LAPORAN PROYEK

Netflix-Movie Recommendation System Using Apriori Algorithm

Disusun oleh:

1. 12S17004 – Fivin Sadesla Tambunan
2. 12S17026 – Mika Lestari Valentina Manurung
3. 12S17037 – Nita Sophia Winandi Sirait

PROGRAM STUDI SARJANA SISTEM INFORMASI

FAKULTAS INFORMATIKA DAN TEKNIK ELEKTRO

INSTITUT TEKNOLOGI DEL

2020

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB 1
BUSINESS UNDERSTANDING

Tahap pertama pada *framework* CRISP-DM (*Cross Industry Standard Process for Data Mining*) adalah *business understanding* yang diartikan dengan pemahaman proyek. Pada tahap pertama ini diperlukan perspektif bisnis, esensi, dan persyaratan dari kegiatan *data mining* serta kebutuhannya [1]. Pemahaman bisnis mengacu pada sistem rekomendasi film. Pada tahap *business understanding* ini dibutuhkan pemahaman mengenai latar belakang pemilihan proyek dan juga mengenai tujuan pada sistem rekomendasi.

1.1 *Determine Business Objectives*

Kegiatan yang dilakukan pada tahap ini adalah merancang tujuan dari bisnis dan menentukan faktor-faktor utama yang berperan dalam proyek yang dirancang dan untuk memeriksa bahwa proyek telah menghasilkan hasil akhir yang tepat pada saat dilakukan evaluasi. Tujuan dari bisnis tidak sama dengan tujuan langsung dari proyek, tujuan bisnis memiliki artian jangka panjang pada kehidupan nyata. Hasil dari *data mining* menggambarkan pengetahuan yang digunakan sebagai parameter (standar) organisasi terkait saat pembuatan kebijakan pada organisasi tersebut.

Semakin berkembangnya teknologi maka akan semakin banyaknya informasi yang tersedia. Informasi ini dapat diakses melalui pemanfaatan dari teknologi multifungsi yang dikaji agar lebih efisien dan optimal melalui internet. Semakin berkembangnya penggunaan teknologi informasi maka setiap informasi yang diperlukan pengguna telah terdapat di internet dengan bermacam varian. Pada era sekarang, banyak masyarakat yang membutuhkan hiburan untuk mengurangi kepenatan dari rutinitas sehari-hari, hiburan tersebut dapat diperoleh melalui film. Menurut data *British Film Institute* (BFI), dari tahun 2009 hingga 2015, jumlah film *box office* yang dirilis terus bertambah setiap tahunnya (*British Film Institute*, 2016). Pada tahun 2009, 503 film yang dirilis dan tahun 2015 jumlah film yang dirilis adalah 759 film (*British Film Institute*, 2016). Berdasarkan hal tersebut maka dapat disimpulkan bahwa film sudah beredar banyak di internet. Oleh sebab itu, diperlukan suatu sistem yang dapat menyediakan informasi berdasarkan kebutuhan pengguna. Sistem ini biasa disebut sebagai sistem rekomendasi.

Tujuan bisnis pada sistem rekomendasi film yaitu membantu pengguna mencari pilihan film mereka di antara semua jenis film yang berbeda dan juga pengguna tidak perlu menghabiskan

banyak waktu untuk mencari film favorit mereka [2]. Faktor yang mempengaruhi pembuatan sistem rekomendasi film adalah informasi yang diperoleh dari *user* tersebut. *Item* tersebut atau atau film ini dapat disarankan berdasarkan *rating* film yang telah diberikan oleh user atau berdasarkan pengguna lain yang mempunyai kebiasaan yang mirip.

1.2 *Access the Situation*

Tugas ini melibatkan pencarian fakta yang lebih rinci tentang semua sumber daya (*sources*) seperti *Hardware sources* (sumber daya perangkat keras), *data sources* (sumber daya data), dan *personel sources* (sumber daya personal).

1. *Data sources* yang digunakan pada proyek ini adalah *dataset* netflix yang dapat dilihat pada url berikut https://www.kaggle.com/netflix-inc/netflix-prize-data. *Dataset* ini belum pernah digunakan untuk sistem rekomendasi film menggunakan algoritma Apriori.
2. *Hardware sources* yang digunakan pada proyek ini adalah laptop IdeaPad Lenovo 4 GB RAM, Processor Intel Core i5-7200U, Dual Core 2,5 GHZ TurboBoost 3,1 GHZ CD/DVD ROM Drive
3. *Personel sources* pada proyek ini terdiri dari 3 orang mahasiswa yang berperan pada pengerjaan sistem rekomendasi film menggunakan *content based* mulai dari tahap *business understanding*, *data understanding*, *data preparation*, *modeling*, *evaluation,* sampai tahap akhir *deployment.*

1.3 *Determine Data Mining Goals*

Merupakan suatu tahapan yang mampu untuk mengubah pengetahuan pada domain bisnis menjadi sebuah definisi *problem data mining* serta menetapkan tujuan *data mining* (proyek). Tujuan *data mining* pada pengerjaan proyek ini adalah menemukan pengetahuan (*discovering knowledge*) mengenai pola (*pattern*) *item* berdasarkan riwayat pencarian pengguna sehingga dapat diketahui dan diprediksi *item* (film) yang mungkin diminati pengguna.

