum harus terlebih dahulu ditentukan. *Support* minimum ini digunakan untuk menentukan apakah suatu *sequence* merupakan *frequent sequence*.



Gambar 5. Flowchart Garis Besar Algoritma SPADE.

Dalam format data pada Gambar 1 setiap *item* dikelompokkan menjadi suatu *id-list* seperti pada Gambar 2. Setiap *id-list* ini yang akan menjadi sebuah *element* dalam sebuah *sequence*. Dalam setiap proses pengubahan *element*, akan dihitung *support* dengan cara menghitung panjang dari *sid* (*sequence id*) dari setiap *elemen*.

3.2.3 Result Postprocessing

Hasil dari algoritma SPADE akan berupa sekelompok *sequence* yang memiliki *support*. Untuk dapat dijadikan data sebagai dasar dari sistem rekomendasi maka data *sequence* harus di proses

terlebih dahulu. Data akan dibersihkan menjadi tabel yang dapat dibaca oleh *sql*. Pada tahap ini dilakukan pembersihan dari data *sequence* yang di dapat dari algoritma SPADE. Proses pertama dari pembersihan ini untuk membersihkan *sequence* dan mengambil *sequence movieId* yang ditemukan. Setelah dibersihkan data *sequence* akan dipisah ke dalam kolom yang berbeda. Hal ini agar mempermudah data yang didapat untuk dimasukkan ke dalam tabel *sql*. Setelah itu kolom *support* akan dihapus. Kemudian hasil akhir dari tabel yang ada akan di simpan.

3.2.4 Recommendation System

Sistem rekomendasi dibuat menggunakan web. Hasil dari *postprocessing* akan di ubah menjadi tabel pada *sql*. Halaman rekomendasi akan ditampilkan saat *user* telah memilih salah satu film yang ada. Data rekomendasi didasarkan atas *post processing*. Halaman web akan mencari data dan menampilkan elemen 1 dari *sequence* yang ada. Rekomendasi akan ditampilkan ketika *user* telah memilih salah satu film yang tertera pada halaman awal web.

4. HASIL DAN PEMBAHASAN

4.1 Pengujian Standar Sistem

4.1.1 Input Program

Input dari program ini adalah data *ratings* yang didapat dari *movielens*. Data yang akan digunakan sebagai *input* dari tahap *preprocessing* memiliki empat kolom yaitu *userId*, *movieId*, *rating*, *timestamp*.

	userId	movieId	rating	timestamp
0	1	307	3.5	1256677221
1	1	481	3.5	1256677456
2	1	1091	1.5	1256677471
3	1	1257	4.5	1256677460
4	1	1449	4.5	1256677264

Gambar 6. Data Input *Preprocessing*.

Pada Gambar 6 ditunjukkan format *input* data pada tahap *preprocessing*. Kolom data *timestamp* berupa detik yang dihitung sejak tengah malam pada tanggal 1 januari 1970. Kolom data *rating* memiliki jarak dari 0.5 sampai 5. Data yang digunakan berjumlah 27753440 *ratings*. Data yang di *inputkan* berupa *file csv*.

4.1.2 Preprocessing

Pada proses *preprocessing*, terdapat dua tahap yaitu *filtering* dan *formatting*. Saat dilakukan *filtering* data *input* difilter sesuai dengan kriteria yang ditetapkan, data yang digunakan adalah data yang memiliki *rating* lebih dari tiga. Setelah itu kolom *rating* dihilangkan. Kolom *timestamp* pada data *input* diubah menjadi kolom *eventId*. kolom *rating* sudah dihilangkan. Pada hasil *filtering*, data *rating* yang tersisa adalah 22759420 *ratings* dan dikonversi ke *csv*. Setelah proses *filtering* data akan kemudian melalui tahap *formatting*. . Setiap baris dari data yang memiliki *eventId* dan *userId* yang sama dengan baris lain akan digabung menjadi satu baris.

	sequenceID	eventID	SIZE	items
1	1	1256677221	1	307
2	1	1256677239	1	2478
3	1	1256677243	1	3698
4	1	1256677260	1	3020
5	1	1256677264	1	1449

Gambar 7. Contoh Hasil Output Dari *Preprocessing*.

