TUGAS AKHIR – KS184822

KLASIFIKASI *MULTI-LABEL* PADA ARTIKEL JURNAL SCIENCEDIRECT DENGAN METODE *K-NEAREST NEIGHBOR* (KNN) DAN *SUPPORT VECTOR MACHINE* (SVM)

RIFQI RABBANIE
NRP 062116 4000 0096

Dosen Pembimbing
Dr. Dra. Kartika Fithriasari, M.Si.
Dra. Wiwiek Setya Winahju, M.S.

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

TUGAS AKHIR – KS184822

KLASIFIKASI *MULTI-LABEL* PADA ARTIKEL JURNAL SCIENCEDIRECT DENGAN METODE *K-NEAREST NEIGHBOR* (KNN) DAN *SUPPORT VECTOR MACHINE* (SVM)

RIFQI RABBANIE
NRP 062116 4000 0096

Dosen Pembimbing
Dr. Dra. Kartika Fithriasari, M.Si.
Dra. Wiwiek Setya Winahju, M.S.

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

FINAL PROJECT – KS184822

MULTI-LABEL CLASSIFICATION ON JOURNAL ARTICLE OF SCIENCEDIRECT USING CLASSIFER K-NEAREST NEIGHBOR (KNN) AND SUPPORT VECTOR MACHINE (SVM)

RIFQI RABBANIE
SN 062116 4000 0096

SUPERVISORS
Dr. Dra. Kartika Fithriasari, M.Si.
Dra. Wiwiek Setya Winahju, M.S.

UNDERGRADUATE PROGRAMME
DEPARTEMEN OF STATISTICS
FACULTY OF SCIENTICS AND DATA ANALYTICS
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2020

LEMBAR PENGESAHAN

KLASIFIKASI *MULTI-LABEL* PADA ARTIKEL JURNAL SCIENCEDIRECT DENGAN METODE *K-NEAREST NEIGHBOR* (KNN) DAN *SUPPORT VECTOR MACHINE* (SVM)

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:
Rifqi Rabbanie
NRP. 062116 4000 0096

Disetujui oleh Pembimbing:
Dr. Dra. Kartika Fithriasari, M.Si. ( )
NIP. 19691212 199303 2 002
Dra. Wiwiek Setya Winahju, M.S. ( )
NIP. 19560424 198303 2 001

Mengetahui,
Kepala Departemen Statistika

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

SURABAYA, JULI 2020

(*Halaman sengaja dikosongkan*)

KLASIFIKASI *MULTI-LABEL* PADA ARTIKEL JURNAL SCIENCEDIRECT DENGAN METODE *K-NEAREST NEIGHBOR* (KNN) DAN *SUPPORT VECTOR MACHINE* (SVM)

Nama Mahasiswa	**: Rifqi Rabbanie**
NRP	**: 062116 4000 0096**
Departemen	**: Statistika**
Dosen Pembimbing	**: Dr. Dra. Kartika Fithriasari, M.Si.**
	Dra. Wiwiek Setya Winahju, M.S.

Abstrak

ScienceDirect adalah *platform online* yang menyediakan akses terbitan yang terbanyak di dunia. Jurnal yang diterbitkan oleh ScienceDirect dikelompokkan menjadi empat bidang, yaitu *Physical Sciences and Engineering, Life Sciences, Health Sciences,* dan *Social Sciences and Humanities*. Meskipun begitu, ScienceDirect masih belum mengelompokkan artikel jurnal yang tersimpan di *platform* miliknya. Selain itu, sangat memungkinkan artikel jurnal yang disimpan termasuk ke dalam dua bidang atau lebih yang berbeda, sehingga perlu untuk diklasifikasikan secara *multi-label*. Data yang digunakan dalam penelitian ini berupa abstrak dari artikel jurnal dengan kata kunci "*Data Mining*" dan setiap jurnal diklasifikasikan berdasarkan abstrak tersebut. Sebelum data diklasifikasikan, terlebih dulu melalui tahapan *pre-processing* pada data teks abstrak tersebut, seperti proses *case folding, delete punctuation, remove number, tokenization, remove stopwords,* dan *lemmatization*. Setelah itu data tersebut diklasifikasikan secara *multi-label* dengan pendekatan *problem transformation Label Powerset*, yaitu dengan mentransformasi data kategori setiap artikel jurnal yang semula *multi-label* menjadi *multi-class* yang selanjutnya diklasifikasikan dengan KNN dan SVM. Kinerja klasifikasi KNN dan SVM diukur dengan nilai *hamming loss* dan didapatkan kesimpulan bahwa berdasakan nilai *hamming loss,* SVM memberikan hasil ketepatan klasifikasi yang lebih baik jika dibandingkan dengan KNN.

Kata Kunci: Grid Search, Hamming Loss, Label Powerset, Multi Label, Support Vector Machine.

(*Halaman sengaja dikosongkan*)

MULTI-LABEL CLASSIFICATION ON JOURNAL ARTICLE OF SCIENCEDIRECT USING CLASSIFIER K-NEAREST NEIGHBOR (KNN) AND SUPPORT VECTOR MACHINE (SVM)

Name	**: Rifqi Rabbanie**
Student Number	**: 062116 4000 0096**
Department	**: Statistics**
Supervisors	**: Dr. Dra. Kartika Fithriasari, M.Si.**
	Dra. Wiwiek Setya Winahju, M.S.

Abstract

ScienceDirect is an online platform that provides the most publications in the world. Journals published by ScienceDirect are grouped into four fields, namely Physiology and Engineering, Life Sciences, Health Sciences, and Social Sciences and Humanities. Even so, ScienceDirect still hasn't grouped journal articles stored on its platform. Also, it is very possible for saved article entries can include in two or more different fields, so it needs to be classified as multi-labeled. The data used in this study is in the form of abstracts from the meanings of journals with the keyword "Data Mining" and each journal is classified based on the abstract. Before the data is classified, it first goes through the pre-processing steps in the abstract text data, such as the case of folding, delete punctuation, re-move number, tokenization, remove stopwords, and lemmatization. After that, the data is classified in a multi-label manner with the Label Powerset problem transformation approach, namely by transforming the data categories of each journal article that was originally multi-label into a multi-class which is then classified with KNN and SVM. The KNN and SVM classification performance are measured by the value of hamming loss and it can be concluded that based on the value of a hamming loss, SVM gives better classification results when compared with KNN.

Key Words : Grid Search, Hamming Loss, Label Powerset, Multi Label, Support Vector Machine.