1.4 *Produce Project Plan*

Tahapan yang dilakukan disini adalah memaparkan rancangan kerja yang ditujukan untuk mencapai tujuan dari data mining sehingga mampu untuk mencapai tujuan bisnis, kemudian menentukan teknik dan t*ools* yang selanjutnya akan dipergunakan.

Project plan :

Perencanaan proyek agar dapat menyelesaikan tujuan data mining serta mencapai tujuan bisnis adalah sebagai berikut :

Tabel 1.1. Perencanaan Proyek

Tahap	Waktu	Kegiatan
Business Understanding	5 hari	Menetapkan tujuan bisnis, menilai situasi, menetapkan sasaran *data mining*, dan menyusun beberapa rencana proyek.
Data Understanding	1 minggu	Mengumpulkan data, memahami data, melakukan eksplorasi data, dan mengidentifikasi kualitas data.
Data Preparation	2 minggu	Memilih *dataset* akhir dari data mentah, melakukan *cleaning data*, mengimplementasikan data, melakukan integrasi data, serta membuat format data
Modeling	4 minggu	Memilih metode pemodelan dan sejumlah parameter yang disesuaikan untuk memperoleh nilai yang optimum.
Evaluation	2 minggu	Melakukan evaluasi hasil, *review* proses, dan menentukan langkah selanjutnya.
Deployment	3 minggu	Menyusun rencana penerapan, menyusun rencana pemantauan (*monitoring*) dan pemelihataan (*maintenance*), membuat laporan akhir

Teknik :

Teknik atau metode yang digunakan pada proyek ini adalah menggunakan algoritma Apriori. Algoritma Apriori ini termasuk dalam jenis metode asosiasi pada penambangan data (*data mining*) untuk penambangan *frequent itemset*. Algoritma ini menggunakan dua langkah "*join*" dan "*prune*" untuk mengurangi ruang pencarian. Algoritma ini merupakan pendekatan berulang untuk menemukan *frequent itemset* [3].

Probabilitas *item* I tidak *frequent* adalah jika:

1. P (I) < *minimum support threshold* , maka I tidak *frequent*
2. P (I + A) < *minimum support threshold*, maka I + A tidak *frequent*, di mana A juga termasuk dalam *itemset*.
3. Jika *itemset* memiliki nilai kurang dari *minimum support* maka semua *superset*-nya juga akan berada di bawah *minimum support*, dan karenanya dapat diabaikan. Properti ini disebut properti Antimonotone.

Langkah-langkah Algoritma Apriori pada *data mining* adalah:

1. *Join Step* :

   Langkah ini menghasilkan (K + 1) itemset dari K-*itemsets* dengan menggabungkan setiap item dengan *item* itu sendiri.
2. *Prune Step* :

   Langkah ini memindai hitungan setiap *item* dalam *database*. Jika kandidat *item* tidak memenuhi *minimum support*, maka *item* tersebut dianggap *infrequent* dan oleh karena itu dihapus. Langkah ini dilakukan untuk mengurangi ukuran dari kandidat *itemset.*

Tools :

Tools yang digunakan untuk proyek ini adalah python

BAB 2
DATA UNDERSTANDING

Data Understanding diartikan sebagai pemahaman data merupakan tahap pengumpulan data dan meneliti data untuk dapat mengidentifikasi dan memahami apa yang dapat dilakukan dengan data tersebut Pemahaman data merujuk pada *dataset* Netflix. Hal lain yang dilakukan pada tahap ini adalah mempelajari tentang format data (form dan laporan) dan tahapan yang lebih dalam (bentuk fisik data).