Pada Gambar 7 ditunjukkan data yang telah melalui tahap *formatting*. Kolom *movieId* yang telah digabung menjadi kolom *items*. Kolom *size* berisi jumlah *movieId* yang terdapat pada kolom *items*. Data yang dihasilkan kemudian akan disimpan dalam format *text*.

4.1.3 SPADE

Setelah melalui *preprocessing*, data akan kemudian dibaca dan diolah menggunakan algoritma SPADE. Parameter penentu hasil pada *library* SPADE yang digunakan adalah *support*, *maxlen*, dan *maxsize*. Sebagai standar hasil percobaan, variabel yang ditetapkan adalah *support* 0.3, *maxlen* 3, dan *maxsize* 3.

```

set of 3 sequences with
most frequent items:
  296      318      356 (Other)
   1         1         1         1
most frequent elements:
 {296}   {318}   {356} (Other)
   1     1     1     1
element (sequence) size distribution:
sizes
1
3
sequence length distribution:
lengths
1
3

```

Gambar 8. Ringkasan Dari Hasil Pengolahan Data Menggunakan SPADE.

Pada Gambar 8 ditunjukkan bahwa dengan standar parameter yang digunakan data yang dihasilkan oleh algoritma SPADE. Data yang dihasilkan hanya memiliki 3 set *sequence*. Panjang *sequence* yang dihasilkan hanya memiliki satu *element* dan dalam satu elemen hanya terdapat satu *item*.

4.2 Pengujian Parameter

4.2.1 Penyetelan Parameter Maxsize

Parameter *maxsize* berpengaruh pada jumlah *item* dalam satu *element*. Dalam suatu *sequence* terdapat beberapa *element*. Di dalam sebuah *element* terdapat beberapa *items*. Isi standar percobaan pertama parameter *maxsize* adalah 3. Hal ini memungkinkan algoritma SPADE untuk menghasilkan *sequence* yang memiliki *element* yang berisi 1 sampai 3 *item*.

Tabel 2. Hasil Percobaan Pada Parameter Maxsize.

maxsize	Jumlah item per element	Jumlah yang ditemukan	Jumlah sequence yang ditemukan dengan panjang 2 atau lebih
1	1	63	0
2	1	63	0
	2	0	
3	1	63	0
	2	0	
	3	0	
4	1	63	0
	2	0	
	3	0	
	4	0	

Pada Tabel 2 Dapat dilihat bahwa pengisian parameter menghasilkan data yang sama. Oleh karena itu parameter *maxsize* diisi dengan 2 sebagai hasil dari penyetelan Parameter yang optimal.

4.2.2 Penyetelan Parameter Support

Dalam proses mengolah data, algoritma SPADE menggunakan *support* untuk menentukan apakah sebuah *sequence* termasuk ke dalam sebuah *frequent sequence*. Dalam pengujian standar program pertama *support* yang digunakan adalah 0.3. Jumlah pengurangan yang akan dilakukan adalah 0.1, *support* yang akan dicobakan adalah 0.2 dan 0.1.

```

set of 235 sequences with
most frequent items:
 4993  4995 108932  50872  51255 (other)
   1    1    1        1    1    230
most frequent elements:
{101864} {102125} {102445} {102903} {103042} (other)
   1        1        1        1        1    230
element (sequence) size distribution:
sizes
 1
235
sequence length distribution:
lengths
 1
235
    
```

Gambar 9. Hasil Pengolahan Data Dengan Support 0.2.