(*Halaman sengaja dikosongkan*)

KATA PENGANTAR

Dengan mengucapkan Alhamdulillah segala puji dan syukur penulis panjatkan atas kehadirat Allah SWT, karena berkat rahmat dan hidayah-Nya penyusunan Laporan Tugas Akhir yang berjudul "*Klasifikasi Multi-Label pada Artikel Jurnal ScienceDirect dengan Metode K-Nearest Neighbor (KNN) dan Support Vector Machine (SVM)*" ini dapat diselesaikan guna memenuhi salah satu persyaratan dalam menyelesaikan pendidikan di Departemen Statistika Institut Teknologi Sepuluh Nopember Surabaya.

Perjalanan panjang telah penulis lalui dalam rangka perampungan penulisan skripsi ini. Banyak hambatan yang dihadapi dalam penyusunannya, namun berkat kehendak-Nya sehingga penulis berhasil menyelesaikan penyusunan skripsi ini. Selain itu selesainya Laporan Tugas Akhir ini tidak lepas dari bantuan, bimbingan, serta dukungan dari berbagai pihak. Oleh karena itu, penulis ingin mengucapkan terima kasih kepada :

1. Ibu Dr. Dra. Kartika Fithriasari, M.Si. selaku Pembimbing I sekaligus sebagai Kepala Departemen Statistika ITS dan Ibu Dra. Wiwiek Setya Winahju, M.S. selaku Pembimbing II. Terima kasih atas segala bimbingan, ajaran, dan ilmu baru yang penulis dapatkan selama penyusunan skripsi. Dengan segala kesibukan masing-masing dalam pekerjaan maupun pendidikan, masih bersedia membimbing dan menuntun penulis dalam penyusunan Laporan Tugas Akhir ini. Terima kasih dan mohon maaf bila ada kesalahan yang telah penulis lakukan.
2. Ibu Dr. Santi Wulan Purnami, S.Si., M.Si. selaku Sekretaris Departemen 1 Bidang Akademik dan Kemahasiswaan, yang membantu penulis selama penyelesaian Tugas Akhir.
3. Prof. Drs. Nur Iriawan, MIKom., Ph.D. dan juga Ibu Adatul Mukarromah, S.Si., M.Si. selaku dosen penguji yang banyak memberikan saran dan masukan yang membangun selama penyusunan Laporan Tugas Akhir.
4. Ibu Santi Puteri Rahayu, M.Si, Ph.D selaku dosen wali penulis selama berkuliah di Departemen Statistika ITS yang senantiasa membantu penulis selama berkuliah.

5. Segenap dosen dan tenaga kependidikan Departemen Statistika ITS yang telah banyak memberikan ilmu dan membantu penulis selama berkuliah.
6. Kedua orang tua, ayahanda Muhammad Bashori dan ibunda tercinta Susilawati serta adik-adik penulis, Muhammad Reza Aisyi dan Rayya Maulidya yang senantiasa memberikan dukungan serta doa kepada penulis.
7. *Special Thanks for* Oktavia Ramadhani yang telah banyak membantu, mendampingi dan memberikan semangat selama penyusunan Laporan Tugas Akhir ini.
8. Kawan-kawan kontrakan yang senantiasa memberikan dukungan ilmu serta moril kepada penulis.
9. Kawan-kawan *Easy Data Consulting* yang bersedia membantu penulis dalam menyusun Laporan Tugas Akhir.
10. Seluruh kawan-kawan TR16GER yang selalu memberikan dukungan moral kepada penulis .
11. Seluruh pihak yang tidak bisa disebutkan satu per satu yang telah banyak membantu penulis selama ini.

Penulis menyadari bahwa laporan ini masih jauh dari kata sempurna. Oleh karena itu, kritik dan saran diharapkan dapat diberikan dari semua pihak untuk tahap pengembangan selanjutnya. Semoga laporan yang penulis susun dapat bermanfaat dan tidak lupa penulis memohon maaf apabila terdapat banyak kekurangan. Atas perhatian dan dukungannya, penulis menyampaikan terima kasih.

Surabaya, Juli 2020

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

(*Halaman sengaja dikosongkan*)

DAFTAR LAMPIRAN

(*Halaman sengaja dikosongkan*)

BAB I
PENDAHULUAN

1.1 Latar Belakang

Dalam dunia akademik dan penelitian, tidak asing dengan istilah jurnal ilmiah. Jurnal ilmiah seperti sekarang ini telah hadir sejak tahun 1665 sebagai bagian dari tradisi ilmuwan Inggris yang akhirnya menyebar ke mana-mana (Arianto, 2010). Menurut Hakim (dalam Suprayitno, 2019), jurnal ilmiah adalah majalah publikasi yang memuat Karya Tulis Ilmiah (KTI) yang secara nyata mengandung data dan informasi yang berkaitan iptek dan ditulis sesuai dengan kaidah-kaidah penulisan ilmiah serta diterbitkan secara berkala. Jurnal ilmiah diterbitkan sebagai cara atau media diseminasi hasil penelitian dalam disiplin atau sub disiplin ilmu tertentu. Publikasi jurnal ilmiah umumnya dalam bentuk artikel ilmiah. Artikel ilmiah merupakan tulisan yang berisi laporan sistematis mengenai hasil kajian atau hasil penelitian bagi akademisi ataupun masyarakat tertentu, yang merupakan audiens khusus dengan tujuan menyampaikan hasil kajian dan kontribusi penulis artikel kepada mereka untuk dikaji kembali dan didiskusikan, baik secara lisan maupun tulisan (Suryoputro, Riadi, & Sya'ban, 2012). Artikel ilmiah meliputi laporan penelitian dan non-penelitian, *review* literatur, obituari, laporan kasus, dan editorial. Artikel ilmiah yang ditulis dalam jurnal ilmiah diproduksi oleh individu dalam komunitas ilmuwan dan biasanya disajikan bagi komunitas ilmuwan juga, seperti mahasiswa, guru, dosen, peneliti, ilmuwan, dan sebagainya. Hal ini menunjukkan jurnal memegang peranan penting sebagai sarana komunikasi akademik antara para ilmuwan, khususnya dalam peningkatan pemahaman mengenai ilmu berdasarkan *review* dari ilmuwan. Selain itu, jurnal juga memegang peranan penting dalam penyebaran hasil-hasil penelitian serta sebagai pertukaran informasi untuk menghasilkan ide-ide baru dalam ilmu pengetahuan dan teknologi.