2.1 *Collect Initial Data*

Dataset yang digunakan pada proyek berasal dari Kaggle https://www.kaggle.com/netflix-inc/netflix-prize-data . *Dataset* yang akan digunakan akan di-*download* secara manual dan disimpan dalam sebuah folder. Data tersebut terdiri atas 7 file dengan format csv dan txt yaitu *combined data_1*.txt, *combined_data*_2.txt, *combined_data*_3.txt, *combined_data*_4.txt, *movie*_titles.csv, qualifying.txt, dan probe.txt. Data yang dikumpulkan berisi informasi dari film netflix, yan terdiri atas *Cust_Id, Rating, Movie_Id, Movie_title , Year_release, date.* Berikut adalah penjelasan mengenai deskripsi data yang akan dikumpulkan:

1. *Cust_Id* : berisi ID user yang berlangganan netflix
2. *Rating :* berisi *rating* terhadap *movie* yang diberikan oleh *user*
3. *Movie_Id* : berisi ID movie yang bersifat unik
4. *Movie_ttitle :* berisi judul movie
5. *Year_release* : berisi tahun *movie* dirilis
6. *date* : berisi tanggal pemberian *rating* terhadap *movie* yang diberikan oleh *user*

2.2 *Describe Data*

Tugas ini melibatkan deskripsi terhadap isi data. Data berasal dari lebih dari satu *file* dengan ekstensi .txt selanjutnya akan dilakukan proses integrasi data atau penggabungan data. Data tersedia dalam 7 file yaitu *combined_data*_1.txt, *combined_data*_2.txt, *combined_data*_3.txt, *combined_data*_4.txt, *movie*_titles.txt, *probe*.txt, dan *qualifying*.txt. Data berisi film yang rilis dari tahun 1890 sampai 2005. Pengguna dipilih secara acak dan setidaknya memberikan *rating* untuk 1 *movie*. Setiap pengguna diwakili oleh sebuah id (*CustomerId*) dan *movie* diwakili oleh sebuah id (*movieId*). Adapun deskripsi data yang akan digunakan dapat dilihat pada tabel berikut:

Tabel 2.1. Combined_data.txt 1 sampai 4

Variable Name	*Variable Description*
Cust_Id	ID user yang berlangganan netflix
Rating	*Rating* terhadap *movie* yang diberikan oleh *user*

Tabel 2.2. Movie_titles.csv

Variable Name	*Variable Description*
Movie_Id	ID movie yang bersifat unik
Year_release	*Rating* terhadap *movie* yang diberikan oleh *user*
Movie_title	Judul *movie*

Probe.txt adalah file yang berisi kumpulan data yang disediakan oleh Netflix untuk digunakan untuk data *test* sebagai kumpulan prediksi yang memenuhi syarat. Adapun isi dari file probe.txt pada Tabel berikut :

Tabel 2.3. Probe.txt

Variable Name	*Variable Description*
Movie_Id	ID *movie* yang bersifat unik
Cust_Id	ID *user* yang berlangganan netflix

Qualifying.txt adalah file yang berisi bentuk *Dataset* yang memenuhi syarat yang disediakan oleh Netflix. Pada file teks tidak terdapat baris yang kosong. Adapun isi dari file qualifying.txt pada Tabel berikut :

Tabel 2.4. Qualifying.txt

Variable Name	*Variable Description*
Movie_Id	ID *movie* yang bersifat unik
Cust_Id	ID *user* yang berlangganan netflix

2.3 *Explore Data*

Tugas ini membahas mengenai *data mining* dengan menggunakan teknik kueri, visualisasi, dan pelaporan. Eksplorasi data juga termasuk distribusi atribut kunci (misalnya, atribut target dari tugas prediksi) hubungan antara pasangan atau sejumlah kecil atribut, hasil agregasi sederhana, properti sub-populasi yang signifikan, dan analisis statistik sederhana. Analisis ini dapat secara langsung membahas tujuan *data mining*; mereka juga dapat berkontribusi

atau menyempurnakan deskripsi data dan laporan kualitas, dan dimasukkan ke dalam transformasi dan langkah-langkah persiapan data lainnya yang diperlukan untuk analisis lebih lanjut. Adapun analisis yang telah dilakukan pada *dataset*, yaitu:

```
print("Total data ")
print("\nTotal ratings :",df.shape[0])
print("Total Users   :", len(np.unique(df.user)))
print("Total movies  :", len(np.unique(df.movie)))
```

