TUGAS AKHIR – KS184822

KLASIFIKASI MULTILABEL GENRE FILM MENGGUNAKAN TRANSFORMASI DAN *MULTILABEL K-NEAREST NEIGHBOR* (ML-KNN)

HERVIANA MAYU NABILA
NRP 062116 4000 0075

Dosen Pembimbing
Dr. Dra. Kartika Fithriasari., M.Si
Adatul Mukarromah, S.Si., M.Si.

PROGRAM STUDI SARJANA
DEPARTEMEN STATISTIKA
FAKULTAS SAINS DAN ANALITIKA DATA
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2020

TUGAS AKHIR – KS184822

KLASIFIKASI MULTILABEL GENRE FILM MENGGUNAKAN TRANSFORMASI DAN *MULTILABEL K-NEAREST NEIGHBOR* (MLKNN)

HERVIANA MAYU NABILA
NRP 062116 4000 0075

Dosen Pembimbing
Dr. Dra. Kartika Fithriasari., M.Si
Adatul Mukarromah, S.Si., M.Si.

PROGRAM STUDI SARJANA
DEPARTEMEN STATISTIKA
FAKULTAS SAINS DAN ANALITIKA DATA
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2020

FINAL PROJECT – KS184822

MOVIE GENRE MULTILABEL CLASSIFICATION USING PROBLEM TRANSFORMATION AND MULTILABEL K-NEAREST NEIGHBOR (ML-KNN)

HERVIANA MAYU NABILA
SN 062116 4000 0075

Supervisors
Dr. Dra. Kartika Fithriasari., M.Si
Adatul Mukarromah, S.Si., M.Si.

UNDERGRADUATE PROGRAMME
DEPARTMENT OF STATISTICS
FACULTY OF SCIENCE AND DATA ANALYTICS
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2020

(Halaman ini sengaja dikosongkan)

LEMBAR PENGESAHAN

KLASIFIKASI MULTILABEL GENRE FILM MENGGUNAKAN TRANSFORMASI DAN MULTILABEL K-NEAREST NEIGHBOR (ML-KNN)

TUGAS AKHIR

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

Oleh :
Herviana Mayu Nabila
NRP. 062116 4000 0075

Disetujui oleh Pembimbing Tugas Akhir:
Dr. Dra. Kartika Fithriasari, M.Si. ( )
NIP. 19741213 199802 2 001
Adatul Mukarromah, S.Si., M.Si. ( )
NIP. 19800418 200312 2 001

Mengetahui,
Kepala Departemen

Dr. Dra. Kartika Fithriasari, M.Si.
NIP. 19691212 199303 2 002

SURABAYA, JULI 2020

(Halaman ini sengaja dikosongkan)

KLASIFIKASI MULTILABEL GENRE FILM MENGGUNAKAN TRANSFORMASI DAN *MULTILABEL K-NEAREST NEIGHBOR* (ML-KNN)

Nama Mahasiswa	**:**	**Herviana Mayu Nabila**
NRP	**:**	**062116 4000 0075**
Departemen	**:**	**Statistika-FSAD-ITS**
Dosen Pembimbing	**:**	**Dr. Dra. Kartika Fithriasari., M.Si**
		Adatul Mukarromah, S.Si., M.Si.

Abstrak

Film merupakan bentuk media massa yang mampu memberikan nilai hiburan pada masyarakat. Berkembangnya dunia digital membuat masyarakat dapat dengan mudah mengakses situs online yang menyediakan informasi mengenai genre film, salah satunya melalui situs rating film seperti IMDb dan TMDb. Pengelompokkan genre film dilakukan secara manual dan tidak efisien karena membutuhkan banyak waktu dan tenaga ahli, sehingga klasifikasi genre film secara otomatis dapat menjadi solusi. Suatu film tidak hanya memiliki satu genre saja namun dapat memiliki lebih dari satu genre, sehingga klasifikasi genre film dikategorikan sebagai klasifikasi multilabel. Pada penelitian tugas akhir ini dilakukan klasifikasi genre drama, action, adventure, thriller, dan comedy berdasarkan teks sinopsis film dengan membandingkan pendekatan transformasi dan adaptasi algoritma. Metode transformasi yang digunakan adalah Label Powerset (LP), dan metode adaptasi algortima yang digunakan adalah Multilabel K-Nearest Neighbor (ML-KNN). Data yang digunakan berupa teks sinopsis dan genre film yang diambil dari situs The Movie Database (TMDb). Hasil penelitian menunjukkan bahwa metode Multilabel K-Nearest Neughbor (ML-KNN) menghasilkan hamming loss lebih kecil dibandingkan metode K-Nearest Neighbor dengan transformasi Label Powerset (LP).

Kata kunci: Genre Film, Label Powerset, ML-KNN, Multilabel

(Halaman ini sengaja dikosongkan)

MOVIE GENRE CLASSIFICATION USING PROBLEM TRANSFORMATION AND MULTILABEL K-NEAREST NEIGHBOR (ML-KNN)

Name	**:**	**Herviana Mayu Nabila**
Student Number	**:**	**062116 4000 0075**
Department	**:**	**Statistics**
Supervisors	**:**	**Dr. Dra. Kartika Fithriasari., M.Si**
		Adatul Mukarromah, S.Si., M.Si.

Abstract

A movie is a media that is able to provide entertaiment value to the public. Movies can be devided into several categories or called movie genres. Movie gere is useful for people to watch movies based on the type they like. The development of digital world allows people to easily access online sites that provide information about movie genres, one which is through a movie rating websites such as IMDb and TMDb. Manual genre classification is ineffiecient because it requires a lot of time, so automatic movie genre classification can be a solution. A movie does not only have one genre but can have several genres at once, so the classification of movie genre is categorized as multilabel classification. In multilabel classification, a data can be categorized to more than one label. This study will comparing the problem transformation method called Binary Relevance (BR) and adaptation algorithm Method named Multilabel K-Nearest Neighbor (ML-KNN) on classfying drama, action, adventure, thriller, and comedy movie genre. The data used in this study is synopsis text and movie genres that taken from The Movie Database (TMDb) sites using TMDb API. The study conducted shows that movie genre multilabel classification using Multilabel K-Nearest Neighbor (ML-KNN) has a lower hamming loss than Label Powerset (LP).

Keywords: Label Powerset, ML-KNN, Multilabel, Movie Genre

(Halaman ini sengaja dikosongkan)

KATA PENGANTAR

Puji syukur penulis panjatkan atas rahmat dan hidayah yang diberikan Allah SWT sehingga penulis dapat menyelesaikan laporan Tugas Akhir yang berjudul "Klasifikasi Multilabel Genre Film Menggunakan Transformasi dan *Multilabel K-Nearest Neighbor* (ML-KNN)" dengan lancar.

Penulis menyadari bahwa Tugas Akhir ini dapat terselesaikan tidak terlepas dari bantuan dan dukungan berbagai pihak. Oleh karena itu, penulis menyampaikan terima kasih kepada:

1. Dr. Dra. Kartika Fithriasari, M.Si. selaku Kepala Departemen Statistika FSAD dan Dr. Santi Wulan, S.Si., M.Si. selaku Sekretaris Departemen Bidang Akademik yang telah menyediakan fasilitas untuk menyelesaikan Tugas Akhir ini..
2. Dr. Dra. Kartika Fithriasari, M.Si. dan Adatul Mukarromah S.Si., M.Si., selaku dosen pembimbing penulis dalam menyelesaikan laporan Tugas Akhir ini dengan sabar dan tak lupa memberikan semangat dan motivasi kepada penulis dalam menyelesaikan penelitian ini.
3. Dr. Sutikno, S.Si., M.Si., selaku dosen wali penulis selama masa studi yang telah banyak memberikan saran dan arahan dalam proses belajar di Departemen Statistika
4. Dr. Irhamah, S.Si, M.Si. dan Dra. Wiwiek Setya Winahju, M.S. selaku dosen penguji yang selalu sabar dalam mengomentari serta memberikan masukan dan saran dalam penyelesaian tugas akhir.
5. Seluruh dosen Departemen Statistika ITS yang telah memberikan ilmu dan pengetahuan yang tak ternilai harganya, serta segenap karyawan Departemen Statistika ITS.
6. Kedua orang tua dan keluarga, atas segala do'a, nasehat, kasih sayang, dan dukungan yang diberikan kepada penulis demi kesuksesan dan kebahagiaan penulis.
7. Teman-teman Statistika ITS ∑27 angkatan 2016, yang selalu memberikan dukungan kepada penulis selama ini.

8. Semua teman, relasi dan berbagai pihak yang tidak bisa penulis sebutkan namanya satu persatu yang telah membantu dalam penulisan laporan ini.

Besar harapan penulis untuk mendapatkan kritik dan saran yang membangun sehingga Tugas Akhir ini dapat memberikan manfaat bagi semua pihak yang terkait.

Surabaya, 2020

Penulis

DAFTAR ISI

DAFTAR GAMBAR

(Halaman ini sengaja dikosongkan)

DAFTAR TABEL

(Halaman ini sengaja dikosongkan)

DAFTAR LAMPIRAN

(Halaman ini sengaja dikosongkan)

BAB I
PENDAHULUAN

1.1 Latar Belakang

Kemajuan teknologi informasi khususnya perkembangan internet di Indonesia menciptakan era digital dimana informasi, komunikasi, hiburan, bahkan kebutuhan sehari-hari dapat diakses dengan mudah. Jumlah masyarakat Indonesia yang terhubung ke internet terus meningkat. Berdasarkan hasil Polling Indonesia yang bekerja sama dengan Asosiasi Penyelenggara Jasa Internet Indonesia (APJII), pada tahun 2018 jumlah pengguna internet di Indonesia sudah mencapai 171,17 juta jiwa atau setara dengan 64,8% dari total penduduk Indonesia (Franedya, 2019). Kemajuan teknologi dapat berdampak pada industri hiburan termasuk industri film. Film merupakan salah satu bentuk media massa yang mampu memberikan nilai hiburan pada masyarakat. Secara harfiah, film (*cinema*) adalah *cinematographie* yang berasal dari kata *cinema* (gerak), *tho* atau *phytos* (cahaya) dan *graphie* atau *grhap* (tulisan, gambar, citra), sehingga film diartikan melukis gerak dengan cahaya (Riadi, 2012). Film berperan sebagai sarana komunikasi yang digunakan untuk menyebarkan hiburan yang menyajikan cerita, peristiwa, musik, drama, humor dan sajian teknis lainnya pada masyarakat umum.

Pemanfaatan era digital membuka peluang usaha bagi penyedia jasa untuk memfasilitasi masyarakat sebagai penikmat film. Permasalahan yang sering dialami masyarakat ketika ingin menonton film adalah ingin mengetahui informasi dari film tersebut seperti ringkasan dana lur cerita, pemain dan sutradara, serta genre dari film tersebut. Sehingga banyak situs *online* yang menyediakan informasi seputar film, mulai dari genre, sinopsis, pemain, rating, dan *review* film. Situs penyedia informasi ini digemari masyarakat karena mudah untuk diakses melalui internet kapanpun dan dimanapun. Situs rating film juga dapat digunakan sebagai media promos film. Film yang masuk ke situs-situs rating film otomatis akan lebih dikenal oleh masyarakat. Salah satu contoh situs rating film adalah Internet Movie Database (IMDb).

Internet Movie Database (IMDb) merupakan situs *online* yang menyediakan informasi mengenai film dan TV *show* dari seluruh dunia, termasuk daftar pemeran, ringkasan alur cerita, dan ulasan serta penilaian dari penggemar. IMDb diluncurkan secara *online* pada tahun 1990, dan telah menjadi anak perusahaan dari Amazon.com sejak tahun 1998 (IMDb, 2019). Selain IMDb, situs *review* film lainnya adalah The Movie Database (TMDb). TMDb adalah *database* film dan TV yang dibangun oleh komunitas yang telah didirikan dari tahun 2008, sampai saat ini jumlah film yang ada di TMDb mencapai 514 ribu film (TMDb, 2019).

Film dapat dibedakan menjadi beberapa kategori atau lebih dikenal dengan sebutan genre film. Genre film dapat diartikan sebagai bentuk, kategori, atau klasifikasi tertentu dari beberapa film yang memiliki kesamaan bentuk, latar, tema, suasana, dan lainnya (Dirks, 2010). Genre film diantaranya adalah *melodrama, western, horror, comedy*, dan *action-adventure.* Genre film digunakan oleh situs rating film untuk mengkategorikan film sehingga memudahkan pengguna mencari film dengan genre tertentu. Saat ini, penentuan genre film di IMDb dan TMDb dilakukan secara manual dengan cara menonton langsung keseluruhan film, kemudian memberikan genre sesuai dengan adegan yang ada pada film tersebut. Pemberian genre secara manual dirasa tidak efisien karena membutuhkan banyak waktu dan tenaga ahli, mengingat jumlah film yang tersedia mencapai ratusan ribu. Pemberian genre secara otomatis dapat membantu mengurangi atau menggantikan peran manusia dalam memberi genre pada film. Pengklasifikasian genre film dapat diselesaikan dengan berbagai cara, salah satunya ialah menggunakan data berupa teks atau biasa dikenal dengan *text mining. Text mining* adalah suatu langkah dari analisis teks yang dilakukan secara otomatis oleh komputer untuk menggali informasi dari suatu rangkaian teks (Han dan Kamber, 2006). Pada penelitian ini akan dilakukan klasifikasi genre film dengan data berupa teks sinopsis atau ringkasan film. Sinopsis adalah ringkasan pendek yang menggambarkan isi film (Eneste, 2005).

Menurut Tim Dirks (2010), suatu film tidak hanya memiliki satu genre saja melainkan dapat memiliki lebih dari satu genre sekaligus, sehingga klasifikasi genre film dikategorikan sebagai klasifikasi multilabel. Klasifikasi multilabel ialah klasifikasi di mana satu data dapat dihubungkan dengan lebih dari satu label (Manning dkk., 2009). Sebagai contoh, film Deadpool memiliki tiga genre yaitu *action, adventure* dan *comedy.* Terdapat dua macam pendekatan utama terhadap klasifikasi multilabel, yang pertama yaitu *problem transformation,* di mana objek multilabel diubah menjadi *single label* dahulu, kemudian diklasifikasikan seperti layaknya objek *single label* biasa. Pendekatan kedua adalah adaptasi algoritma, yaitu menggunakan algoritma yang di-sesuaikan untuk melakukan klasifikasi secara langsung dari objek multilabel (Tsoumakas dan Katakis, 2009).