Kemajuan teknologi informasi dan komunikasi memberikan dampak yang signifikan terhadap publikasi jurnal. Pemanfaatan teknologi informasi telah mengubah paradigma jurnal konvensio-

nal yang dipublikasi dalam bentuk cetak menjadi publikasi jurnal secara elektronik. Jurnal secara elektronik atau yang sering dikenal dengan istilah *e-journal*, menurut Smith (dalam Jamaluddin, 2015) merupakan setiap jurnal yang tersedia secara *online*, baik secara elektronik maupun tercetak. *E-journal* mengalami pengembangan yang pesat seiring berkembangnya teknologi internet dan mampu menarik minat bagi para pembaca dan peneliti karena dapat diakses dan jurnal pun dapat diterbitkan dengan mudah. Ini memungkinkan untuk penerbitan yang lebih efektif dan efisien. Pembaca yang tertarik dengan sebuah jurnal, dapat dengan mudah untuk mengakses jurnal tersebut kapan pun dan di mana pun mereka berada, bahkan mereka mampu mencetak sendiri jurnal tersebut jika dibutuhkan. Sementara bagi seorang peneliti, hasil penelitiannya dapat dengan mudah diterbitkan dan disebarluaskan kepada pembaca (Miswan, 2002). Oleh karena itu, jurnal elektronik ini secara perlahan menggeser peran jurnal dalam media cetak yang dianggap terlalu lama dan memakan biaya yang besar untuk penerbitan dan pemanfaatannya. Hal ini membuat jurnal elektronik memiliki informasi lebih mutakhir dan lebih banyak dimanfaatkan dalam ilmu pengetahuan sehingga menjadikan jurnal elektronik ini sebagai sumber belajar dan informasi yang sangat penting.

Dalam perkembangannya, jurnal elektronik semakin banyak diminati dan berdampak kepada semakin berkembang pula pangkalan data (*database*) jurnal elektronik yang mampu memuat jurnal ilmiah dari berbagai penerbit dan dikumpulkan menjadi satu data sehingga memudahkan pembaca ataupun pustakawan untuk mencari artikel dari berbagai jurnal secara tepat (Arianto, 2010). Pangkalan data dalam arti luas merupakan suatu sistem pengelolaan rekam dokumen yang berisi informasi bibliografis, abstrak, maupun artikel lengkap, baik secara manual maupun otomatis yang dapat diakses dengan komputer. Pangkalan data berfungsi menghimpun data dan memelihara informasi. Data yang terdapat dalam sebuah pangkalan data tersebut disimpan secara terorganisir dan saling berelasi sehingga memudahkan ketika diakses lebih dari satu pengguna(Kusmayadi, 2008). Pangkalan data berisi jurnal elektro-

nik dapat berupa situs web yang menyediakan akses jurnal-jurnal elektronik. Ada banyak sekali situs web yang mempublikasikan jurnal elektronik, salah satunya yaitu ScienceDirect. ScienceDirect merupakan *platform* terkemuka dari Elsevier, penerbit *online* terkemuka yang berpusat di Belanda dalam literatur ilmiah *peer review*, dimana ilmuwan lain (*peer*) mengevaluasi dan memberikan kredibilitas riset sebelum mengizinkan dipublikasi dalam media cetak elektronik.

Sampai pada tahun 2017, ScienceDirect masih menjadi *platform online* dengan jumlah terbitan jurnal terbanyak di dunia, terdiri dari 3.976 jurnal dan 48.124 buku (Saputra, 2018). ScienceDirect sebagai salah satu pangkalan data pengetahuan jurnal, berperan besar dalam penyebaran pengetahuan yang menyediakan informasi ilmiah, teknis, dan medis terbesar di dunia. Informasi pengantar yang terdapat di situs web ScienceDirect terkait dengan cakupan isi literatur dan menawarkan banyak keuntungan bagi pengguna, khususnya bagi pustakawan dan peneliti (Nashihuddin & Rahayu, 2013). Sebagai seorang peneliti, tentunya menginginkan hasil penelitiannya dapat dijadikan referensi untuk penelitian yang lebih lanjut dan bisa bermanfaat bagi orang banyak. Hal ini bisa memberikan angka kredit yang besar bagi karier seorang peneliti. Sedangkan sebagai seorang pustakawan, tentunya untuk mencari rujukan informasi mengenai bidang tertentu menjadi lebih mudah karena hampir tersedia semua di dalam pangkalan data yang terdapat di ScienceDirect. Jurnal-jurnal yang terdapat di ScienceDirect dikelompokkan menjadi empat bagian utama, yaitu : *Physical Sciences and Engineering*, *Life Sciences*, *Health Sciences*, dan *Social Sciences and Humanities*. Sehingga bagi pustakawan ataupun pembaca dengan mudah bisa mencari sebuah jurnal maupun artikel berdasarkan keempat bidang tersebut.

Sebagai salah satu pangkalan data terbesar di dunia yang menyediakan jurnal ilmiah, ScienceDirect mengelompokkan bidang jurnal menjadi empat bidang, yaitu *Physical Sciences and Engineering*, *Life Sciences*, *Health Sciences*, dan *Social Sciences and Humanities*. Sementara ini, pengelompokan dilakukan hanya pada jur-

nal publikasi saja, dengan kata lain ScienceDirect belum terdapat pengelompokan artikel berdasarkan disiplin ilmu tertentu. Sementara ini, ketika mencari sebuah artikel ilmiah, seperti artikel penelitian, dapat dilakukan dengan fitur *search engine* dengan menuliskan kata kunci, penulis, ataupun judul artikel yang ingin dicari. Beberapa *platform* pangkalan data yang menyediakan jurnal, melakukan pengelompokan berdasarkan artikel jurnal, selain mengelompokkan jurnal tersebut ke dalam bidang disiplin ilmu. Pengelompokan artikel jurnal berdasarkan bidang disiplin ilmu juga dirasa sangat dibutuhkan, selain bagi pembaca juga bagi pihak *developer* ScienceDirect. Sebagai pembaca/peneliti, pengelompokan artikel berdasarkan disiplin ilmu akan memudahkan dalam mencari sebuah artikel penelitian. Sebagai *developer*, dengan adanya pengelompokan berdasarkan artikel jurnal yang dapat dilakukan secara otomatis tentunya akan membantu dalam melakukan pengelompokan dan pemberian label pada jurnal yang akan dipublikasi menjadi lebih efisien. Permasalahan klasifikasi tersebut dapat diselesaikan dengan menggunakan metode *text mining. Text mining* merupakan dimana seseorang berhadapan dengan kumpulan dokumen berupa teks dan kemudian berusaha untuk mengekstrak informasi yang berguna dari sebuah sumber data melalui identifikasi dan eksplorasi data (Feldmen & Sanger, 2007).

