TUGAS AKHIR – KS184822

KLASIFIKASI *MULTI-LABEL* BERITA *ONLINE* MENGGUNAKAN *PROBLEM TRANSFORMATION* DENGAN METODE *K-NEAREST NEIGHBOR*

OKTAVIA RAMADHANI
NRP 062116 4000 0067

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 *MULTI-LABEL* BERITA *ONLINE* MENGGUNAKAN *PROBLEM TRANSFORMATION* DENGAN METODE *K-NEAREST NEIGHBOR*

OKTAVIA RAMADHANI
NRP 062116 4000 0067

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

ONLINE NEWS MULTI-LABEL CLASSIFICATION USING THE PROBLEM TRANSFORMATION WITH K-NEAREST NEIGHBOR

OKTAVIA RAMADHANI
SN 062116 4000 0067

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

LEMBAR PENGESAHAN

KLASIFIKASI *MULTI-LABEL* BERITA *ONLINE* MENGGUNAKAN *PROBLEM TRANSFORMATION* DENGAN METODE *K-NEAREST NEIGHBOR*

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:
Oktavia Ramadhani
NRP. 062116 4000 0067

Disetujui oleh Pembimbing:
Dr. Dra. Kartika Fithriasari, M.Si. ( )
NIP. 19691212 199303 2 002
Adatul Mukarromah, S.Si., M.Si ( )
NIP. 19800418 200312 2 001

Mengetahui,
Kepala Departemen Statistika

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

SURABAYA, JULI 2020

(*Halaman ini sengaja dikosongkan*)

KLASIFIKASI *MULTI-LABEL* BERITA *ONLINE* MENGGUNAKAN *PROBLEM TRANSFORMATION* DENGAN METODE *K-NEAREST NEIGHBOR*

Nama Mahasiswa : Oktavia Ramadhani
NRP : 062116 4000 0067
Departemen : Statistika
Dosen Pembimbing : Dr. Dra. Kartika Fithriasari, M.Si.
Adatul Mukarromah, S.Si., M.Si.

Abstrak

Detik.com merupakan salah satu portal berita *online* paling populer di Indonesia. Portal berita *online* ini memiliki banyak kategori utama. Berita yang tersaji tidak selalu memuat satu kategori utama saja, akan tetapi dianggap hanya masuk dalam satu kategori utama saja. Permasalahan tersebut dapat diselesaikan dengan melakukan klasifikasi *multi-label*. Oleh karena itu, pada penelitian ini dilakukan klasifikasi *multi-label* menggunakan tiga metode *problem transformation*, yakni *Binary Relevance, Label Powerset,* dan *Classifier Chain* dengan metode klasifikasi dasar yakni *K-Nearest Neighbor*. Data yang digunakan dalam penelitian ini adalah data judul berita pada portal berita *online* detik.com yang terkategori secara *multi-label* dalam enam kategori, yakni detikFinance, detikOto, detikHot, detikInet, detikTravel, dan detikNews. Berdasarkan hasil analisis, didapatkan bahwa pada kasus ini metode *problem transformation* terbaik menggunakan metode klasifikasi dasar *K-Nearest Neighbor* yakni metode *Binary Relevance*. Hal tersebut didasari atas nilai *hamming loss* yang dihasilkan oleh metode *Binary Relevance* lebih kecil dibandingkan dengan metode *Label Powerset* dan *Classifier Chain*.

Kata Kunci: Hamming Loss, K-Nearest Neighbor, Klasifikasi Multi-Label, Problem Transformation

(*Halaman ini sengaja dikosongkan*)

ONLINE NEWS MULTI-LABEL CLASSIFICATION USING THE PROBLEM TRANSFORMATION WITH K-NEAREST NEIGHBOR

Name	**: Oktavia Ramadhani**
Student Number	**: 062116 4000 0067**
Department	**: Statistics**
Supervisors	**: Dr. Dra. Kartika Fithriasari, M.Si.**
	Adatul Mukarromah, S.Si., M.Si.

Abstract

Detik.com is one of the most popular online news portals in Indonesia. Detik.com has various main categories. Every news presented in this news portal does not always contain only one main category, but it is classified only into one main category. This kind of problem can be solved by doing multi-label classification to classify news into one or more labels. Therefore, in this study, a multi-label classification was carried out using three problem transformation methods, which is Binary Relevance, Label Powerset, and Classifier Chain, with the K-Nearest Neighbor as the base classifier. The data used in this study is the news headlines data obtained through the online news portal detik.com. The news headlines are categorized as multi-label in six categories (detikFinance, detikOto, detikHot, detikInet, detikTravel, and detikNews). Based on the analysis results, it can be concluded that in this case, the best problem transformation method with K-Nearest Neighbor as the base classifier is the Binary Relevance method. It is based on the value of the hamming loss given by Binary Relevance method is smaller than the Label Powerset and Classifier Chain methods.

Keywords: Hamming Loss, K-Nearest Neighbor, Multi-Label Classification, Problem Transformation

(*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 berjudul "Klasifikasi Multi-Label Berita *Online* menggunakan *Problem Transformation* dengan Metode *K-Nearest Neighbor*" dengan lancar.

Proses penyusunan laporan Tugas Akhir ini tidak lepas dari bantuan dan dukungan dari berbagai pihak. Oleh karena itu, penulis menyampaikan terima kasih kepada:

1. Ibu Dr. Dra. Kartika Fithriasari, M.Si. selaku Kepala Departemen Statistika ITS sekaligus dosen pembimbing yang telah memberikan fasilitas, sarana, dan prasarana sekaligus dengan sangat sabar memberikan arahan, bimbingan, saran, masukan, serta motivasi kepada penulis selama penyusunan Tugas Akhir.
2. Ibu Dr. Santi Wulan Purnami, S.Si., M.Si. selaku Sekretaris Departemen I yang telah memberikan fasilitas, sarana, dan prasarana guna penyelesaian Tugas Akhir ini.
3. Ibu Adatul Mukarromah, S.Si., M.Si. selaku dosen co-pembimbing yang telah meluangkan waktu dan dengan sangat sabar memberi bimbingan, saran, masukan, dan motivasi kepada penulis selama penyelesaian Tugas Akhir.
4. Bapak Prof. Drs. Nur Iriawan, MIKom., Ph.D. dan Ibu Dra. Wiwiek Setya Winahju, M.S. selaku dosen penguji yang sabar dalam memberikan banyak komentar serta saran dan masukan selama penyelesaian Tugas Akhir ini.
5. Bapak Dr. Bambang Widjanarko Otok, M.Si. selaku dosen wali selama masa studi penulis yang telah banyak memberikan saran dan arahan dalam proses belajar di Departemen Statistika ITS.
6. Seluruh dosen Statistika ITS yang telah memberikan ilmu dan pengetahuan yang tak ternilai harganya, serta segenap karyawan Departemen Statistika ITS.
7. Kedua orang tua penulis yakni Bapak Mashudi dan Ibu Siti Masita Schu serta adik penulis Kevin Surya Huditara atas segala doa, nasihat, cinta, dan kasih sayang yang diberikan kepada

penulis sehingga menjadi motivasi bagi penulis dalam menghadapi kesulitan dalam penyelesaian Tugas Akhir ini.

8. Rifqi Rabbanie yang telah menjadi teman diskusi dan bertukar pandangan bagi penulis, selalu siap menemani dan membantu penulis, serta memberikan semangat dan motivasi kepada penulis selama masa studi sampai dengan Tugas Akhir ini selesai.
9. Teman-teman di bawah bimbingan Ibu Kartika Fithriasari, yang telah menjadi tempat berkeluh kesah serta memberikan bantuan dan dukungan kepada penulis selama penyusunan Tugas Akhir ini.
10. Fitria, Naufal, Mas Antok, Mbak Ulfa, dan Mas Azmi yang mulai dari STATION 2018 sampai dengan penyusunan Tugas Akhir ini menjadi tempat berbagi keluh kesah serta memberikan motivasi dan saran kepada penulis.
11. Teman-teman PSDM BEM ITS Gelora Aksi dan PSDM BEM ITS Kolaborapi yang senantiasa menghibur dan memberikan semangat kepada penulis serta menjadi teman berpikir dan berkembang penulis selama perkuliahan sampai dengan selesai penyusunan Tugas Akhir ini.
12. Teman-teman #CABSFORLYF yang senantiasa memberikan semangat serta dukungan moral kepada penulis selama penyusunan Tugas Akhir ini.
13. Semua teman, relasi, serta 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, Juli 2020

Penulis

DAFTAR ISI

Halaman

DAFTAR GAMBAR

Halaman

DAFTAR TABEL

Halaman

DAFTAR LAMPIRAN

Halaman

(*Halaman ini sengaja dikosongkan*)

BAB I
PENDAHULUAN

1.1 Latar Belakang

Berita adalah informasi baru yang disajikan dalam penulisan yang jelas, aktual, dan menarik (Horne, 2007). Menurut KBBI, berita merupakan cerita atau keterangan mengenai kejadian atau peristiwa yang hangat. Peristiwa-peristiwa yang terjadi dapat dijadikan suatu berita dan setiap berita memiliki nilainya masing-masing. Nilai berita dapat berubah sesuai dengan faktor-faktor tertentu. Salah satu faktor tersebut adalah faktor waktu, dimana semakin baru peristiwa maka beritanya akan semakin menarik publik untuk membacanya. Faktor-faktor lainnya yang dapat mempengaruhi nilai berita yaitu faktor jarak, nama, keanehan, pengaruh suatu berita, sentuhan manusiawi (*human interest*), universal, dan konflik atau ketegangan. Berita sudah menjadi kebutuhan masyarakat dan diperlukan agar seseorang memiliki wawasan yang luas dan mengetahui situasi dunianya (Musfah, 2018). Selain dapat memperoleh wawasan yang luas, membaca berita secara intens juga dapat menambah kosakata baru bagi pembacanya.

Berita dulunya dipublikasikan melalui radio, televisi, dan media cetak. Akan tetapi, kini cukup banyak portal berita *online* yang bermunculan seiring dengan berkembangnya teknologi informasi dan komunikasi. Beberapa media cetak pun ikut beralih ke media *online* dalam penyampaian berita karena keterbatasan waktu penyampaian dan perlu menunggu waktu untuk terbit keesokan harinya. Hadirnya berita di televisi, walaupun banyak membantu, tidak bisa setiap waktu memberikan berita secepat mungkin karena terkadang terhalang dengan jadwal acara lainnya. Melalui media *online* yang memiliki kemudahan akses, berita yang bersifat aktual akan lebih cepat diterima oleh para pembaca. Beberapa portal berita *online* berbahasa Indonesia yang cukup terkenal yaitu detik.com, kompas.com, okezone.com dan vivanews.com (Nova, 2011).

Pada awal mula kemunculannya, berita online disajikan secara statis dan hanya berupa salinan dari versi cetak. Namun, pada

tahun 1998 detik.com mengenalkan langgam berita baru: "ringkas *to the point*". Oleh karena itu, berita dapat dengan cepat dan mudah diterima oleh para pembaca tanpa harus menunggu terbitan besok atau siaran *breaking news* di televisi. Portal berita *online* sempat mengalami kejatuhan pada tahun 2002-2003. Akan tetapi, mereka tampil kembali dengan lebih atraktif selepas tahun 2003. Portal berita *online* ini hadir dengan memberikan ruang interaksi antar pembaca di dalam situsnya. Pembaca memiliki keleluasaan dalam memberikan komentar pada berita serta ruang diskusi dalam forum (Margianto & Syaefullah). Berdasarkan survei yang dilakukan oleh DailySocial.id (2017) terhadap 1.012 responden yang merupakan pengguna *smartphone* di Indonesia, diketahui bahwa media *online* menjadi sumber informasi bagi 90,22% responden serta 55,43% responden sudah tidak membeli koran.

Portal berita *online* dapat merilis ratusan berita setiap harinya. Oleh karena itu, berita pada portal berita *online* umumnya diberi label sesuai dengan topiknya agar pembaca dapat dengan mudah memilih berita mana yang ingin dibaca. Portal berita *online* detik.com memiliki banyak kategori utama dengan domain tersendiri, yaitu detikNews, detikFinance, detikFintech, detikOto, dan lain sebagainya. Portal berita *online* kompas.com juga memiliki banyak segmen atau kategori di dalamnya, yakni Entertainment, Bola, Tekno, *Female*, *Health*, Properti dan lain sebagainya (Nova, 2011). Pengelompokkan berita akan sulit apabila harus dilakukan secara manual sehingga diperlukan suatu metode yang dapat digunakan untuk mengelompokkan setiap berita yang ada di portal berita *online* secara otomatis (Aggarwal & Zhai, 2012). Berita pada portal berita *online* umumnya berbentuk teks sehingga proses pengelompokkan berita secara otomatis ke dalam kategori tertentu perlu diselesaikan dengan proses *text mining*.

Text mining adalah proses mengekstraksi informasi dari sumber data melalui identifikasi dan eksplorasi pola yang menarik. Secara fungsional, proses kerja *text mining* mengikuti model umum dari proses kerja *data mining*. Hal yang membedakan *text mining* dengan *data mining* yaitu data yang digunakan pada *data mining*

menggunakan data yang terstruktur sedangkan *text mining* menggunakan data yang tidak terstruktur (Feldman & Sanger, 2007). Adapun salah satu contoh dari aplikasi *text mining* adalah metode *supervised learning*. Metode *supervised learning* adalah metode *machine learning* yang memanfaatkan data *training* untuk mempelajari *classifier* atau fungsi regresi yang dapat digunakan untuk menghitung prediksi pada data baru yang belum diketahui. Masalah *supervised learning* terkadang disebut juga sebagai klasifikasi.

Klasifikasi pola sederhana dapat mengelompokkan data sampel dalam satu kategori atau label. Klasifikasi ini disebut juga klasifikasi *single-label*. Akan tetapi, permasalahan yang mungkin muncul pada klasifikasi *single-label* adalah permasalahan di mana suatu data sampel memiliki kemungkinan untuk tidak hanya terkategori dalam satu label saja. Klasifikasi *multi-label* tersebut memungkinkan suatu data sampel untuk diberi lebih dari satu label. Algoritma yang digunakan antara *single-label* dan *multi-label* berbeda. Algoritma dari klasifikasi sederhana berdasarkan *tree*, *neural network*, *support vector machine*, yang dirancang untuk memberikan nilai tunggal sebagai *output*. Algoritma ini tidak dapat digunakan secara langsung untuk mengatasi masalah klasifikasi *multi-label*. Terdapat dua cara yang dapat dilakukan untuk mengatasi permasalahan tersebut, antara lain mengubah kumpulan data *multi-label* sehingga memungkinkan untuk dilakukan klasifikasi menggunakan algoritma klasifikasi yang dikenal, seperti melakukan *training* dengan *binary classifier* untuk masing-masing label atau melakukan adaptasi algoritma klasifikasi sederhana sehingga diperoleh algoritma baru dengan kemampuan untuk mengaitkan data sampel dengan lebih dari satu label atau kategori tertentu.

Detik.com merupakan salah satu portal berita *online* terpopuler di Indonesia. Survei yang telah dilakukan oleh DailySocial.id (2017) menunjukkan bahwa detik.com merupakan portal berita *online* Indonesia yang paling familiar bagi para respondennya. Namun, saat ini proses pengelompokkan berita pada portal berita *online* detik.com masih menggunakan klasifikasi *single-label* padahal berita tersebut memiliki kemungkinan untuk tidak hanya terkatego-

ri dalam satu label saja. Berdasarkan permasalahan tersebut, maka klasifikasi *multi-label* merupakan proses yang tepat karena berita-berita pada portal berita *online* akan terkategori ke dalam satu atau beberapa kategori tertentu. Penelitian sebelumnya mengenai klasifikasi *multi-label* untuk data berita pernah dilakukan oleh Isnaini dkk (2019) dengan menggunakan metode *problem transformation Binary Relevance* (BR) dengan metode klasifikasi dasar *K-Nearest Neighbor* (KNN). Pada penelitian tersebut, diperoleh nilai kebaikan klasifikasinya menggunakan nilai *hamming loss* yakni sebesar 0,1116. Penelitian klasifikasi *multi-label* untuk data artikel berita juga pernah dilakukan oleh Pambudi dkk (2019) menggunakan metode *problem transformation* BR dengan *Pseudo Nearest Neighbor Rule* sebagai metode klasifikasi dasar dan diperoleh nilai *hamming loss* sebesar 0,1645. Penelitian lainnya juga pernah dilakukan oleh Prawira dkk (2018) menggunakan metode *Multinomial Naïve Bayes* dan didapatkan *hamming loss* sebesar 0,18.

Pada penelitian ini, metode *problem transformation* yang akan digunakan yaitu metode BR, *Label Powerset* (LP), dan *Classifier Chain (CC)*. LP dan CC adalah metode *problem transformation* yang mempertimbangkan dependensi pada label. Metode tersebut berbeda dengan metode BR yang menganggap bahwa setiap label adalah independen. Pada penelitian ini, BR juga digunakan sebagai komparasi karena metode *problem transformation* tersebut merupakan metode *problem transformation* yang banyak digunakan untuk permasalahan klasifikasi *multi-label*. Metode klasifikasi dasar yang akan digunakan pada penelitian ini yaitu metode KNN. Berdasarkan ketiga metode *problem transformation* tersebut, akan ditentukan metode *problem transformation* terbaik dengan metode klasifikasi dasar KNN untuk proses klasifikasi berita pada portal berita *online* detik.com.