Penelitian sebelumnya terkait klasifikasi multilabel dengan pendekatan transformasi pernah dilakukan oleh Pushpa dan Karpagavalli (2017). Penelitian ini membandingkan transfor-masi *Binary Relevance* (BR), *Label Powerset* (LP), dan *Classifier Chain* (CC) dengan metode klasifikasi yang digunakan adalah J48, *Naïve Bayes, SMO, AdaBoostMI,* dan *Zero R.* Hasil dari penelitian ini menunjukkan bahwa metode transformasi *Label Powerset* (LP) adalah yang terbaik dengan akurasi paling tinggi. Klasifikasi multilabel dengan membandingkan metode *Mallows,* ML-KNN, dan BR(C4.5) dilakukan oleh Cheng dan Hullermeier (2008). Menggunakan 7 jenis *dataset* yang berbasis *music, vision, biology, multimedia,* dan *text.* Pada penelitian ini didapatkan hasil klasifikasi terbaik untuk jenis *dataset* berbasis teks yaitu metode ML-KNN dengan nilai *rank-loss* paling kecil yaitu sebesar 0.068 dan nilai *hamming loss* kedua terkecil yaitu sebesar 0.087.

Wiraguna, Al Faraby, dan Adiwijaya (2019) juga melakukan klasifikasi multilabel pada Hadist Bukhari dalam terjemahan bahasa Indonesia menggunakan pendekatan transformasi *Binary Relevance* (BR), *Label Powerset* (LP), *Classifier Chain* (CC) dengan metode klasifikasi *Random Forest.* Hasil penelitian me-nunjukkan bahwa nilai *hamming loss* paling rendah didapatkan

dengan metode *Binary Relevance* (BR) tanpa proses *stemming* yaitu sebesar 0,0663. Isnaini, Adiwijaya, Mubarok, Abu Bakar (2019) melakukan klasifikasi berita Indonesia menggunakan KNN dengan pendekatan transformasi *Binary Relevance* (BR). Hasil klasifikasi terbaik adalah KNN dengan *manhattan distance* dan k = 11 didapatkan nilai *hamming loss* sebesar 11.16%.

Klasifikasi genre film menggunakan sinopsis secara *single label* pernah dilakukan oleh Muslimah, Indriati, dan Wihandika (2019). Genre film yang digunakan adalah genre *action, horror, romance, thriller,* dan *family.* Dalam penelitian tersebut klasifikasi dilakukan menggunakan metode *improved* KNN dengan *cosine similarity* dan pembobotan kata TF-IDF. Berdasarkan hasil penelitian pada klasifikasi genre film menggunakan *improved* KNN didapatkan tingkat akurasi sebesar 88%. Pengklasifikasian dokumen teks secara *single label* pernah dilakukan oleh Nugraha, Al Faraby, dan Adiwijaya (2018) menggunakan metode KNN dengan *information gain,* diperoleh hasil klasifikasi terbaik menggunakan metode KNN tanpa *information gain* dengan akurasi mencapi 92%.

Oleh karena itu, penelitian ini bertujuan untuk melakukan klasifikasi genre film secara multilabel berdasarkan teks sinopsis menggunakan pendekatan tranformasi yaitu *Label Powerset* (LP) menggunakan metode klasifikasi K-*Nearest Neighbor* (KNN) dan pendekatan adaptasi algoritma menggunakan metode *Multilabel* K-*Nearest Neighbor* (ML-KNN).

1.2 Perumusan Masalah

Pemberian genre film pada situs rating film masih dilakukan secara manual dengan cara menonton langsung film dan memberi genre yang sesuai dengan adegan yang ada pada film tersebut. Hal tersebut menjadi masalah karena pemberian genre secara manual dirasa tidak efisien karena membutuhkan banyak waktu dan tenaga ahli. Situs rating dan *review* film yang mengalami permasalahan tersebut diantaranya TMDb dan IMDb, di mana genre film dibutuhkan untuk mengelompokkan film dan mempermudah masyarakat mencari film dengan jenis yang diinginkan.

Suatu film dapat masuk ke beberapa genre sekaligus, sehingga klasifikasi genre film tergolong ke dalam klasifikasi multilabel. Ketepatan pengklasifikasian genre pada film penting untuk diperhatikan agar masyarakat sebagai penikmat film dapat mudah mencari film yang sesuai dengan genre yang diinginkan. Klasifikasi multilabel dapat diselesaikan dengan dua pendekatan yakni transformasi dan adaptasi algoritma. Metode yang digunakan untuk pendekatan transformasi adalah *Label Powerset* (LP) dan untuk pendekatan adaptasi algoritma adalah *Multilabel K-Nearest Neighbor* (ML-KNN).

1.3 Tujuan

Tujuan yang ingin dicapai dalam penelitian ini adalah sebagai berikut.

1. Mengklasifikasikan genre *action, adventure, comedy, drama,* dan *thriller* pada film berdasarkan sinopsis dengan pendekatan transformasi *Label Powerset* (LP) dan metode klasifikasi *K-Nearest Neighbor* (KNN) serta pendekatan adaptasi algoritma *Multilabel K-Nearest Neighbor* (ML-KNN).
2. Mendapatkan hasil klasifikasi terbaik dalam hal mengklasifikasikan genre *action, adventure, comedy, drama,* dan *thriller* pada film berdasarkan sinopsis antara pendekatan transformasi *Label Powerset* (LP) serta pendekatan adaptasi algoritma ML-KNN.

1.4 Manfaat Penelitian

Manfaat yang diharapkan pada penelitian ini sebagai berikut.

1. Bagi keilmuan statistika
   Dapat menjadi informasi tambahan untuk penelitian-penelitian selanjutnya dalam melakukan klasifikasi multilabel berdasarkan teks menggunakan pendekatan transormasi dan adaptasi algoritma ML-KNN.
2. Bagi pembaca
   Dapat menjadi informasi tambahan kepada pembaca, khususnya yang melakukan penelitian dalam bidang *text mining* dan

klasifikasi multilabel menggunakan pendekatan transformasi dan pendekatan adaptasi algoritma.

3. Bagi penyedia situs rating dan *review* film *online*
   Dapat menjadi informasi tambahan bagi situs *online* rating film seperti IMDb dan TMDb untuk mempertimbangkan pengklasifikasian genre film secara otomatis.

1.5 Batasan Masalah

Batasan permasalahan dalam penelitian ini yaitu data yang digunakan adalah film yang diambil pada *website* TMDb (The Movie Database) dengan kategori 'Most Popular Movies'. Genre film yang digunakan yaitu *action, comedy, drama, horror,* dan *science fiction.*

BAB II
TINJAUAN PUSTAKA

2.1 Klasifikasi Multilabel

Klasifikasi multilabel merupakan klasifikasi di mana suatu data dapat dikategorikan ke dalam lebih dari satu label (Manning dkk., 2009). Klasifikasi multilabel berbeda dengan klasifikasi *single label*. Pada klasifikasi multilabel, setiap *dataset* atau sekumpulan atribut pada *dataset* terkait dengan beberapa label yang terdiri dari beberapa kelas biner (*binary class*). Sedangkan dalam klasifikasi *single label* setiap *dataset* atau kumpulan atribut dalam *dataset* hanya terkait dengan satu label, baik itu *binary class* atau *multiclass* (Herrera dkk., 2016). Perbedaan klasifikasi *single label* dengan klasifikasi multilabel dapat dilihat pada Gambar 2.1.

Binary-class Dataset

X_1	X_2	…	X_m	Y
X_{11}	X_{12}	…	X_{1m}	1
X_{21}	X_{22}	…	X_{2m}	0
…	…	…	…	1
X_{n1}	X_{n2}	…	X_{nm}	0

Multi-class Dataset

X_1	X_2	…	X_m	Y
X_{11}	X_{12}	…	X_{1m}	C1
X_{21}	X_{22}	…	X_{2m}	C2
…	…	…	…	C3
X_{n1}	X_{n2}	…	X_{nm}	C4

Multi-label *dataset*

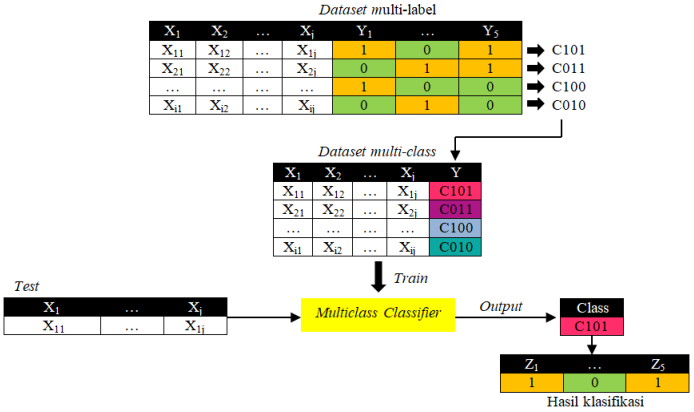
X_1	X_2	…	X_m	Y_1	…	Y_k
X_{11}	X_{12}	…	X_{1m}	1	0	1
X_{21}	X_{22}	…	X_{2m}	0	1	1
…	…	…	…	1	0	0
X_{n1}	X_{n2}	…	X_{nm}	0	1	0

Gambar 2.1 Ilustrasi *Dataset* Biner, *Multiclass*, dan Multilabel

Menurut Tsoumakas dan Katakis (2009), ada dua macam pendekatan utama terhadap klasifikasi multilabel. Pendekatan pertama yaitu *problem transformation*, di mana objek multilabel diubah menjadi *single label* dahulu, kemudian diklasifikasikan seperti layaknya objek *single label*. Ide dari *problem transformation* adalah bagaimana membentuk multilabel *dataset* dan bisa menghasilkan kumpulan data yang bisa diproses oleh klasifikasi biner maupun *multiclass. Problem transformation* menggunakan metode klasifikasi *single label* yang umum digunakan dan tidak melakukan modifikasi khusus dalam pembentukan *dataset* (Read

dkk., 2012). Jenis transformasi yang biasa digunakan diantaranya *Binary Relevance*, *Label Powerset*, *Classifier Chains*. Pendekatan kedua adalah adaptasi algoritma, yaitu menggunakan algoritma yang disesuaikan untuk melakukan klasifikasi secara langsung dari objek multilabel (Tsoumakas dan Katakis, 2009). Algoritma ini merupakan adaptasi dari algoritma klasifikasi tunggal seperti KNN pada *lazy learning.* Beberapa algoritma yang ditawarkan pada adaptasi algoritma diantaranya ML-KNN, ML-RBF, ML-C4.5, ML-Naïve Bayes (Zhang dan Zhou, 2014). Pada penelitian ini pendekatan yang digunakan adalah pendekatan transformasi menggunakan *Label Powerset* (LP) dengan algortima *K-Nearest Neighbor* (KNN) serta pendekatan adaptasi algoritma *Multilabel K-Nearest Neighbor* (ML-KNN).

Transformasi *Label Powerset* (LP) merupakan metode kombinasi label yang merubah masalah multilabel menjadi *single label* dengan *multiclass*. Cara kerja *Label Powerset* adalah tiap kombinasi genre yang ada dijadikan sebagai kategori genre baru dan digunakan pada proses pelatihan (Tsoumakas dan Katakis, 2009). Ilustrasi transformasi *Label Powerset* ditampilkan pada Gambar 2.2 berikut ini.



Gambar 2.2 Ilustrasi Transformasi *Label Powerset*

Metode transformasi *Label Powerset* (LP) dinilai lebih efektif karena memperhatikan hubungan antar genre. LP merubah klasifikasi multilabel menjadi klasifikasi *multiclass* dengan menambahkan kategori genre baru untuk tiap kombinasi genre yang unik. Algoritma *Label Powerset* didefinisikan sebagai berikut:

1. Tiap kombinasi genre pada *dataset* multilabel dijadikan kategori genre baru, sehingga *dataset* berubah menjadi *single label multiclass*.
2. Klasifikasi menggunakan metode *single label multiclass*.
3. Hasil prediksi dari *dataset multiclass* kemudian diubah menjadi hasil klasifikasi multilabel (Herrera dkk., 2016).

2.1.1 Text Preprocessing

Text mining adalah suatu langkah dari analisis teks yang dilakukan secara otomatis oleh komputer untuk mencari informasi yang berkualitas dari suatu rangkaian teks yang terangkum dalam sebuah dokumen (Han dan Kamber, 2006). Ide awal pembuatan *text mining* adalah untuk menemukan pola-pola informasi yang dapat digali dari suatu teks yang tidak terstruktur. *Text mining* melibatkan *text preprocessing* seperti kategorisasi teks, ekstrasi informasi, dan ekstrasi kata (Feldman dan Sanger, 2007). *Text preprocessing* merupakan tahapan awal dalam *text mining*, dimana file teks mentah akan dirubah menjadi rangkai unit bahasa yang sangat jelas (Herbrich dan Graepel, 2010). *Text preprocessing* digunakan untuk mengubah bentuk dokumen menjadi data yang terstruktur sesuai kebutuhannya supaya dapat diolah lebih lanjut dalam proses *text mining*. Tahapan *preprocessing* dalam klasifikasi bertujuan untuk meningkatkan akurasi klasifikasi data. Berikut merupakan tahapan dari *text preprocessing.*

a. *Case Folding*

   Case folding adalah tahapan yang berfungsi untuk mengubah *font*, serta mengubah semua huruf menjadi huruf kecil (*lowercase*). Langkah ini merupakan langkah dasar yang paling banyak digunakan dalam *natural language processing* (Lutfi dkk., 2018).

b. *Data Cleaning*
   Data Cleaning adalah tahapan membersihkan data teks dari hal-hal yang tidak dibutuhkan seperti angka, simbol, dan tanda baca.
c. *Lemmatizing*
   Tahapan *Lemmatizing* adalah mengubah kata menjadi kata dasar sesuai kamus. Pada tahap ini dilakukan proses mengubah berbagai bentuk kata ke dalam suatu representasi yang sama.
d. *Stopwords Removal*
   Stopwords Removal adalah tahapan pengambilan kata-kata yang penting dengan membuang kata yang berada di dalam s*topwords*. *Stopwords* adalah kata umum yang biasa muncul dalam jumlah besar dan dianggap tidak memiliki makna.