Text mining dapat digunakan untuk mengklasifikasikan teks, dimana klasifikasi teks merupakan proses untuk membentuk kelas-kelas (label) dari sekumpulan dokumen berdasarkan pada kelompok kelas yang sudah diketahui sebelumnya atau disebut juga *supervised learning* (Darujati & Gumelar, 2012). Salah satu metode alternatif yang dapat digunakan untuk mengklasifikasikan artikel-artikel jurnal pada ScienceDirect yang memiliki kemungkinan dalam satu artikel termasuk ke dalam dua bidang disiplin ilmu tertentu adalah dengan metode klasifikasi *multi-label.* Klasifikasi *multi-label* memiliki prinsip dasar yang sama dengan *single-label*, hanya saja klasifikasi *multi-label* memiliki dua atau lebih label yang akan diprediksi (Pushpa & Karpagavalli, 2017). Metode klasifikasi ini dapat menjadi salah satu alternatif untuk melakukan pengelompok-

an artikel-artikel jurnal yang terdapat pada ScienceDirect yang memungkinkan artikel-artikel tersebut dikelompokkan atau mendapatkan label lebih dari satu.

Penelitian mengenai *multi-label* sebelumnya pernah dilakukan Isnaini dkk (2019) mengenai klasifikasi *multi-label* pada topik berita di Indonesia dengan menggunakan metode *K-Nearest Neighbor* (KNN). Hasil dari penelitian tersebut menunjukkan bahwa KNN dapat digunakan untuk melakukan klasifikasi *multi-label* dan menunjukkan kinerja *hamming-loss* terkecil sebesar 11,16% dengan menggunakan nilai parameter tetangga k = 11. Nilai *hamming loss* tersebut tergolong kecil sehingga dapat dikatakan bahwa model klasifikasi yang digunakan sudah baik. Selain itu, penelitian lain pernah dilakukan oleh Pushpa dan Karpagavalli (2017) yang melakukan klasifikasi *multi-label* pada Phoneme Tamil dan menggunakan 3 transformasi data *multi-label,* yaitu *Binary Relevance* (BR), *Label Powerset* (LP), dan *Classifier Chain* (CC) dengan 5 *base classifier* yang berbeda. Hasil yang didapatkan adalah metode transformasi yang terbaik LP dan *classifier* yang memiliki kinerja yang terbaik adalah *Support Vector Machine* (SVM). Dalam penelitian ini, akan dilakukan klasifikasi *multi-label* pada artikel jurnal ScienceDirect dengan menggunakan metode transformasi data *Label Powerset* (LP) dengan menggunakan dua metode *base classifier,* yaitu KNN dan SVM.

Sebelum melakukan klasifikasi *multi-label* pada artikel jurnal ScienceDirect, terlebih dahulu perlu dilakukan *pra-processing* data sehingga data yang akan dianalisis dapat terstruktur dan mudah untuk dilakukan analisis. Data yang digunakan berbentuk teks, sehingga perlu dilakukan *text pra-processing*, yaitu Teknik yang diperlukan untuk menyiapkan data mentah yang tidak terstruktur menjadi data yang terstruktur untuk operasi *text mining. Text preprocessing* yang dilakukan adalah *tokenization, case folding, removing stop word, delete punctuation, removing numbers,* dan *lemmatization* sehingga kita bisa mendapatkan kata kunci atau *term* yang nantinya akan digunakan dalam analisis klasifikasi *multi-label*.

1.2 Rumusan Masalah

Artikel dari jurnal-jurnal yang terdapat di dalam ScienceDirect dapat dikelompokkan ke dalam 4 bidang, yaitu *Physical Sciences and Engineering*, *Life Sciences, Health Sciences*, dan *Social Sciences and Humanities*. Sebagai salah satu pangkalan data penyedia jurnal ilmiah terbesar perlu memberikan pelayanan yang maksimal kepada peneliti, pustakawan, maupun pembaca agar dapat dengan mudah menemukan artikel jurnal yang beririsan antara dua bidang yang ada, atau bahkan lebih. Oleh karena itu, penelitian ini akan dilakukan *text mining* dan melakukan klasifikasi artikel-artikel jurnal menjadi *multi-label* ke dalam dua atau lebih kategori yang beririsan dengan menggunakan *classifier K-Nearest Neighbor* (KNN) dan *Support Vector Machine* (SVM). Maka permasalahan utama yang akan dibahas dalam penelitian ini adalah bagaimana hasil model klasifikasi *multi-label* pada artikel jurnal ScienceDirect dengan menggunakan metode *classifier K-Nearest Neighbor* (KNN) dan *Support Vector Machine* (SVM).

1.3 Tujuan Penelitian

Permasalahan yang ada di ScienceDirect menjadi tujuan penelitian ini dan penjabaran tujuan penelitian ini adalah sebagai berikut.

1. Mendapatkan karakteristik artikel jurnal ScienceDirect yang dicari dengan *keyword* "*Data Mining*" yang diterbitkan pada tahun 2019 berdasarkan label yang terdiri dari : *Physical Sciences and Engineering*, *Life Sciences*, *Health Sciences*, dan *Social Sciences and Humanities*.
2. Mendapatkan hasil model klasifikasi *multi-label* dari artikel jurnal pada ScienceDirect dengan metode *classifier K-Nearest Neighbor* (KNN) dan *Support Vector Machine* (SVM).

1.4 Manfaat Penelitian

Hasil dari penelitian ini diharapkan dapat memberikan kemudahan bagi pihak Elsevier selaku produser dan pengembang pangkalan data ScienceDirect untuk memudahkan dalam melakukan label atau pemberian kategori bidang penelitian secara otomatis, khu-

susnya artikel-artikel dan jurnal yang dapat dikategorikan menjadi dua atau lebih bidang penelitian. Selain itu juga membantu pustakawan dan pembaca untuk mencari artikel ataupun jurnal yang berkaitan dengan bidang tertentu yang beririsan yang ingin dirujuk atau dibaca untuk menambah pengetahuan. Selain itu, dalam bidang ilmu pengetahuan, diharapkan penelitian ini dapat menambah wawasan dari penulis dan menjadi referensi bagi penelitian yang lebih lanjut.

1.5 Batasan Masalah

Batasan masalah dalam penelitian ini adalah data yang digunakan merupakan data abstrak dari artikel jurnal yang terdapat di ScienceDirect dengan kata kunci "*Data Mining*" yang terbit selama tahun 2019 dan diambil dari hasil pencarian tersebut secara acak.