1.2 Rumusan Masalah

Portal berita *online* detik.com mampu menyajikan ratusan berita setiap harinya. Oleh karena itu, berita-berita di dalamnya diberi label sesuai dengan topik agar pembaca dapat dengan mudah

memilih berita mana yang ingin dibaca. Portal berita *online* detik.com memiliki banyak kategori utama dengan domain tersendiri, yaitu detikFinance, detikOto, detikHot, detikInet, detikTravel, dan detikNews. Pada portal berita *online* detik.com, suatu berita hanya diberi satu label atau kategori saja padahal berita tersebut memiliki kemungkinan untuk terkategori dalam lebih dari satu label. Oleh karena itu, pada penelitian ini dilakukan klasifikasi *multi-label* untuk mengategorikan berita ke dalam satu label atau lebih menggunakan beberapa metode klasifikasi multi-label, yakni metode *Binary Relevance* dengan metode klasifikasi dasar *K-Nearest Neighbor* (BRKNN), metode *Label Powerset* dengan metode klasifikasi dasar *K-Nearest Neighbor* (LPKNN), dan metode *Classifier Chain* dengan metode klasifikasi dasar *K-Nearest Neighbor* (CCKNN). Permasalahan utama yang akan dibahas dalam penelitian ini ialah bagaimana karakteristik dari data judul berita pada berita *online* detik.com, bagaimana hasil perbandingan ketepatan klasifikasi menggunakan metode BRKNN, LPKNN, dan CCKNN, dan kata kunci apa yang paling sering menjadi topik utama pemberitaan.

1.3 Tujuan

Tujuan yang ingin dicapai dalam penelitian ini berdasarkan rumusan masalah yang telah diuraikan adalah sebagai berikut.

1. Mendapatkan karakteristik judul berita pada portal berita *online* detik.com.
2. Mendapatkan serta membandingkan hasil ketepatan klasifikasi *multi-label* berita pada portal berita *online* detik.com menggunakan metode BRKNN, LPKNN, dan CCKNN.
3. Mendapatkan kata kunci topik utama pemberitaan di masing-masing kombinasi kategori berita yang disivualisasikan dengan *word cloud*.

1.4 Manfaat

Hasil dari penelitian ini diharapkan dapat bermanfaat dalam bidang klasifikasi teks secara *multi-label* secara umum dengan menggunakan metode BRKNN, LPKNN, dan CCKNN. Penelitian ini juga diharapkan dapat memberikan tambahan referensi bagi pe-

nyedia portal berita *online* terkait klasifikasi *multi-label* berita *online* sehingga klasifikasi secara *multi-label* dapat dilakukan secara otomatis.

1.5 Batasan Masalah

Pada penelitian ini, terdapat beberapa batasan masalah yang digunakan. Batasan masalah pada penelitian ini adalah sebagai berikut.

1. Data yang digunakan merupakan berita pada portal berita *online* detik.com.
2. Data berita diambil dari enam kategori yang ada di detik.com, yakni detikFinance, detikOto, detikHot, detikInet, detikTravel, dan detikNews.
3. Berita yang dipilih adalah berita yang dirilis pada rentang waktu 20 Januari 2020 hingga 26 Januari 2020.

BAB II
TINJAUAN PUSTAKA

2.1 *Text Mining*

Text mining adalah proses atau upaya untuk mengekstraksi informasi dari sumber data melalui identifikasi dan eksplorasi pola yang menarik. Secara fungsional, proses kerja *text mining* mengikuti model umum dari proses kerja *data mining*. Hal yang membedakan *text mining* dengan *data mining* yaitu data yang digunakan pada *data mining* menggunakan data yang terstruktur sedangkan *text mining* menggunakan data yang tidak terstruktur (Feldman & Sanger, 2007). Data yang terstruktur umumnya dikelola oleh sistem *database* sedangkan data yang tidak terstruktur umumnya dikelola melalui *search engine*. Sebuah *search engine* memungkinkan pengguna untuk memperoleh informasi yang bermanfaat dari koleksi dengan mudah menggunakan *keyword query*.

Adapun beberapa contoh dari aplikasi *text mining* adalah metode *supervised learning* dari data teks, *metode unsupervised learning* dari data teks, dan *text summarization*. Metode *supervised learning* adalah metode *machine learning* umum yang memanfaatkan data *training* untuk mempelajari *classifier* atau fungsi regresi yang dapat digunakan untuk menghitung prediksi pada data baru yang belum diketahui. Masalah *supervised learning* ini terkadang disebut juga klasifikasi. Metode *unsupervised learning* tidak memerlukan data *training*. Dua metode *unsupervised learning* yang umum digunakan adalah *clustering* dan *topic modelling* (Aggarwal & Zhai, 2012).

2.2 Klasifikasi *Multi-Label*

Pada *machine learning*, khususnya *supervised learning*, salah satu metode yang paling sering digunakan adalah metode klasifikasi. Metode klasifikasi menggunakan suatu kumpulan data untuk memperoleh model yang mampu memberi label kepada sampel yang baru dan tidak digunakan dalam proses *training*. Secara sederhana, metode klasifikasi dirancang dari suatu kumpulan data

di mana setiap sampel nantinya akan dikaitkan dengan satu dan hanya satu kelas atau label. L merupakan himpunan label yang diterapkan pada suatu kumpulan data, X_i semua sampel memiliki kelas tertentu, dan l_a dan l_b adalah indeks dari kedua label. Klasifikasi sederhana harus memenuhi syarat sebagai berikut.

$$L = \{l_1, l_2, ..., l_k\}, |L| = k > 1 \tag{2.1}$$

$$X_{l_a} \cap X_{l_b} = \varnothing, \forall l_a, l_b, a \neq b \tag{2.2}$$

Syarat pada persamaan (2.1) menunjukkan bahwa himpunan L setidaknya harus memiliki dua label atau lebih karena apabila tidak memenuhi syarat tersebut maka seluruh sampel akan diklasifikasikan pada satu label yang sama. Persamaan (2.2) menunjukkan bahwa setiap label merupakan *disjoint subset*s atau himpunan yang saling lepas, atau dengan kata lain masing-masing sampel sesuai dengan satu label saja. Ketika $k = 2$ maka klasifikasinya adalah biner atau label pada klasifikasi tersebut umumnya diidentifikasikan sebagai *true* atau *false*. Salah satu penerapannya yaitu seperti pada pesan elektronik yang pesannya diberi label, yaitu *spam* atau bukan *spam*. Apabila $k > 2$ maka umumnya disebut sebagai klasifikasi *multi-class* (Charte dkk, 2012).

Pada klasifikasi *multi-label*, syarat pada persamaan (2.2) tidak terpenuhi karena sampel dapat diklasifikasikan pada lebih dari satu label. Persamaan (2.3) menunjukkan bahwa *classifier* pada klasifikasi *multi-label* apabila telah dilatih maka akan mencakup sebuah *set Y* yang merupakan *subset* atau himpunan bagian dari L. Syarat pada persamaan (2.2) mengarah ke persamaan (2.4) yang menjelaskan bahwa himpunan bagian dari label data sampel belum tentu merupakan himpunan yang saling lepas.

$$Y = f(X_i), Y \subseteq L \tag{2.3}$$

$$\neg \forall l_a, l_b, X_{l_a} \cap X_{l_b} = \varnothing \tag{2.4}$$

Algoritma dari klasifikasi sederhana berdasarkan *tree*, *neural network*, *support vector machine*, yang dirancang untuk memberikan nilai tunggal sebagai *output*. Algoritma ini tidak dapat digunakan secara langsung untuk mengatasi masalah klasifikasi *mul-*

ti-label. Terdapat dua cara yang dapat dilakukan untuk mengatasinya, antara lain:

a. Metode *Problem Transformation*, yakni dengan mengubah *dataset* sehingga memungkinkan untuk menggunakan algoritma klasifikasi yang dikenal, seperti melakukan *training* dengan *binary classifier* untuk setiap label, atau
b. Metode *Algorithm Adaptation*, yakni dengan mengadaptasi algoritma klasifikasi sederhana dengan menambahkan kemampuan untuk berurusan dengan fakta bahwa setiap sampel dapat dikaitkan dengan beberapa label.

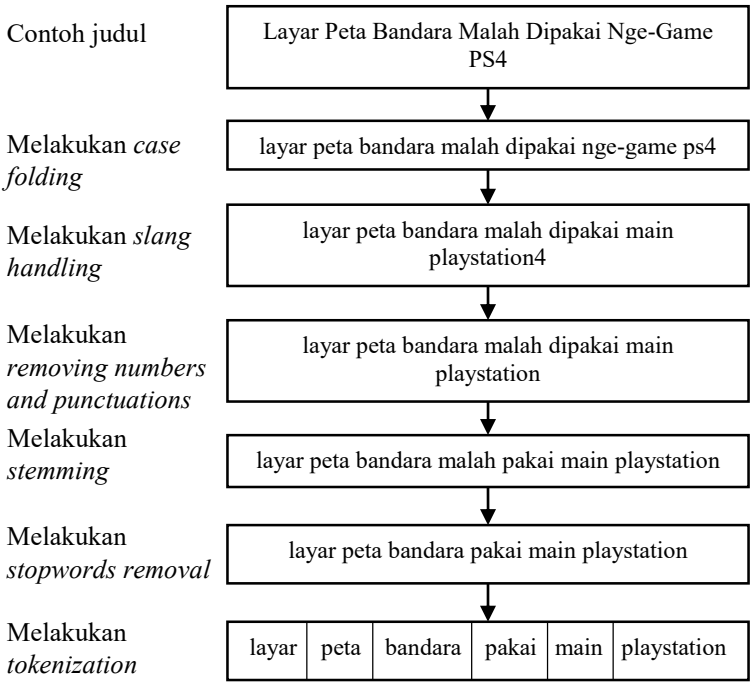
2.2.1 *Pre-Processing*

Pre-processing merupakan tahap awal dalam melakukan pengolahan data teks. Proses ini merupakan proses pengubahan bentuk data teks menjadi data yang lebih terstruktur sesuai dengan kebutuhan. Tahap *pre-processing* dapat meningkatkan nilai akurasi dari klasifikasi data. Tahapan dalam *pre-processing* meliputi proses *case folding*, *slang handling*, *removing numbers and punctuations*, *stemming*, *stopwords removal*, dan *tokenization*.

a. *Case Folding*, merupakan proses mengkonversi karakter teks menjadi huruf kecil (Mittal dkk, 2013).
b. *Slang handling*, yakni proses mengganti *slang* dengan kata yang baku dan telah tercantum pada kamus. *Slang* merupakan penggunaan kata informal dan tidak baku dalam standar suatu bahasa atau dialek tetapi dianggap dapat diterima dan dipahami dalam suatu kelompok tertentu (Mittal dkk, 2013).
c. *Removing numbers and punctuations,* merupakan proses penghapusan angka dan tanda baca (Hidayatullah & Ma'Arif, 2017).
d. *Stemming*, merupakan proses menghilangkan imbuhan pada kata sehingga diperoleh kata dasarnya saja. Dalam praktiknya, *stemming* lebih sering digunakan karena cenderung lebih cepat dan lebih mudah diimplementasikan daripada *lemmatization* (Bholat dkk, 2015).

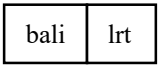
e. *Stopwords Removal*, merupakan proses penghapusan kosakata yang bukan termasuk kata unik atau memiliki kontribusi sedikit dalam membedakan teks satu dengan teks (Bholat dkk, 2015).
f. *Tokenization*, merupakan proses identifikasi kata kunci yang bermakna (Verma dkk, 2014). Proses ini dilakukan dengan cara memecah keseluruhan teks yang semula berbentuk kalimat menjadi kata per kata.

Penjabaran hasil *output* pada setiap tahap *pre-processing* di atas akan ditunjukkan melalui simulasi *pre-processing* terhadap salah satu judul berita yang diambil dari portal berita *online* detik.com, yakni berita berjudul "Layar Peta Bandara Malah Dipakai Nge-Game PS4".



Gambar 2.1 Simulasi *Pre-Processing*

Pada judul berita berikutnya, yakni judul "Bali Bakal Punya LRT di 2022", akan dilakukan tahap *pre-processing* dengan langkah-langkah yang sama sehingga diperoleh hasil sebagai berikut.



Gambar 2.2 Contoh Hasil *Pre-Processing*

Berdasarkan kedua contoh hasil *pre-processing* pada judul berita di atas, diperoleh struktur data setelah *pre-processing* sebagaimana ditunjukkan pada Tabel 2.1.

Tabel 2.1 Contoh Struktur Data Setelah *Pre-Processing*

Berita ke-	**Kata ke-*j* yang muncul pada berita (t_j)**							
	bali	bandara	layar	lrt	main	pakai	peta	playstation
1	0	1	1	0	1	1	1	1
2	1	0	0	1	0	0	0	0

Pembentukan struktur data setelah *pre-processing* adalah sebagaimana pada Tabel 2.1, yaitu menjadikan setiap kata menjadi suatu variabel t_j. Apabila terdapat tambahan kata pada judul berita baru, maka kata tersebut akan diletakkan pada baris yang sama dan di Kolom berikutnya. Namun, apabila pada judul berita berikutnya muncul kata yang sama dengan kata yang telah ada di struktur data sebelumnya, maka kata yang sama tersebut tidak akan dimasukkan lagi pada struktur data. Nilai setiap kata pada Tabel 2.1 merupakan jumlah kemunculan kata pada judul berita ke-*i*.

2.2.2 *Term Frequency Inverse Document Frequency*

Term Frequency Inverse Document Frequency (TF-IDF) merupakan metode pembobotan yang dilakukan untuk menentukan seberapa relevan suatu kata dalam dokumen tertentu. Kata dengan angka TF-IDF yang tinggi menunjukkan bahwa terdapat hubungan yang kuat antara kata tersebut dengan dokumen dimana kata tersebut muncul (Ramos, 2003). Bobot kata ke-*j* diperoleh dengan mengalikan nilai *Term Frequency* kata ke-*j* pada dokumen ke-*i* dengan nilai *Inverse Document Frequency* sebagaimana pada persamaan (2.5) dan (2.6) berikut (Ghag & Shah, 2014),

$$X_{ij} = TF\left(t_j, \mathrm{d}_i\right) \times IDF\left(t_j\right) \tag{2.5}$$

$$IDF\left(t_j\right) = \log\left(\frac{n}{DF\left(t_j\right)}\right) \tag{2.6}$$

dengan,

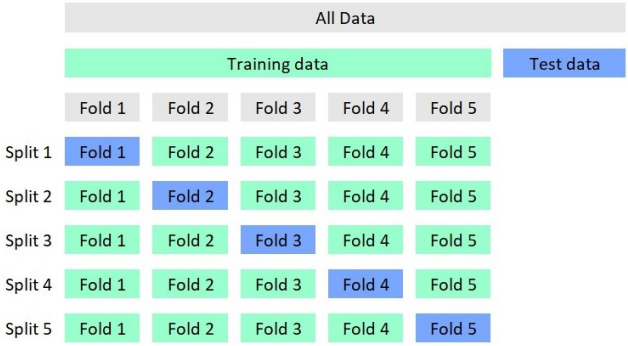
t_j : kata ke-j
d_i : dokumen ke-i
X_{ij} : bobot TF-IDF pada kata ke-j dokumen ke-i
$TF(t_j,d_i)$: *Term Frequency* atau jumlah kemunculan kata ke-j pada dokumen ke-i
$IDF(t_j)$: *Inverse Document Frequency*
n : jumlah dokumen
$DF(t_j)$: jumlah dokumen yang berisikan kata ke-j

2.2.3 *K-Fold Cross Validation*

K-fold cross validation merupakan metode yang umum digunakan untuk mengevaluasi kinerja klasifikasi. Metode ini digunakan untuk mempartisi data menjadi data *training* dan data *testing*. Metode ini banyak digunakan oleh peneliti karena mampu mengurangi bias yang terjadi dalam pengambilan sampel dengan membagi data menjadi sejumlah q himpunan bagian yang disebut *fold* (Gokgoz & Subasi, 2015). Selain itu, pengulangan penentuan data *training* dan data *testing* juga dilakukan sebanyak q kali. Setiap 1 dari q himpunan bagian digunakan sebagai data *testing* maka himpunan q-1 lainnya disatukan untuk membentuk suatu data *training* (Schneider, 1997). Nilai q yang direkomendasikan untuk digunakan adalah q = 5 atau q = 10 (Rodríguez dkk, 2009).

Ilustrasi pembagian data *training* dan data *testing* menggunakan metode *k-fold cross validation* dengan q = 5 terdapat pada Gambar 2.3. Berdasarkan Gambar 2.3 dapat dilihat bahwa data terbagi menjadi 5 himpunan bagian atau *fold* dan dilakukan pengulangan penentuan data *training* dan *testing* sebanyak 5 kali. Pada penentuan data *training* dan data *testing* pertama, *fold* 1 menjadi data *testing* sedangkan 4 *fold* lainnya menjadi data *training*. Pada penentuan data *training* dan data *testing* kedua, *fold* 2 menjadi data

testing sedangkan 4 *fold* lainnya menjadi data *training*. Proses tersebut dilakukan berulang sampai dengan penentuan data *training* dan data *testing* kelima.