2.1.2 Pembobotan Kata

Pembobotan kata yang lazim digunakan adalah TF-IDF (*Term Frequency-Inverse Document Frequency*). TF-IDF adalah suatu cara untuk memberikan bobot hubungan suatu kata (*term*) terhadap dokumen. TF berfungsi menghitung jumlah sebuah kata dalam suatu berkas. Dalam TF, semakin besar kemunculan suatu kata dalam dokumen maka semakin besar pula bobotnya. IDF mengurangi kata dengan melihat frekuensi kemunculan kata pada suatu dokumen. Tujuan dari TF-IDF adalah untuk menemukan jumlah kata yang diketahui (TF) setelah dikalikan dengan be-berapa banyak dokumen dimana suatu kata tersebut muncul (IDF). Berikut merupakan rumus TF-IDF (Zhang dkk., 2008).

$$IDF_j = \log\left(\frac{n}{df_j}\right) \tag{2.1}$$

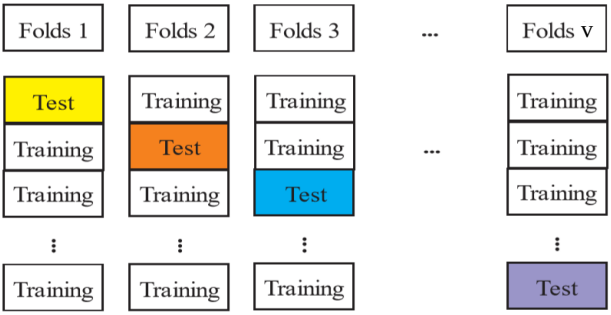
$$w_{i,j} = TF_{i,j} \times IDF_j \tag{2.2}$$

dimana $w_{i,j}$ adalah bobot kata ke-*j* pada sinopsis ke-*i*, *n* adalah jumlah seluruh sinopsis, $TF_{i,j}$ adalah jumlah kemunculan kata ke-

j pada sinopsis ke-*i*, dan df_j adalah jumlah dokumen yang mengandung kata-*j* (Zhang dkk., 2008).

2.1.3 K-fold Cross Validation

K-fold cross validation adalah sebuah teknik yang digunakan untuk mempartisi data menjadi data *training* dan data *testing*. *K-fold cross validation* secara berulang membagi data menjadi data *training* dan data *testing*, dimana setiap data memiliki kesempatan menjadi data *testing* (Gokgoz dan Subasi, 2015). Metode ini dapat mengurangi bias yang terjadi dalam pengambilan sampel. Berikut merupakan ilustrasi pembagian data menggunakan *K-fold cross validation* dimana v merupakan besar angka partisi data yang digunakan untuk pembagian *training testing*.



Gambar 2.3 Ilustrasi Pembagian Data

2.1.4 *K-Nearest Neighbor* (KNN)

Metode *K-Nearest Neighbor* (KNN) merupakan salah satu pendekatan yang sederhana untuk diimplementasikan. Metode KNN memiliki tingkat efisiensi yang tinggi dan dalam beberapa kasus memberikan tingkat akurasi yang tinggi dalam hal peng-klasifikasian (Hamamoto dkk., 1997). Prinsip kerja metode KNN adalah melakukan klasifikasi berdasarkan kedekatan lokasi (jarak) suatu data dengan data yang lain (Prasetyo, 2012). Dekat atau jauhnya lokasi (jarak) bisa dihitung melalui besaran jarak yang telah ditentukan yakni jarak *Euclidean*, jarak *Minkowski*, dan

jarak *Mahalanobis*. Namun dalam penerapannya seringkali menggunakan jarak *Euclidean* karena memiliki tingkat akurasi yang tinggi (Dasarathy, 1990). Jarak *Euclidean* memperlakukan semua peubah adalah bebas (tidak berkorelasi). Jarak *Euclidean* adalah besarnya jarak suatu garis lurus yang menghubungkan antar objek. Rumus jarak *Euclidean* adalah sebagai berikut (Yang dan Liu, 1999):

$$D = \sqrt{\sum_{j=1}^{f} (x_{aj} - x_{bj})^2} \tag{2.3}$$

Keterangan:

D : jarak *Euclidian*

f : jumlah varibel bebas

x_{aj} : data ke-*j* pada data pertama

x_{bj} : data ke-*j* pada data kedua

Dalam metode KNN, ada beberapa hal penting yang mempengaruhi kinerja klasifikasi diantaranya pemilihan nilai k (jumlah tetangga terdekat). Tahapan algoritma KNN diuraikan sebagai berikut.

1. Menghitung jarak antara data *training* dan data yang ingin diketahui genrenya menggunakan jarak *Euclidian* kemudian mengurutkan dari jarak yang terdekat,
2. Menentukan jumlah k (*nearest neighbor*) yang akan digunakan untuk penentuan genre kelas. Nilai k ditentukan oleh pengguna baik dengan metode coba-coba atau menggunakan metode lainnya seperti metode optimasi.
3. Menentukan genre kelas data uji berdasarkan genre dengan suara mayoritas sesuai langkah 3.

2.1.5 *Multilabel K-Nearest Neighbor* (ML-KNN)

Multilabel *K-Nearest Neighbor* adalah adaptasi algoritma dari *K-Nearest Neighbor* untuk kasus multilabel. Algoritma ini ditemu-kan oleh Zhang & Zhou (2018) dan menjadi algoritma multilabel "*lazy learning*" pertama. Fase pertama metode ini adalah proses perhitungan *prior probabilities* dan *posterior*

probabilities. Proses ini dihitung hanya dengan menggunakan data *training*. Metode KNN diimplementasikan pada tahap *posterior probabilities distribution*. Prediksi probabilitas dilakukan dengan memperhatikan aturan Bayesian.

Algoritma ML-KNN dimulai dengan membangun model yang terdiri dari dua bagian informasi (Herrera dkk., 2016):

1. Probabilitas prior untuk setiap genre, yaitu jumlah kemunculan tiap genre dibagi dengan total data. Probabilitas prior didefinisikan sebagai berikut.

$$P(H_1^l) = \frac{s + \sum_{i=1}^{n} \vec{y}_{x_i}}{s \times 2 + n} \tag{2.4}$$

$$P(H_0^l) = 1 - P(H_1^l) \tag{2.5}$$

Keterangan:

H_1^l : Kejadian ketika x_i memiliki genre *l*

H_0^l : Kejadian ketika x_i memiliki genre bukan *l*

n : Jumlah data

$\vec{y}_{x_i}$: Set genre data ke-*i*

s : Parameter *smoothing* (bernilai 1)

2. Probabilitas kondisional untuk setiap label, yaitu proporsi data *x* dengan genre yang dipertimbangkan, dimana k *nearest neighbor* memiliki genre yang sama. Probabilitas kondisional didefinisikan sebagai berikut.

$$P(\mathrm{E}_p^l \mid H_1^l) = \frac{s + c[p]}{s \times (k+1) + \sum_{p=0}^{k} c[p]} \tag{2.6}$$

$$P(\mathrm{E}_p^l \mid H_0^l) = \frac{s + c'[p]}{s \times (k+1) + \sum_{p=0}^{k} c'[p]} \tag{2.7}$$

Keterangan:

E_p^l : kejadian dimana diantara *k* terdapat *p* tetangga yang memiliki genre *l*

$c[p]$: jumlah data dengan genre l dimana diantara k tetangga terdapat p tetangga dengan genre l

n : Jumlah data

Probabilitas ini dihitung secara independen untuk tiap genre. Setelah proses pelatihan selesai, maka selanjutnya *classifier* dapat memprediksi genre data baru dengan langkah sebagai berikut.

1. Menghitung k tetangga terdekat menggunakan jarak *Euclidian* untuk mengukur jarak antara data *testing* dengan data *training.*
2. Masing-masing genre yang ada pada k tetangga terdekat digunakan untuk menghitung probabilitas *maximum a posteriori* (MAP) berdasarkan probabilitas kondisional yang telah didapatkan sebelumnya.
3. Menghitung *maximum a posteriori* (MAP) untuk menentukan set genre baru. Probabilitas tiap genre baru diurutkan sehingga menghasilkan peringkat genre.

Diberikan variabel x dan genre terkaitnya y, dan k *nearest neighbor* dipertimbangkan dalam metode ML-KNN. Diberikan $\vec{y}_x$ sebagai vektor label dari x, dimana komponen ke-l $\vec{y}_x(l)$ bernilai 1 jika $l \in y$ dan 0 lainnya. Sebagai tambahan, diberikan $N(x)$ menunjukkan himpunan k tetangga terdekat dari x yang teridentifikasi pada data *training.* Maka, berdasarkan set genre tetangganya, vektor perhitungan keanggotaan dapat didefinisikan sebagai berikut (Zhang dan Zhou, 2007).

$$\vec{C}_x(l) = \sum_{\alpha \in N(x)} \vec{y}_\alpha(l) \tag{2.8}$$

dengan $\vec{C}_x(l)$ adalah banyaknya tetangga dari x yang masuk pada genre ke-l. Untuk setiap variabel t pada data *testing*, ML-KNN mengidentifikasi k tetangga terdekatnya pada data *training.* Diberikan H_1^l merupakan kejadian ketika t mempunyai label l, sedangkan H_0^l merupakan kejadian ketika t mempunyai label yang bukan l. Selanjutnya, diberikan $E_p^l (p \in \{0,1,...,k\})$ menunjukkan kejadian dimana diantara k tetangga terdekat dari t

terdapat *p*-variabel yang memiliki label *l*. Maka, berdasarkan vektor perhitungan keanggotaan $\vec{C}_x(l)$, vektor kategori $\vec{y}_t$ ditetapkan berdasarkan *maximum a posteriori principle*. Vektor kategori $\vec{y}_t$ didefinisikan sebagai berikut (Zhang dan Zhou, 2007).

$$\vec{y}_t(l) = \arg\max_{b\in\{0,1\}} P(H_b^l \mid E^l_{\vec{C}_x(l)}) \tag{2.9}$$

menggunakan *bayesian rule*, persamaan 2.4 dapat dituliskan menjadi:

$$\begin{aligned}\vec{y}_t(l) &= \arg\max_{b\in\{0,1\}} \frac{P(H_b^l)\,\mathrm{P}(E^l_{\vec{C}_x(l)} \mid H_b^l)}{\mathrm{P}(E^l_{\vec{C}_x(l)})} \\ &\propto \arg\max_{b\in\{0,1\}} P(H_b^l)\,\mathrm{P}(E^l_{\vec{C}_x(l)} \mid H_b^l)\end{aligned} \tag{2.10}$$

2.2 Evaluasi Hasil Klasifikasi

Tahapan evaluasi adalah tahapan untuk mengetahui tingkat akurasi dan kinerja dari hasil klasifikasi. Pengukuran ketepatan klasifikasi dilakukan untuk melihat performa klasifikasi yang telah dilakukan. Evaluasi performa klasifikasi multilabel berbeda dengan klasifikasi *single label*. Pada klasifikasi multilabel, evaluasi menjadi lebih rumit karena hasil klasifikasi bisa menjadi benar semua, benar sebagian, maupun salah semua (Boutell dkk., 2004).

Hamming loss adalah ukuran yang paling umum digunakan untuk mengevaluasi hasil klasifikasi multilabel. *Hamming loss* merupakan ukuran banyaknya label yang salah dibanding dengan jumlah seluruh label yang ada. Semakin kecil *hamming loss* maka semakin baik hasil klasifikasi. Nilai *hamming loss* berada pada rentang 0 sampai 1. *Hamming loss* didefinisikan sebagai berikut.

$$\text{Hamming Loss} = \frac{1}{n}\frac{1}{l}\sum_{i=1}^{n}\left|y_i \Delta \hat{y}_i\right| \tag{2.11}$$

Keterangan:

L : Jumlah genre

n : Jumlah data

$|y_i \Delta \hat{y}_i|$: Perbedaan simetris antara set genre asli y_i dengan set genre prediksi $\hat{y}_i$

Sebagai contoh perhitungan nilai *hamming loss*, diberikan 5 data dengan genre asli dan genre prediksinya ditunjukkan pada Tabel 2.1 sebagai berikut.

Tabel 2. 1 Contoh Perhitungan *Hamming Loss*

Film	Genre Asli y_i	Genre Prediksi $\hat{y}_i$	$\lvert y_i \Delta \hat{y}_i \rvert$
1	0,1,1,0,0	1,1,1,0,0	1
2	1,1,0,0,0	1,0,0,0,0	1
3	0,1,1,0,1	0,1,0,0,1	0
4	0,1,0,1,0	1,0,0,1,1	3
5	1,1,1,0,0	0,1,1,1,0	2

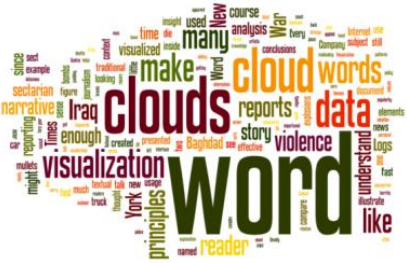
$$Hamming\ Loss = \frac{1}{n}\frac{1}{l}\sum_{i=1}^{n}|y_i \Delta \hat{y}_i|$$

$$= \frac{1}{5}\frac{1}{5}(1+1+0+3+2) = 0,28$$

Maka didapatkan nilai *hamming loss* sebesar 0,28 atau 28%, artinya terdapat 28% bagian genre yang diprediksi salah.

2.3 *Word Cloud*

Word cloud merupakan salah satu metode visualisasi dokumen teks yang sering digunakan. *Word cloud* adalah representasi grafis dari dokumen teks dengan melakukan plotting kata-kata yang sering muncul kedalam ruang dua dimensi. Melalui *word cloud*, dapat diketahui seberapa besar frekuensi dari kata yang muncul melalui besar kecilnya ukuran huruf kata tersebut. Semakin besar ukuran kata, maka semakin besar frekuensi kata tersebut muncul dalam dokumen (Castella dan Sutton, 2013).



Gambar 2. 4 Ilustrasi *Word Cloud*

2.4 Genre Film

Kamus Besar Bahasa Indonesia mengartikan genre sebagai jenis, tipe, atau kelompok sastra atas dasar bentuknya. Genre film adalah bentuk, kategori atau klasifikasi tertentu dari beberapa film yang memiliki kesamaan bentuk, latar, tema, suasana. Menurut Tim Dirks (2010), genre film utama yaitu *action, adventure, comedy, crime, drama, epics/historical, horror, musical, science-fiction, war,* dan *western.* Kemudian dari genre film utama tersebut dapat dikembangkan menjadi beberapa sub genre seperti *romance, superhero, fantasy, mystery,* dst.

(Halaman ini sengaja dikosongkan)

BAB III
METODE PENELITIAN

3.1 Sumber Data

Data yang digunakan dalam penelitian tugas akhir ini adalah data sekunder yang diperoleh dari situs The Movie Database (TMDb). Data diambil pada tanggal 5 Mei tahun 2020 dari situs https://www.themoviedb.org/ menggunakan TMDb API. Data yang digunakan berupa teks sinopsis dan genre film. Jumlah data yang diambil sebanyak 1846 film pada kategori 'Most Popular' di situs TMDb. Genre yang digunakan adalah *action*, *adventure*, *comedy*, *drama*, dan *thriller*. Data sinopsis dan genre film dapat dilihat pada Lampiran 1.