(*Halaman sengaja dikosongkan*)

BAB II
TINJAUAN PUSTAKA

2.1 Klasifikasi *Multi-label*

Dalam *data mining*, tugas klasifikasi yaitu dapat memetakan dan mengelompokkan data sesuai dengan jumlah label yang akan diprediksi pada setiap sampel ke dalam *single-label* ataupun *multi-label*. Klasifikasi *single-label* digunakan untuk memprediksi satu label saja dari sebuah data atau sampel. Sedangkan dalam klasifikasi *multi-label* ini memiliki prinsip dasar yang sama dengan *single-label*, hanya saja klasifikasi *multi-label* memiliki dua atau lebih label yang akan diprediksi (Pushpa & Karpagavalli, 2017). Dalam penelitian ini, dilakukan pengelompokan artikel jurnal dimana artikel tersebut dapat berasosiasi dengan beberapa label dan dapat dituliskan secara matematis sebagai berikut.

$$Y = f(X_i), Y \subseteq L \tag{2.1}$$

$$\neg \forall l_a, l_b, X_{l_a} \cap X_{l_b} = \varnothing \tag{2.2}$$

dimana Y merupakan label hasil prediksi dari $f(X_i)$ dan Y merupakan sub set atau sama dengan L. L merupakan kumpulan kategori atau label yang terdapat dalam data, yaitu $L=\{l_1, l_2, ..., l_c\}$ dan $a \neq b$. Persamaan (2.1) dan (2.2) menunjukkan indikasi sebuah kumpulan data merupakan kumpulan data *multi-label.* Sebelum data yang berupa abstrak dari artikel jurnal diolah, data tersebut perlu diolah untuk menjadi terstruktur sehingga dapat diolah dan dianalisis. Tahapan yang perlu dilakukan untuk mengubah data menjadi lebih terstruktur yaitu dengan melakukan *text pre-processing.* Kemudian dilakukan pembobotan pada setiap term yang dihasilkan dari *text pre-processing* dengan *Term Frequency – Inverse Document Frequency* (TF-IDF). Setelah itu melakukan transformasi data menjadi data *multi-class* dengan metode *Label Powerset* (LP) dan membagi data menjadi data *traning* dan data *testing* dengan *K-Fold Cross Validation*.

2.1.1 *Text Pre-Processing*

Penelitian yang dilakukan ini berkaitan dengan data yang berupa teks abstrak artikel jurnal ilmiah, sehingga proses penggalian

informasi dilakukan terhadap data yang berupa kumpulan data teks. Proses dimana seseorang berhadapan dengan kumpulan dokumen berupa teks dan kemudian berusaha untuk mengekstrak informasi yang berguna dari sebuah sumber data melalui identifikasi dan eksplorasi data disebut sebagai *text mining* (Feldmen & Sanger, 2007). *Text mining* dapat digunakan untuk mengklasifikasikan teks dan mengelompokkan teks. Salah satu tahapan yang diperlukan dalam *text mining* adalah proses mempersiapkan data, dalam dunia sains komputasi dikenal dengan istilah *pre-processing data*. Data yang berupa teks, perlu diolah dan diproses agar lebih terstruktur dan dapat digunakan untuk analisis yang lebih lanjut. Teknik yang diperlukan untuk menyiapkan data mentah yang tidak terstruktur menjadi data terstruktur untuk operasi *text mining* disebut juga sebagai *text pre-processing* (Feldmen & Sanger, 2007). Beberapa *pre-processing* untuk data teks yang dilakukan dalam penelitian ini yaitu *case folding delete punctuation*, *remove number*, *tokenization*, *removing stop words*, dan *lemmatization*.

1. *Case Folding*, yaitu proses untuk mengubah seluruh huruf dalam teks dokumen menjadi non kapital (Prihatin, 2016).
2. *Delete Punctuation*, merupakan proses untuk menghapus semua karakter non-alpabet, misalnya simbol seperti titik (.), koma (,), dan simbol sejenis serta menghapus spasi (Mujilahwati, 2016).
3. *Remove Number*, proses untuk menghapus angka yang terdapat pada teks karena tidak digunakan dalam analisis lebih lanjut dan dapat mempengaruhi kinerja klasifikasi (Mujilahwati, 2016).
4. *Tokenization*, merupakan proses untuk membagi data teks *input* menjadi unit-unit kecil yang disebut token. Token atau yang biasa disebut juga term bisa berupa suatu kata, angka, atau tanda baca. Pada penelian ini, tanda baca dan angka dihilangkan sehingga tidak dianggap sebagai token (Prihatin, 2016).
5. *Removing Stop Words*, merupakan proses untuk menghilangkan kata-kata yang tidak memiliki arti yang relevan yang berakhir sebagai pemisah kata per kata. Kata-kata yang masuk dalam *stopword* contohnya adalah "*between*", "*and*", "*or*", "*an*", "*a*", "*the*", dan lain sebagainya yang dianggap tidak memiliki mak-

na. Sehingga kata tersebut dibuang dan tidak ikut diproses pada tahap selanjutnya(Khan, Baharudin, Lee, & Khan, 2010).

6. *Lemmatization,* merupakan normalisasi dimana berbagai varian atau jenis morfologis suatu kata dipetakan ke dalam kata dasar yang sama dengan memperhatikan kamus sehingga dapat dianalisis sebagai satu item (istilah atau konsep) (Liu, Christiansen, Baumgartner Jr., & Verspoor, 2012). *Lemmatization* hampir sama seperti *stemming*, yang membedakan adalah pada *stemming* lebih banyak memotong akhir kata dan sering membuang imbuhan, namun pada *lemmatization* menghasilkan kata dasar sesuai kamus. Sebagai contoh kata "*studies*" pada *stemming* akan menghasilkan output "*studi*", sedangkan pada *lemmatization* menghasilkan output "*study*".

2.1.2 *Terms Frequency-Inverse Document Frequency* (TF-IDF)

Metode TF-IDF merupakan suatu cara untuk memberikan bobot hubungan suatu kata (*term*) terhadap dokumen (Nurjannah, Hamdani, & Astuti, 2013). Metode ini menggabungkan dua konsep untuk perhitungan bobot, yaitu frekuensi kemunculan sebuah kata di dalam sebuah dokumen tertentu dan *inverse* frekuensi dokumen yang mengandung kata tersebut. Frekuensi kemunculan kata di dalam dokumen yang diberikan menunjukkan seberapa penting kata itu di dalam dokumen tersebut. Frekuensi dokumen yang mengandung kata tersebut menunjukkan seberapa umum kata tersebut. Sehingga bobot hubungan antara sebuah kata dan sebuah dokumen akan tinggi apabila frekuensi kata tersebut tinggi di dalam dokumen dan frekuensi keseluruhan dokumen yang mengandung kata tersebut yang rendah pada kumpulan dokumen. Berikut ini merupakan rumus TF-IDF :

$$idf_{d,t} = \log\left(\frac{D}{df_t}\right) \tag{2.3}$$

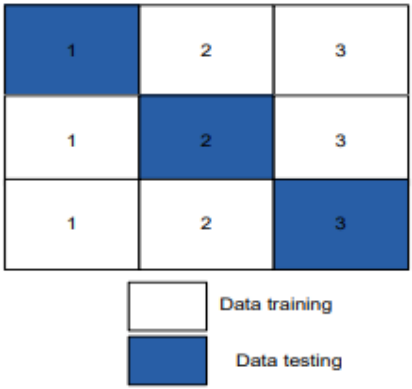
$$W_{d,t} = tf_{d,t} \times idf_{d,t} \tag{2.4}$$

dengan,

D = Total dokumen
df_t = Jumlah dokumen yang mengandung *term* t
$W_{d,t}$ = Bobot *term* t terhadap dokumen d
$tf_{d,t}$ = Frekuensi kemunculan *term* t dalam dokumen d

2.1.3 *K-fold Cross Validation*

Cross-validation atau dapat disebut estimasi rotasi adalah sebuah teknik validasi model untuk menilai bagaimana hasil statistik analisis akan menggeneralisasi kumpulan data independen. Teknik ini digunakan untuk melakukan prediksi model dan memperkirakan seberapa akurat sebuah model prediktif ketika dijalankan dalam praktiknya.