Gambar 2.3 Ilustrasi Pembagian Data dengan 5 *Fold*
(Sumber: miro.medium.com)

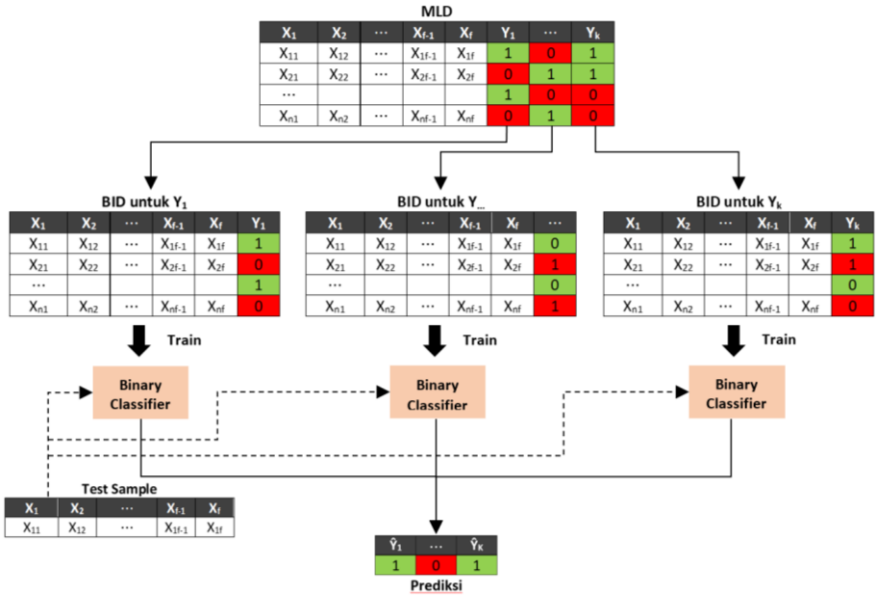
2.2.4 *Binary Relevance*

Binary Relevance (BR) merupakan salah satu metode *problem transformation* untuk kasus *multi-label*. Metode BR dilakukan dengan menerapkan *binary classifier* pada setiap label dan *output* yang dihasilkan akan digabungkan untuk membangun kumpulan label hasil prediksi (Herrera dkk, 2016).

Metode BR adalah metode yang sederhana untuk melakukan klasifikasi *multi-label*. Akan tetapi, metode ini juga memiliki kelemahan yakni mengabaikan adanya korelasi di antara label yang ada karena menganggap bahwa setiap label independen. Gambar 2.4 merupakan ilustrasi dari proses kerja metode BR. Penjabaran proses kerja metode BR adalah sebagai berikut.

1. Data *training multi-label* diuraikan menjadi data-data *training single-label binary* sebanyak k untuk setiap label Y_1, Y_2, ..., Y_k.
2. Melatih *binary classifier* sebanyak k dengan menerapkan algoritma klasifikasi *binary* terhadap data-data *training single-label binary* yang telah terbentuk.

3. Mendapatkan hasil prediksi setiap label ($\hat{Y}_1$, $\hat{Y}_2$, ..., $\hat{Y}_k$) untuk data *testing* menggunakan masing-masing *binary classifier*.
4. Menggabungkan hasil prediksi setiap label ($\hat{Y}_1$, $\hat{Y}_2$, ..., $\hat{Y}_k$) menjadi hasil prediksi yang berbentuk *multi-label*.



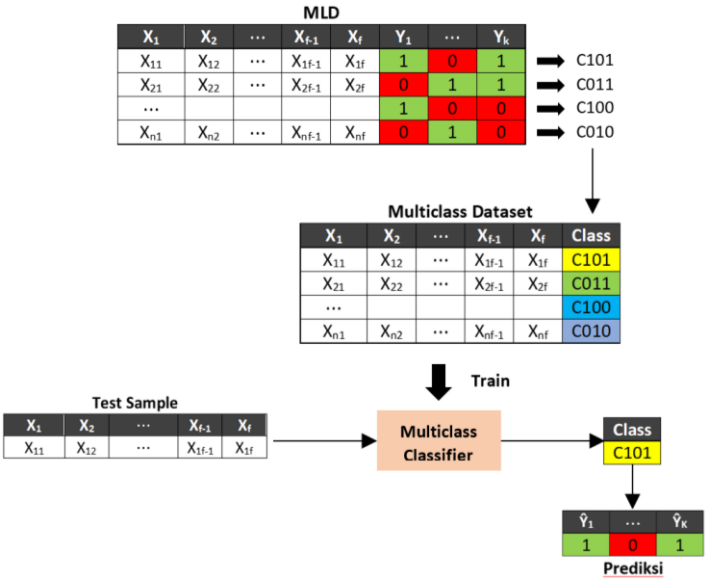
Gambar 2.4 Ilustrasi *Binary Relevance*
(Sumber: Herrera dkk, 2016)

2.2.5 *Label Powerset*

Label Powerset (LP) merupakan metode *problem transformation* untuk kasus *multi-label* yang lebih sederhana dari BR karena tidak perlu melatih banyak model serta menggabungkan masing-masing *output* yang dihasilkan menjadi sekumpulan label (Herrera dkk, 2016). LP secara langsung mengubah kumpulan data yang *multi-label* menjadi satu kumpulan data yang *single-label*, dengan cara mentransformasikan setiap kombinasi label yang berbeda pada data *multi-label* menjadi kelas yang berbeda pada data *single-label* (Spolaor dkk, 2013).

Berdasarkan Gambar 2.5 dapat dilihat bahwa proses metode LP berbeda dengan BR karena pada metode LP cukup melatih satu model saja. Berikut adalah penjabaran proses kerja metode LP.

1. Data *training multi-label* diuraikan menjadi satu data *training single-label multi-class*, yakni dengan mentransformasikan setiap kombinasi label yang berbeda pada data *multi-label* menjadi kelas yang berbeda pada data *single-label*.
2. Melatih *multi-class classifier* menggunakan algoritma klasifikasi *multi-class* pada data *training single-label multi-class*.
3. Mendapatkan kelas hasil prediksi untuk data *testing* menggunakan *multi-class classifier*.
4. Mentransformasikan kelas hasil prediksi *single-label* menjadi kombinasi hasil prediksi ($\hat{Y}_1$, $\hat{Y}_2$, ..., $\hat{Y}_k$) yang berbentuk *multi-label*.



Gambar 2.5 Ilustrasi *Label Powerset*
(Sumber: Herrera dkk, 2016)

2.2.6 *Classifier Chain*

Classifier Chain (CC) merupakan metode *problem transformation* untuk kasus *multi-label* selain BR. Metode CC melatih model sejumlah *k* atau sejumlah label yang ada, seperti metode BR. *Classifier* pertama dilatih menggunakan data *input* yang asli. *Output* label pertama kemudian ditambahkan sebagai atribut *input* yang baru untuk melatih *classifier* kedua. Proses tersebut dilakukan terus menerus sampai dengan *classifier* ke-*k*. *Classifier* yang berantai tersebut diharapkan mampu memperhitungkan dependensi yang mungkin ada pada label (Herrera dkk, 2016). Penjabaran proses kerja metode CC adalah sebagai berikut (Heider dkk, 2013).

1. Melatih *binary classifier* pertama menggunakan input asli X_j.
2. Melatih *binary classifier* kedua menggunakan input (*expanded*) yakni X_i, Y_1.
3. Mengulang proses nomor 2 sampai diperoleh *binary classifier* ke-*k* dengan menggunakan input X_j, Y_1, ..., Y_{k-1}.
4. Mendapatkan hasil prediksi $\hat{Y}_1$, $\hat{Y}_2$, ..., $\hat{Y}_k$ menggunakan *binary classifier* yang telah dilatih dengan input X_j, $\hat{Y}_1$, ..., $\hat{Y}_{k-1}$.
5. Menggabungkan hasil prediksi setiap label ($\hat{Y}_1$, $\hat{Y}_2$, ..., $\hat{Y}_k$) menjadi hasil prediksi yang berbentuk *multi-label*.

2.2.7 *K-Nearest Neighbor*

KNN merupakan salah satu metode *supervised learning* yang telah lama digunakan dalam klasifikasi. Algoritma yang digunakan pada metode KNN merupakan salah satu algoritma klasifikasi sederhana. KNN mengklasifikasikan objek baru berdasarkan label dari *c* objek terdekatnya.

Algoritma pada metode KNN membutuhkan beberapa *input*, yaitu sekumpulan data *training*, jarak (metrik) untuk menghitung kesamaan objek, dan nilai *c*. Pada umumnya, nilai *c* yang digunakan adalah bilangan ganjil untuk menghindari mayoritas yang imbang pada *c* objek terdekat (Adeniyi dkk, 2016). Berdasarkan tiga input tersebut, objek baru akan diklasifikasikan dengan mengikuti algoritma sebagai berikut.

1. Menghitung jarak antara keseluruhan data *training* dengan objek baru yang hendak diklasifikasikan. Penentuan jarak antara suatu data dengan data yang lain pada algoritma KNN dapat dilakukan dengan berbagai perhitungan jarak. Pada penelitian ini, akan digunakan perhitungan jarak *Euclidean* karena perhitungan jarak ini merupakan perhitungan jarak yang sering digunakan untuk metode klasifikasi KNN. Jarak *Euclidean* merupakan akar dari jumlah selisih kuadrat antara nilai-nilai yang berlawanan dalam vektor. Persamaan (2.7) berikut adalah persamaan yang digunakan untuk menghitung jarak *Euclidean* (Prasath dkk, 2017),

$$ED(X_a, X_b) = \sqrt{\sum_{j=1}^{f} \left| X_{aj} - X_{bj} \right|^2} \tag{2.7}$$

   dengan,
   $ED(X_a,X_b)$: jarak *Euclidean* antara vektor X_a dan vektor X_b
   f : jumlah variabel bebas
   X_{aj} : data ke-j pada vektor X_a
   X_{bj} : data ke-j pada vektor X_b
2. Melakukan identifikasi c data *training* yang hasil perhitungan jaraknya terdekat dengan objek tersebut.
3. Setelah diperoleh c data *training* terdekat, objek diklasifikasikan atau diberi label sesuai dengan label yang paling banyak (mayoritas) dari c data *training* tersebut (Gorunescu, 2011).

2.2.8 Pengukuran Ketepatan Klasifikasi *Multi-Label*

Evaluasi algoritma klasifikasi *multi-label* jauh lebih rumit daripada evaluasi klasifikasi *single-label*. Salah satu pengukuran ketepatan klasifikasi *multi-label* yakni *hamming loss*. *Hamming loss*, mengevaluasi seberapa banyak *input* mengalami misklasifikasi, seperti yang tertulis pada persamaan (2.8), dimana Δ berarti terdapat perbedaan simetris antara (Y_i) yang merupakan *set* label aktual untuk dokumen ke-i dengan ($\hat{Y}_i$) yang merupakan *set* label prediksi untuk dokumen ke-i, n adalah jumlah data yang dianalisis, dan k adalah jumlah label atau *output* yang memungkinkan. Sema-

kin kecil nilai *hamming loss* maka semakin baik kinerjanya. Perhitungan ini membagi jumlah kesalahan prediksi label dengan jumlah label yang memungkinkan. Oleh karena itu, penilaian akan berbeda apabila jumlah kesalahan sama tetapi jumlah label pada kumpulan data *multi-label* berbeda (Herrera dkk, 2016).

$$HammingLoss = \frac{1}{n}\frac{1}{k}\sum_{i=1}^{n}\left|Y_i \Delta \hat{Y}_i\right| \tag{2.8}$$

Berikut merupakan ilustrasi perhitungan nilai *hamming loss* menggunakan contoh label aktual dan label prediksi sebagaimana ditampilkan pada Tabel 2.2. Langkah pertama yang dilakukan untuk memperoleh nilai *hamming loss* yakni menentukan jumlah data (*n*) dan jumlah label (*k*). Berdasarkan Tabel 2.2, didapatkan jumlah data (*n*) sebanyak 5 dan jumlah label (*k)* sebanyak 5. Langkah selanjutnya setelah menentukan *n* dan *k* adalah menghitung nilai $\left|Y_i \Delta \hat{Y}_i\right|$ yang ditunjukkan juga pada Tabel 2.2.

Tabel 2.2 Ilustrasi Perhitungan Nilai *Hamming Loss*

i	**Label Aktual**					**Label Prediksi**					$\left\|Y_i \Delta \hat{Y}_i\right\|$
	Y_1	Y_2	Y_3	Y_4	Y_5	$\hat{Y}_1$	$\hat{Y}_2$	$\hat{Y}_3$	$\hat{Y}_4$	$\hat{Y}_5$	
1	1	0	0	0	0	1	0	0	0	0	0
2	1	0	0	1	1	1	0	0	1	$0^{(*)}$	1
3	0	0	0	1	0	1	0	0	$0^{(*)}$	$1^{(*)}$	2
4	1	0	0	0	0	1	0	0	0	$1^{(*)}$	1
5	1	0	1	0	1	1	0	1	0	1	0

Keterangan :
$^{(*)}$ Label prediksi yang mengalami misklasifikasi

Setelah itu, dilakukan perhitungan nilai *hamming loss* menggunakan persamaan (2.8). Proses perhitungan tersebut adalah sebagai berikut.

$$HammingLoss = \frac{1}{n}\frac{1}{k}\sum_{i=1}^{n}\left|Y_i \Delta \hat{Y}_i\right| = \frac{1}{5}\frac{1}{5}(0+1+2+1+0) = 0{,}16$$

2.2.9 *Word Cloud*

Word cloud merupakan salah satu metode visualisasi dokumen teks yang paling popular dan sering digunakan. *Word cloud*

memberikan presentasi grafis dokumen teks dengan melakukan *plotting* kata-kata yang paling banyak muncul dalam dokumen teks dalam ruang dua dimensi (Castellà & Sutton, 2014). Pada *word cloud*, frekuensi suatu kata ditunjukkan oleh ukuran *font* kata tersebut. Semakin besar ukuran *font* kata maka semakin besar frekuensi kemunculan kata tersebut. Gambar 2.6 berikut merupakan salah satu contoh hasil visualisasi data teks dengan *word cloud*.

Gambar 2.6 Visualisasi Data Teks dengan *Word Cloud*
(Sumber: Castellà & Sutton, 2014)

2.3 Detik.com

Detik.com merupakan portal berita *online* yang berisi berita dan artikel online di Indonesia. Detik.com merupakan salah satu portal berita *online* terpopuler di Indonesia. Detik.com mulai *online* dengan sajian lengkap pada 9 Juli 1998 oleh Budiono Darsono, Yayan Sopyan, Abdul Rahman, dan Didi Nugrahadi. Pada tahun 1998, detik.com mengenalkan suatu langgam berita baru: ringkas *to the point*. Oleh karena itu, berita dapat dengan cepat dan mudah diterima oleh pembaca tanpa harus menunggu terbitan koran besok atau siaran *breaking news* di televisi (Margianto & Syaefullah).

Pada 3 Agustus 2011, portal berita *online* detik.com diakuisisi oleh CT Corp. Chairul Tanjung, pemilik CT Corp, membeli detik.com secara total (100 persen) dengan nilai US$60 juta atau apabila dikurskan nominalnya menjadi Rp 521-540 miliar. Setelah diambil alih, maka jajaran direksi akan diisi oleh pihak-pihak dari TransCorp, sebagai perpanjangan tangan CT Corp di ranah media

(Noviyanto, 2011). Berikut adalah logo dari portal berita *online* detik.com.

Gambar 2.7 Logo Detik.com

Portal berita *online* detik.com mempunyai beragam kategori dengan domain yang berbeda-beda, antara lain sebagai berikut:

a. detikNews, yang memuat informasi seputar berita politik atau peristiwa tertentu,
b. detikFinance, yang memuat berita seputar ekonomi dan keuangan,
c. detikHot, yang berisikan informasi seputar artis atau selebriti,
d. detikInet, yang memuat berita mengenai teknologi,
e. detikSport, yang memuat berita seputar olahraga,
f. detikOto, yang berisikan informasi seputar otomotif,
g. detikTravel, yang memuat berita tentang liburan dan pariwisata,
h. detikFood, memuat informasi tentang makanan dan kuliner,
i. detikHealth, yang memuat informasi dan artikel yang berkaitan dengan kesehatan,
j. Wolipop, berisikan informasi tentang wanita dan gaya hidup,
k. 20Detik, yang memuat original konten video mulai dari news sampai dengan *lifestyle*.

BAB III
METODOLOGI PENELITIAN

3.1 Sumber Data

Data yang digunakan dalam penelitian ini adalah data judul berita yang diperoleh melalui portal berita *online* detik.com pada 20 Januari 2020 sampai dengan 26 Januari 2020. Data yang akan digunakan pada penelitian sebanyak 1.405 judul berita. Judul berita tersebut terkategori secara *multi-label* dalam enam kategori, yakni kategori detikFinance, detikOto, detikHot, detikInet, detikTravel, dan detikNews sebagaimana terlampir pada Lampiran 1.

3.2 Struktur Data dan Variabel Peenelitian

Variabel yang digunakan dalam penelitian ini dapat dilihat pada Tabel 3.1.