Pada Gambar 9 ditunjukkan bahwa hasil pemrosesan data dengan menggunakan *support* 0.2 belum menemukan *sequence* dengan panjang 2 *element*. Hal ini dikarenakan data yang digunakan sangat banyak dan *support* 0.2 belum cukup rendah.

Pada Gambar 10 ditunjukkan bahwa hasil pengolahan data dengan menggunakan *support* 0.1 memunculkan hasil yang berjumlah 2357 *sequence* dengan panjang 2 *element*. *Sequence* yang dihasilkan pada *support* 0.1 cukup banyak namun *support* 0.1 terlihat sangat rendah oleh karena itu *support* akan dinaikkan menjadi 1.25 dengan hasil 390 *sequence* yang memiliki panjang 2 atau lebih.

```

set of 3043 sequences with
most frequent items:
111759  99114 104841 122882 106782 (other)
 151    116    114    108    103    4808
most frequent elements:
{111759} {99114} {104841} {122882} {106782} (other)
 151    115    114    108    103    4777
element (sequence) size distribution:
sizes
 1  2
718 2325
sequence length distribution:
lengths
 1  2
686 2357
    
```

Gambar 10. Hasil pengolahan data dengan support 0.1.

4.2.3 Penyetelan Parameter Maxlen

Parameter *Maxlen* berpengaruh pada jumlah *element* dalam sebuah *sequence*.

Tabel 3. Hasil Percobaan Pada Parameter Maxlen

Isi parameter	Sequence dengan panjang 1	Sequence dengan panjang 2
2	512	390
3	512	390
4	512	390
5	512	390

Pada Tabel 3 dituliskan hasil percobaan parameter *maxlen*. Dari hasil percobaan tersebut semua parameter menghasilkan data yang sama. Oleh karena itu parameter *maxlen* akan diisi dengan 2.

4.2.4 Menetapkan Filter Minimum Size

Dikarenakan rekomendasi data yang dihasilkan terlalu sedikit filter *minimum size* ditetapkan. Filter ini berfungsi untuk membatasi hanya *sequence* data yang memiliki *size* lebih dari yang ditetapkan yang dapat diolah.

Tabel 4. Hasil Percobaan Pada Parameter Minimum Size.

Isi parameter	Sequence dengan panjang 1	Sequence dengan panjang 2
10	512	390
20	518	477
30	530	400
40	483	79

Pada Tabel 4 dituliskan hasil percobaan dengan pengisian 10 sampai 40. Pada pengisian *minimum size* 20, data *sequence* dengan panjang 2 atau lebih yang dihasilkan lebih tinggi dari hasil percobaan lain. Oleh karena itu angka 20 akan ditetapkan sebagai pengisian optimal dari parameter *minimum size*.

4.3 Pengujian Filter Data

Setelah mendapatkan pengaturan yang optimal dari masing-masing parameter pengujian terhadap filtrasi *rating* dari data *input* dilakukan. Pada standar pengujian program filter *rating* yang

digunakan adalah lebih dari 3. Data yang dihasilkan dari proses standar *filtering* ini berjumlah 7984265 *ratings*.

Pada percobaan dengan filter rating 3 dihasilkan *frequent sequence* yang memiliki panjang 2 atau lebih sebanyak 477 buah. Pada percobaan dengan filter rating 3.5 dihasilkan *frequent sequence* yang memiliki panjang 2 atau lebih sebanyak 60 buah. Pada percobaan dengan filter rating 4 dihasilkan *frequent sequence* yang memiliki panjang 2 atau lebih sebanyak 7 buah. Oleh karena itu *filter rating* ditetapkan pada 3.