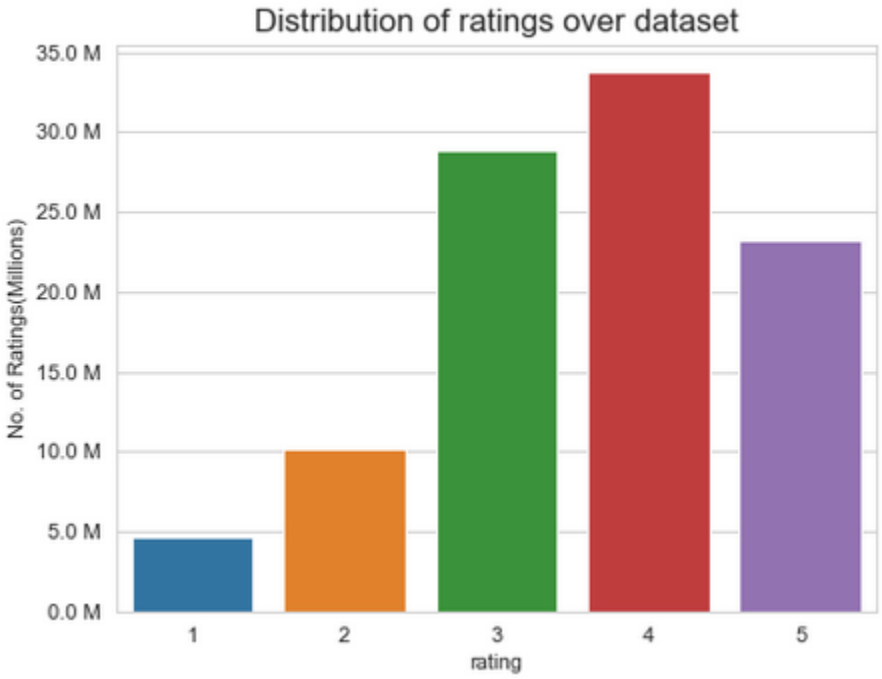
Gambar 2.1. *Cek Jumlah Baris Data*

Hasil pengecekan jumlah baris data yang terdapat pada *dataset*, yaitu :

Total ratings : 100480507

Total Users : 480189

Total movies : 17770



Gambar 2.2. Distribusi dari R*ating*

Dapat dilihat dari grafik diagram sebaran dari rating yang telah diberikan oleh *user*. Adapun kesimpulan yang bisa diperoleh berdasarkan diagram tersebut adalah *Movie* dengan rating 1 sekitar 4 juta, rating 2 10 juta, raring 3 sekitar 29 juta, rating 4 sekitar 33 juta, dan rating 5 sekitar 23 juta.

```
no_of_rated_movies_per_user =
df.groupby(by='user')['rating'].count().sort_values(ascending=False)

no of rated movies per user.head()
```

Gambar 2.3. Analisis Rating oleh *User*

Potongan program diatas digunakan untuk melihat jumlah rating yang pernah diberikan oleh setiap pengguna. Adapun hasilnya dapat dilihat pada gambar berikut.

```
user
305344     17653
387418     17436
2439493    16565
1664010    15813
2118461    14831
Name: rating, dtype: int64
```

Gambar 2.4. Hasil Analisis Rating

Misalnya *user* dengan Id 305344 telah memberikan total rating 17653.

2.4 *Verify Data Quality*

Tahap ini berisi evaluasi kualitas dan integritas data. Nilai yang hilang sering terjadi, terutama saat mengumpulkan data yang sangat besar dan membutuhkan waktu lama. Oleh sebab itu, perlu dilakukan pemeriksaan atribut yang hilang atau kosong dan penilaian semua nilai atribut. *Data cleaning* dibutuhkan untuk menghilangkan data yang *noise* dan data yang tidak sesuai atau data yang tidak diperlukan. Pembersihan data akan berdampak pada kinerja sistem *data mining* karena jumlah dan kompleksitas data telah berkurang.

BAB 3
DATA PREPARATION

Tahapan ini dilakukan untuk memastikan kualitas dari data yang akan digunakan. Pada tahap ini masalah yang perlu diatasi adalah apabila terdapat *noisy data* dan *missing values.* Proses pembersihan data atau *cleaning* dilakukan untuk menemukan apakah terdapat anomali data pada data yang akan digunakan.

3.1 *Dataset Description*

Pada tahap ini mendeskripsikan *dataset*. *Dataset* yang digunakan tersedia secara *online* di Kaggle di url https://www.kaggle.com/netflix-inc/netflix-prize-data. *Dataset* yang disediakan oleh Netflix ini bertujuan untuk menentukan apakah ada yang dapat memprediksi rating pengguna sebuah film lebih baik dari Netflix. *Dataset* berisi 7 file dengan format csv dan txt yaitu *combined data_1*.txt, *combined_data*_2.txt, *combined_data*_3.txt, *combined_data*_4.txt, *movie*_titles.csv, qualifying.txt, dan probe.txt. Adapun total baris data yang terdapat pada *dataset* yaitu :

Total ratings : 100480507

Total Users : 480189

Total movies : 17770

3.2 *Select Data*

Tugas ini adalah memilih data yang akan digunakan. Sumber data potensial yang memiliki atribut dan kualitas data yang sesuai dengan tujuan keseluruhan proyek. Dalam sumber data yang digunakan, hanya *combined data_1*.txt, *combined_data*_2.txt, *combined_data*_3.txt, *combined_data*_4.txt, *movie*_titles.txt. Adapun variabel pada dataset dapat dilihat pada **Error! Reference source not found.**.