Tabel 3.1 Variabel Penelitian

No.	Variabel	Keterangan	Skala
1	Y_1	0 = tidak termasuk label detikFinance 1 = termasuk label detikFinance	Nominal
2	Y_2	0 = tidak termasuk label detikOto 1 = termasuk label detikOto	Nominal
3	Y_3	0 = tidak termasuk label detiHot 1 = termasuk label detikHot	Nominal
4	Y_4	0 = tidak termasuk label detikInet 1 = termasuk label detikInet	Nominal
5	Y_5	0 = tidak termasuk label detikTravel 1 = termasuk label detikTravel	Nominal
6	Y_6	0 = tidak termasuk label detikNews 1 = termasuk label detikNews	Nominal
7	X_j	bobot kata ke-*j* yang muncul pada data berita	Rasio

Tahapan *pre-processing* (meliputi *case folding*, *slang handling*, *removing numbers and punctuations*, *stemming*, *stopwords removal*, dan *tokenization*) kemudian akan diterapkan terhadap data. Struktur data yang telah melalui tahap tersebut ditunjukkan pada Tabel 3.2.

Tabel 3.2 Struktur Data Penelitian

No.	Y_1	Y_2	⋯	Y_6	X_1	X_2	⋯	X_{3213}
1	$Y_{1,1}$	$Y_{1,2}$	⋯	$Y_{1,6}$	$X_{1,1}$	$X_{1,2}$	⋯	$X_{1,3213}$
2	$Y_{2,1}$	$Y_{2,2}$	⋯	$Y_{2,6}$	$X_{2,1}$	$X_{2,2}$	⋯	$X_{2,3213}$
3	$Y_{3,1}$	$Y_{3,2}$	⋯	$Y_{3,6}$	$X_{3,1}$	$X_{3,2}$	⋯	$X_{3,3213}$
4	$Y_{4,1}$	$Y_{4,2}$	⋯	$Y_{4,6}$	$X_{4,1}$	$X_{4,2}$	⋯	$X_{4,3213}$
5	$Y_{5,1}$	$Y_{5,2}$	⋯	$Y_{5,6}$	$X_{5,1}$	$X_{5,2}$	⋯	$X_{5,3213}$
⋮	⋮	⋮	⋱	⋮	⋮	⋮	⋱	⋮
1405	$Y_{1405,1}$	$Y_{1405,2}$	⋯	$Y_{1405,6}$	$X_{1405,1}$	$X_{1405,2}$	⋯	$X_{1405,3213}$

Keterangan :
$X_{i,j}$: bobot kata ke-j pada berita ke-i
$Y_{i,k}$: kategori k untuk berita ke-i

3.3 Langkah Analisis

Langkah-langkah yang digunakan dalam penelitian ini adalah sebagai berikut.

1. Menyiapkan data berita dari portal berita *online* detik.com.
   a. Memperoleh URL halaman "Indeks" dari setiap kategori berita dan tanggal yang telah ditentukan untuk proses *scraping* data.
   b. Melakukan *scraping* data menggunakan *software Parsehub* dengan menginputkan URL halaman "Indeks" yang telah diperoleh pada tahap (a) sehingga didapatkan data berisi URL dan judul setiap artikel berita yang rilis sesuai kategori dan tanggal yang telah ditentukan dengan format .csv.
   c. Melakukan proses pemberian label terhadap berita yang telah diperoleh dari proses *scraping*.
2. Melakukan *pre-processing* teks menggunakan *software* Python dengan *syntax* sebagaimana ditunjukkan pada Lampiran 3. *Syntax* tersebut merupakan modifikasi dari *syntax* yang diambil dari laporan tugas akhir Mochamad Ihsan Ananto yang berjudul "*Klasifikasi Kategori Pengaduan Masyarakat melalui Kanal Lapor! menggunakan Artificial Neural Network (ANN)*". Berikut merupakan tahap *pre-processing* yang dilakukan:
   a. *Case folding*, yakni proses mengubah karakter teks menjadi huruf kecil.

b. *Slang handling*, yaitu proses mengganti *slang* yang ada pada data teks menjadi kata yang baku.
c. *Removing numbers and punctuations,* merupakan penghapusan angka dan tanda baca yang ada pada data teks.
d. *Stemming*, yakni proses menghilangkan imbuhan pada kata sehingga diperoleh kata dasarnya saja. Proses ini dilakukan menggunakan *library* Sastrawi pada *software* Python.
e. *Stopwords removal*, yakni proses penghapusan kosakata yang bukan termasuk kata unik atau memiliki kontribusi sedikit dalam membedakan teks satu dengan teks lainnya. Proses ini dilakukan menggunakan *software* Python. Daftar *stopwords* diambil dari tesis F. Tala yang berjudul "*A Study of Stemming Effect on Information Retrieval in Bahasa Indonesia*".
f. *Tokenization*, yakni proses memecah keseluruhan teks yang semula berbentuk kalimat menjadi kata per kata.

3. Melakukan pembobotan kata dengan TF-IDF sesuai persamaan (2.5) dan (2.6) menggunakan *software* Python dengan *syntax* sebagaimana ditunjukkan pada Lampiran 4.
4. Melakukan pembagian data *training* dan *testing* menggunakan *k-fold cross validation* dengan *fold* sebanyak 5. *K-fold cross validation* dilakukan menggunakan *software* Python dengan *syntax* sebagaimana ditunjukkan pada Lampiran 5, Lampiran 6, dan Lampiran 7. *Syntax* tersebut merupakan modifikasi dari *syntax* yang diambil dari laporan tugas akhir Muhammad Abid As Sarofi yang berjudul "*Klasifikasi Genre Musik berdasarkan Mel Frequency Cepstrum Coefficient (MFCC) dengan menggunakan Metode Support Vector Machine (SVM) dan Random Forest (RF)*".
5. Melakukan klasifikasi menggunakan metode *problem transformation* BRKNN (*syntax* terlampir pada Lampiran 5), LPKNN (*syntax* terlampir pada Lampiran 6), dan CCKNN (*syntax* terlampir pada Lampiran 7).
   Adapun langkah-langkah klasifikasi menggunakan metode BR-KNN, yakni:

a. Melakukan transformasi terhadap kumpulan data *training multi-label* menjadi kumpulan data *training single-label binary* se-banyak label yang ada (*k*).
b. Melakukan klasifikasi untuk data *testing* dengan tahapan sebagai berikut.
   - Menentukan nilai *c*.
   - Menghitung jarak *Euclidean* dari objek terhadap data *training* yang diberikan sesuai dengan persamaan (2.7).
   - Menentukan *c* data *training* yang paling dekat dari objek.
   - Mengklasifikasikan objek sesuai dengan mayoritas label pada *c* data *training* yang telah diperoleh.
c. Melakukan transformasi terhadap hasil prediksi menjadi kumpulan hasil prediksi *multi-label*.

Sedangkan langkah-langkah klasifikasi menggunakan metode LPKNN, yakni:

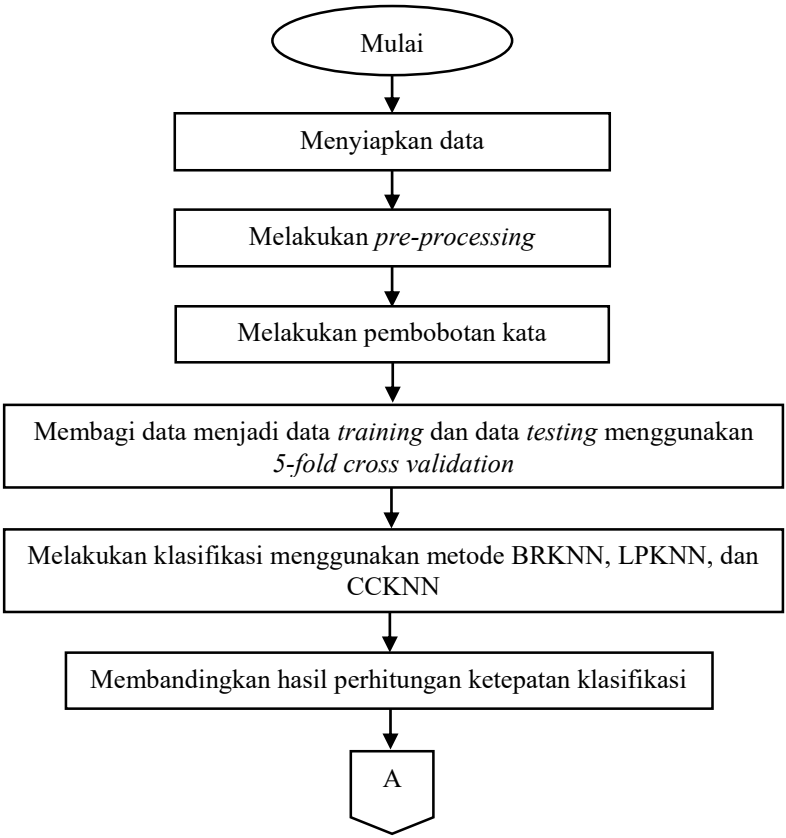
a. Melakukan transformasi pada kumpulan data *training multi-label* menjadi data *training single-label multi-class*.
b. Melakukan klasifikasi untuk data *testing* sebagaimana yang dijelaskan pada klasifikasi menggunakan metode BRKNN.
c. Mentransformasikan hasil prediksi menjadi kumpulan hasil prediksi *multi-label*.

Berikut langkah-langkah klasifikasi menggunakan metode CC-KNN, yakni:

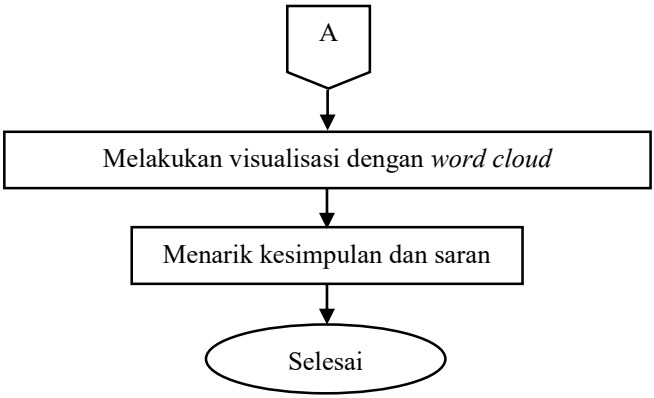
a. Melakukan klasifikasi label pertama dengan data *input* yang asli menggunakan metode klasifikasi dasar KNN sebagaimana yang dijelaskan pada klasifikasi menggunakan metode BRKNN.
b. Melakukan klasifikasi label kedua dengan data *input* yang asli ditambah dengan *input* baru yakni *output* label pertama menggunakan metode yang sama dengan proses (a).
c. Mengulangi proses (b) sampai diperoleh hasil klasifikasi label ke-*k* (sejumlah label yang ada).
d. Menggabungkan hasil prediksi dari setiap *classifier* menjadi kumpulan prediksi *multi-label*.

6. Membandingkan hasil perhitungan ketepatan klasifikasi menggunakan metode BRKNN, LPKNN, dan CCKNN berdasarkan nilai *hamming loss* sesuai dengan persamaan (2.8).
7. Melakukan visualisasi kata kunci topik utama pemberitaan di masing-masing kombinasi kategori berita dengan *word cloud*. *Syntax* visualisasi *word cloud* dapat dilihat pada Lampiran 8.
8. Menarik kesimpulan dan saran.

Langkah-langkah analisis di atas dapat digambarkan dengan diagram alir yang disajikan pada Gambar 3.1.



Gambar 3.1 Diagram Alir Penelitian



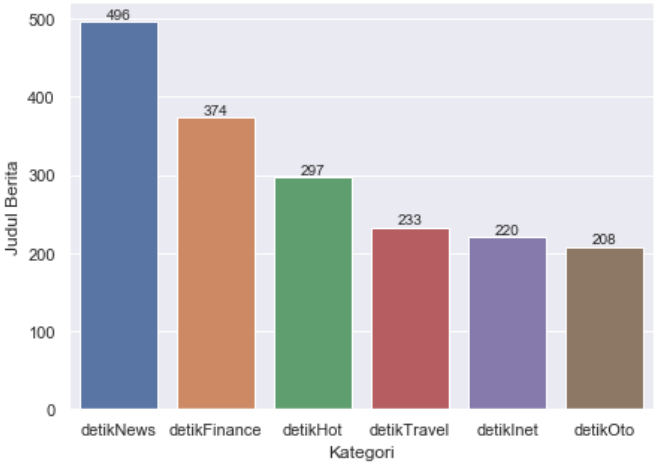
Gambar 3.1 Diagram Alir Penelitian (lanjutan)

BAB IV
ANALISIS DAN PEMBAHASAN

Pada penelitian ini dilakukan klasifikasi *multi-label* judul berita menggunakan metode *problem transformation Binary Relevance*, *Label Powerset*, dan *Classifier Chain* dengan metode klasifikasi dasar *K-Nearest Neighbor*. Data yang digunakan adalah data pada portal berita *online* detik.com pada 20 Januari 2020 sampai dengan 26 Januari 2020 dengan jumlah berita sebanyak 1.405 berita. Berita terbagi secara *multi-label* ke dalam enam kategori seperti ditunjukkan pada Lampiran 1. Setelah dilakukan klasifikasi, akan dilakukan pengukuran ketepatan klasifikasi dengan nilai *hamming loss*.

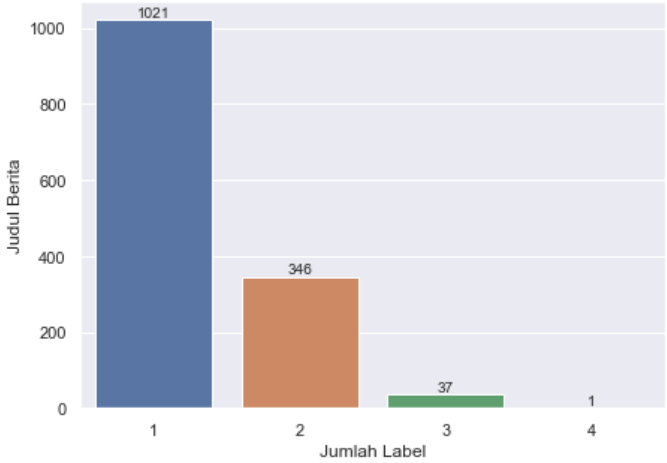
4.1 Karakteristik Data Judul Berita

Detik.com merupakan salah satu portal berita *online* yang paling populer di Indonesia. Data judul berita pada portal berita *online* detik.com terkategori dalam enam kategori berita. Data tersebut memuat judul serta kategori berita. Jumlah berita pada setiap kategori yang ada dapat dilihat pada Gambar 4.1 sebagai berikut.



Gambar 4.1 Jumlah Berita Tiap Kategori

Gambar 4.1 menunjukkan jumlah berita tiap label yang ada di portal berita *online* detik.com pada 20 Januari 2020 sampai dengan 26 Januari 2020. Berdasarkan Gambar 4.1, diketahui bahwa berita terbanyak terdapat pada kategori detikNews, yakni sebanyak 496 berita. Hal tersebut disinyalir karena kategori detikNews mempunyai cakupan berita yang lebih luas sehingga jumlah berita dengan kategori detikNews lebih banyak dibandingkan dengan kategori yang lain. Sedangkan berita paling sedikit terdapat pada kategori detikOto dengan jumlah berita sebanyak 208 berita. Setiap berita dapat dikategorikan ke dalam lebih dari satu kategori (bersifat *multi-label*) dan setiap berita dapat memiliki jumlah label yang berbeda-beda. Jumlah berita pada masing-masing jumlah label dapat dilihat pada Gambar 4.2 sebagai berikut.



Gambar 4.2 Jumlah Berita Tiap Jumlah Label

Mayoritas dari judul berita memiliki label sebanyak satu, yaitu sebanyak 1.021 judul berita. Hal tersebut ditunjukkan sebagaimana pada Gambar 4.2. Sedangkan berita dengan label sebanyak empat memiliki judul berita yang paling sedikit, yakni hanya satu berita saja. Jumlah berita dengan kategori lebih dari satu yakni sebanyak 384 berita.

4.2 Klasifikasi Judul Berita

Proses selanjutnya yakni melakukan klasifikasi *multi-label* judul berita menggunakan *problem transformation* dengan metode klasifikasi dasar *K-Nearest Neighbor*. Tahapan-tahapan yang dilakukan yakni melakukan *pre-processing*, mendapatkan ketepatan klasifikasi, serta membandingkan ketepatan klasifikasi menggunakan metode *problem transformation* dengan metode klasifikasi dasar *K-Nearest Neighbor*.

4.2.1 *Pre-Processing* Data

Sebelum melakukan proses klasifikasi terhadap data, akan dilakukan *pre-processing* data terlebih dahulu. Tahapan *pre-processing* yang akan dilakukan dalam penelitian ini meliputi *case folding*, *slang handling*, *removing number and punctuations*, *stemming*, *stopwords removal*, dan *tokenization* menggunakan *syntax* pada Lampiran 3. Berikut adalah contoh data sebelum dan sesudah dilakukan setiap tahapan *pre-processing* data.

1. *Case Folding*

Case folding adalah proses mengkonversikan karakter teks menjadi huruf kecil. Contoh data sebelum dan sesudah dilakukan tahap *case folding* dapat dilihat pada Tabel 4.1.