4.4 Analisa Hasil Parameter Optimal

Setelah pengujian parameter dilakukan parameter yang optimal untuk menemukan data rekomendasi dapat ditentukan. Filter *rating* data yang digunakan adalah lebih dari 3. Untuk pengisian parameter pada algoritma SPADE, *supprot* akan diisi dengan 0.125, *maxlen* diisi dengan 2, dan *maxsize* diisi dengan 2.

	rule	support	confidence
1	<{4993}> => <{5952}>	0.105479	0.527727
2	<{296}> => <{593}>	0.123182	0.403457
3	<{318}> => <{593}>	0.117489	0.347730
4	<{296}> => <{589}>	0.101288	0.331748
5	<{318}> => <{589}>	0.102485	0.303321

Gambar 11. Contoh Hasil Analisa Rules.

Confidence merupakan sebuah nilai yang berjarak antara 0 dan 1. Nilai *Confidence* menggambarkan peluang dinamakan *element* dua akan muncul jika *element* pertama sudah dipilih. Sebagai contoh pada Gambar 11 baris pertama jika *user* telah memilih *movieId* 4993 maka terdapat 52.77% *user* akan memilih *movieId* 5952.

	support	confidence
count	458.000000	458.000000
mean	0.137280	0.390848
std	0.010114	0.048629
min	0.125704	0.300448
25%	0.129456	0.355670
50%	0.135084	0.387097
75%	0.142589	0.418848
max	0.180113	0.563025

Gambar 12. Ringkasan Hasil Analisa.

Pada Gambar 12 digambarkan ringkasan dari hasil analisa *rules*. Dalam ringkasan ini dapat dilihat bahwa rata – rata atau mean dari *confidence* yang diperoleh adalah 0.3908. Hal ini berarti *rules* yang dihasilkan dari algoritma SPADE memiliki rata-rata keyakinan untuk memprediksi *item* selanjutnya adalah 39.08%.

4.5 Menampilkan Rekomendasi

Frequent sequence yang dihasilkan oleh algoritma SPADE akan kemudian dijadikan dasar untuk rekomendasi. Tahap *postprocessing* data yang dihasilkan dari tahap pengolahan menguakkan algoritma SPADE dibersihkan untuk perubahan ke format *sql*. Data *input* dari *proceesing* berupa tabel yang berisi *sequence* dan *support*. Data *sequence* memiliki karakter seperti <> dan {} yang tidak dibutuhkan. Karena itu data *sequence* harus dibersihkan dan dipisah menjadi dua kolom yang berbeda. Kolom *support* juga dihilangkan.

Pada Gambar 13 digambarkan hasil dari tahap *postprocessing*. Hasil *output* dari tahap ini berupakah tabel dengan dua kolom. Kedua kolom berisi *movieId*. Tabel akan kemudian dikonversi ke dalam *sql*. Berdasarkan data yang telah diubah menjadi *sql* data kemudian dimasukkan ke dalam *database*. Data akan kemudian menjadi dasar dari sistem rekomendasi yang digunakan oleh *user*. Pada halaman utama *list* film dari halaman utama ini didapat dari *element* pertama dari hasil rekomendasi. Halaman kedua akan berisi rekomendasi dari film yang dipilih.

	pertama	kedua
100	4993	5952
101	296	593
102	318	593
103	296	589
104	318	589

Gambar 13. Hasil Data Output Dari Tahap *Postprocessing*.

4.6 Hasil Kuesioner

Kuesioner ini diberikan pada 22 orang yang menyatakan suka menonton film. Rata –rata jenis kelamin yang disurvei adalah laki –laki selain itu mereka juga berumur sekitar 18 – 30 tahun.

- Kelengkapan film : 2.59/5
- Variasi rekomendasi yang ditampilkan : 2.636 /5
- Keinginan untuk menonton film : 3/5
- Kepuasan pengguna : 2.545/5