Tabel 3.1. Deskripsi Variabel Pada Dataset

Variable Name	***Variable Description***
Movie_title	Judul *movie*
Movie_ID	ID *movie*
Year_release	Tahun *movie* dirilis
User_ID	ID *User*
rating	*Rating* terhadap *movie* yang diberikan oleh *User*

Variable Name	*Variable Description*
date	Tanggal pemberian *Rating* terhadap *movie* yang diberikan oleh *User*

3.3 *Clean Data*

Setelah dilakukan eksplorasi data pada subbab 2.3 selanjutnya adalah melakukan *data preprocessing*. Pada tahap ini dilakukan pembersihan terhadap data dan tindakan pembersihan seperti apa yang akan dilakukan untuk mendukung teknik pemodelan. Tahap ini termasuk mengidentifikasi *clean subset* dari data, memperbarui nilai yang kosong, atau memperkirakan data yang hilang (*missing values)*.

Dataset yang dihasilkan sangat besar sehingga terjadi kesalahan *memory* pada kernel saat kode program dijalankan. Oleh karena itu dilakukan *data cleaning* untuk mengurangi volume data dengan meningkatkan kualitas data di bawah ini:

1 Menghapus film dengan *review* yang terlalu sedikit (karena relatif tidak populer)

2 Menghapus cust_Id yang memberikan terlalu sedikit ulasan (karena relatif kurang aktif)

Sebelum melakukan *data cleaning* maka jumlah variabel pada dataset adalah sebagai berikut

Tabel 3.2. Dataset Sebelum Melakukan *Data Cleaning*

Nama Variabel	Jumlah
Movie	4.499
Customer	470.758
Ratings	24.053.764

Setelah melakukan *data cleaning* maka jumlah variabel pada dataset adalah sebagai berikut :

Tabel 3.3. Dataset Setelah Melakukan *Data Cleaning*

Nama Variabel	Jumlah
Movie	1.350
Customer	143.458
Ratings	17.337.458

3.4 *Integrate Data*

Pada sub bab 3.2 pemilihan data, kumpulan data yang digunakan terpisah yaitu *combined data_1*.txt dan *movie_titles.csv* Metode yang digunakan dalam menggabungkan data adalah

Merging data. Penggabungan 2 kumpulan data dengan *record* yang sama. Data tersebut digabungkan dengan menggunakan *key* (seperti *cust_id)*. Data yang dihasilkan bertambah dalam kolom.

Tabel berikut menampilkan jumlah data sebelum digabung (*merge dataset*).

Tabel 3.4. Sebelum *Merge Dataset*

Nama Variabel	Jumlah
Cust_Id	143.458
Rating	17.337.458

Tabel berikut menampilkan jumlah data setelah digabung (*merge dataset*).

Tabel 3.5. Setelah *Merge Dataset*

Nama Variabel	Jumlah
Cust_Id	143.458
Rating	17.337.458
Movie_Id	1.350

BAB 4
MODELING

Pada tahap *modeling* (pemodelan), data yang telah dipersiapkan dan dianalisis pada *data preparation* selanjutnya dibawa ke *modeling*, dan hasilnya mulai menjelaskantentang masalah bisnis yang ditimbulkan selama *business understanding*. Pemodelan biasanya dilakukan dalam beberapa iterasi. Biasanya, penambang data menjalankan beberapa model menggunakan parameter *default* dan kemudian menyempurnakan parameter atau kembali ke *data preparation.*

4.1 *Selecting Modeling Technique*

Menentukan model yang paling sesuai biasanya akan didasarkan pada pertimbangan berikut:

1. Tipe data yang tersedia untuk *mining*.
2. Tujuan *data mining*.
3. Persyaratan pemodelan khusus.

4.1.1 *Modeling Techniques*

Teknik pemodelan yang digunakan pada proyek ini adalah algoritma apriori sesuai dengan tujuan *data mining* proyek yaitu menggali pengetahuan (*discovering knowledge*) tentang pola (*pattern*) item berdasarkan riwayat pencarian pengguna sehingga dapat diketahui dan diprediksi item (film) yang mungkin diminati pengguna.

Algoritma apriori adalah algoritma yang umum digunakan untuk menambang aturan asosiasi data. Algoritma apriori digunakan untuk mengetahui kumpulan data yang sering muncul. Menemukan pola set ini membantu untuk membuat beberapa keputusan.