Tabel 4.1 Contoh Data Sebelum dan Sesudah *Case Folding*

No.	Sebelum *Case Folding*	Sesudah *Case Folding*
1.	DPR Rapat Bareng BUMN Pangan, Bahas Sektor Pertanian	dpr rapat bareng bumn pangan, bahas sektor pertanian
2.	DPR Rapat 7 Jam dengan Menkes dan BPJS Kesehatan, Ini Hasilnya	dpr rapat 7 jam dengan menkes dan bpjs kesehatan, ini hasilnya
3.	PM Kanada Bantah Bicara ke Ratu soal Biaya Keamanan Harry-Meghan	pm kanada bantah bicara ke ratu soal biaya keamanan harry-meghan

Perbedaan antara data sebelum dan sesudah diterapkan proses *case folding* dapat dilihat pada Tabel 4.1. Pada data yang sudah diterapkan proses *case folding*, kata yang mengandung huruf kapital tidak

lagi ditemukan. Contohnya, frasa "DPR Rapat" telah berubah menjadi "dpr rapat". Setelah dilakukan proses *case folding*, tahap yang dilakukan adalah *slang handling*.

2. *Slang Handling*

Slang handling, yakni proses mengganti *slang* dengan kata yang baku dan telah tercantum pada kamus. *Slang* adalah penggunaan kata tidak baku dalam standar suatu bahasa atau dialek tetapi dianggap dapat diterima dan dipahami dalam suatu kelompok tertentu. Tabel 4.2 menunjukkan contoh data sebelum dan sesudah diterapkannya proses *slang handling* terhadap data teks.

Tabel 4.2 Contoh Data Sebelum dan Sesudah *Slang Handling*

No.	Sebelum *Slang Handling*	Sesudah *Slang Handling*
1.	dpr rapat bareng bumn pangan, bahas sektor pertanian	dpr rapat dengan bumn pangan, bahas sektor pertanian
2.	dpr rapat 7 jam dengan menkes dan bpjs kesehatan, ini hasilnya	dpr rapat 7 jam dengan menkes dan bpjs kesehatan, ini hasilnya
3.	pm kanada bantah bicara ke ratu soal biaya keamanan harry-meghan	pm kanada bantah bicara ke ratu soal biaya keamanan harry-meghan

Perbedaan antara sebelum dan sesudah diterapkannya *slang handling* terhadap data teks dapat dilihat pada Tabel 4.2. Pada Tabel 4.2, ditunjukkan bahwa kata "bareng" yang tidak termasuk kata baku telah digantikan dengan kata "dengan" yang merupakan kata baku dan telah tercantum dalam kamus. Tahap berikutnya yang dilakukan setelah diterapkannya proses *slang handling* adalah tahap *removing number and punctuations*.

3. *Removing Numbers and Punctuations*

Removing numbers and punctuations adalah proses penghapusan angka dan tanda baca yang ada pada data teks. Contoh data sebelum dan sesudah dilakukan *removing numbers and punctuations* ditampilkan pada Tabel 4.3. Tabel 4.3 menunjukkan bahwa angka "7" serta tanda baca "," dan "-" telah dihapuskan dari data

teks. Tahap selanjutnya setelah *removing numbers and punctuations* yakni *stemming*.

Tabel 4.3 Contoh Data Sebelum dan Sesudah *Removing Numbers and Punctuations*

No.	Sebelum *Removing Numbers and Punctuations*	Sesudah *Removing Numbers and Punctuations*
1.	dpr rapat dengan bumn pangan, bahas sektor pertanian	dpr rapat dengan bumn pangan bahas sektor pertanian
2.	dpr rapat 7 jam dengan menkes dan bpjs kesehatan, ini hasilnya	dpr rapat jam dengan menkes dan bpjs kesehatan ini hasilnya
3.	pm kanada bantah bicara ke ratu soal biaya keamanan harry-meghan	pm kanada bantah bicara ke ratu soal biaya keamanan harry meghan

4. *Stemming*

Stemming merupakan proses menghilangkan imbuhan pada kata sehingga diperoleh kata dasarnya saja. Contoh data sebelum dan sesudah proses *stemming* dapat dilihat pada Tabel 4.4.

Tabel 4.4 Contoh Data Sebelum dan Sesudah *Stemming*

No.	Sebelum *Stemming*	Sesudah *Stemming*
1.	dpr rapat dengan bumn pangan bahas sektor pertanian	dpr rapat dengan bumn pangan bahas sektor tani
2.	dpr rapat jam dengan menkes dan bpjs kesehatan ini hasilnya	dpr rapat jam dengan menkes dan bpjs sehat ini hasil
3.	pm kanada bantah bicara ke ratu soal biaya keamanan harry meghan	pm kanada bantah bicara ke ratu soal biaya aman harry meghan

Tabel 4.4 menunjukkan bahwa terdapat perbedaan antara data sebelum dan sesudah diterapkan proses *stemming*, sebagai contoh kata “pertanian” yang berubah menjadi “tani”, kata “kesehatan” menjadi “sehat”, dan kata “hasilnya” menjadi “hasil”. Tahap berikutnya setelah proses *stemming* yakni proses *stopwords removal*.

5. *Stopwords Removal*

Stopwords removal merupakan proses penghapusan kosa-kata yang bukan termasuk kata unik atau memiliki kontribusi sedikit dalam membedakan teks satu dengan teks lainnya. Contoh data sebelum dan sesudah proses *stopwords removal* dapat dilihat pada Tabel 4.5. Berdasarkan Tabel 4.5, dapat dilihat perbedaan antara sebelum dan sesudah dilakukan proses *stopwords removal*. Beberapa kata pada contoh yang dihilangkan yakni "dengan", "dan", "ini", "ke", dan "soal". Tahap selanjutnya setelah proses *stopwords removal* adalah proses *tokenization*.

Tabel 4.5 Contoh Data Sebelum dan Sesudah *Stopwords Removal*

No.	Sebelum ***Stopwords Removal***	Sesudah ***Stopwords Removal***
1.	dpr rapat dengan bumn pangan bahas sektor tani	dpr rapat bumn pangan bahas sektor tani
2.	dpr rapat jam dengan menkes dan bpjs sehat ini hasil	dpr rapat jam menkes bpjs sehat hasil
3.	pm kanada bantah bicara ke ratu soal biaya aman harry meghan	pm kanada bantah bicara ratu biaya aman harry meghan

6. *Tokenization*

Tokenization merupakan merupakan proses identifikasi kata kunci yang bermakna dengan cara memecah keseluruhan teks yang semula berbentuk kalimat menjadi kata per kata. Contoh data sebelum dan sesudah proses *tokenization* dapat dilihat pada Tabel 4.6.

Tabel 4.6 Contoh Data Sebelum dan Sesudah *Tokenization*

No.	Sebelum ***Tokenization***	Sesudah ***Tokenization***
1.	dpr rapat bumn pangan bahas sektor tani	'dpr' 'rapat' 'bumn' 'pangan' 'bahas' 'sektor' 'tani'
2.	dpr rapat jam menkes bpjs sehat hasil	'dpr' 'rapat' 'jam' 'menkes' 'bpjs' 'sehat' 'hasil'
3.	pm kanada bantah bicara ratu biaya aman harry meghan	'pm' 'kanada' 'bantah' 'bicara' 'ratu' 'biaya' 'aman' 'harry' 'meghan'

Pemecahan teks yang semula berbentuk kalimat menjadi kata per kata ditunjukkan pada Tabel 4.6. Hasil dari proses *tokenization* lalu digunakan sebagai kata kunci dari data judul berita. Langkah selanjutnya adalah membentuk struktur data baru dengan masing-masing kata kunci tersebut yang menjadi variabelnya dan diketahui frekuensi kemunculannya di setiap judul berita dengan menerapkan *syntax* pada Lampiran 4. Hasil dari proses tersebut ditunjukkan pada Tabel 4.7

Tabel 4.7 *Count Vectorizer* Kata dalam Judul Berita

No	abbas	⋯	bpjs	⋯	grab	⋯	kelas	⋯	koordinasi	⋯	zx
1	0	⋯	1	⋯	0	⋯	1	⋯	0	⋯	0
2	0	⋯	0	⋯	1	⋯	0	⋯	1	⋯	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
23	0	⋯	1	⋯	0	⋯	0	⋯	0	⋯	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
74	0	⋯	1	⋯	0	⋯	0	⋯	0	⋯	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
299	0	⋯	0	⋯	0	⋯	0	⋯	0	⋯	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
858	0	⋯	0	⋯	1	⋯	0	⋯	0	⋯	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
871	0	⋯	0	⋯	0	⋯	1	⋯	0	⋯	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1405	0	⋯	0	⋯	0	⋯	0	⋯	0	⋯	0

Tabel 4.7 menunjukkan hasil perhitungan frekuensi kemunculan kata pada setiap judul berita. Pada judul berita pertama, kata “bpjs” dan “kelas” muncul sebanyak satu kali. Sedangkan kata “abbas”, “grab”, “koordinasi”, dan “zx” tidak muncul pada judul berita pertama. Pada judul berita kedua, kata “grab” dan “koordinasi” muncul sebanyak satu kali. Sedangkan kata “abbas”, “bpjs”, “kelas”, dan “zx” tidak muncul pada judul berita kedua. Jumlah kata yang didapatkan dari data judul berita ini sebanyak 3.213 kata. Tabel 4.8 menampilkan frekuensi kemunculan kata tertinggi pada data judul berita. Setelah dilakukan proses *count vectorizer* dengan hasil sebagaimana yang ditampilkan pada Tabel 4.7, dilakukan perhitungan pembobotan untuk masing-masing kata menggunakan TF-IDF. Hasil perhitungan pembobotan dengan TF-IDF ditunjukkan pada Lampiran 2.

Tabel 4.8 Frekuensi Kemunculan Kata Tertinggi dalam Judul Berita

No.	Kata	Frekuensi	No.	Kata	Frekuensi
1	mobil	57	11	pakai	28
2	indonesia	43	12	ri	28
3	imlek	37	13	garuda	27
4	foto	35	14	jakarta	27
5	honda	33	15	corona	26
6	anak	32	16	tewas	26
7	orang	32	17	dunia	25
8	china	31	18	virus	25
9	kota	29	19	dpr	23
10	harga	28	20	jiwasraya	23

4.2.2 Klasifikasi Data menggunakan *Problem Transformation Binary Relevance* dengan Metode Klasifikasi Dasar KNN (BRKNN)

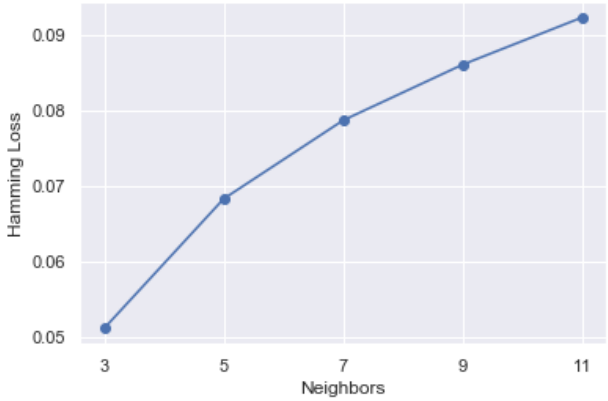
Binary Relevance (BR) adalah salah satu metode *problem transformation* untuk kasus *multi-label*. BR dilakukan dengan menerapkan *binary classifier* pada setiap label dan *output* yang dihasilkan akan digabungkan untuk membangun kumpulan label hasil prediksi. Metode klasifikasi dasar yang digunakan adalah metode KNN dengan nilai parameter *c* 3, 5, 7, 9, dan 11 serta perhitungan jarak *Euclidean*. Pembagian data *training* dan data *testing* dilakukan menggunakan *5-fold cross validation* dengan *syntax* yang terlampir pada Lampiran 5. Nilai *hamming loss* yang dihasilkan oleh BRKNN untuk data *training* ditunjukkan oleh Tabel 4.9.

Tabel 4.9 Ketepatan Klasifikasi Data *Training* BRKNN

Fold	***Hamming Loss***				
	3-NN	**5-NN**	**7-NN**	**9-NN**	**11-NN**
1	0,05308	0,06880	0,07844	0,08778	0,09223
2	0,05116	0,06702	0,07903	0,08823	0,09490
3	0,04730	0,06732	0,07711	0,08511	0,09401
4	0,05323	0,06732	0,07800	0,08244	0,08808
5	0,05086	0,07103	0,08096	0,08689	0,09253
Mean	**0,05113**(*)	**0,06830**	**0,07871**	**0,08609**	**0,09235**

Keterangan:
(*) Nilai *mean hamming loss* terkecil

Apabila divisualisasikan, nilai *hamming loss* yang didapatkan dari data *training* adalah sebagaimana pada Gambar 4.3. Metode klasifikasi dasar *3-Nearest Neighbor* menghasilkan nilai *hamming loss* terkecil, yakni senilai 0,05113. Hal tersebut ditunjukkan oleh Tabel 4.9 serta visualisasi pada Gambar 4.3. Oleh karena itu, pada kasus ini metode klasifikasi dasar *3-Nearest Neighbor* merupakan yang terbaik untuk metode *problem transformation* BR.



Gambar 4.3 Visualisasi Ketepatan Klasifikasi Data *Training* BRKNN

Selanjutnya, metode *problem transformation* BR dengan metode klasifikasi dasar *3-Nearest Neighbor* (BR3NN) digunakan untuk memprediksi data *testing*. Hasil perhitungan ketepatan klasifikasi untuk data *testing* menggunakan nilai *hamming loss* dapat dilihat pada Tabel 4.10.

Tabel 4.10 Ketepatan Klasifikasi BR3NN pada Data *Testing*

Fold	***Hamming Loss***
1	0,11329
2	0,12337
3	0,10142(*)
4	0,12989
5	0,11329
Mean	**0,11625**

Keterangan:
(*) *Fold* dengan nilai *hamming loss* terkecil

Nilai *hamming loss* yang dihasilkan oleh BR3NN sebagaimana ditunjukkan pada Tabel 4.10 adalah sebesar 0,11625. Nilai tersebut dapat diartikan bahwa dari prediksi label data *testing* yang dihasilkan terdapat 11,625% misklasifikasi pelabelan. Nilai *hamming loss* terkecil diperoleh pada *fold* ke-3, yakni sebesar 0,10142.

4.2.3 Klasifikasi Data menggunakan *Problem Transformation Label Powerset* dengan Metode Klasifikasi Dasar KNN (LPKNN)

Label Powerset (LP) juga salah satu metode *problem transformation* untuk kasus *multi-label*. Berbeda dengan BR, LP secara langsung mengubah kumpulan data yang *multi-label* menjadi satu kumpulan data yang *single-label* dengan cara mentransformasikan setiap kombinasi label yang berbeda pada data *multi-label* menjadi kelas yang berbeda pada data *single-label*. Sama halnya dengan BRKNN sebelumnya, metode klasifikasi dasar yang akan digunakan adalah KNN dengan nilai parameter *c* 3, 5, 7, 9, dan 11 serta perhitungan jarak *Euclidean*. Klasifikasi dilakukan menggunakan *5-fold cross validation* untuk membagi data menjadi data *training* dan data *testing* dengan *syntax* yang tercantum pada Lampiran 6. Nilai *hamming loss* yang dihasilkan metode LPKNN untuk data *training* ditunjukkan oleh Tabel 4.11.

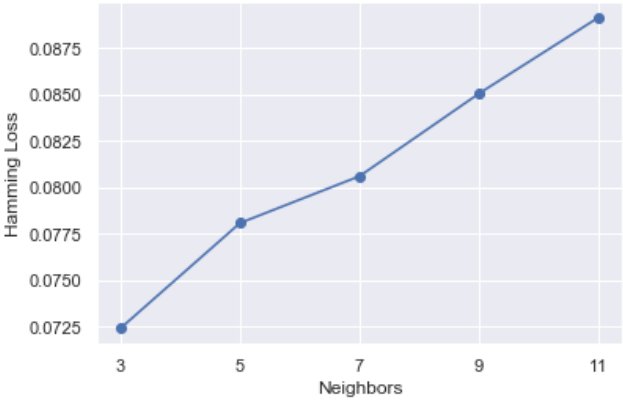
Tabel 4.11 Ketepatan Klasifikasi Data *Training* LPKNN

Fold	***Hamming Loss***				
	3-NN	**5-NN**	**7-NN**	**9-NN**	**11-NN**
1	0,07473	0,08037	0,08066	0,08749	0,09253
2	0,06940	0,07740	0,08081	0,08363	0,08808
3	0,07236	0,07340	0,07814	0,08170	0,08852
4	0,07192	0,07607	0,08230	0,08556	0,08541
5	0,07370	0,08319	0,08111	0,08689	0,09119
Mean	**0,07242**(*)	**0,07808**	**0,08060**	**0,08505**	**0,08915**

Keterangan:
(*) Nilai *mean hamming loss* terkecil

Apabila divisualisasikan, nilai *hamming loss* yang diperoleh untuk data *training* ditunjukkan oleh Gambar 4.4. Nilai *hamming loss*

yang ditunjukkan pada Tabel 4.11 dan Gambar 4.4 menunjukkan bahwa metode klasifikasi dasar *3-Nearest Neighbor* menghasilkan *hamming loss* terkecil, yaitu sebesar 0,07242. Oleh karena itu, pada kasus judul berita ini metode klasifikasi dasar yang terbaik untuk metode LP adalah *3-Nearest Neighbor*.