3.2 Variabel Penelitian

Variabel penelitian yang digunakan adalah bobot tiap kata yang dihitung menggunakan TF-IDF. Variabel penelitian tersebut tercantum pada Tabel 3.1.

Tabel 3.1 Variabel Penelitian

Variabel	Keterangan	Skala
X_j	Bobot kata yang muncul pada sinopsis	Rasio
Y_1	Genre *Drama*	Nominal
Y_2	Genre *Action*	Nominal
Y_3	Genre *Adventure*	Nominal
Y_4	Genre *Thriller*	Nominal
Y_5	Genre *Comedy*	Nominal

Struktur data setelah dilakukan *text preprocessing* dan penghitungan bobot kata menggunakan TF-IDF dalam penelitian ini terdiri dari variabel prediktor yaitu bobot kata yang ada pada sinopsis dan variabel respon yaitu genre film. Struktur data pada penelitian ini seperti pada Tabel 3.2.

Tabel 3. 2 Struktur Data Penelitian

No	X_1	X_2	…	X_j	Y_1	Y_2	…	Y_5
1	$X_{1,1}$	$X_{1,2}$	…	$X_{1,j}$	$Y_{1,1}$	$Y_{1,2}$	…	$Y_{1,5}$
2	$X_{2,1}$	$X_{2,2}$	…	$X_{2,j}$	$Y_{2,1}$	$Y_{2,2}$	…	$Y_{2,5}$
…	…	…	…	…	…	…	…	…
1846	$X_{1846,1}$	$X_{1846,2}$	…	$X_{1846,j}$	$Y_{1846,1}$	$Y_{1846,2}$	…	$Y_{1846,5}$

3.3 Langkah Penelitian

Langkah penelitian yang digunakan dalam tugas akhir ini adalah sebagai berikut.

1. Mengumpulkan data sinopsis dengan cara melakukan *crawling* dari situs TMDb (The Movie Databes) menggunakan TMDb API dengan tahapan sebagai berikut.
   a. Membuat akun di situs TMDb dengan alamat *website* www.themoviedb.org
   b. Memilih menu *setting* kemudian menu ‘API’ dan menekan pilihan ‘*request API key*’. Kemudian melengkapi formulir hingga mendapatkan API *key.*
   c. Setelah mendapatkan API *key*, masuk ke halaman situs developers.themoviedb.org. Kemudian memilih menu ‘Get Popular’ pada menu ‘Movies’.
   d. Mengumpulkan data sinopsis dan genre film dengan bantuan *software* Python dengan *syntax* di Lampiran 6 menggunakan *package* json untuk mengambil data 2000 film secara otomatis dengan cara memasukkan API *key* yang didapatkan sebelumnya.
2. Melakukan *text preprocessing* pada sinopsis film dengan bantuan *software* Python menggunakan *package* Natural Language Toolkit (NLTK) dengan *syntax* pada Lampiran 7. Berikut tahapan *text preprocessing* yang dilakukan.

a. *Case folding*, yaitu mengubah kata menjadi huruf kecil.
b. *Data Cleaning*, yaitu menghapus angka, simbol, dan tanda baca.
c. *Lemmatizing*, yaitu mengubah kata menjadi kata dasar.
d. *Stopwords Removal*, yaitu menghapus kata yang tidak dibutuhkan (*stopwords*).

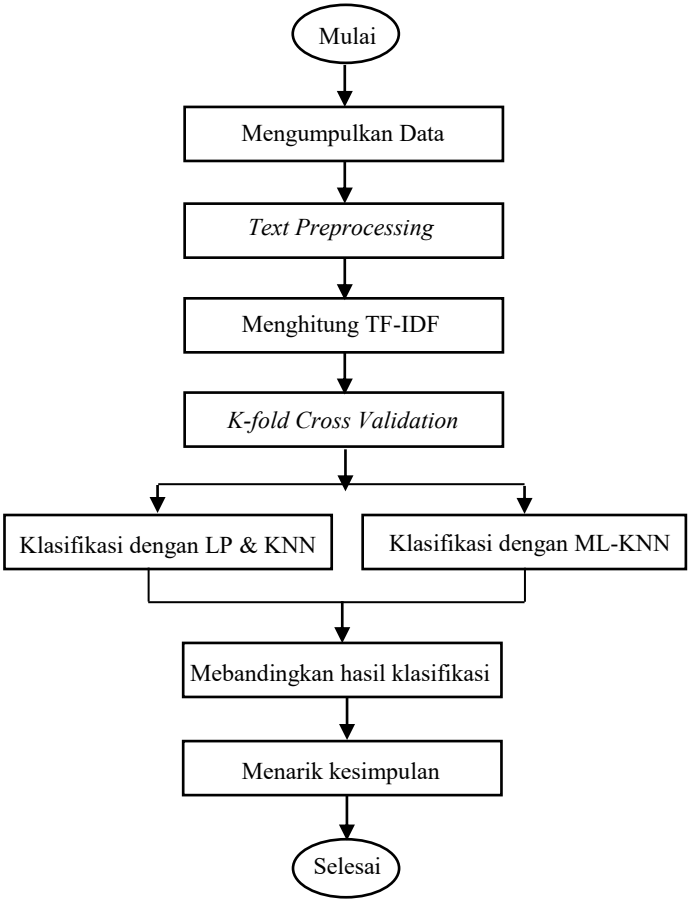
3. Melakukan pembobotan terhadap kata hasil *preprocessing* menggunakan TF-IDF dengan cara menghitung frekuensi tiap kata kemudian menghitung bobot kata sesuai persamaan 2.2. TF-IDF dihitung dengan bantuan *software* Python menggunakan *package* scikit-learn dengan *syntax* pada Lampiran 9.
4. Membagi data menjadi *training* dan *testing* dengan *K-fold Cross-Validation* dengan jumlah *fold* sebanyak 5. Tahap ini dilakukan dengan bantuan *software* Python menggunakan *package* scikit-learn dengan *syntax* di Lampiran 10. Algoritma KCV adalah membagi data menjadi 5 bagian, kemudian dari masing-masing bagian data dibagi lagi menjadi data *training* sebanyak 80% dan *testing* 20%.
5. Melakukan klasifikasi dengan menggunakan transformasi *Label Powerset* dan metode klasifikasi KNN dengan bantuan *software* Python menggunakan *package* scikit-multilearn dengan *syntax* pada Lampiran 10. Algoritma metode LP-KNN didefinisikan sebagai berikut.
   a. Merubah setiap kombinasi genre menjadi kategori genre baru.
   b. Melakukan lasifikasi dataset secara *multiclass* menggunakan KNN dengan tahapan sebagai berikut:
      - Menghitung jarak *Euclidian* antara data *training* dan *testing* menggunakan persamaan 2.3 dan mengurutkan dari jarak yang terdekat.
      - Menentukan k (*nearest neighbor*).

- Menentukan genre baru dari mayoritas genre pada *nearest neighbor*.

c. Hasil klasifikasi kemudian diubah menjadi hasil klasifikasi multilabel.
d. Menghitung nilai *hamming loss* menggunakan persamaan 2.11 untuk mengevaluasi hasil klasifikasi.

6. Melakukan klasifikasi genre film menggunakan metode ML-KNN dengan bantuan *software* Python menggunakan *package* scikit-multilearn dengan *syntax* pada Lampiran 11. Algoritma ML-KNN ditunjukkan sebagai berikut.
   a. Menghitung probabilitas prior dan probabilitas kondisional untuk setiap genre.
   b. Menghitung jumlah k tetangga terdekat dan jarak *Euclidian* antara data *training* dengan data *testing.*
   c. Menghitung *maximum a posteriori* (MAP) untuk menentukan set genre baru berdasarkan genre yang ada pada k *nearest neighbor*dan probabilitas kondisional.
   d. Menghitung nilai *hamming loss* untuk mengevaluasi hasil klasifikasi.
7. Membandingkan kinerja klasifikasi berdasarkan nilai *hamming loss* yang paling kecil dari masing-masing metode.
8. Melakukan visualisasi hasil klasifikasi dengan membuat *word cloud* untuk setiap kombinasi genre menggunakan bantuan *software* Python dengan *syntax* terlampir pada Lampiran 5.
9. Memberikan kesimpulan dan saran terkait hasil analisis yang telah dilakukan.

Langkah-langkah analisis secara umum digambarkan pada diagram alir pada Gambar 3.1.



Gambar 3.1 Diagram Alir Penelitian

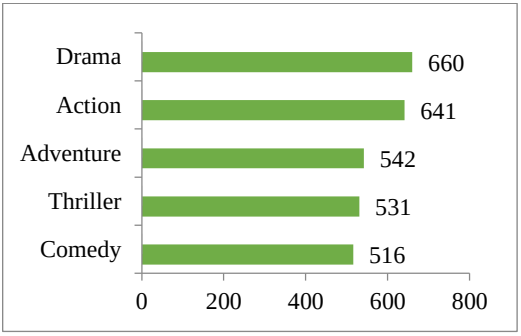
(Halaman ini sengaja dikosongkan)

BAB IV
ANALISIS DAN PEMBAHASAN

Pada penelitian ini dilakukan klasifikasi genre film secara multilabel. Penelitian ini membandingkan dua metode yaitu transformasi *Label Powerset* (LP) dengan metode klasifikasi *K-Nearest Neighbor* (KNN) dan *Multilabel K-Nearest Neighbor* (ML-KNN). Kebaikan hasil klasifikasi diukur berdasarkan nilai *hamming loss*, semakin kecil nilai *hamming loss* maka semakin baik.

4.1 Karakteristik Data

Data pada penelitian ini adalah sinopsis dan genre film yang diperoleh dari situs *rating* film The Movie Database (TMDb) dengan menggunakan TMDb API. Jumlah data awal yang diambil adalah 2000 film. Jumlah genre yang ada pada situs TMDb adalah sebanyak 19 genre. Penelitian ini menggunakan 5 genre dengan film paling banyak yaitu *drama, action, adventure, thriller*, dan *comedy* sehingga total data yang digunakan pada penelitian ini sebanyak 1846. Jumlah film pada masing-masing genre secara independen ditampilkan pada Gambar 4.1 dimana satu film dapat memiliki lebih dari satu genre.

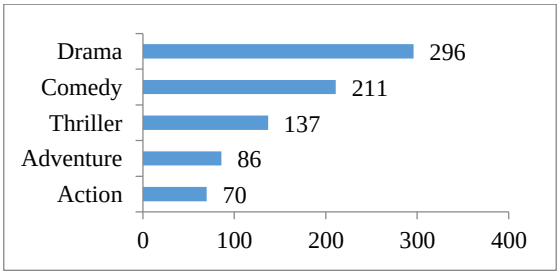


Gambar 4.1 Jumlah Film Tiap Genre

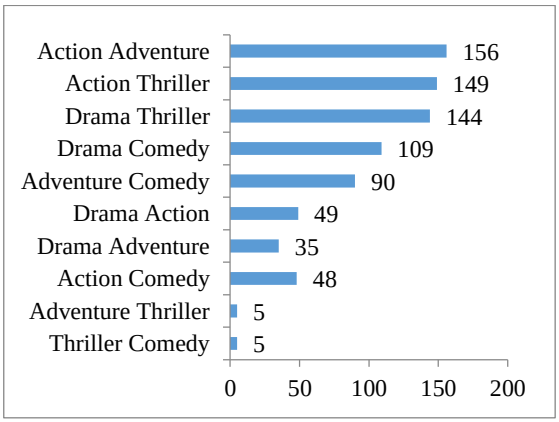
Genre *drama, action, adventure, thriller*, dan *comedy* adalah genre yang paling banyak muncul dibandingkan genre lainnya.

Jumlah film dengan genre tersebut relatif sama, berada di kisaran 500 – 600 film dimana genre paling banyak adalah *drama* dengan 660 film dan yang paling sedikit adalah genre *comedy* dengan 516 film. Setiap film dapat memiliki satu sampai empat genre. Pada penelitian ini terdapat 28 kombinasi genre yang terbentuk dari 5 genre yang ada, dengan rincian satu genre sebanyak 5, kombinasi dua genre sebanyak 10, kombinasi tiga genre sebanyak 9, dan kombinasi empat genre sebanyak 4 kombinasi. Jumlah film pada masing-masing kombinasi genre ditampilkan pada Gambar 4.2 sebagai berikut.

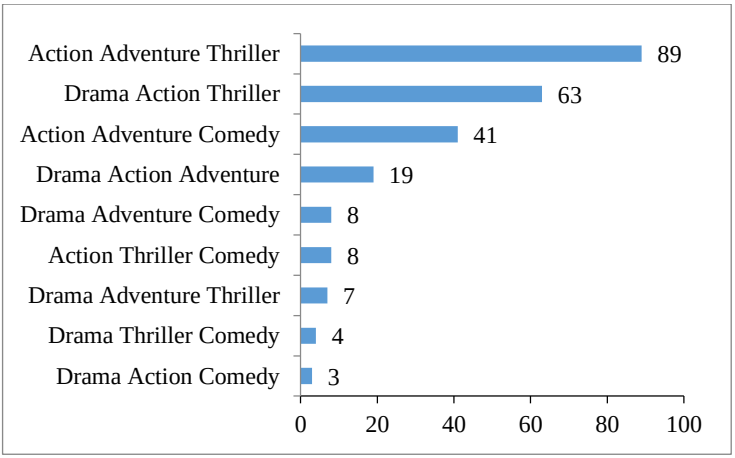
(a)



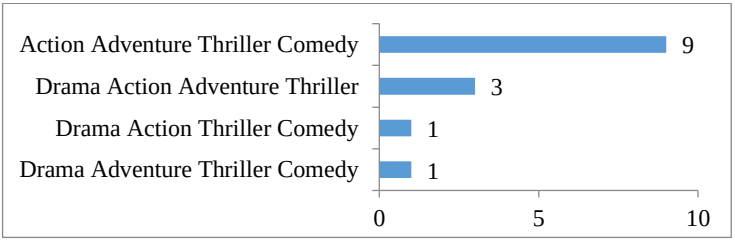
(b)



(c)



(d)



Gambar 4.2 Jumlah Film dengan Kombinasi (a) 1 Genre (b) 2 Genre (c) 3 Genre (d) 4 Genre

Gambar 4.2 menunjukkan jumlah film dengan masing-masing kombinasi genre. Total film dengan satu genre yaitu 800 film, kombinasi dua genre yaitu sebanyak 790 film, kombinasi tiga genre sebanyak 242 film, serta kombinasi 4 genre sebanyak 14 flm. dua genre. Genre paling banyak adalah genre *drama* dengan 296 film, sedangkan kombinasi dua genre paling banyak adalah *action-adventure* dengan 156 film, untuk kombinasi tiga genre yang terbanyak adalah *action-adventure-thriller* sebanyak 89 film, dan kombinasi empat genre paling banyak adalah *action-adventure-thriller-comedy* dengan 9 film.