Gambar 2.1 Contoh Penerapan *3-fold cross validation*
(Sumber Gambar : Tempola, Muhammad, & Khairan, 2018)

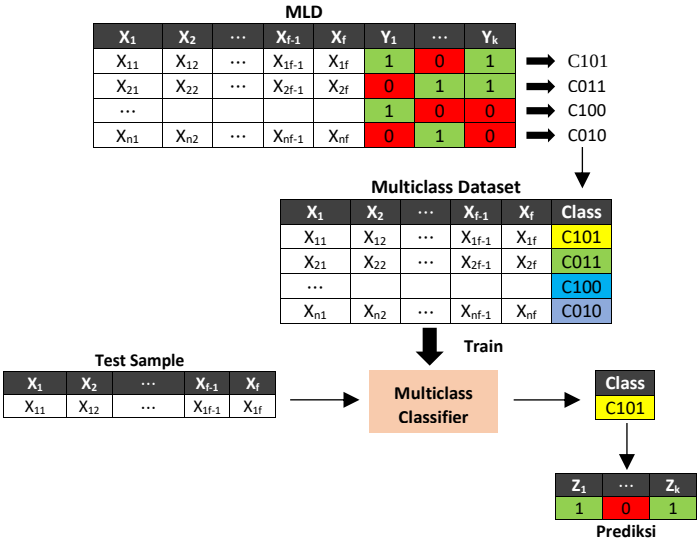
Salah satu teknik dari validasi silang adalah *k-fold cross validation*, yang mana memecah data menjadi *k* bagian set data dengan ukuran yang sama. Penggunaan *k-fold cross validation* untuk menghilangkan bias pada data (Tempola, Muhammad, & Khairan, 2018). *Training* dan *testing* dilakukan sebanyak *k* kali. Percobaan pertama, sub set data S_1 diperlakukan sebagai data *testing* dan subset lainnya diperlakukan sebagai data *training.* Selanjutnya, percobaan kedua, subset S_2 diperlakukan sebagai data *testing*, sementara yang lain

(S_1,S_3,...,S_k) sebagai data *training* dan begitu seterusnya. Ilustrasi penggunaan *k-fold cross validation* disajikan pada Gambar 2.1, dimana *k* yang digunakan adalah 3. Sehingga ilustrasi tersebut menunjukkan 3-*fold cross validation* dan setiap data akan dieksekusi sebanyak 3 kali. Setiap subset data, yakni D1, D2, dan D3 mempunyai kesempatan sebagai data *training* dan *testing*.

2.1.5 Metode *Problem Transformation Label Powerset*

Secara umum, metode klasifikasi *multi-label* terbagi dalam dua cara, yaitu metode transformasi data *multi-label* dan metode adaptasi algoritma *multi-label* (Pushpa & Karpagavalli, 2017). Dalam penelitian ini akan menggunakan metode transformasi data, di mana data *multi-label* ditransformasi ke dalam *single-label* sehingga bisa diselesaikan dengan pendekatan *single-label.* Salah satu pendekatan yang paling alami dalam transformasi data adalah *Label Powerset* (LP) dimana membangkitkan kelas baru untuk setiap kombinasi label dan kemudian me-mecahkan masalah menggunakan pendekatan klasifikasi *multi-class* (Pushpa & Karpagavalli, 2017). Pendekatan dengan LP mempertimbangkan setiap kombinasi unik dari label dalam kumpulan data *multi-label* sebagai nilai satu kelas dalam *multi-class* Cara ini lebih memperhitungkan korelasi antar label (Spolaor, Cherman, Monard, & Lee, 2013). Ilustrasi transformasi data *multi-label* dengan metode LP dapat dilihat pada Gambar 2.2. Langkah-langkah yang dilakukan dalam transformasi *Label Powerset* adalah seperti berikut.

1. Kombinasi unik dari kategori dokumen teks *multi-label* ditransformasi menjadi *multi-class* dengan memberikan kode berupa angka 0 hingga sebanyak kombinasi unik yang terbentuk.
2. Data hasil dari kombinasi yang berupa *multi-class* tersebut dilatih dengan menggunakan metode analisis yang lebih lanjut. Dalam penelitian ini, data dilatih dengan menggunakan K-NN dan SVM.
3. Mendapatkan hasil prediksi kategori *multi-class* dengan metode K-NN dan SVM. Kategori hasil prediksi dalam bentuk *multi-class* ditransformasi kembali ke dalam bentuk *multi-label.*



Gambar 2.2 Ilustrasi *Label Powerset*
(Sumber: Herrera, Charte, Rivera, & del Jesus, 2016)

2.1.6 *K-Nearest Neighbor* (KNN)

K-Nearest Neighbor adalah salah satu metode yang paling populer digunakan untuk klasifikasi teks. K-NN adalah metode untuk mengklasifikasikan objek berdasarkan sampel dari data latih terdekat dalam sebuah *space* untuk menentukan kelompok *k* objek (Sreemathy & Balamurugan, 2012). Sehingga dalam menentukan hasil klasifikasi KNN melihat jarak terdekat dari objek masing-masing kelompok. Jarak tersebut diperoleh dari hasil kedekatan antara data masukan dengan data yang berada dalam kelompok berdasarkan sejumlah fitur yang ada. Tahapan untuk menjalankan KNN sebagai berikut.

1. Menentukan parameter *p* sebagai jumlah tetangga terdekat.
2. Menghitung jarak objek dengan masing-masing data *training*. Perhitungan jarak yang paling umum adalah dengan mengguna-

kan persamaan *euclidian distance* (Sreemathy & Balamurugan, 2012) seperti di bawah ini,

$$\text{dist}(x_a, x_b) = \sqrt{\sum_{j=1}^{m} (x_{aj} - x_{bj})^2} \quad (2.5)$$

dimana x_{aj} merupakan objek baru pada term ke-*j* dan x_{bj} merupakan data *training* pada term ke-*j* dengan *m* adalah jumlah data (*term*).