4.1.2 *Modeling Assumptions*

Semua asumsi data serta manipulasi data apa pun yang dilakukan untuk memenuhi persyaratan model akan didokumentasikan. Pemodelan algoritma apriori pada proyek ini akan menghasilkan aturan seperti berikut : jika seseorang merekomendasikan film-film ini, mereka juga akan merekomendasikan film ini. Agar dapat menghasilkan aturan tersebut maka, data perlu dimanipulasi untuk menentukan apakah seseorang merekomendasikan sebuah film. Hal ini dilakukan dengan menciptakan fitur baru yang akan bernilai benar jika orang memberikan *review* baik untuk film. Contoh kode program yang digunakan untuk fitur ini terdapat pada Gambar 4.1.

```
merge_dataset["Populer"] = merge_dataset["Rating"] > 3
merge_dataset[10:15]
merge_dataset[merge_dataset["Cust_Id"] == 1][:5]
```

Gambar 4.1. Kode Program Movie dengan Rating Terpopuler

Pada Gambar 4.1, kode program bertujuan untuk merancang kondisi yang mengindikasi *movie* dengan *rating* terpopuler sehingga menghasilkan fitur baru dengan menampilkan dataset movie dengan rating terpopuler.

4.2 *Generate Test Design*

Sebagai langkah terakhir sebelum benar-benar membangun model, maka terlebih dahulu mempertimbangkan bagaimana hasil model akan diuji. Ada dua bagian untuk menghasilkan desain pengujian yang komprehensif :

1. Mendeskripsikan kriteria“ *goodness* ”model
2. Menentukan data yang akan menguji kriteria ini

Keunggulan model dapat diukur dengan beberapa cara. Pada proyek ini, model akan diukur dengan mengevaluasi aturan asosiasi menggunakan konsep yang sama seperti pada klasifikasi yaitu menghitung *test set confidence*, yaitu confidence masing-masing aturan asosiasi pada test set.

4.2.1 *Test Design*

Desain pengujian (test design) merupakan gambaran langkah-langkah yang akan dilakukan untuk menguji model yang dihasilkan. Pada proyek ini, langkah-langkah untuk menguji model adalah sebagai berikut :

1. Mengestrak test data yaitu record yang tidak digunakan dalam training set.
2. Menghitung instance yang benar di mana premisnya mengarah ke kesimpulan.
3. Menghitung confidence setiap aturan dari jumlah yang benar.
4. Mencetak aturan asosiasi terbaik dengan judul.

4.3 *Build Model*

Pada proses pembuatan model, terdapat tiga bagian informasi yang akan digunakan dalam keputusan *data mining*:

4. Parameter settings

   Pengaturan parameter mencakup catatan terkait parameter yang memberikan hasil terbaik.
5. Models

   Model aktual yang diproduksi.

6. Model Description

   Deskripsi hasil model, termasuk kinerja dan masalah data yang terjadi selama pelaksanaan model dan eksplorasi hasil-hasilnya.

4.3.1 *Parameter Settings*

Sebagian besar teknik pemodelan memiliki berbagai parameter atau pengaturan yang dapat disesuaikan untuk mengendalikan proses pemodelan. Pada proyek ini, digunakan parameter *value*s untuk mendapatkan nilai aktual.

4.3.2 *Models*

Setelah menentukan parameter yang diibutuhkan maka selanjutnya adalah mengeksekusi model untuk menghasilkan *result* yang terlihat.

4.3.3 *Models Description*

Dalam penambangan data (data mining) dari *frequent itemset*, jenis metode asosiasi termasuk algoritma Apriori. Analisis aturan asosiasi adalah teknik untuk mengungkap bagaimana item dikaitkan satu sama lain. Satu set *item* disebut frequent jika memenuhi nilai ambang minimum untuk *support* dan *confidence.* Langkah pertama saat pemodelan menggunakan algoritma apriori adalah membuat initial *frequent itemset*. Berikut merupakan program kode yang digunakan untuk mendapatkan *initial frequent itemset*.

```
frequent_itemsets = {}
min_support = 50
frequent_itemsets[1] = dict((frozenset((Movie_Id,)), row["Populer"])
                            for Movie_Id, row in
num_favorable_by_movie.iterrows()
                            if row["Populer"] > min_support)

print("Terdapat {} movies dengan lebih dari {} favorable
reviews".format(len(frequent_itemsets[1]), min_support))
```