Gambar 4.4 Visualisasi Ketepatan Klasifikasi Data *Training* LPKNN

Selanjutnya, dilakukan prediksi label data *testing* menggunakan metode *problem transformation* LP dengan metode klasifikasi dasar *3-Nearest Neighbor* (LP3NN). Tabel 4.12 berikut menunjukkan nilai ketepatan klasifikasi untuk data *testing*.

Tabel 4.12 Ketepatan Klasifikasi LP3NN pada Data *Testing*

Fold	***Hamming Loss***
1	0,13582
2	0,14353
3	0,11981(*)
4	0,15718
5	0,13582
Mean	**0,13843**

Keterangan:
(*) *Fold* dengan nilai *hamming loss* terkecil

Metode LP3NN menghasilkan nilai *hamming loss* untuk data *testing* sebesar 0,13843. Hal tersebut dapat dilihat pada Tabel 4.12.

Nilai *hamming loss* tersebut menunjukkan bahwa dari prediksi label data *testing* yang dihasilkan terdapat 13,843% label yang mengalami misklasifikasi. Nilai *hamming loss* yang terkecil didapatkan pada *fold* ke-3, yakni sebesar 0,11981.

4.2.4 Klasifikasi Data menggunakan *Problem Transformation Classifier Chain* dengan Metode Klasifikasi Dasar KNN (CCKNN)

Classifier Chain (CC) merupakan metode *problem transformation* lainnya kasus *multi-label.* Seperti BR, CC melatih model sejumlah *k* atau sejumlah label yang ada. Hal yang membedakan adalah *classifier* pertama pada CC menggunakan data *input* yang asli dan kemudian *output* label pertama digunakan sebagai atribut input yang baru untuk melatih *classifier* kedua. Proses tersebut dilakukan terus menerus sampai *classifier* ke-*k*. Metode klasifikasi dasar, nilai parameter *c*, serta perhitungan jarak yang digunakan sama dengan BRKNN dan LPKNN. Klasifikasi dilakukan dengan menerapkan *5-fold cross validation* dengan *syntax* pada Lampiran 7. Nilai *hamming loss* yang dihasilkan oleh CCKNN dengan *5-fold cross validation* untuk data *training* ditunjukkan oleh Tabel 4.13.

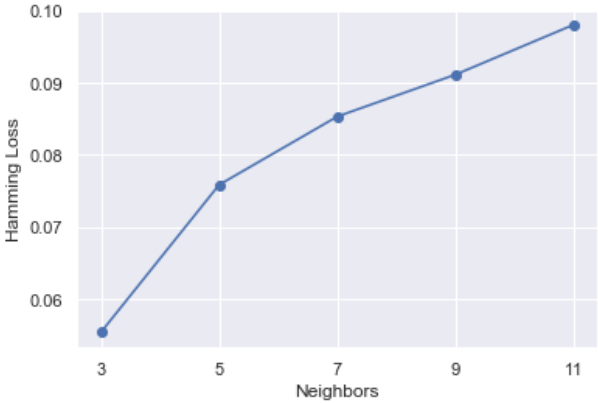
Tabel 4.13 Ketepatan Klasifikasi Data *Training* CCKNN

Fold	***Hamming Loss***				
	3-NN	**5-NN**	**7-NN**	**9-NN**	**11-NN**
1	0,05620	0,07770	0,08630	0,09223	0,10202
2	0,05724	0,07547	0,08571	0,09015	0,09698
3	0,04982	0,07740	0,08407	0,09149	0,09994
4	0,05827	0,07206	0,08259	0,08689	0,09149
5	0,05560	0,07666	0,08793	0,09490	0,09979
Mean	**0,05543**(*)	**0,07586**	**0,08532**	**0,09113**	**0,09804**

Keterangan:
(*) Nilai *mean hamming loss* terkecil

Visualisasi *nilai hamming loss* menggunakan metode CCKNN untuk data *training* dapat dilihat pada Gambar 4.5. Nilai *hamming loss* yang ditunjukkan oleh Tabel 4.13 dan Gambar 4.5 menunjukkan bahwa metode klasifikasi dasar *3-Nearest Neighbor* memiliki

nilai *hamming loss* terkecil, yakni sebesar 0,05543. Oleh karena itu, pada kasus ini metode klasifikasi dasar yang terbaik untuk metode *problem transformation* CC adalah *3-Nearest Neighbor*.



Gambar 4.5 Visualisasi Ketepatan Klasifikasi Data *Training* CCKNN

Metode *problem transformation* CC dengan metode klasifikasi dasar *3-Nearest Neighbor* (CC3NN) lalu digunakan untuk memprediksi data *testing*. Hasil perhitungan ketepatan klasifikasi untuk data *testing* menggunakan nilai *hamming loss* dapat dilihat pada Tabel 4.14.

Tabel 4.14 Ketepatan Klasifikasi CC3KNN pada Data *Testing*

Fold	***Hamming Loss***
1	0,11803
2	0,12811
3	0,10083(*)
4	013998
5	0,11210
Mean	**0,11981**

Keterangan:
(*) *Fold* dengan nilai *hamming loss* terkecil

Nilai *hamming loss* yang dihasilkan oleh metode CC3NN yakni sebesar 0,11981. Hal tersebut ditunjukkan pada Tabel 4.14. Nilai *hamming loss* sebesar 0,11981 tersebut menunjukkan bahwa dari

hasil label prediksi data *testing* terdapat 11,981% label yang mengalami misklasifikasi. Nilai *hamming loss* yang terkecil didapatkan pada *fold* ke-3, yakni sebesar 0,10083.

4.2.5 Perbandingan Nilai Ketepatan Klasifikasi menggunakan *Problem Transformation* dengan Metode Klasifikasi Dasar KNN

Setelah dilakukan klasifikasi baik menggunakan metode BRKNN, LPKNN, maupun CCKNN, langkah selanjutnya adalah membandingkan ketepatan klasifikasi yang telah dihasilkan oleh ketiga metode tersebut. Nilai ketepatan klasifikasi yang akan dibandingkan yakni nilai ketepatan klasifikasi yang terkecil pada masing-masing metode BRKNN, LPKNN, dan CCKNN. Perbandingan ketepatan klasifikasi ketiga metode *problem transformation* dengan metode klasifikasi dasar KNN tersebut dapat dilihat pada Tabel 4.15.

Tabel 4.15 Perbandingan Ketepatan Klasifikasi

Problem Transformation **dan Metode Klasifikasi Dasar**	***Hamming Loss***	
	Training	***Testing***
Binary Relevance dan *3-NN*	0,05113(*)	0,11625(*)
Label Powerset dan *3-NN*	0,07242	0,13843
Classifier Chain dan *3-NN*	0,05543	0,11981

Keterangan :
(*) Nilai *mean hamming loss* terkecil

Berdasarkan Tabel 4.15, hasil perbandingan ketiga metode *problem transformation* didapatkan bahwa metode BRKNN dengan metode klasifikasi dasar *3-Nearest Neighbor* memiliki nilai *hamming loss* untuk data *training* maupun data *testing* yang lebih kecil dibandingkan dua metode *problem transformation* lainnya, yakni berturut-turut sebesar 0,05113 dan 0,11625. Nilai *hamming loss* untuk data *testing* sebesar 0,11625 menunjukkan bahwa dari keseluruhan label prediksi yang dihasilkan untuk data *testing* menggunakan metode BRKNN dengan metode klasifikasi dasar *3-Nearest Neighbor*, 11,625% dari label prediksinya mengalami misklasifikasi. Beberapa contoh hasil klasifikasi menggunakan BRKNN

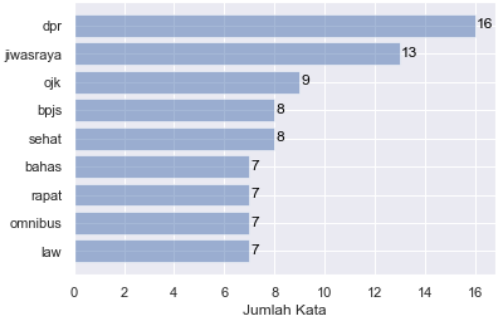
dengan metode klasifikasi dasar *3-Nearest Neighbor* ditunjukkan pada Lampiran 9.

4.3 Visualisasi *Word Cloud*

Visualisasi data teks menggunakan *word cloud* digunakan untuk mengetahui kata-kata yang paling sering muncul pada data. Pada penelitian ini, *word cloud* digunakan untuk mendapatkan kata kunci yang sering muncul pada judul berita berdasarkan kombinasi kategori berita. Kombinasi kategori berita tersebut merupakan hasil prediksi dari klasifikasi *multi-label* menggunakan metode *problem transformation Binary Relevance* dengan metode klasifikasi dasar *3-Nearest Neighbor*. Ukuran *font* pada *word cloud* menunjukkan frekuensi kemunculan kata. Semakin besar ukuran *font* kata maka semakin besar frekuensi kemunculan kata tersebut. Visualisasi *word cloud* dilakukan dengan menerapkan *syntax* yang terlampir pada Lampiran 8. Berikut ini adalah beberapa *word cloud* untuk setiap kombinasi kategori berita.

(a)

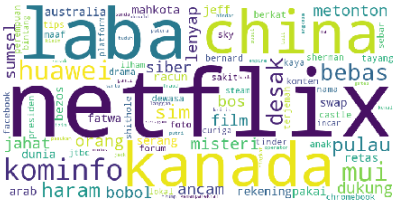
(b)



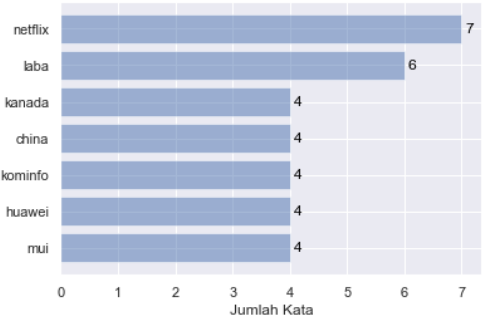
Gambar 4.6 (a) *Word Cloud* dan (b) *Bar Chart* Kategori detikFinance dan detikNews

Gambar 4.6 menunjukkan bahwa kata yang paling sering muncul pada judul berita di portal berita *online* detik.com dengan kombinasi kategori detikFinance dan detikNews adalah kata "dpr", "jiwasraya", dan "ojk". Hal tersebut sesuai dengan banyaknya pemberitaan mengenai kasus dugaan korupsi pada PT Asuransi Jiwasraya. Kasus tersebut kemudian melibatkan OJK dan DPR. Salah satu judul berita yang berkaitan dengan kata-kata tersebut yakni, "Buntut Masalah Jiwasraya, Komisi XI DPR Panggil OJK Besok".

(a)



(b)



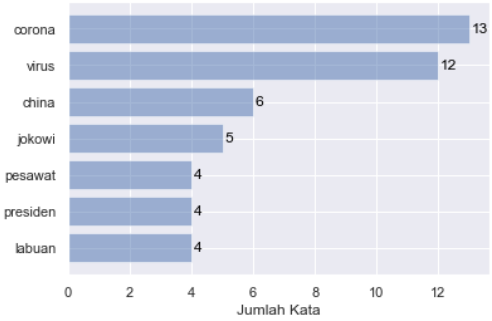
Gambar 4.7 (a) *Word Cloud* dan (b) *Bar Chart* Kategori detikInet dan detikNews

Kata yang paling sering muncul pada judul berita di portal berita *online* detik.com dengan kombinasi kategori detikInet dan detikNews sebagaimana ditunjukkan pada Gambar 4.7 adalah kata "netflix", "laba", "kanada", "china", "kominfo", "huawei", dan "mui". Kata "netflix", "kominfo", dan "mui" menjadi topik utama pada kombinasi label ini berkaitan dengan adanya pemberitaan mengenai konten dewasa pada Netflix serta adanya wacana fatwa haram terkait menonton Netflix. Sedangkan kata "laba" menjadi

topik utama selain kata “netflix” karena pemberitaan mengenai laba-laba beracun yang meneror Australia. Kata “kanada”, “china”, serta “huawei” menjadi topik utama lainnya berkaitan dengan China yang mendesak Kanada untuk membebaskan bos Huawei.

(a)

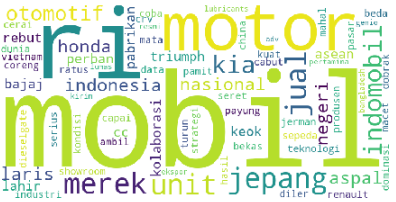
(b)



Gambar 4.8 (a) *Word Cloud* dan (b) *Bar Chart* Kategori detikTravel dan detikNews

Pada judul berita di portal berita *online* detik.com dengan kombinasi kategori detikTravel dan detikNews, kata yang paling sering muncul adalah kata “corona”, “virus” dan “china”. Hal tersebut ditunjukkan oleh Gambar 4.8. Kata-kata tersebut menjadi topik utama pada kombinasi label ini berkaitan dengan semakin mewabahnya virus corona yang berasal dari Wuhan, China. Beberapa judul yang berkaitan dengan kata-kata tersebut yakni “Virus Corona dan Imlek yang Menyedihkan di China” dan “Korut Bentengi Diri dari Virus Corona, Turis Dilarang Masuk!”.

(a)



(b)

Gambar 4.9 (a) *Word Cloud* dan (b) *Bar Chart* Kategori detikFinance dan detikOto

Kata "mobil", "motor" dan "ri" merupakan kata yang paling sering muncul pada judul berita di portal berita *online* detik.com dengan kombinasi kategori detikFinance dan detikOto sebagaimana yang ditunjukkan oleh Gambar 4.9. Kata-kata tersebut menjadi topik utama pada kombinasi label tersebut berkaitan dengan beragamnya pemberitaan mengenai industri otomotif di Indonesia. Beberapa contoh judul berita yang mengandung kata-kata tersebut adalah "Dua Merek Mobil Pamit, Coreng Industri Otomotif RI di Mata Dunia?" dan "Penjualan Motor di RI 6 Juta Unit, di Jepang Cuma 300 Ribu Unit". Beberapa *word cloud* untuk setiap kombinasi kategori berita lainnya dapat dilihat pada Lampiran 10.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan analisis dan pembahasan yang telah dilakukan, maka dapat diperoleh kesimpulan sebagai berikut.

1. Berita terbanyak dalam portal berita *online* detik.com yang rilis pada 20 Januari 2020 sampai dengan 26 Januari 2020 terdapat pada kategori detikNews, yakni sebanyak 496 berita. Hal tersebut disinyalir karena kategori detikNews memiliki cakupan berita yang lebih luas sehingga jumlah berita dengan kategori detikNews lebih banyak dibandingkan dengan kategori yang lain. Sedangkan berita paling sedikit terdapat pada kategori detikOto dengan jumlah berita sebanyak 208 berita. Selain itu, berita paling banyak memiliki satu label saja, yakni sebanyak 1.021 berita. Sedangkan berita paling sedikit memiliki empat label, yakni hanya satu berita saja. Jumlah berita dengan kategori lebih dari satu yakni sebanyak 384 berita.
2. Metode *Binary Relevance* dengan metode klasifikasi dasar *3-Nearest Neighbor* didapatkan hasil ketepatan klasifikasi dengan nilai *hamming loss* untuk data *training* dan data *testing* berturut-turut sebesar 0,05113 dan 0,11625. Metode *Label Powerset* dengan metode klasifikasi dasar *3-Nearest Neighbor* didapatkan hasil ketepatan klasifikasi dengan nilai *hamming loss* untuk data *training* dan data *testing* berturut-turut sebesar 0,07242 dan 0,13843. Metode *Classifier Chain* dengan metode klasifikasi dasar *3-Nearest Neighbor* didapatkan hasil ketepatan klasifikasi dengan nilai *hamming loss* untuk data *training* dan data *testing* berturut-turut sebesar 0,05543 dan 0,11981. Perbandingan antara ketiga metode *problem transformation*, yakni *Binary Relevance*, *Label Powerset*, dan *Classifier Chain* didapatkan bahwa hasil *Binary Relevance* dengan metode klasifikasi dasar *3-Nearest Neighbor* lebih baik dibandingkan kedua metode *problem transformation* yang lain.

3. Berdasarkan hasil visualisasi *word cloud*, didapatkan bahwa kata kunci yang sering muncul pada kombinasi kategori detikFinance dan detikNews adalah "dpr" berkaitan dengan pemberitaan mengenai penyikapan DPR terhadap permasalahan PT Jiwasraya (Persero). Kata kunci yang sering muncul pada kombinasi kategori detikInet dan detikNews yakni "netflix" berkaitan dengan konten dewasa pada Netflix serta adanya wacana fatwa haram terkait menonton Netflix. Lalu kata kunci yang sering muncul pada kombinasi kategori detikTravel dan detikNews adalah "corona" berkaitan dengan mewabahnya virus corona di seluruh dunia. Sedangkan kata "mobil" menjadi kata kunci yang sering muncul pada kombinasi kategori detikFinance dan detikOto berkaitan dengan pemberitaan mengenai perkembangan industri mobil di Indonesia.