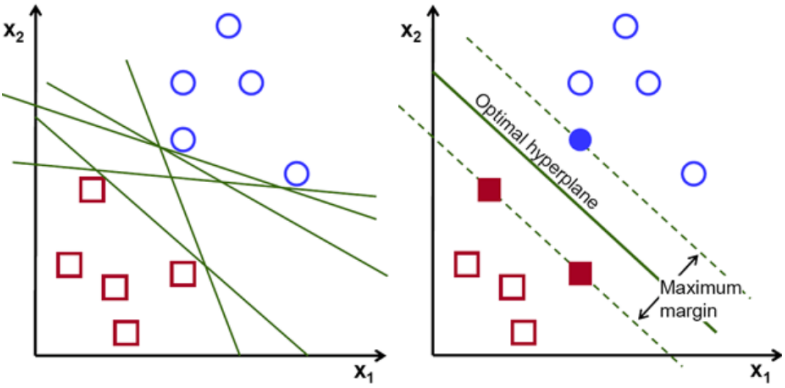
3. Menentukan *p* data *training* terdekat dari objek berdasarkan hasil perhitungan jarak *euclidean* menggunakan persamaan (2.5).
4. Menentukan kategori dari objek baru berdasarkan mayoritas label dari *p* tetangga terdekat.

2.1.7 *Support Vector Machine* (SVM)

Support Vector Machine (SVM) adalah metode yang mempelajari area yang memisahkan antar kategori dalam sebuah observasi. Teknik SVM digunakan untuk mendapatkan fungsi pemisah (*hyperplane*) yang optimal untuk memisahkan observasi yang memiliki nilai variabel target berbeda (Williams, 2011). Pemisah ini berupa *line* pada data *two dimension* dan berupa *flat plane* pada data *multiple dimension*. Meskipun dalam prosesnya cenderung diperlukan waktu yang lama, akurasi yang dihasilkan oleh SVM cenderung tinggi dan tidak terjadi kasus *overfitting*.

SVM dapat digunakan pada data yang dapat dipisah secara linier maupun tidak dapat dipisah secara linier. Salah satu contoh kasus dasar pada klasifikasi adalah data yang bisa dipisahkan secara linier oleh suatu garis lurus. Ilustrasi data yang dapat dipisahkan oleh garis lurus dapat dilihat pada Gambar 2.3 (a), dimana terdapat banyak garis yang bisa digunakan untuk memisahkan data tersebut ke dalam kelas yang bersesuaian. Gambar 2.3 (b), dimana dapat diketahui bahwa pemisah pada tersebut merupakan pemisah yang lebih baik untuk digunakan karena memiliki margin yang lebih besar,

sehingga data tersebut dapat secara akurat dipisahkan ke dalam kelasnya masing-masing.



(a) (b)

Gambar 2.3 (a) Ilustrasi Pemisah yang Dapat Digunakan untuk Memisahkan Data yang Secara Linear dan (b) Pemisah dengan Margin yang Paling Besar

Gambar 2.3 (a) menggambarkan bahwa terdapat banyak garis pemisah yang bisa dibuat untuk memisahkan data kategori +1 (lingkaran warna biru) dengan kategori -1 (kotak warna merah). Namun akan didapatkan suatu garis pemisah yang terbaik, yang dapat meminimalkan kesalahan klasifikasi, yaitu garis yang menghasilkan margin terbesar. Margin adalah jarak antara *hyperplane* atau pemisah dengan data terdekat dari masing-masing kelas data. Persamaan untuk *hyperplane* dapat ditulis sebagai berikut.

$$\mathbf{W}.\mathbf{X} + b = 0 \qquad (2.6)$$

$\mathbf{W} = \{w_1, w_2, ..., w_m\}$ adalah vektor pembobot sebanyak *m*, dimana *m* adalah banyak variabel **X**, dan *b* adalah suatu konstanta atau biasa disebut dengan bias. Persamaan (2.6) dapat dimodifikasi sehingga didapat persamaan untuk setiap sisi margin berikut.

$$L_1: \mathbf{W}.\mathbf{X} + b \geq 1 \text{ untuk } y_l = +1, \qquad (2.7)$$

$$L_2: \mathbf{W}.\mathbf{X} + b \leq -1 \text{ untuk } y_l = -1 \qquad (2.8)$$

Data yang berada pada daerah L_1 akan dikategorikan ke dalam kelas +1, sedangkan data yang berada pada daerah L_2 akan dikategori-

kan ke dalam kelas -1. Jika persamaan tersebut dikalikan dengan masing-masing nilai kelasnya, y_l, maka didapat persamaan sebagai berikut.

$$y_l\left(\mathbf{W}.\mathbf{X}+b\right)\geq 1,\forall_l \tag{2.9}$$

Data yang berada pada tepat pada tepi margin, yaitu L_1 dan L_2, disebut sebagai *support vectors*.

Berdasarkan persamaan (2.9) dapat diketahui bahwa jarak terdekat *hyperplane* ke tepi margin adalah $\frac{1}{||\mathbf{W}||}$, dimana $\left|||\mathbf{W}|\right|$ adalah *Euclidean norm* dari **W,** dengan rumus :

$$\mathbf{W}=\sqrt{w_1^2+w_2^2+\cdots+w_m^2} \tag{2.10}$$

sehingga jarak antar tepi margin adalah $\frac{2}{||\mathbf{W}||}$. Karena akan didapatkan *hyperplane* yang memberikan margin maksimal, yaitu dengan mencari nilai $max\frac{2}{||\mathbf{W}||}$, hal ini kongruen dengan mencari nilai minimal dari $min\frac{||\mathbf{W}||}{2}$, atau $min\ \frac{||\mathbf{W}||^2}{2}$. Maka fungsi obyektif pada permasalahan ini adalah

$$\min\frac{\left\|\mathbf{W}\right\|^2}{2} \tag{2.11}$$

dengan fungsi batasan sebagai berikut.

$$y_l\left(\mathbf{W}.\mathbf{X}_l+b\right)\geq 1 \tag{2.12}$$

Kemudian dengan menggunakan *Lagrange Multiplier*, didapat persamaan sebagai berikut.

$$L_{pd}=\frac{\left\|\mathbf{W}\right\|^2}{2}-\sum_{m=1}^{L}a_l\left[y_l\left(\mathbf{W}.\mathbf{X}_l+b\right)-1\right] \tag{2.13}$$