4.2 *Text Preprocessing*

Text preprocessing dilakukan pada data sinopsis film untuk membersihkan data dari hal-hal yang tidak diperlukan dalam penelitian. Jumlah kata sebelum *text preprocessing* terdapat sebanyak 553.511 kata. Tahapan *text preprocessing* yakni *case folding, data cleaning, lemmatizing* dan *stopwords removal. Text preprocessing* dilakukan menggunakan bantuan *software* Python *package* Natural Language Toolkit (NLTK) dengan *syntax* pada Lampiran 7. Tahapan yang pertama adalah *case folding* yaitu tahapan yang berfungsi untuk mengubah semua huruf pada data sinopsis menjadi huruf kecil (*lowercase*). Hasil setelah dilakukan *case folding* ditunjukkan pada Tabel 4.1 sebagai berikut.

Tabel 4.1 Contoh Proses *Case Folding*

Sebelum *Case Folding*	**Setelah *Case Folding***
[RUMORED] The sixth and final installment of the Scary Movie franchise that ignores the fifth film. A parody of Scream 4 and various other horror movies, sequels and reboots that were released between the late 2010's and 2020. Takes place 15 years after the fourth film.	[rumored] the sixth and final installment of the scary movie franchise that ignores the fifth film. a parody of scream 4 and various other horror movies, sequels and reboots that were released between the late 2010's and 2020. takes place 15 years after the fourth film.
Tyler Rake, a fearless mercenary who offers his services on the black market, embarks on a dangerous mission when he is hired to rescue the kidnapped son of a Mumbai crime lord…	tyler rake, a fearless mercenary who offers his services on the black market, embarks on a dangerous mission when he is hired to rescue the kidnapped son of a mumbai crime lord…
Sisterhood is tested, rivalries heat up and new bonds are formed when students go back to their performing arts school to compete for an all expense paid summer scholarship program to a prestigious Conservatory of Fine Arts.	sisterhood is tested, rivalries heat up and new bonds are formed when students go back to their performing arts school to compete for an all expense paid summer scholarship program to a prestigious conservatory of fine arts.

Tahapan selanjutnya yakni *data cleaning* yang merupakan proses membersihkan data sinopsis dari hal-hal yang tidak diperlukan dalam penelitian. Tahapan *data cleaning* yang dilakukan yaitu menghapus tanda baca, simbol, angka dan spasi pada data sinopsis. Hasil setelah dilakukan *data cleaning* ditampilkan pada Tabel 4.2 sebagai berikut.

Tabel 4.2 Contoh Proses Data Cleaning

Sebelum *Data Cleaning*	**Setelah *Data Cleaning***
[rumored] the sixth and final installment of the scary movie franchise that ignores the fifth film. a parody of scream 4 and various other horror movies, se-quels and reboots that were released between the late 2010's and 2020. takes place 15 years after the fourth film.	rumored the sixth and final installment of the scary movie franchise that ignores the fifth film a parody of scream and various other horror movies sequels and reboots that were released between the late s and takes place years after the fourth film
tyler rake, a fearless mercenary who offers his services on the black market, embarks on a da-ngerous mission when he is hired to rescue the kidnapped son of a mumbai crime lord…	tyler rake a fearless mercenary who offers his services on the black market embarks on a dangerous mission when he is hired to rescue the kidnapped son of a mumbai crime lord
sisterhood is tested, rivalries heat up and new bonds are formed when students go back to their performing arts school to compete for an all expense paid summer scholarship program to a prestigious conservatory of fine arts	sisterhood is tested rivalries heat up and new bonds are formed when students go back to their performing arts school to compete for an all expense paid summer scholarship program to a prestigious conservatory of fine arts

Setelah dilakukan proses *data cleaning* kemudian dilanjutkan ke tahapan *lemmatizing* untuk merubah kata menjadi kata dasar sesuai kamus. Kamus *lemmatizing* yang digunakan adalah *wordnetlemmatizer* dari Natural Language Toolkit (NLTK). Hasil *lemmatizing* ditampilkan pada Tabel 4.3.

Tabel 4. 3 Contoh Proses *Lemmatizing*

Sebelum *Lemmatizing*	**Setelah *Lemmatizing***
rumored the sixth and final install-ment of the scary movie franchise that ignores the fifth film a parody of scream and va-rious other horror movies sequels and reboots that were released between the late s and takes place years after the fourth film	rumored the sixth and final installment of the scary movie franchise that ignores the fifth film a parody of scream and va-rious other horror movie sequel and reboots that were released between the late s and take place year after the fourth film
tyler rake a fearless mercenary who offers his services on the black market embarks on a dangerous mission when he is hired to rescue the kidnapped son of a mumbai crime lord.	tyler rake a fearless mercenary who offer his service on the black market embarks on a dangerous mission when he is hired to rescue the kidnapped son of a mumbai crime lord.
sisterhood is tested rivalries heat up and new bonds are formed when students go back to their performing arts school to com-pete for an all expense paid sum-mer scholarship program to a prestigious conservatory of fine arts.	sisterhood is tested rivalry heat up and new bond are formed when student go back to their performing art school to compete for an all expense paid summer scholarship program to a prestigious conservatory of fine art.

Tahapan terakhir adalah *stopwords removal* yang merupakan proses menghapus kata-kata yang tidak relevan. Daftar *stopwords* yang digunakan adalah *English Stopwords* dari Natural Language Toolkit (NLTK).

Tabel 4.4 Contoh Proses *Stopwords Removal*

Sebelum *Stopwords Removal*	**Setelah *Stopwords Removal***
rumored the sixth and final install-ment of the scary movie franchise that ignores the fifth film a parody of scream and va-rious other horror movie sequel and reboots that were released between the late s and take place year after the fourth film	rumored sixth final installment scary movie franchise ignores fifth film parody scream vari-ous horror movie sequel reboots released late take place year fourth film

Tabel 4.5 Contoh Proses *Stopwords Removal* (Lanjutan)

Sebelum *Stopwords Removal*	**Setelah *Stopwords Removal***
tyler rake a fearless mercenary who offer his service on the black market embarks on a dangerous mission when he is hired to rescue the kidnapped son of a mumbai crime lord	tyler rake fearless mercenary offer service black market embarks dangerous mission hired rescue kidnapped son mumbai crime lord
sisterhood is tested rivalry heat up and new bond are formed when student go back to their performing art school to compete for an all expense paid summer scholarship program to a prestigious conservatory of fine art.	sisterhood tested rivalry heat new bond formed student go back performing art school compete expense paid summer scholarship program prestigious conservatory fine art.

Jumlah kata pada data sinopsis setelah dilakukan *text preprocessing* adalah sebanyak 317.106 kata. Setelah melakukan *text preprocessing* kemudian dibentuk struktur data baru dimana masing-masing kata yang ada pada semua sinopsis menjadi variabel prediktor. Selanjutnya dihitung frekuensi kemunculan masing-masing kata pada tiap sinopsis yang akan digunakan untuk penghitungan bobot kata (TF-IDF). Perhitungan TF-IDF dilakukan dengan bantuan *software* Python menggunakan *package* scikit-learn dengan *syntax* pada Lampiran 9. Frekuensi kemunculan kata pada tiap sinopsis ditunjukkan pada Tabel 4.6.

Tabel 4.6 Frekuensi Kemunculan Kata dalam Sinopsis

Sinopsis	abandoned	…	life	…	world	…	zoo
1	0	…	0	…	0	…	0
2	0	…	0	…	0	…	0
3	0	…	0	…	0	…	0
4	0	…	1	…	2	…	0
5	0	…	0	…	0	…	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1843	0	…	0	…	0	…	0
1844	0	…	0	…	0	…	0
1845	0	…	2	…	0	…	0
1846	1	…	0	…	0	…	0

Tabel 4.6 menunjukkan perhitungan frekuensi kata pada tiap sinopsis film. Berdasarkan Tabel 4.6 dapat dilihat bahwa pada sinopsis ke-1846 kata ‘abandoned’ disebutkan sebanyak satu kali, kata ‘life’ disebutkan sebanyak satu kali pada sinopsis ke-4 dan dua kali pada sinopsis ke-1845, kata ‘world’ disebutkan sebanyak dua kali pada sinopsis ke-4. Selanjutnya menghitung IDF (*invers document frequency*) menggunakan persamaan 2.1 yang didapatkan dari log dikali perbandingan total sinopsis dengan jumlah sinopsis yang mengandung kata tersebut. Kemudian dari menghitung bobot tiap kata pada tiap sinopsis sesuai persamaan 2.2 dengan cara mengalikan frekuensi kemunculan kata dengan nilai IDF. Berikut hasil perhitungan bobot kata (TF-IDF) ditunjukkan pada Tabel 4.7 berikut.

Tabel 4.7 TF-IDF Kata dalam Sinopsis

Sinopsis	abandoned	…	life	…	world	…	zoo
1	0	…	0	…	0	…	0
2	0	…	0	…	0	…	0
3	0	…	0	…	0	…	0
4	0	…	0,074980	…	0,154724	…	0
5	0	…	0	…	0	…	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1842	0	…	0	…	0	…	0
1843	0	…	0	…	0	…	0
1844	0	…	0	…	0	…	0
1845	0	…	0.158151	…	0	…	0
1846	0,131233	…	0	…	0	…	0

Hasil perhitungan TF-IDF pada data sinopsis kemudian digunakan sebagai variabel prediktor untuk melakukan klasifikasi genre film secara multilabel menggunakan metode *K-Nearest Neighbor* (KNN) dengan transormasi *Label Powerset* (LP) dan *Multilabel K-Nearest Neighbor* (ML-KNN).

4.3 Klasifikasi Multilabel Genre Film

Klasifikasi genre film dilakukan menggunakan metode *K-Nearest Neighbor* (KNN) dengan transformasi *Label Powerset* (LP) dan metode *Multilabel K-Nearest Neighbor* (ML-KNN).

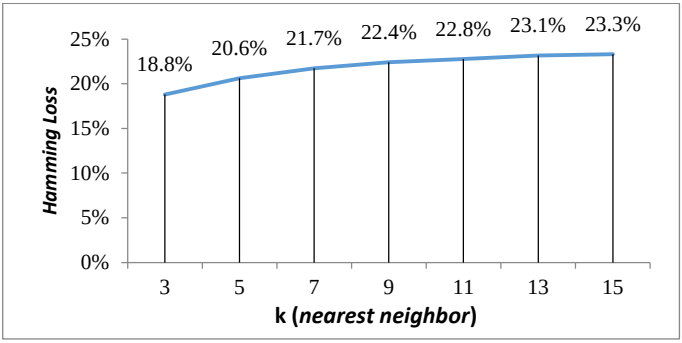
Pada penelitian ini digunakan *K-Fold Cross Validation* untuk mengurangi bias terkait pengambilan sampel dari data dengan cara membagi data ke dalam sejumlah bagian yang telah ditentukan atau disebut dengan *fold.* Jumlah *fold* yang digunakan pada penelitian ini adalah sebanyak 5 *fold*. Pada 5-*fold cross validation*, data dibagi menjadi ke dalam 5 *fold* lalu kemudian dibagi kembali menjadi data *training* dan data *testing* dengan perbandingan 80:20. Pembagian data *training* dan *testing* ini dilakukan dengan *software* Python menggunakan *package* scikit-learn dengan *syntax* di Lampiran 10. Kebaikan hasil klasifikasi diukur berdasarkan nilai *hamming loss.* Berikut pembahasan lebih lanjut untuk masing-masing metode.

4.2.1 Klasifikasi dengan Transformasi *Label Powerset*

Metode klasifikasi pertama yang akan digunakan pada penelitian ini untuk mengklasifikasikan genre film adalah *K-Nearest Neighbor* (KNN). KNN merupakan metode yang digunakan untuk klasifikasi secara *single label*, sehingga pada metode ini perlu dilakukan transformasi data dari multilabel menjadi *single label*. Transformasi yang digunakan pada penelitian ini adalah *Label Powerset* (LP) yaitu mengubah setiap kombinasi label menjadi *single label* dengan *multiclass.* Total kombinasi genre yang didapatkan adalah sebanyak 28 kombinasi. Kemudian dilakukan klasifikasi menggunakan KNN dengan jumlah k (*nearest neighbor*) yang digunakan adalah 3 sampai 15.

Algoritma KNN dimulai dengan menghitung jarak antar data menggunakan *Euclidian distance* berdasarkan persamaan 2.3 dan mengurutkan dari jarak terkecil. Selanjutnya dicari k *nearest neighbor* berdasarkan jarak *Euclidian* yang paling kecil dan menentukan genre baru berdasarkan mayoritas genre yang dimiliki oleh k *nearest neighbor.* Hasil klasifikasi KNN *multi-class* diubah menjadi set genre multilabel dan dievaluasi dengan cara menghitung nilai *hamming loss* berdasarkan persamaan 2.11. Semakin kecil nilai *hamming loss* maka semakin baik performa hasil klasifikasi. Klasifikasi LP-KNN dilakukan dengan bantuan *software* Python menggunakan *package* scikit-multilearn dengan

syntax pada Lampiran 10. Berikut merupakan nilai rata-rata *hamming loss* untuk masing-masing k ditampilkan pada Gambar 4.3.



Gambar 4.3 *Hamming Loss* LP-KNN

Rata-rata nilai *hamming loss* setiap *fold* untuk masing-masing jumlah k pada metode LP-KNN ditampilkan pada Gambar 4.3. Semakin kecil nilai *hamming loss* maka semakin baik hasil klasifikasi. Hasil performa pada Gambar 4.3 menunjukkan bahwa KNN dengan jumlah 3 *nearest neighbor* memberikan hasil paling baik dengan *error* yang minimum yaitu rata-rata *hamming loss* sebesar 18,79%. Artinya dari hasil klasifikasi terdapat 18,79% bagian genre film yang diprediksi salah. Terdapat tren naik pada performa hasil klasifikasi dengan k dari 3 sampai 15, semakin besar nilai k maka nilai rata-rata *hamming loss* semakin besar. Sehingga jumlah k yang digunakan pada metode KNN dengan transformasi *Label Powerset* (LP) adalah 3. Nilai *hamming loss* untuk tiap *folds* dilampirkan pada Lampiran 2.