4.7 Analisis Hasil Pengujian

Hasil akhir dari pengujian parameter dapat menghasilkan jumlah *sequence* yang cukup banyak, namun *sequence* yang dihasilkan belum memiliki panjang yang diinginkan. Parameter *support* berperan penting pada pengolahan data. Selain mempengaruhi jumlah data yang dihasilkan, *support* juga mempengaruhi kualitas *sequence* yang dihasilkan. Parameter *minimum size* berfungsi untuk menurunkan *noise* dari data input. Pada data input terdapat *sequence rating* yang memiliki panjang element terlalu pendek. Dengan menghilangkan *sequence rating* pendek, rata – rata *support* akan mengingkat. Namun pengisian parameter *minimum size* terlalu tinggi akan menyebabkan data menjadi basis terhadap orang yang *merating* banyak film. Hasil kuesioner menyatakan rekomendasi film kurang memuaskan. Variasi film yang ditampilkan kurang memuaskan. Pengguna mengharapkan lebih banyak variasi terhadap film yang ditawarkan.

5. KESIMPULAN

Dari hasil pengujian aplikasi, dapat diambil beberapa kesimpulan antara lain:

- Berdasarkan analisa hasil dari percobaan parameter, *support* merupakan parameter yang paling berpengaruh terhadap hasil pengolahan.
- Berdasarkan analisa hasil dari sistem data yang dihasilkan sebagai hasil pengolahan data dengan menggunakan SPADE sebagai dasar rekomendasi, model memiliki 39.08% *confidence* dalam merekomendasikan film.
- Berdasarkan analisa hasil dari kuesioner yang dilaksanakan kelengkapan film dan variasi rekomendasi film kurang memenuhi kepuasan pengguna dengan nilai rata – rata 2.59 dari 5 untuk kelengkapan film dan 2.636 dari 5 untuk variasi rekomendasi film.
- Berdasarkan analisa hasil kuesioner yang dilaksanakan keinginan pengguna untuk menonton film yang direkomendasikan sudah mencukupi dengan nilai rata – rata 3 dari 5.
- Berdasarkan analisa hasil, SPADE tidak cocok untuk digunakan dalam kasus *dataset* rating film.

Saran yang diberikan untuk penyempurnaan dan pengembangan lebih lanjut untuk aplikasi ini adalah sebagai berikut:

- Sistem dibuat lebih linear, sehingga program dapat dijalankan secara otomatis.
- Klasifikasi data dapat dibuat sebagai pengganti filter rating agar meningkatkan hasil analisa data.
- Menambahkan *now trending movie* pada sistem rekomendasi bagi user yang belum pernah melihat film.
- Mengurangi variasi film pada data *input*.

- Membuat *user interface* yang lebih mudah digunakan.
- SPADE lebih cocok digunakan untuk dataset yang memiliki variasi lebih sedikit dan sequence yang panjang.

6. DAFTAR REFERENSI

- [1] Harper, F. M., & Konstan, J. A. 2015. *The movielens datasets: ACM transactions on interactive intelligent systems*. New York: Association for Computing Machinery.
- [2] Jannach, D., Zanker, M., Felfernig, A., & Friedrich, G. 2010. *Recommender systems: An introduction*. Cambridge: Cambridge University Press.
- [3] Khandagale, S., Mallade, S., Kharat, K., & Bansode, V. 2016. *Food recommendation system using Sequential Pattern Mining*. Pune, India: Cummins College of Engineering for Women.
- [4] Ricci, F., Rokach, L., & Shapira, B. 2015. *Recommender systems: Introduction and challenges. Recommender systems handbook*. New York: Springer.
- [5] Ricci, F., Rokach, L., & Shapira, B. 2011. *Introduction to recommender systems handbook. Recommender systems handbook*. Boston, MA: Springer.
- [1] Sand, S. 2017. *Video recommendation systems: Finding a suitable recommendation approach for an application without sufficient data*. Oslo: University of Oslo.
- [7] Wright, AP., Wright, AT., McCoy, AB., Sitting, DF. 2015. *The use of Sequential Pattern Mining to predict next prescribed medication*. San Diego, USA: Elsevier Science.
- [8] Zaki, MJ. 2001. *SPADE: An efficient algorithm for mining frequent sequences*. Norwell, MA, USA: Kluwer Academic Publishers.