Gambar 4.2. Kode Program Mendapatkan *Initial Frequent Itemset*

Kemudian, membuat *candidate itemset* menggunakan *superset* di *frequent itemset* yang tersedia. Selanjutnya menguji *candidate itemset* untuk melihat apakah *frekuen* atau tidak, namun apabila tidak maka akan dihapus.

```
import sys
# Mengatur fungsi dan counting dictionary
from collections import defaultdict

def find_frequent_itemsets(favorable_reviews_by_customers, k_1_itemsets,
min_support):
```

```
    counts = defaultdict(int)

# Melakukan iterasi pada semua pengguna dan ulasan mereka
    for user, reviews in favorable_reviews_by_customers.items():
# Melihat setiap itemset yang ditemukan sebelumnya
         for itemset in k_1_itemsets:
            if itemset.issubset(reviews):
# Membuat superset dari pengguna yang tidak ada dalam itemset
                for other_reviewed_movie in reviews - itemset:
                    current_superset = itemset |
frozenset((other_reviewed_movie,))
                    counts[current_superset] += 1
# Mengakhiri fungsi kita dengan menguji candidate itemset
    return dict([(itemset, frequency) for itemset, frequency in
counts.items()
                 if frequency >= min_support])
# Membuat frequent itemset dan menyimpan mereka di dalam dictionary sesuai
dengan panjangnya
for k in range(2, 20):
    cur_frequent_itemsets =
find_frequent_itemsets(favorable_reviews_by_customers,
                           frequent_itemsets[k-1],
                           min_support)
# Memecah loop sebelumnya jika tidak menemukan frequent itemset yang baru
    if len(cur_frequent_itemsets)==0:
        print("Did not find any frequent itemsets of length {}".format(k))
        sys.stdout.flush()
        break
```

Gambar 4.3. Kode Program Menguji *Candidate itemset*

Selanjutnya adalah mengembalikan semua *frequent itemset* yang ditemukan.

```
else:
        print("I found {} frequent itemsets of length
{}".format(len(cur_frequent_itemsets), k))
        frequent_itemsets[k] = cur_frequent_itemsets
del frequent_itemsets[1]
```

Gambar 4.4. Mengembalikan Semua *Frequent Itemset*

Setelah algoritma Apriori selesai, kemudian mengkstrak aturan asosiasi (*association rules*)

```
# Mencetak lima aturan teratas dengan mengurutkan confidence dictionary
from operator import itemgetter

sorted_confidence = sorted(rule_confidence.items(), key=itemgetter(1),
reverse=True)
for index in range(5):
    print("Rule #{0}".format(index + 1 ))
    (premise, conclusion) = sorted_confidence[index][0]
    print("Rule: If a person recommends {0} they will also recommend
{1}".format(premise, conclusion))
    print(" - Confidence: {0:.3f}\n".format(rule_confidence[(premise,
conclusion)]))
```

Gambar 4.5. Mengekstrak *Association Rules*

Berikut merupakan hasil pemodelan *movie recommendation* yang dihasilkan oleh aturan asosiasi pada algoritma apriori :

```
Rule #1
Rule: If a person recommends The Last Samurai, Kill Bill: Vol. 2, Finding
Nemo (Widescreen) they will also recommend 1905
 - Confidence: 1.000

Rule #2
Rule: If a person recommends Kill Bill: Vol. 2, Finding Nemo (Widescreen),
Man on Fire they will also recommend 1905
 - Confidence: 1.000

Rule #3
Rule: If a person recommends 50 First Dates, Lord of the Rings: The
Fellowship of the Ring, The Bourne Supremacy they will also recommend 1905
 - Confidence: 1.000

Rule #4
Rule: If a person recommends Kill Bill: Vol. 2, Finding Nemo (Widescreen),
American Beauty they will also recommend 1905
 - Confidence: 1.000

Rule #5
Rule: If a person recommends The Italian Job, Pirates of the Caribbean:
The Curse of the Black Pearl, Ray they will also recommend 2372
 - Confidence: 1.000
```

Gambar 4.6. Hasil Pemodelan *Movie Recommendation*

4.4 *Assess Model*

Pemodelan dilakukan dengan pembuatan *itemset* dengan setiap film secara individual dan menguji apakah *itemset* tersebut *frekuen*. Model dinilai menggunakan konsep yang sama seperti pada klasifikasi yaitu menggunakan set data yang tidak digunakan untuk *training* dan mengevaluasi aturan yang kita temukan berdasarkan performa aturan tersebut dalam *test set* ini. Untuk dapat melakukan *assess model* maka terlebih dahulu menghitung *test set confidence*, yaitu *confidence* masing-masing aturan asosiasi *pada test set*. Sebelum menghitung *confidence* maka pertama sekali harus mendapatkan nilai *support* terlebih dahulu. Nilai *support* (s) adalah persentase untuk jumlah kasus seperti kombinasi *item.* Untuk menghitung *suppo*rt digunakan rumus berikut :

$$Support, s(X \rightarrow Y) = \frac{(X \cup Y)}{N}$$

Gambar 4.7. Rumus Menghitung *Support*

X∪Y = jumlah transaksi untuk X dan Y,

N = jumlah seluruh transaksi.