4.2.2 Klasifikasi dengan Metode ML-KNN

Metode klasifikasi yang digunakan selanjutnya yaitu adaptasi algoritma dari KNN yaitu *Multilabel K-Nearest Neighbor* (ML-KNN). Metode ini merupakan adaptasi algoritma dari KNN yang sudah disesuaikan untuk mengklasifikasikan data multilabel. Pada penelitan ini perhitungan jarak yang digunakan untuk metode

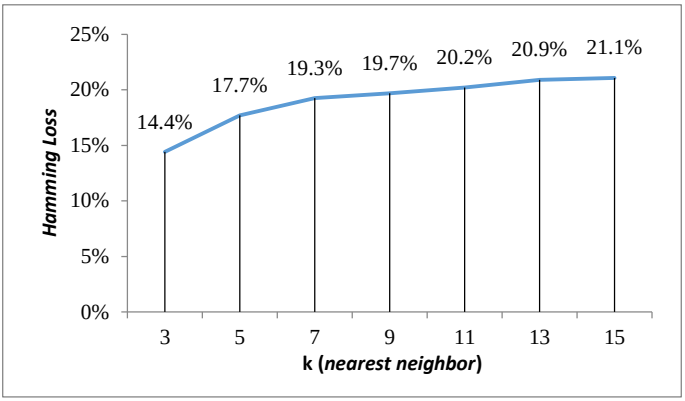
ML-KNN adalah *euclidian distance* dan jumlah k yang digunakan adalah 3 sampai 15. Algoritma ML-KNN dimulai dengan menghitung probabilitas prior untuk setiap genre berdasarkan persamaan 2.4 dan 2.5. probabilitas prior didapatkan dari jumlah film dengan genre *l* dibagi total seluruh film. Berikut hasil perhitungan probabilitas prior untuk tiap genre ditunjukkan pada Tabel 4.8.

Tabel 4. 8 Probabilitas Prior

Genre	Probabilitas Prior
Drama	$P(H_1^{drama}) = 0{,}358$
	$P(H_0^{drama}) = 0{,}642$
Action	$P(H_1^{action}) = 0{,}347$
	$P(H_0^{action}) = 0{,}653$
Adventure	$P(H_1^{adventure}) = 0{,}294$
	$P(H_0^{adventure}) = 0{,}706$
Thriller	$P(H_1^{thriller}) = 0{,}287$
	$P(H_0^{thriller}) = 0{,}713$
Comedy	$P(H_1^{comedy}) = 0{,}279$
	$P(H_0^{thriller}) = 0{,}721$

Tabel 4.8 menampilkan hasil perhitungan probabilitas prior yang didapatkan dari jumlah film dengan genre tertentu dibagi dengan total seluruh film. Berdasarkan Tabel 4.8 didapatkan peluang suatu film memiliki genre *drama* adalah 0,358 dan peluang suatu film memiliki genre bukan *drama* adalah 0,642, dan seterusnya untuk setiap genre. Setelah didapatkan probabilitas prior untuk masing-masing genre, langkah selanjutnya adalah menghitung *Euclidian distance* berdasarkan persamaan 2.3 untuk mencari k *nearest neighbor* untuk setiap data. Selanjutnya menghitung probabilitas kondisional untuk tiap genre yakni proporsi film dengan genre tertentu dimana terdapat k *nearest neighbor* yang memiliki

genre yang sama. Probabilitas kondisional dihitung menggunakan persamaan 2.6 dan 2.7. Setelah mendapatkan probabilitas prior dan probabilitas kondisional kemudian dapat dicari prediksi set genre baru menggunakan persamaan 2.10. Hasil klasifikasi dievaluasi dengan menghitung nilai *hamming loss* berdasarkan persamaan 2.11. Klasifikasi genre film metode ML-KNN dilakukan dengan bantuan *software* Python *package* scikit-multilearn dengan *syntax* pada Lampiran 11. Berikut nilai rata-rata *hamming loss* untuk tiap k ditampilkan pada Gambar 4.4.



Gambar 4.4 *Hamming Loss* ML-KNN

Gambar 4.4 menunjukkan performa klasifikasi untuk masing-masing k. Hasil performa pada Gambar 4.4 menunjukkan bahwa ML-KNN dengan k sebanyak 3 memberikan hasil terbaik dan *error* yang minimum yaitu rata-rata *hamming loss* sebesar 14,4%, artinya dari hasil klasifikasi terdapat 14,4% bagian genre film yang diprediksi salah. Sama seperti sebelumnya, untuk metode ML-KNN juga terdapat tren naik pada performa hasil klasifikasi dengan nilai k dari 3 sampai 15, semakin besar nilai k maka nilai *hamming loss* semakin besar. Sehingga jumlah k yang digunakan pada metode *Multilabel K-Nearest Neighbor* (ML-KNN) adalah 3. Nilai *hamming loss* untuk tiap *fold* dilampirkan pada Lampiran 3.

4.2.3 Perbandingan Performa Klasifikasi Antar Metode

Setelah didapatkan jumlah tetangga terbaik untuk kedua metode yaitu LP-KNN dan ML-KNN, dilakukan perbandingan kebaikan metode berdasarkan nilai *hamming loss* untuk memilih metode mana yang paling baik dalam mengklasifikasikan genre film secara multilabel. Pemilihan metode terbaik untuk klasifikasi genre film dilakukan dengan cara membandingkan nilai *hamming loss* data *testing* pada masing-masing *folds* untuk kedua metode LP-KNN dan ML-KNN dengan k sebanyak 3. Nilai *hamming loss* masing-masing *fold* ditampilkan pada Tabel 4.8.

Tabel 4.9 *Hamming Loss* tiap *Fold*

Fold	LP-KNN		ML-KNN	
	Train	**Test**	**Train**	**Test**
1	18,17%	32,65%	13,89%	30,11%
2	18,77%	30,73%	14,06%	28,62%
3	19,09%	31,54%	14,52%	29,16%
4	19,38%	32,79%	15,14%	29,54%
5	18,55%	32,52%	14,46%	29,16%
Average	18,79%	32,05%	**14,44%**	**29,32%**

Tabel 4.9 menunjukkan bahwa metode ML-KNN menghasilkan preforma lebih baik dengan *error* lebih kecil yaitu rata-rata *hamming loss* data *training* sebesar 14,44% dan data *testing* sebesar 29,32%, artinya terdapat 29,32% bagian genre film yang diprediksi salah. Sedangkan metode KNN dengan transformasi LP menghasilkan *error* yang sedikit lebih besar yaitu rata-rata *hamming loss* data *training* sebesar 18,79% dan data *testing* sebesar 32,05%, artinya terdapat 32,05% bagian genre film yang diprediksi salah. Maka dapat disimpulkan bahwa metode ML-KNN dengan k sebanyak 3 dapat mengklasifikasikan genre film secara multilabel lebih baik dibandingkan metode KNN dengan transformasi LP.

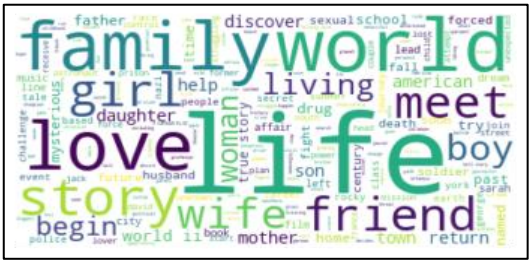
4.4 Visualisasi *Word Cloud*

Hasil klasifikasi genre film dengan metode terbaik yaitu ML-KNN dengan k sebanyak 3 dan kemudian divisualisasikan untuk melihat kata kunci tiap genre. Sinopsis film untuk tiap kombinasi

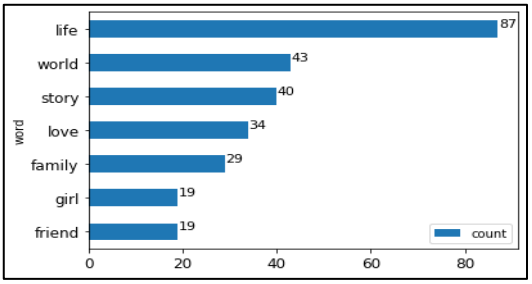
genre divisualisasikan menggunakan *word cloud* yang merupakan representasi grafis dari dokumen teks ke dalam ruang dua dimensi dengan melakukan *plotting* kata yang sering muncul. Semakin besar ukuran kata dalam *word cloud*, menunjukkan frekuensi kata tersebut paling banyak, sehingga dapat diketahui bahwa pengklasifikasian genre telah sesuai. Pembuatan *word cloud* dilakukan menggunakan bantuan *software* Python dengan *syntax* terlampir pada Lampiran 5.

Genre drama adalah genre pada film yang biasanya bercerita tentang permasalahan kehidupan, keluarga, maupun percintaan. Gambar 4.5 di bawah ini menunjukkan kata yang paling sering muncul pada genre drama adalah *life* sebanyak 87 kata. Kata terbanyak selanjutnya adalah *world, story, love, family*. Sehingga dapat dikatakan bahwa model dapat mengklasifikasikan film genre *drama* dengan baik karena kata-kata terebut sesuai dengan genre drama yang biasa menampilkan film yang bercerita tentang kehidupan, percintaan, dan keluarga.

(a)



(b)

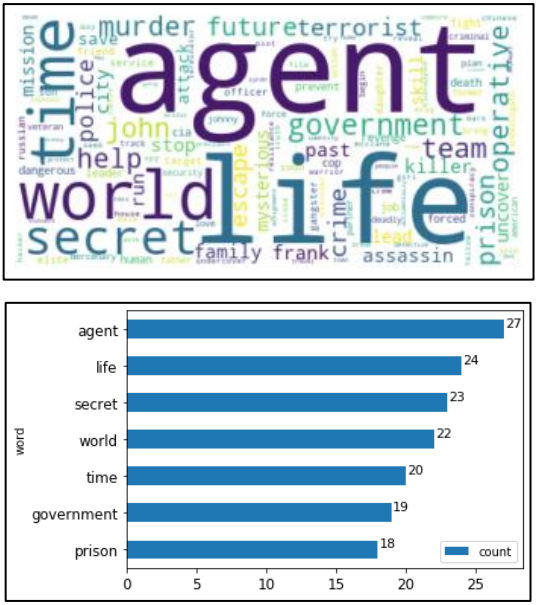


Gambar 4. 5 (a) *Word Cloud* (b) *Bar Chart* Genre *Drama*

Selanjutnya visualisasi *word cloud* untuk genre *action thriller* ditampilkan pada Gambar 4.6. Genre *action* merupakan genre film yang biasa menampilkan adegan aksi seperti berkelahi dan membunuh, sedangkan genre *thriller* adalah genre film yang biasa menampilkan adegan yang menegangkan dan mengejutkan. Gambar 4.6 menampilkan kata *agent, life, secret, world, time, government,* dan *prison* sebagai kata yang paling sering muncul pada genre *action thriller.* Kata *'agent'* pada film merujuk pada agen pembunuhan, agen polisi, dan lainnya. Model dapat meng-klasifikasikan film genre *action thriller* dengan baik karena kata-kata tersebut sesuai dengan genre *action thriller* yang biasa menampilkan film dengan adegan aksi yang menegangkan.

(a)

(b)

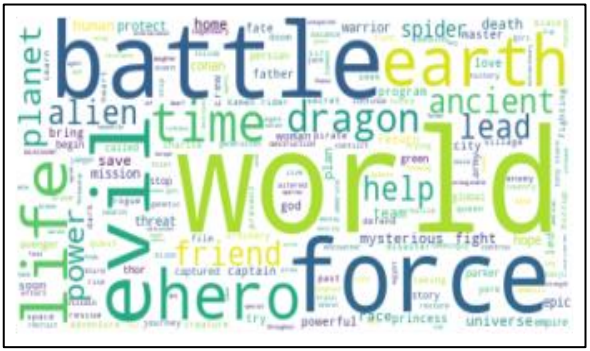


Gambar 4. 6 (a) *Word Cloud* (b) *Bar Chart* Genre *Action Thriller*

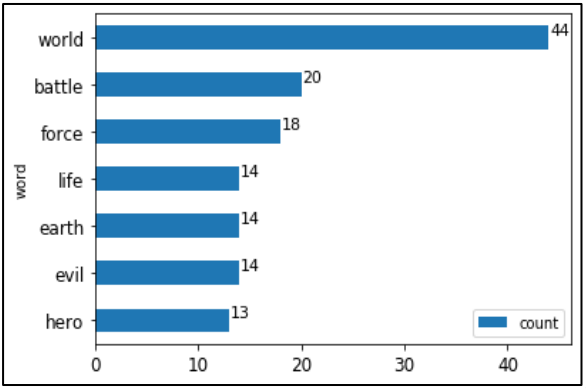
Genre *adventure* merupakan genre film yang berisi adegan petualangan. Visualisasi *word cloud* untuk genre *action adventure* ditampilkan pada Gambar 4.7. Berdasarkan Gambar 4.7 dapat

dilihat bahwa genre *action adventure* didominasi oleh kata *world* sebanyak 44 kata. Kata selanjutnya adalah *battle, force, life, earth, evil,* dan *hero.* Kata *world* dan *earth* identik dengan genre *adventure* yang biasa bercerita tentang petualangan di dunia khususnya di bumi. Kata *battle, force, evil,* dan *hero* identik dengan genre *action* yang biasa menampilkan adegan aksi seperti pertarungan. Model dapat mengklasifikasikan film genre *action adventure* dengan baik karena kata-kata tersebut sesuai dengan film genre *action adventure* yang biasanya bercerita tentang petualangan dengan adegan aksi.

(a)



(b)

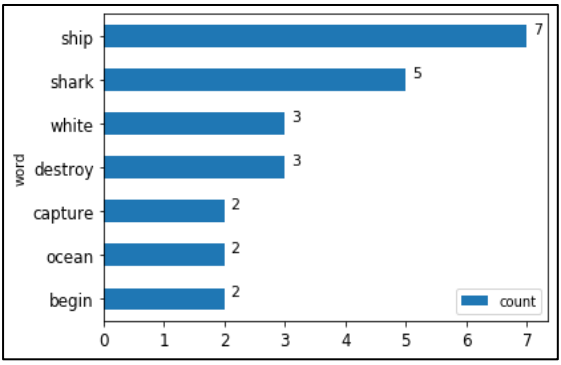


Gambar 4.7 (a) *Word Cloud* (b) *Bar Chart* Genre *Action Adventure*

Visualisasi kata kunci untuk genre film *adventure thriller* ditampilkan pada Gambar 4.8 di bawah ini. Genre film *adventure thriller* merupakan genre film yang bercerita tentang petualangan dengan adegan yang menegangkan. Gambar 4.8 menampilkan kata terbanyak untuk genre *adventure thriller* adalah *ship, shark, destroy, ocean.* Kata *ship* dan *ocean* identik dengan petualangan di laut, kata *shark* dan *destroy* identik dengan adegan yang menegangkan. Kata-kata tersebut sesuai dengan genre *adventure thriller* yang bercerita tentang petualangan dengan adegan yang menegangkan.