5.2 Saran

Berdasarkan hasil analisis yang telah dijelaskan, maka saran yang dapat diberikan sebagai pertimbangan penelitian selanjutnya adalah menambahkan data baru terkait judul berita pada portal berita *online* detik.com agar kategori lainnya yang tidak masuk dalam penelitian ini dapat ditambahkan pada penelitian selanjutnya.

DAFTAR PUSTAKA

Adeniyi, D. A., Wei, Z., & Yongquan, Y. (2016). Automated Web Usage Data Mining and Recommendation System using K-Nearest Neighbor (KNN) Classification Method. *Applied Computing and Informatics*, 90-108.

Aggarwal, C. C., & Zhai, C. (2012). *Mining Text Data.* Boston: Springer.

Bholat, D., Hansen, S., Santos, P., & Schonhardt-Bailey, C. (2015). *Text Mining for Central Banks.* London: Bank of England.

Castellà, Q., & Sutton, C. (2014). Word Storms: Multiples of Word Clouds for Visual Comparison of Documents. *Proceedings of the 23rd International Conference on World Wide Web*, 665-676.

Charte, F., Rivera, A., del Jesus, M. J., & Herrera, F. (2012). Improving Multi-label Classifiers via Label Reduction with Association Rules. *International Conference on Hybrid Artificial Intelligence Systems* (pp. 188-199). Berlin: Springer.

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

Ghag, K., & Shah, K. (2014). SentiTFIDF – Sentiment Classification using Relative. *International Journal of Advanced Computer Science and Applications (IJACSA) Vol. 5 No. 2*, 36-43.

Gokgoz, E., & Subasi, A. (2015). Comparison of Decision Tree Algorithms for EMG Signal Classification using DWT. *Biomedical Signal Processing and Control*, 138-144.

Gorunescu, F. (2011). *Data Mining: Concepts, Models and Techniques.* Berlin: Springer.

Heider, D., Senge, R., Cheng, W., & Hüllermeier, E. (2013). Multilabel Classification for Exploiting Cross-Resistance Information in HIV-1 Drug Resistance Prediction. *Bioinformatics, 29.16*, 1946-1952.

Herrera, F., Charte, F., Rivera, A. J., & del Jesus, M. J. (2016). *Multilabel Classification: Problem Analysis, Metrics and Techniques.* Cham: Springer.

Hidayatullah, A. F., & Ma'Arif, M. R. (2017). Pre-processing Tasks in Indonesian Twitter Messages. *Journal of Physics: Conference Series, 801*(1), 012072.

Horne, M. v. (2007). *Menjadi Penulis: Membina Jemaat yang Menulis.* Jakarta: Gunung Mulia.

Isnaini, N., Adiwijaya, Mubarok, M. S., & Bakar, M. Y. (2019). A Multi-Label Classification on Topics of Indonesian News. *Journal of Physics: Conference Series, 1192*.

Margianto, J. H., & Syaefullah, A. (n.d.). *Media Online: Antara Pembaca, Laba, dan Etika Problematika Praktik Jurnalisme Online di Indonesia.* Jakarta Pusat: Aliansi Jurnalis Independen (AJI) Indonesia.

Mittal, N., Agarwal, B., Agarwal, S., Agarwal, S., & Gupta, P. (2013). A Hybrid Approach for Twitter Sentiment Analysis. *10th International Conference on Natural Language Processing (ICON-2013)*, (pp. 116-120). Noida.

Musfah, J. (2018). *Analisis Kebijakan Pendidikan Mengurai Krisis Karakter Bangsa.* Jakarta: Kencana.

Nova, H. A. (2011). *Panduan Lengkap Internet lewat Ponsel Java.* Jakarta: PT Elex Media Komputindo.

Noviyanto. (2011, Agustus 4). *Home - News - LensaIndonesia.com*. Retrieved from Detikcom Resmi Dibeli Chairul Tanjung Transcorp Rp 540 Miliar: https://www.lensaindonesia.com

Pambudi, R. A., Adiwijaya, & Mubarok, M. S. (2019). Multi-Label Classification of Indonesian News Topics using Pseudo Nearest Neighbor Rule. *Journal of Physics: Conference Series, 1192*.

Prasath, V. B., Alfeilat, H. A., Hassanat, A. B., Lasassmeh, O., Tarawneh, A. S., Alhassanat, M. B., & Salman, H. S. (2017). Distance and Similarity Measures Effect on the Performance of K-Nearest Neighbor Classifier--A Review. *arXiv preprint arXiv:1708.04321*.

Prawira, I. M., Adiwijaya, & Mubarok, M. S. (2018). Klasifikasi Multi-Label Pada Topik Berita Berbahasa Indonesia Menggunakan Multinomial Naïve Bayes. *eProceedings of Engineering*.

Ramos, J. (2003). Using TF-IDF to Determine Word Relevance in Document Queries. *Proceedings of the First Instructional Conference on Machine Learning*, (pp. 133-142).

Rodríguez, J. D., Pérez, A., & Lozano, J. A. (2009). Sensitivity Analysis of k-Fold Cross Validation in Prediction Error Estimation. *IEEE transactions on pattern analysis and machine intelligence, 32.3*, 569-575.

Schneider, J. (1997). *Cross Validation*. Retrieved from Carnegie Mellon School of Computer Science: https://www.cs.-cmu.edu/~schneide/tut5/node42.html

Spolaor, N., Cherman, E. A., Monard, M. C., & Lee, H. D. (2013). A Comparison of Multi-Label Feature Selection Methods. *Electronic Notes in Theoretical Computer Science Vol. 292*, 135-151.

Verma, T., Renu, & Gaur, D. (2014). Tokenization and Filtering Process in RapidMiner. *International Journal of Applied Information Systems (IJAIS) Volume 7 No. 2*, 16-18.

(*Halaman ini sengaja dikosongkan*)

LAMPIRAN

Lampiran 1. Data Berita Portal Berita *Online* Detik.com

Tanggal	Judul Berita	detikFinance	detikOto	detikHot	detikInet	detikTravel	detikNews
1/20/2020	Peserta BPJS Kesehatan Kelas III Bisa Pindah ke PBI	1	0	0	0	0	0
1/20/2020	Grab Siap Berkoordinasi dengan Kemenhub Bahas Penyesuaian Tarif Ojol	1	0	0	0	0	1
1/20/2020	Soal Cost Recovery atau Gross Split, Ini Kata Kontraktor Migas	1	0	0	0	0	0
1/20/2020	BPJS Kesehatan Bantah Terima Insentif Ratusan Juta/Bulan	1	0	0	0	0	0
1/20/2020	Bocoran Skenario PNS Pusat Hijrah ke Ibu Kota Baru 2024	1	0	0	0	0	1
1/20/2020	Ibu Kota Dipindah ke Kaltim, Ini Dua Opsi Buat PNS	1	0	0	0	0	1
1/20/2020	Pemerintah Tepis Omnibus Law Hilangkan Cuti Hamil	1	0	0	0	0	1
1/20/2020	Aceh Termiskin di Sumatera, Pemprov Genjot Pertanian & UMKM	1	0	0	0	0	1
1/20/2020	Terawan Tunggu BPJS Kesehatan Transparansi Data Keuangan	1	0	0	0	0	1
1/20/2020	Chevron Sebut Banyak yang Curi Minyak di Blok Rokan	1	0	0	0	0	0
1/20/2020	Cari Dana Buat Bayar Utang, PP Properti Terbitkan Obligasi Rp 1,2 T	1	0	0	0	0	0
1/20/2020	DPR Rapat 7 Jam dengan Menkes dan BPJS Kesehatan, Ini Hasilnya	1	0	0	0	0	1
1/20/2020	Reformasi Industri Asuransi hingga Perbaiki Citra Pasar Modal	1	0	0	0	0	0
1/20/2020	Heboh Ditolak Buruh, Apa Sih Omnibus Law Itu?	1	0	0	0	0	1

Lampiran 1. Data Berita Portal Berita *Online* Detik.com (lanjutan)

1/20/2020	Pasokan Seret Bikin Harga Cabai Melonjak Jelang Imlek	1	0	0	0	0	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1/23/2020	Korban Jiwa Virus Corona di China Meningkat Nyaris 2 Kali Lipat dalam Sehari	0	0	0	0	0	1
1/23/2020	Fakta tentang Laba-laba Paling Beracun di Dunia yang Mengancam Australia	0	0	0	1	0	1
1/23/2020	Trump Sebut Beberapa Negara Akan Ditambahkan ke Daftar Larangan Masuk AS	0	0	0	0	1	1
1/23/2020	Seorang Mahasiswa Iran Dilarang Masuk ke AS dan Dipulangkan	0	0	0	0	0	1
1/23/2020	Trump Sebut Boeing sebagai 'Kekecewaan Besar'	1	0	0	0	0	1
1/23/2020	Jerman Kembalikan Karya Seni yang Dicuri Nazi ke Keluarga Prancis	0	0	1	0	0	1
1/23/2020	PM Kanada Bantah Bicara ke Ratu soal Biaya Keamanan Harry-Meghan	0	0	1	0	0	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1/26/2020	Ramahnya Will Smith saat Jadi Driver Taksi Online	0	1	1	0	0	0
1/26/2020	Kabar 'Penjegal' Avanza hingga Rush dari China	0	1	0	0	0	0
1/26/2020	Hyundai Tucson Tampang Baru Dibanderol Mulai Rp 464 Juta	0	1	0	0	0	0
1/26/2020	Antisipasi Pabrikan Mobil di China Hadapi Wabah Virus Corona	0	1	0	0	0	1
1/26/2020	Tampang Baru SUV Mahindra Usai Disemprit Jeep	0	1	0	0	0	0
1/26/2020	Wuhan, Motor City yang Lumpuh Karena Virus Corona	0	1	0	0	0	1
1/26/2020	Kok Bisa Duo Marquez, Rossi dan Vinales Kompakan ke Jakarta?	0	1	0	0	0	0

Lampiran 2. Hasil Pembobotan TF-IDF

No.	abal	⋯	bmw	bni	bobol	⋯	bocah	bocor	⋯	zx
1	0	⋯	0	0	0	⋯	0	0	⋯	0
2	0	⋯	0	0	0	⋯	0	0	⋯	0
3	0	⋯	0	0	0	⋯	0	0	⋯	0
4	0	⋯	0	0	0	⋯	0	0	⋯	0
5	0	⋯	0	0	0	⋯	0	0.381957	⋯	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
181	0	⋯	0	0	0	⋯	0	0	⋯	0
182	0	⋯	0	0.456464	0	⋯	0	0	⋯	0
183	0	⋯	0	0	0	⋯	0	0	⋯	0
184	0	⋯	0	0	0	⋯	0	0	⋯	0
185	0	⋯	0	0	0	⋯	0	0	⋯	0
186	0	⋯	0	0	0	⋯	0	0	⋯	0
187	0	⋯	0	0	0	⋯	0	0	⋯	0
188	0	⋯	0	0	0	⋯	0	0.386216	⋯	0
189	0	⋯	0	0.530197	0	⋯	0	0	⋯	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
439	0	⋯	0	0	0	⋯	0	0.331202	⋯	0
440	0	⋯	0	0	0	⋯	0	0	⋯	0
441	0	⋯	0	0	0	⋯	0	0	⋯	0
442	0	⋯	0	0	0	⋯	0	0	⋯	0
443	0	⋯	0.418517	0	0	⋯	0	0	⋯	0
444	0	⋯	0	0	0	⋯	0	0	⋯	0
445	0	⋯	0	0	0	⋯	0	0	⋯	0
446	0	⋯	0.418855	0	0	⋯	0	0	⋯	0
447	0	⋯	0	0	0	⋯	0	0	⋯	0
448	0	⋯	0	0	0	⋯	0	0	⋯	0
449	0	⋯	0	0	0	⋯	0	0.44514	⋯	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
861	0	⋯	0	0	0	⋯	0	0.381481	⋯	0
862	0	⋯	0	0	0	⋯	0	0	⋯	0
863	0	⋯	0	0	0	⋯	0	0	⋯	0
864	0	⋯	0	0	0	⋯	0	0	⋯	0
865	0	⋯	0	0	0	⋯	0	0	⋯	0
866	0	⋯	0	0	0	⋯	0	0	⋯	0
867	0	⋯	0	0	0.410328	⋯	0	0	⋯	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1401	0	⋯	0	0	0	⋯	0	0	⋯	0
1402	0	⋯	0	0	0	⋯	0	0	⋯	0
1403	0	⋯	0	0	0	⋯	0	0	⋯	0
1404	0	⋯	0	0	0	⋯	0	0	⋯	0
1405	0	⋯	0	0	0	⋯	0	0	⋯	0

Lampiran 3. *Syntax Pre-Processing*

a. *Case Folding*

```
import string, nltk, re, ast

news_title = news['title_name']
news_title = news_title.astype(str)
news_title.head()

# Lowercase
datalower = []
for line in news_title:
    result = line.lower()
    datalower.append(result)
print(datalower)
```

b. *Slang Handling*

```
# Slang Handling
kata = ast.literal_eval(open('D:/ITS/DATA
AKADEMIK/SEMESTER 8/OneDrive/TUGAS
AKHIR/Run/similarity.txt','r').read())

from collections import OrderedDict
def replace_all(datalower, dic):
    for i, j in dic.items():
        datalower = datalower.replace(i, j)
    return datalower
dic = OrderedDict(kata)

dataedit = []
for line in datalower:
    result = replace_all(line, dic)
    dataedit.append(result)
print(dataedit)
```

Lampiran 3. *Syntax Pre-Processing* (lanjutan)

c. *Removing Numbers and Punctuations*

```
# Remove Punctuation [!”#$%&’()*+,-./:;<=>?@[\]^`{|}~]
datanopunct = []
for line in dataedit:
    result = re.sub(r"[^\w\s]"," ",line)
    datanopunct.append(result)
print(datanopunct)

# Remove Number
datanonumber = []
for line in datanopunct:
    result = re.sub("\d"," ",line)
    datanonumber.append(result)
print(datanonumber)

# Clear Space Enter
dataclearspace = []
for line in datanonumber:
    result = re.sub(r"\s+"," ",line)
    dataclearspace.append(result)
print(dataclearspace)
```

d. *Stemming*

```
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory

# Stemming
factory = StemmerFactory()
stemmer = factory.create_stemmer()
datastemmed = map(lambda x: stemmer.stem(x), dataclearspace)
datastemmed = list(datastemmed)
print(datastemmed)
```

Lampiran 3. *Syntax Pre-Processing* (lanjutan)

e. *Stopwords* dan *Tokenization*

```
from nltk.tokenize import word_tokenize

# Stopwords and Tokenization
stopwords = open('D:/ITS/DATA AKADEMIK/SEMESTER
8/OneDrive/TUGAS
AKHIR/Run/stopwords_id.txt','r').read().split()
datatokenized = []
df = []
for line in datastemmed:
    word_token = nltk.word_tokenize(line)
    word_token = [word for word in word_token if not word in
stopwords]
    datatokenized.append(word_token)
    df.append(" ".join(word_token))
print(datatokenized)
```

Lampiran 4. *Syntax Count Vectorizer* dan TF-IDF

```
from pandas import DataFrame
from sklearn.feature_extraction.text import CountVectorizer

# Count Vectorizer
vector = CountVectorizer(min_df = 1)
df_counted = vector.fit_transform(df)
df_counted = df_counted.toarray()
feature_names = vector.get_feature_names()
feature_counted = DataFrame(df_counted, columns = feature_names)
feature_counted.to_excel('count_vectorizer.xlsx')
feature_counted.head()

from sklearn.feature_extraction.text import TfidfTransformer

# TF-IDF
feature_tfidf = TfidfTransformer(use_idf =
True).fit_transform(feature_counted)
feature_tfidf = DataFrame(feature_tfidf.A, columns = feature_names)
feature_tfidf.to_excel('TFIDF.xlsx')
feature_tfidf.head()

X = feature_tfidf
Y = DataFrame(news.iloc[:,1:])

print('X shape : ', X.shape)
print('Y shape : ', Y.shape)
```

Lampiran 5. *Syntax K-Fold Cross Validation* BRKNN

```
from sklearn.model_selection import KFold
from sklearn.neighbors import KNeighborsClassifier
from skmultilearn.problem_transform import BinaryRelevance
import numpy as np
from sklearn.metrics import hamming_loss

kf = KFold(n_splits = 5, random_state = 0, shuffle = True)
kf.get_n_splits(X)
print(kf)
print('\n')

neighbors = [3,5,7,9,11]
hamming_loss_avg_BRKNN = []

for n in neighbors:
    print('BRKNN with k = ',str(n))
    hamming_loss_test = []
    hamming_loss_train = []

    for train_index, test_index in kf.split(X):
        X_train, X_test = X.iloc[train_index], X.iloc[test_index]
        y_train, y_test = Y.iloc[train_index], Y.iloc[test_index]
        BRKNN = BinaryRelevance(KNeighborsClassifier(n_neighbors = n,
         p = 2))
        BRKNN.fit(X_train, y_train)
        y_pred = BRKNN.predict(X_test)
        y_predt = BRKNN.predict(X_train)

        hamming_loss_test_fold = hamming_loss(y_test, y_pred)
        hamming_loss_train_fold = hamming_loss(y_train, y_predt)
        print("Hamming Loss Test :", hamming_loss_test_fold)
        print("Hamming Loss Train :", hamming_loss_train_fold)
        hamming_loss_test.append(hamming_loss_test_fold)
        hamming_loss_train.append(hamming_loss_train_fold)

    print("Average Hamming Loss Test :", np.mean(hamming_loss_test))
    print("Average Hamming Loss Train :", np.mean(hamming_loss_train))
```