Kemudian, menghitung nilai *confidence.* Untuk menghitung *confidence.* digunakan rumus berikut :

```
Confidence,c(X->Y)=Support_count(XUY)/Support_count(X)
```

Gambar 4.8. Rumus Menghitung *Confidence*

X∪Y = jumlah transaksi untuk X dan Y,

X = jumlah transaksi untuk X.

Nilai *confident* yang tinggi untuk mendeskripsikan banyaknya Y yang muncul dalam transaksi untuk X.

BAB 5
EVALUATION

Evaluasi hasil pemodelan bertujuan untuk memastikan bahwa hasil pemodelan sudah tepat dan sesuai. Dua jenis hasil yang dihasilkan oleh *data mining* :

1. Model terakhir yang dipilih di fase CRISP-DM sebelumnya.
2. Setiap kesimpulan atau kesimpulan yang diambil dari model itu sendiri serta dari data proses penambangan. Ini dikenal sebagai temuan.

Saat melakukan evaluasi, kita menghitung *confidence* setiap aturan dari jumlah yang benar.

```
test_confidence = {candidate_rule:
                   (correct_counts[candidate_rule] /
                   float(correct_counts[candidate_rule] +
incorrect_counts[candidate_rule]))
                   for candidate_rule in rule_confidence}
```

Gambar 5.1. Kode Program Menghitung *Confidence* Setiap *Rule*

Selanjutnya, mencetak aturan asosiasi terbaik.

```
for index in range(5):
    print("Rule #{0}".format(index + 1))
    (premise, conclusion) = sorted_confidence[index][0]
    premise_names = ", ".join(get_merge_dataset_title(idx) for idx in
premise)
    conclusion_name = get_merge_dataset_title(conclusion)
    print("Rule : if a person recommends {0} they will also recommend {1}"
          .format(premise_names, conclusion_name))
    print(" - Train confidence:
{0:.3f}".format(rule_confidence.get((premise, conclusion), -1)))
    print("- Test Confidence:
{0:.3f}\n".format(test_confidence.get((premise, conclusion), -1)))
```

Gambar 5.2 Kode Program Mencetak Aturan Asosiasi

Aturan asosiasi terbaik dapat dilihat pada gambar berikut :

```
Rule #1
Rule : if a person recommends The Last Samurai, Kill Bill: Vol. 2, Finding
Nemo (Widescreen) they will also recommend Pirates of the Caribbean: The
Curse of the Black Pearl
 - Train confidence: 1.000
- Test Confidence: 1.000

Rule #2
Rule : if a person recommends Kill Bill: Vol. 2, Finding Nemo (Widescreen),
Man on Fire they will also recommend Pirates of the Caribbean: The Curse of
the Black Pearl
```

```
 - Train confidence: 1.000
- Test Confidence: 1.000

Rule #3
Rule : if a person recommends 50 First Dates, Lord of the Rings: The
Fellowship of the Ring, The Bourne Supremacy they will also recommend
Pirates of the Caribbean: The Curse of the Black Pearl
 - Train confidence: 1.000
- Test Confidence: 1.000

Rule #4
Rule : if a person recommends Kill Bill: Vol. 2, Finding Nemo (Widescreen),
American Beauty they will also recommend Pirates of the Caribbean: The
Curse of the Black Pearl
 - Train confidence: 1.000
- Test Confidence: 1.000

Rule #5
Rule : if a person recommends The Italian Job, Pirates of the Caribbean:
The Curse of the Black Pearl, Ray they will also recommend The Bourne
Supremacy
 - Train confidence: 1.000
- Test Confidence: 1.000
```

Gambar 5.3. Hasil Aturan Asosiasi

Confidence selalu antara 0 dan 1. Jika terdapat nilai -1 maka menunjukkan bahwa aturan tertentu tidak ditemukan dalam *test set*.

DAFTAR PUSTAKA

[1] p. Chapman, . J. Clinton , . R. Kerber , T. Khabaza , T. Reinartz, C. Shearer dan R. Wirth, “CRISP-DM 1.0,” 2000.

[2] S. Agrawal dan P. Jain, “An Improved Approach for Movie Recommendation System,” *International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC),* p. 336, 2017.

[3] T. Badriyah, . R. Fernando and I. Syarif, "Sistem Rekomendasi Content Based Filtering Menggunakan Algoritma Apriori," *Konferensi Nasional Sistem Informasi ,* 2018.