(a)

(b)



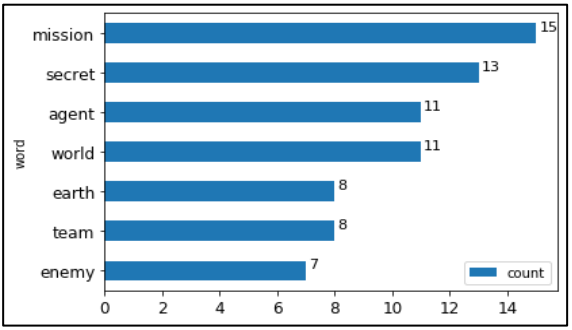
Gambar 4.8 (a) *Word Cloud* (b) *Bar Chart* Genre *Adventure Thriller*

Genre *action adventure thriller* merupakan gabungan antara adegan aksi, petualangan, serta adegan yang menegangkan atau mengejutkan. Gambar 4.9 merupakan visualisasi kata kunci untuk

genre *action adventure thriller*. Gambar 4.9 menunjukkan kata yang paling sering muncul adalah *mission, secret, agent, world, earth, team,* dan *enemy*. Kata *mission, agent, team, enemy* identik dengan adegan aksi yang menegangkan. Sedangkan kata *world, earth, secret* identik dengan cerita tentang petualangan yang menegangkan. Model dapat mengklasifikasikan film genre *action adventure thriller* dengan baik karena kata-kata tersebut sesuai dengan genre *action adventure thriller* yang biasanya banyak menceritakan petualangan yang penuh dengan adegan aksi yang menegangkan. Kata kunci dan visualisasi untuk kombinasi genre lainnya dapat dilihat pada Lampiran 4 dan Lampiran 5.

(a)

(b)



Gambar 4.9 (a) *Word Cloud* (b) *Bar Chart* Genre *Action Adventure Thriller*

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Kesimpulan yang diperoleh dari penelitian ini adalah sebagai berikut.

1. Metode *K-Nearest Neighbor* dengan transformasi *Label Powerset* (LP) mampu mengklasifikasikan data genre film dengan baik pada k sebanyak 3 dengan nilai rata-rata *hamming loss* 18,79% pada data *training* dan 32,05% pada data *testing*, artinya terdapat 32,05% bagian genre film yang diprediksi salah. Metode *Multilabel K-Nearest Neighbor* (ML-KNN) mampu mengklasifikasikan data genre film secara multilabel dengan baik pada k sebanyak 3 dengan nilai rata-rata *hamming loss* sebesar 14,44% pada data *training* dan 29,32% pada data *testing*, artinya terdapat 29,32% bagian genre film yang diprediksi salah.
2. Metode yang mampu mengklasifikasikan genre film secara multi label dengan hasil performa yang lebih baik adalah metode *Multilabel K-Nearest Neighbor* (ML-KNN) dengan k sebanyak 3.

5.2 Saran

Berdasarkan kesimpulan yang diperoleh, dapat dirumuskan saran sebagai pertimbangan penelitian selanjutnya adalah melengkapi daftar kata *stopwords* yang banyak terdapat dalam sinopsis seperti nama tokoh dan tempat kejadian dalam film. Mempertimbangkan penggunaan *feature selection* pada sinopsis untuk meningkatkan performa hasil klasifikasi.

(Halaman ini sengaja dikosongkan)

DAFTAR PUSTAKA

Boutell, M.R., Luo, J., Shen, X., dan Brown, C.M. (2004), "Learning Multilabel Scene Classification", *Pattern Recognition*, hal. 1757-1771.

Castella, Q. dan Sutton, C. (2013), *Word Storm: Multiples of Word Clouds for Visual Comparison of Documents*, Cornell University, New York.

Cheng, W. dan Hüllermeier, E. (2008), "Instance-Based Label Ranking using the Mallows Model", *ECCBR, The 9th European Conference on Case-Based Reasoning*, Trier.

Dasarathy, B.V. (1990), "Nearest Neighbours (NN) Norms: NN Pattern Classification Techniques", *IEEE Computer Society Press*.

Dirks, T. (2010), *Film Genres Origin & Type*. [Online] diakses dari: https://www.filmsite.org/ [pada 20 Januari 2020].

Eneste, P. (2005), *Buku Penyuntingan Naskah*, Gramedia Pustaka Utama, Jakarta.

Fayek, H. (2016), *Speech Processing for Machine Learning: Filter banks, Mel-Frequency Cepstral Coefficients (MFCCs) and What's In-Between*. [Online] diakses dari: https://haythamfayek.com/2016/04/21/speech-processing-for-machine-learning.html.

Feldman, R. dan Sanger, J. (2007), *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*, Cambridge University Press, New York.

Franedya, R. (2019), *CNBC Indonesia*. [Online] diakses dari: https://www.cnbcindonesia.com [pada 21 Januari 2020].

Gokgoz, E. dan Subasi, A. (2015), "Comparison of decision tree algorithms for EMG signal classification using DWT", *Biomedical Signal Processing and Control*, hal. 138-144.

Hamamoto, Y., Uchimura, S., dan Tomita, S. (1997), "A Bootstrap Technique for Nearest Neighbours Classifier Design", *IEEE Transactions On Pattern Analysis and Machine Intelligence*, Vol. 19, No. 1, hal. 73-79.

Hamamoto, Y., Uchimura, S., dan Tomita, S. (1997), "A Bootstrap Technique for Nearest Neighbours Classifier Design", *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, Vol. 19, No. 1, hal. 73-79.

Han, J. dan Kamber, M. (2006), *Data Mining : Concept and Techniques Second Edition*, Morgan Kauffman Publishers.

Herbrich, R. dan Graepel, T. (2010), *Natural Language Processing*, Taylor and Francis Group, United State of America.

Herrera, F., Charte, F., Rivera, A.J., dan Jesus, M.J.d. (2016), *Multilabel Classification Problem Analysis, Metrics and Techniques*, Springer, Granada.

IMDb (2019), *Internet Movie Database*. [Online] diakses dari: https://help.imdb.com/ [pada 21 Januari 2020].

IMDb (2019), *Internet Movie Database*. [Online] diakses dari: https://help.imdb.com/ [pada 21 January 2020].

Isnaini, N., Adiwijaya, Mubarok, M.S., dan Bakar, M.Y.A. (2019), "A multilabel classification on topics of Indonesian news", *The 2nd International Conference on Data and Information Science*.

Lutfi , A.A., Permanasari, A.E., dan Fauziati, S. (2018), "Sentiment Analysis in the Sales Review of Indonesian Marketplace by Utilizing Support Vector Machine", *Journal of Information Systems Engineering and Business Intelligence*, hal. 57-64.

Manning, C.D., Raghavan, P., dan Schutze, H. (2009), *An Introduction to Information Retrieval*, Cambridge University Press, Cambridge.

Muslimah, N., Indriati, dan Wihandika, R.C. (2019), "Klasifikasi Film Berdasarkan Sinopsis dengan Menggunakan Improved K-Nearest Neighbor (K-NN)", *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, Vol. 3, hal. 196-204.

Nugraha, P.D., Faraby, S.A., dan Adiwijaya (2018), "Klasifikasi Dokumen Menggunakan Metode k-Nearest Neighbor (kNN) dengan Information Gain", *e-Proceeding of Engineering* , Vol. 5, No. 1, hal. 1541-1550.

Prasetyo, E. (2012), *Data Mining Konsep dan Aplikasi Menggunakan MATLAB*, ANDI Yogyakarta, Yogyakarta.

Pushpa, M. dan Karpagavalli, S. (2017), "Multilabel Classification: Problem Transformation methods in Tamil Phoneme classification", *Procedia Computer Science*, Vol. 115, hal. 572-579.

Read, J., Bifet, A., Holmes, G., dan Pfahringer, B. (2012), "Scalable and efficient multilabel classification for evolving data streams", *Machine Learning*.

Riadi, M. (2012), *Pengertian, Sejarah dan Unsur-Unsur Film*. [Online] diakses dari: https://www.kajianpustaka.com/ [pada 20 Januari 2020].

Thambi, Sincy V., dkk (2014), "Random Forest Algorithm for Improving the Performance of Speech/Non-Speech Detection", *International Conference on Computational Systems and Communications (ICCSC)*, Trivandum.

TMDb (2019), *The Movie Database*. [Online] diakses dari: https://www.themoviedb.org/about [pada 21 Januari 2020].

Tsoumakas, G. dan Katakis, I. (2009), "Multilabel Classification: An Overview ", *International Journal of Data Warehousing and Mining*.

Wiraguna, A., Faraby, S.A., dan Adiwijaya (2019), "Klasifikasi Topik Multi Label pada Hadis Bukhari dalam Terjemahan Bahasa", *e-Proceeding of Engineering*, Vol. 6, hal. 2144-2153.

Yang, Y. dan Liu, X. (1999), "A re-examination of text categorization methods", *Proceedings of SIGIR-99, 22nd ACM International Conference on Research and Development in Information Retrieval*, hal. 42-49.

Zhang, W., Yoshida, T., dan Tang, X. (2008), "TFIDF, LSI and Multi-word in Information Retrieval and Text Categorization", *Systems, Man and Cybernetics*.

Zhang, M.-L. dan Zhou, Z.-H. (2007), "ML-KNN: A lazy learning approach to multilabel learning", *Pattern Recognintion*, hal. 2038-2048.

Zhang, M.-L. dan Zhou, Z.-H. (2014), "A Review On Multilabel Learning Algorithms", *IEEE Transactions on Knowledge and Data Engineering* , hal. 1819-1837.

LAMPIRAN

Lampiran 1 Data Sinopsis dan Genre Film dari Situs TMDb

No	*Overview*	Genre
1	The near future, a time when both hope and hardships drive humanity to look to the stars and beyond. While a mysterious phenomenon menaces to destroy life on planet Earth, astronaut Roy McBride undertakes a mission across the immensity of space and its many perils to uncover the truth about a lost expedition that decades before boldly faced emptiness and silence in search of the unknown.	Science Fiction, Drama
2	Tyler Rake, a fearless mercenary who offers his services on the black market, embarks on a dangerous mission when he is hired to rescue the kidnapped son of a Mumbai crime lord…	Action, Drama, Thriller
3	[RUMORED] The sixth and final installment of the Scary Movie franchise that ignores the fifth film. A parody of Scream 4 and various other horror movies, sequels and reboots that were released between the late 2010's and 2020. Takes place 15 years after the fourth film.	Comedy, Horror, Thriller
4	Based on the global blockbuster videogame franchise from Sega, Sonic the Hedgehog tells the story of the world's speediest hedgehog as he embraces his new home on Earth. In this live-action adventure comedy, Sonic and his new best friend team up to defend the planet from the evil genius Dr. Robotnik and his plans for world domination.	Action, Comedy, Science Fiction, Family
…	…	…
1846	When Teela's sister is murdered and a powerful relic stolen, Marek and her friends face a sinister new enemy – Kishkumen, a foreign mystic bent on reclaiming the Darkspore for his master Szorlok. Armed with twin maps, Marek and her team race Kishkumen and his horde through creature-infested lands, to a long abandoned underground city – all the while pursued by bounty hunters intent on returning Marek to slavery.	Action, Adventure, Fantasy

Lampiran 2 *Hamming Loss* LP-KNN tiap Fold

k	Fold 1		Fold 2		Fold 3		Fold 4		Fold 5	
	Test	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**	**Train**
3	32,65%	18,17%	30,73%	18,77%	31,54%	19,09%	32,79%	19,38%	32,52%	18,55%
5	30,76%	19,82%	29,70%	20,53%	29,38%	21,06%	33,22%	21,14%	30,30%	20,51%
7	29,78%	21,86%	29,65%	21,25%	27,97%	21,62%	31,87%	21,90%	27,48%	21,95%
9	31,08%	22,60%	29,81%	21,64%	28,24%	23,14%	29,81%	21,80%	27,37%	22,82%
11	31,14%	22,49%	29,59%	22,78%	28,89%	23,29%	29,54%	21,86%	28,35%	23,36%
13	30,86%	23,12%	29,21%	22,65%	28,13%	23,95%	29,65%	22,26%	28,62%	23,75%
15	30,92%	22,76%	29,11%	22,59%	27,26%	24,01%	28,40%	22,80%	27,70%	24,25%

Lampiran 3 Hamming Loss ML-KNN tiap Fold

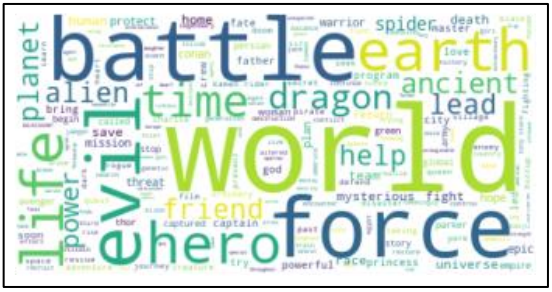
k	Fold 1		Fold 2		Fold 3		Fold 4		Fold 5	
	Test	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**	**Train**
3	30,11%	13,89%	28,62%	14,06%	29,16%	14,52%	29,54%	15,14%	29,16%	14,46%
5	29,03%	17,02%	26,94%	17,45%	26,67%	18,28%	29,92%	17,32%	28,13%	18,46%
7	28,70%	19,09%	27,70%	18,59%	25,42%	19,73%	28,62%	19,28%	28,13%	19,65%
9	29,68%	19,46%	27,59%	19,19%	26,45%	19,96%	27,75%	19,74%	27,32%	20,11%
11	28,86%	19,62%	27,53%	19,59%	26,34%	20,80%	26,88%	19,97%	24,99%	21,02%
13	29,24%	20,33%	27,37%	20,49%	25,75%	21,61%	26,45%	20,66%	24,82%	21,48%
15	28,54%	20,54%	27,91%	20,49%	25,15%	21,60%	27,80%	20,99%	24,55%	21,80%

Lampiran 4 Kata Kunci Tiap Genre

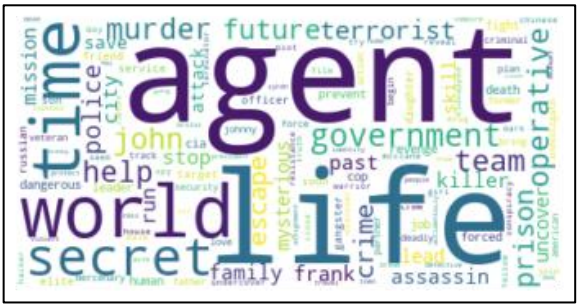
Genre	Kata Kunci
Comedy	life, school, world, friend, family
Thriller	family, life, night, son
Adventure	world, friend, life, adventure, help
Action	world, fight, team, family
Drama	life, world, story, love, family
Thriller Comedy	scream, sinister, bride
Adventure Comedy	world, life, friend, evil, game
Adventure Thriller	ship, shark, destroy, ocean
Action Comedy	city, rescue, student, secret
Action, Thriller	agent, life, secret, world, time
Action. Adventure	world, battle, force, life, earth
Drama Comedy	life, film, story, woman
Drama Thriller	life, daughter, killer
Drama Adventure	love, queen, evil
Drama Action	life, city, family, battle
Action, Adventure, Comedy	world, save, city, plan
Action, Adventure, Thriller	mission, secret, agent, world
Action, Thriller, Comedy	agent, team, criminal
Drama, Adventure, Comedy	life, snow, photo
Drama, Adventure, Thriller	death, wife, killer, train
Drama, Action, Comedy	robbery, execute, family
Drama, Action, Thriller	life, daughter, police, agent
Drama, Action, Adventure	world, life, warrior
Drama, Thriller, Comedy	life, family, stake
Action, Adventure, Thriller, Comedy	cop, death, detective