Lampiran 5. *Syntax K-Fold Cross Validation* BRKNN (lanjutan)

```
    print('\n')
    hamming_loss_avg_BRKNN.append(np.mean(hamming_loss_train))

# Plotting Hamming Loss Average Results
plt.plot(['3','5','7','9','11'], hamming_loss_avg_BRKNN, 'ro-')
plt.xlabel('Neighbors', fontsize = 12)
plt.ylabel('Hamming Loss', fontsize = 12)
plt.show()
```

Lampiran 6. *Syntax K-Fold Cross Validation* LPKNN

```
from sklearn.model_selection import KFold
from sklearn.neighbors import KNeighborsClassifier
from skmultilearn.problem_transform import LabelPowerset
import numpy as np
from sklearn.metrics import hamming_loss

kf = KFold(n_splits = 5, random_state = 0, shuffle = True)
kf.get_n_splits(X)
print(kf)
print('\n')

neighbors = [3,5,7,9,11]
hamming_loss_avg_LPKNN = []

for n in neighbors:
    print('LPKNN with k = ',str(n))
    hamming_loss_test = []
    hamming_loss_train = []

    for train_index, test_index in kf.split(X):
        X_train, X_test = X.iloc[train_index], X.iloc[test_index]
        y_train, y_test = Y.iloc[train_index], Y.iloc[test_index]
        LPKNN = LabelPowerset(KNeighborsClassifier(n_neighbors = n,
        p = 2))
        LPKNN.fit(X_train, y_train)
        y_pred = LPKNN.predict(X_test)
        y_predt = LPKNN.predict(X_train)

        hamming_loss_test_fold = hamming_loss(y_test, y_pred)
        hamming_loss_train_fold = hamming_loss(y_train, y_predt)
        print("Hamming Loss Test :", hamming_loss_test_fold)
        print("Hamming Loss Train :", hamming_loss_train_fold)
        hamming_loss_test.append(hamming_loss_test_fold)
        hamming_loss_train.append(hamming_loss_train_fold)

    print("Average Hamming Loss Test :", np.mean(hamming_loss_test))
    print("Average Hamming Loss Train :",
```

Lampiran 6. *Syntax K-Fold Cross Validation* BRKNN (lanjutan)

```
    np.mean(hamming_loss_train))
    print('\n')
    hamming_loss_avg_LPKNN.append(np.mean(hamming_loss_train))

# Plotting Hamming Loss Average Results
plt.plot(['3','5','7','9','11'], hamming_loss_avg_LPKNN, 'ro-')
plt.xlabel('Neighbors', fontsize = 12)
plt.ylabel('Hamming Loss', fontsize = 12)
plt.show()
```

Lampiran 7. *Syntax K-Fold Cross Validation* CCKNN

```
from sklearn.model_selection import KFold
from sklearn.neighbors import KNeighborsClassifier
from skmultilearn.problem_transform import ClassifierChain
import numpy as np
from sklearn.metrics import hamming_loss

kf = KFold(n_splits = 5, random_state = 0, shuffle = True)
kf.get_n_splits(X)
print(kf)
print('\n')

neighbors = [3,5,7,9,11]
hamming_loss_avg_CCKNN = []

for n in neighbors:
    print('CCKNN with k = ',str(n))
    hamming_loss_test = []
    hamming_loss_train = []

    for train_index, test_index in kf.split(X):
        X_train, X_test = X.iloc[train_index], X.iloc[test_index]
        y_train, y_test = Y.iloc[train_index], Y.iloc[test_index]
        CCKNN = ClassifierChain(KNeighborsClassifier(n_neighbors = n,
        p = 2))
        CCKNN.fit(X_train, y_train)
        y_pred = CCKNN.predict(X_test)
        y_predt = CCKNN.predict(X_train)

        hamming_loss_test_fold = hamming_loss(y_test, y_pred)
        hamming_loss_train_fold = hamming_loss(y_train, y_predt)
        print("Hamming Loss Test :", hamming_loss_test_fold)
        print("Hamming Loss Train :", hamming_loss_train_fold)
        hamming_loss_test.append(hamming_loss_test_fold)
        hamming_loss_train.append(hamming_loss_train_fold)

    print("Average Hamming Loss Test :", np.mean(hamming_loss_test))
    print("Average Hamming Loss Train :",
```

Lampiran 7. *Syntax K-Fold Cross Validation* BRKNN (lanjutan)

```
    np.mean(hamming_loss_train))
    print('\n')
   hamming_loss_avg_CCKNN.append(np.mean(hamming_loss_
   train))

# Plotting Hamming Loss Average Results
plt.plot(['3','5','7','9','11'], hamming_loss_avg_CCKNN, 'ro-')
plt.xlabel('Neighbors', fontsize = 12)
plt.ylabel('Hamming Loss', fontsize = 12)
plt.show()
```

Lampiran 8. *Syntax Word Cloud*

```
from wordcloud import WordCloud
import matplotlib.pyplot as plt

 wordcloud = WordCloud(width = 800, height = 400,
                    max_words = 100,
                    max_font_size = 300,
                    min_font_size = 10,
                    random_state = 0,
                    background_color ='white').generate(result)

# Plot the Word Cloud Image
plt.figure(figsize = (8, 4), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.show()
```

Lampiran 9. Contoh Hasil Klasifikasi *Multi-Label* Judul Berita

Contoh hasil klasifikasi *multi-label* menggunakan BRKNN dengan metode klasifikasi dasar *3-Nearest Neighbor*.

No	Judul Berita	Label Aktual	Label Prediksi
1.	Reformasi Industri Asuransi hingga Perbaiki Citra Pasar Modal	detikFinance	detikFinance
2.	Kinerja 2019 BNI, Bisnis Internasional Semakin Tangguh	detikFinance	detikFinance
3.	Makin Banyak Gedung Perkantoran Kosong di Jakarta	detikFinance	detikFinance
4.	Benarkah Omnibus Law Permudah Kongkalikong dengan Asing?	detikFinance	detikFinance, detikNews
5.	Pengakuan Tata Janeeta Dikaitkan Investasi Bodong MeMiles	detikFinance, detikHot, detikNews	detikFinance, detikHot, detikNews
6.	Arti di Balik Nama Alugoro Kapal Selam Made In RI	detikFinance, detikInet, detikNews	detikFinance, detikInet, detikNews
7.	Punya Alugoro, RI Satu-satunya Pembuat Kapal Selam di ASEAN	detikFinance, detikInet, detikNews	detikFinance, detikInet, detikNews
8.	Pemerintah Tepis Omnibus Law Hilangkan Cuti Hamil	detikFinance, detikNews	detikFinance, detikNews
9.	118.000 PNS Hijrah ke Ibu Kota Baru, Ongkosnya Ditanggung Siapa?	detikFinance, detikNews	detikFinance, detikNews
10.	Saksi Kunci Kasus MeMiles Tengah Diperiksa, Hasilnya?	detikFinance, detikNews	detikFinance, detikHot, detikNews
11.	Usai Diperiksa Kejagung, Saksi Terkait Kasus Jiwasraya Bungkam	detikFinance, detikNews	detikFinance, detikNews
12.	Kontraktor Monas Menyoal Tuduhan Abal-abal dan Kantor Virtual	detikFinance, detikNews	detikFinance, detikNews
13.	BBN-KB Gratis Belum Tentu Bisa Angkat Penjualan Motor Listrik	detikFinance, detikOto, detikNews	detikFinance, detikOto, detikNews
14.	Mulai Akhir Tahun, Kapal Pesiar Bisa Parkir di Labuan Bajo	detikFinance, detikTravel	detikTravel
15.	Menang Lagi, 'Parasite' Buat Sejarah di SAG Awards 2020	detikHot	detikHot

Lampiran 9. Contoh Hasil Klasifikasi *Multi-Label* Judul Berita (lanjutan)

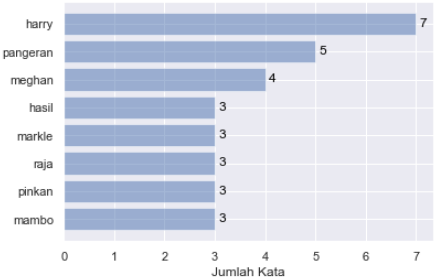
No	Judul Berita	Label Aktual	Label Prediksi
16.	Gokil! Chris Jericho Satukan Musik dengan Gulat di Kapal Pesiar	detikHot	detikHot
17.	Daftar Pemenang Lengkap SAG Awards 2020	detikHot	detikHot
18.	Pengalaman Pertama Dian Sastro Main Film Sekaligus Jadi Produser	detikHot	detikHot
19.	Chanyeol 'EXO' dan Punch Reuni untuk OST 'Dr. Romantic 2'	detikHot	detikHot
20.	Kourtney Kardashian Hingga Victoria Beckham yang Newbie di Tik Tok	detikHot, detikInet	detikInet, detikNews
21.	Pangeran Harry: Aku Akan Selalu Menghormati Nenekku yang Dukung Meghan!	detikHot, detikNews	detikHot, detikNews
22.	Jerman Kembalikan Karya Seni yang Dicuri Nazi ke Keluarga Prancis	detikHot, detikNews	detikHot
23.	Sony GP-VPT2BT, Tongsis yang Bantu Vlog Lebih Mudah	detikInet	detikInet
24.	Desain iPhone 12 Mungkin Mengejutkan	detikInet	detikInet
25.	Penjelasan Lengkap MUI Terkait Netflix	detikInet, detikNews	detikInet, detikNews
26.	Kominfo Sebut Netflix Mau Bayar Pajak, Tapi...	detikInet, detikNews	detikInet, detikNews
27.	Ilham Bintang Rugi Ratusan Juta dari Pembobolan Ponsel dan Rekening	detikInet, detikNews	detikFinance, detikNews
28.	Ngurah Rai Jadi Bandara ke-23 yang Layani GrabCar Airport	detikInet, detikTravel	detikInet, detikTravel
29.	2 Remaja Pemalak Sopir Truk di Jakut Dibekuk Polisi	detikNews	detikNews
30.	Polres Pelabuhan Priok Tangkap 3 Pengoplos Miras, Ribuan Botol Disita	detikNews	detikNews
31.	8 Pengedar Narkoba di Bandung Ditangkap, 114 Gram Sabu Diamankan	detikNews	detikNews

Lampiran 9. Contoh Hasil Klasifikasi Multi-Label Judul Berita (lanjutan)

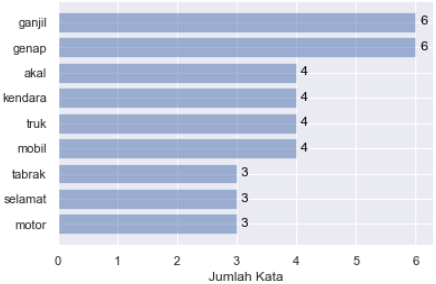
No	Judul Berita	Label Aktual	Label Prediksi
32.	KPK Periksa Ibu Eks Bupati Mojokerto Terkait Kasus Pencucian Uang	detikNews	detikFinance, detikNews
33.	XL7 Sudah Bisa Dipesan Sebelum Launching, Suzuki?	detikOto	detikOto
34.	Mobil Listrik Naik Gunung, Pecahkan Rekor Dunia	detikOto	detikOto
35.	Empat Komponen Potensi Bermasalah, Suzuki Recall Address FI	detikOto	detikOto
36.	Akal-akalan Pengendara biar Lolos dari Ganjil-Genap	detikOto, detikNews	detikOto, detikNews
37.	Mobil SUV Lindas Pria di Bandung, Mobilnya Langsung Kabur	detikOto, detikNews	detikOto
38.	Pemotor Langgar Lalu Lintas Bakal Dipantau CCTV	detikOto, detikNews	detikNews
39.	Bergaya Ala Pemimpin Nazi, Biker Ini Dicari Polisi	detikOto, detikNews	detikNews
40.	Boonpring, Ekowisata nan Asyik di Malang	detikTravel	detikTravel
41.	Negara-negara di Benua Amerika yang Bebas Visa buat Paspor RI	detikTravel	detikTravel
42.	Cuma di Ponorogo, Kemping dengan Pemandangan Secantik Ini	detikTravel	detikTravel
43.	Waspada Virus Corona, Arab Saudi Awasi Turis China	detikTravel, detikNews	detikTravel, detikNews
44.	Wabah Virus Corona, Kota Terlarang di China Tutup!	detikTravel, detikNews	detikTravel, detikNews
45.	Ingin Labuan Bajo Jadi Wisata Super Premium, Ini Jurus Jokowi	detikTravel, detikNews	detikFinance, detikTravel, detikNews

Lampiran 10. Visualisasi *Word Cloud*

a. *Word Cloud* dan *Bar Chart* Kategori detikHot dan detikNews

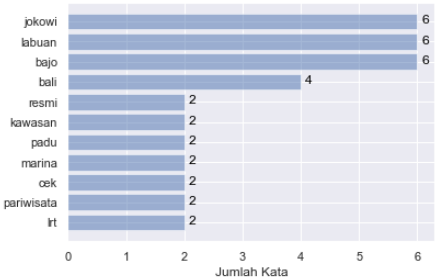


b. *Word Cloud* dan *Bar Chart* Kategori detikOto dan detikNews

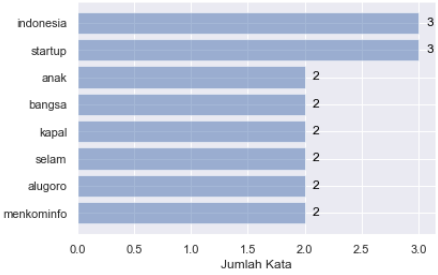


Lampiran 10. Visualisasi *Word Cloud* (lanjutan)

c. *Word Cloud* dan *Bar Chart* Kategori detikFinance, detikTravel, dan detikNews

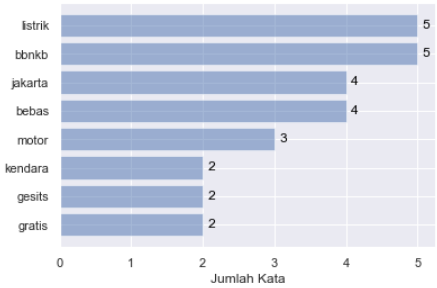


d. *Word Cloud* dan *Bar Chart* Kategori detikFinance, detikInet, dan detikNews

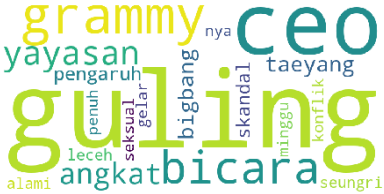


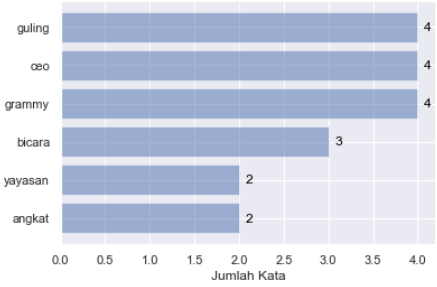
Lampiran 10. Visualisasi *Word Cloud* (lanjutan)

e. *Word Cloud* dan *Bar Chart* Kategori detikFinance, detikOto, dan detikNews



f. *Word Cloud* dan *Bar Chart* Kategori detikFinance dan detikHot





BIODATA PENULIS

Penulis lahir di Surabaya, 1 Oktober 1998 dengan nama lengkap Oktavia Ramadhani dan biasa dipanggil Via. Penulis menempuh pendidikan formal di SD Negeri Manukan Kulon Kawasan Surabaya, SMP Negeri 2 Surabaya, dan SMA Negeri 11 Surabaya. Penulis kemudian diterima sebagai mahasiswa Departemen Statistika ITS melalui jalur masuk SBMPTN pada tahun 2016. Selama masa perkuliahan, penulis aktif menjabat di beberapa organisasi di ITS, yakni sebagai Asisten Direktur Jenderal Pemetaan & Pemantauan Kementerian Pengembangan Sumber Daya Mahasiswa (PSDM) BEM ITS periode 2019-2020, Staf Kementerian PSDM BEM ITS periode 2018-2019, Wakil Ketua II Departemen PSDM HIMASTA-ITS periode 2018-2019, dan Staf PSDM HIMASTA-ITS periode 2017-2018. Selain aktif berorganisasi, penulis juga berkesempatan aktif dalam berbagai kepanitiaan yang ada di ITS, beberapa di antaranya yakni sebagai Staf Kesekretariatan ITS EXPO 2017, Pemandu Integralistik GERIGI ITS 2018, Koordinator Sie Kesekretariatan PRS 2018, dan berbagai kepanitiaan lainnya. Penulis juga pernah berkesempatan menjadi *marketing intern* di PT. Telekomunikasi Indonesia, Tbk Regional II Jakarta. Bagi pembaca yang ingin berdiskusi serta memberi kritik dan saran terkait Tugas Akhir ini, dapat menghubungi *e-mail* penulis pada oktaviaramadhani0110@gmail.com.

(*Halaman ini sengaja dikosongkan*)