Lampiran 5 *Word Cloud* Tiap Kombinasi Genre

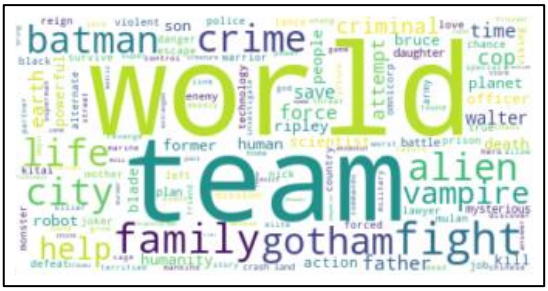
Action Adventure Thriller



Action Adventure



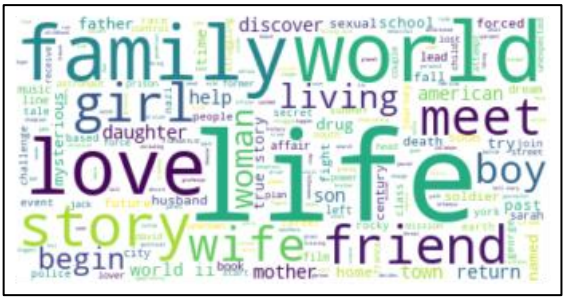
Action Thriller



Action

Adventure Thriller

Comedy



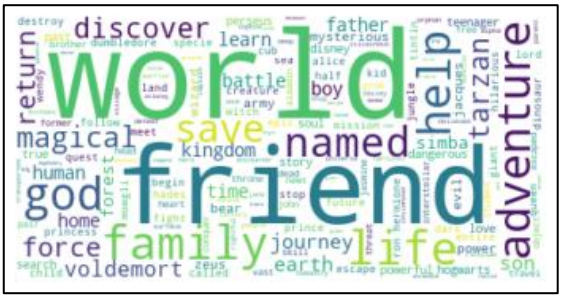
Drama

Action Adventure Comedy

Action Adventure Thriller Comedy

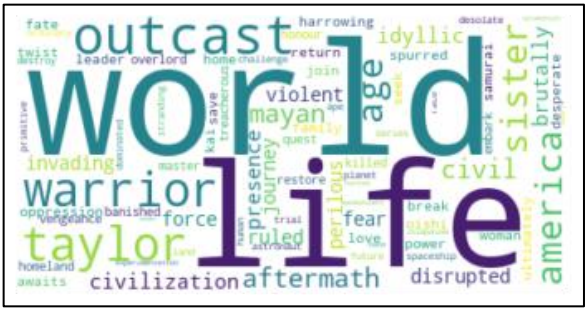
Action Comedy

Action Thriller Comedy



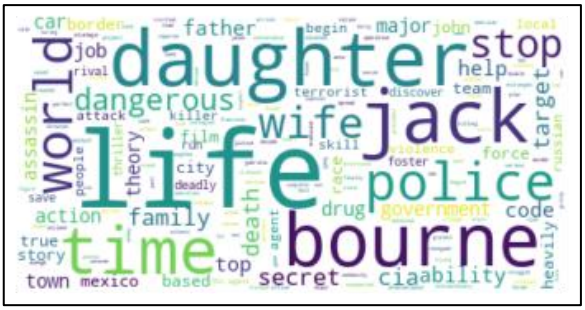
Adventure

Adventure Comedy



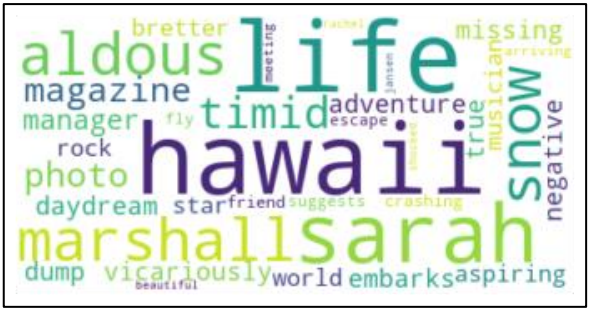
Drama Action Adventure

Drama Adventure Comedy



Drama Action Thriller

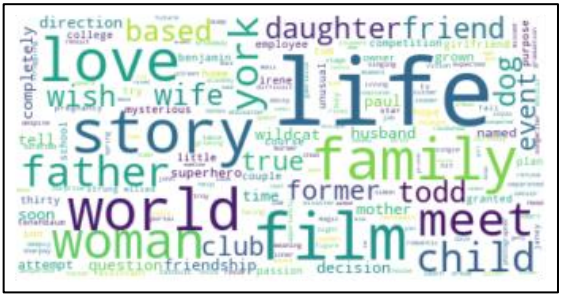
Drama Action



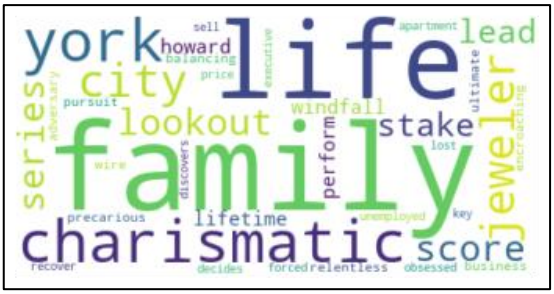
Drama Adventure Comedy

Drama Adventure Thriller

Drama Adventure



Drama Comedy



Drama Thriller Comedy

Drama Thriller

Thriller Comedy

Lampiran 6 Syntax Crawling dan Edit Data

```
import pandas as pd
import numpy as np
import json, requests
from IPython.display import clear_output #counter

overview = pd.DataFrame([])
genre = pd.DataFrame([])
for i in range(1, 101):
    url =
"https://api.themoviedb.org/3/movie/popular?api_key=
aaa47be04fc34431704fc1aa74c28562&language=en-
US&page={}".format(i)
    response = requests.get(url)
    python_json = json.loads(response.text)
    results = pd.DataFrame(python_json['results'])
    overview = np.append(overview,
results['overview'])
    genre = np.append(genre, results['genre_ids'])
    #Print counter
    print("Jumlah Data: {} Film".format(20*i))
    clear_output(wait=True)
overview = pd.DataFrame(overview)
genre = pd.DataFrame(genre)

# mendapatkan index dengan kriteria "no"
indexNames = df[df['Kriteria'] == 'No'].index
# hapus data di indeks dengan kriteria "no"
df.drop(indexNames , inplace=True)
#hapus data yang tidak memiliki sinopsis
df.dropna(subset = ["Overview"], inplace=True)
```

Lampiran 7 Syntax Preprocessing Data

```
import nltk
from bs4 import BeautifulSoup
import string
from nltk.corpus import stopwords
from nltk.tokenize import RegexpTokenizer
from nltk.stem import WordNetLemmatizer
from nltk.stem.porter import PorterStemmer
import nltk
nltk.download('wordnet')

dt=df["Overview"]
```

Lampiran 8 Syntax Preprocessing Data (Lanjutan)

```
#lowercase
dt_lower=[]
for line in dt:
    a=line.lower()
    dt_lower.append(a)
print(dt_lower)

#remove number
import re
dt_angka=[]
for line in dt_lower :
    result=re.sub("\d"," ",line)
    dt_angka.append(result)
print(dt_angka)

#remove punctuation
dt_punct=[]
for line in dt_angka :
    result=re.sub(r"[^\w\s]"," ",line)
    dt_punct.append(result)
print(dt_punct)

#clear space enter
dt_space = []
for line in dt_punct:
    result=re.sub(r"\s+"," ",line)
    dt_space.append(result)
print(dt_space)

#remove stopwords
import nltk
from nltk.corpus import stopwords
stop_words = set(stopwords.words('english'))
#add words that aren't in NLTK list
new_stopwords = ['aamir', 'aaron', 'abagnale',
'abbey', 'abbott', 'abby', 'world's', 'władysław',
'official','zuckerberg', 'état' ,'zugor', 'zune',
'æon', 'état', '運轉手之戀', 'cette', 'les', 'cet']
final_stop_words = stop_words.union(new_stopwords)
dt_stopwords = []
for sentence in dt_space:
    dt_stopwords.append(' '.join(token for token in
nltk.word_tokenize(sentence) if token not in
final_stop_words))
print(dt_stopwords)
```

Lampiran 9 Syntax TF-IDF

```
#count vectorizer
from pandas import DataFrame
from sklearn.feature_extraction.text import
CountVectorizer
vect = CountVectorizer(min_df=1)
X_dtm = vect.fit_transform(dt_stopwords)
X_dtm = X_dtm.toarray()
cv=DataFrame(X_dtm,
columns=vect.get_feature_names())
print(cv)

#tf-idf
from sklearn.feature_extraction.text import
TfidfTransformer
tfidf =
TfidfTransformer(use_idf=True).fit_transform(cv)
tfidf_train = (tfidf.toarray())
print (tfidf_train)
print (tfidf_train.shape)
tf = DataFrame(tfidf.A,
columns=vect.get_feature_names())
print (tf)
```

Lampiran 10 Syntax LP-KNN

```
from skmultilearn.problem_transform import
BinaryRelevance
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import KFold
from sklearn.metrics import label_ranking_loss
from sklearn.metrics import hamming_loss
from sklearn.metrics import zero_one_loss

kf = KFold(n_splits=5,random_state=0, shuffle=True)
kf.get_n_splits(X1)

for train_index, test_index in kf.split(X1):
    print("TRAIN:", train_index, "TEST:",
test_index)
    X_train, X_test = X1[train_index],
X1[test_index]
    y_train, y_test = y1[train_index],
y1[test_index]
    classifier =
LabelPowerset(KNeighborsClassifier(n_neighbors=5))
    classifier.fit(X_train, y_train)
```

Lampiran 10. Syntax LP-KNN (Lanjutan)

```
    classifier.fit(X_train, y_train)
    pred_test = classifier.predict(X_test)
    pred_test = pred_test.toarray()
    pred_train = classifier.predict(X_train)
    pred_train = pred_train.toarray()
    hamming_test = hamming_loss(y_test,pred_test)
    hamming_train =
hamming_loss(y_train,pred_train)
    print("Hamming Loss Test:",hamming_test)
    print("Hamming Loss Train:",hamming_train)
```

Lampiran 11 Syntax ML-KNN

```
from skmultilearn.adapt import MLkNN
from sklearn.metrics import label_ranking_loss
from sklearn.metrics import hamming_loss
from sklearn.metrics import zero_one_loss
KFold(n_splits=5, random_state=0, shuffle=True)
for train_index, test_index in kf.split(X1):
    print("TRAIN:", train_index, "TEST:",
test_index)
    X_train, X_test = X1[train_index],
X1[test_index]
    y_train, y_test = y1[train_index],
y1[test_index]
    classifier = MLkNN(k=5)
    classifier.fit(X_train, y_train)
    pred_test = classifier.predict(X_test)
    pred_test = pred_test.toarray()
    pred_train = classifier.predict(X_train)
    pred_train = pred_train.toarray()
    hamming_test = hamming_loss(y_test,pred_test)
    hamming_train =
hamming_loss(y_train,pred_train)
    print("Hamming Loss Test:",hamming_test)
    print("Hamming Loss Train:",hamming_train)
```

Lampiran 12 Syntax *Worcloud* dan *Bar Chart*

```
text = " ".join(word for word in
genre00101['Overview'])
print ("Genre Adventure Comedy")
wordcloud =
WordCloud(background_color="white").generate(text)
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()

text_token =
genre00101['Overview'].map(my_tokenizer).sum()
from collections import Counter
counter = Counter(text_token)
freq_df =
pd.DataFrame.from_records(counter.most_common(10),
columns=['word','count'])
freq_df.sort_values('count',inplace=True)
ax = freq_df.plot(kind='barh', x ='word',
y='count')
for i in ax.patches:
        ax.text(i.get_width()+.1, i.get_y()+.31, \
            str(round((i.get_width()), 2)),
fontsize=9, color='black')
```

Lampiran 13 Surat Keterangan Pengambilan Data

SURAT PERNYATAAN

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

Nama : Herviana Mayu Nabila

NRP : 06211640000075

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 : https://www.themoviedb.org

Keterangan: *Overview* dan genre film pada kategori 'Most Popular Movie'

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

Mengetahui
Pembimbing Tugas Akhir

Dr. Dra. Kartika Fithriasari, M.Si.
NIP. 19691212 199303 2 002

Surabaya, 14 Juni 2020

Herviana Mayu Nabila
NRP. 06211640000075

*(coret yang tidak perlu)

BIODATA PENULIS

Penulis dilahirkan di Bogor, 28 Mei 1998 dengan nama lengkap Herviana Mayu Nabila dan biasa dipanggil Mayu atau Nabil. Penulis menempuh pendidikan formal di SD Negeri 01 Sibanteng Bogor (2004-2010), MTs Negeri 2 Bogor (2010-2013), SMAN 1 Leuwiliang (2013-2016), dan perguruan tinggi di Departemen Statistika ITS tahun 2016. Selama masa perkuliahan, penulis aktif di berbagai organisasi, kepanitiaan, pelatihan. Organisasi kampus yang pernah diikuti penulis adalah sebaagi Staff Public Relation Statistics Computer Course (SCC) HIMASTA-ITS periode 2017-2018, Staff Ahli PR SCC-HIMASTA ITS periode 2018-2019, dan Staff Kementrian Sosial Masyarakat BEM ITS Gelora Aksi 2018-2019. Selain itu penulis juga berkesempatan mengikuti berbagai kepanitiaan yakni sebagai staff Media Informasi Pekan Raya Statistika 2017, Fasilitator Gernerasi Integralistik (GERIGI) ITS 2017, dan Pemandu Integralistik GERIGI ITS 2018. Bagi pembaca yang ingin berdiskusi, memberikan kritik maupun saran terkait Tugas Akhir ini dapat menghubungi penulis melalui *e-mail:* nabilamayu@gmail.com.