Selanjutnya menggunakan kondisi *Karush-Kuhn-Tucker* (KKT), yaitu:

$$\begin{gathered}\frac{\partial L_{pd}}{\partial \mathbf{W}}=0\leftrightarrow \mathbf{W}-\sum_{l=1}^{L}a_l y_l X_l=0\\ \mathbf{W}=\sum_{l=1}^{L}a_l y_l \mathbf{X}_l\end{gathered} \tag{2.14}$$

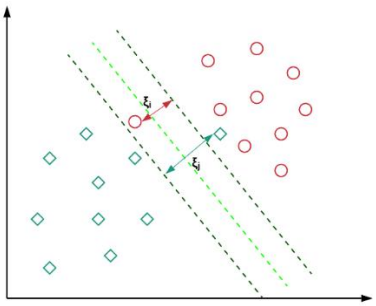
$$\frac{\partial L_{pd}}{\partial b} = 0 \leftrightarrow 0 - \sum_{l=1}^{L} a_l y_l = 0 \leftrightarrow \sum_{l=1}^{L} a_l y_l = 0 \tag{2.15}$$

$$a_l \left[y_l \left(\mathbf{W}.\mathbf{X}_l + b \right) - 1 \right] = 0, \text{ dimana } a_l \geq 0 \tag{2.16}$$

dengan menyubstitusi persamaan (2.14), (2.15) dan (2.16) ke persamaan (2.13), maka akan didapatkan persamaan sebagai berikut,

$$L_d = -\frac{1}{2} \sum_{l_1=1}^{L} \sum_{l_2=1}^{L} a_{l_1} a_{l_2} y_{l_1} y_{l_2} \left(\mathbf{X}_{l_1} . \mathbf{X}_{l_2} \right) + \sum_{l=1}^{L} a_l \tag{2.17}$$

dengan menyubstitusi nilai $y_{l_1}, y_{l_2}, \mathbf{X}_{l_1}, \mathbf{X}_{l_2}$ ke persamaan (2.17), didapat suatu persamaan L_d yang kemudian akan digunakan untuk mendapatkan nilai-nilai a_m (*support vectors*) yang membuat L_d optimum dengan cara mencari turunan parsial L_d terhadap a.



Gambar 2.4 Ilustrasi Data yang Tidak Dapat Dipisah Secara Linear

Kasus data yang bisa dipisahkan secara linier merupakan kasus yang sulit untuk ditemui. Umumnya, terdapat beberapa kelas data yang berada pada daerah kelas data lainnya, kasus ini disebut *linearly non-separable* data atau data yang tidak bisa dipisah secara linier. Salah satu ilustrasi untuk kasus seperti ini dapat dilihat pada Gambar 2.4. Pencarian *hyperplane* yang optimal pada kasus ini akan memperhatikan data-data yang tidak terdapat pada kelasnya (*misclassification error*), yang dilambangkan dengan ξ. Sehingga persamaan (2.9) menjadi

$$y_l \left(\mathbf{W}.\mathbf{X} + b \right) \geq 1 - \xi_l, \forall_l \tag{2.18}$$

dan didapat persamaan *Lagrange Multiplier* sebagai berikut.

$$L_{pd} = \frac{\|\mathbf{w}\|^2}{2} + C\sum_{l=1}^{L}\xi_l - \sum_{l=1}^{L}a_l\left[y_l\left(\mathbf{W}.\mathbf{X}+b\right)-1+\xi_l\right] - \sum_{l=1}^{L}\beta_l\xi_l \qquad (2.18)$$

dengan *C* adalah suatu nilai pengali *Lagrange*. Kemudian dengan menggunakan kondisi *Karush-Kuhn-Tucker* (KKT), yaitu

$$\frac{\partial L_{pd}}{\partial \mathbf{W}} = 0 \leftrightarrow \mathbf{W} - \sum_{l=1}^{L} a_l y_l \mathbf{X}_l = 0$$
$$\mathbf{W} = \sum_{l=1}^{L} a_l y_l \mathbf{X}_l \qquad (2.20)$$

$$\frac{\partial L_{pd}}{\partial b} = 0 \leftrightarrow 0 - \sum_{l=1}^{L} a_l y_l = 0 \leftrightarrow \sum_{l=1}^{L} a_l y_l = 0 \qquad (2.21)$$

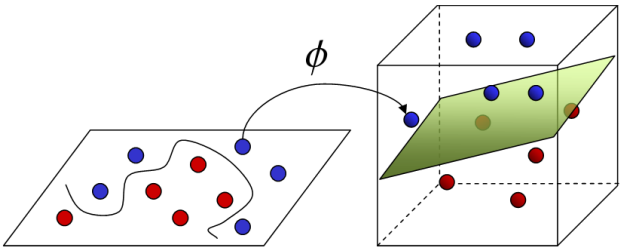
$$\frac{\partial L_{pd}}{\partial \xi_l} = 0 \leftrightarrow 0 + C - a_l - \beta_l = 0 \leftrightarrow C = a_l + \beta_l \qquad (2.22)$$

dengan menyubstitusi persamaan (2.20), (2.21) dan (2.22) ke persamaan (2.19), didapat persamaan sebagai berikut.

$$L_d = -\frac{1}{2}\sum_{l_1=1}^{L}\sum_{l_2=1}^{L} a_{l_1} a_{l_2} y_{l_1} y_{l_2} \left(\mathbf{X}_{l_1}.\mathbf{X}_{l_2}\right) + \sum_{l=1}^{L} a_l \qquad (2.23)$$

Menggunakan langkah yang sama pada kasus data yang bisa terpisah secara linier, yaitu mensubstitusikan nilai $y_{l_1}, y_{l_2}, \boldsymbol{X}_{l_1}, \boldsymbol{X}_{l_2}$ ke persamaan (2.23), didapat suatu persamaan L_d yang kemudian akan digunakan untuk mendapatkan nilai-nilai a_l (*support vectors*) yang membuat L_d optimum dengan cara mencari turunan parsial L_d terhadap a.

Pada kasus nyata, sangat jarang dijumpai data yang dapat terpisah secara linier, oleh karena itu digunakan suatu fungsi kernel (Φ) untuk memetakan data ke dalam ruang vektor yang berdimensi tinggi, seperti yang telah diilustrasikan pada Gambar 2.5, sehingga data hasil transformasi bisa terpisah secara linier.



Gambar 2.5 Ilustrasi Data Non-Linear yang Ditransformasi ke Dalam Ruang Vektor Berdimensi Tinggi dengan Fungsi Kernel

Setelah dilakukan transformasi, selanjutnya menemukan *support vector* dari data yang sudah di transformasi ke ruang yang berdimensi tinggi, yang bisa didapat menggunakan pengembangan dari persamaan (2.17) atau (2.23), yaitu

$$L_d = -\frac{1}{2}\sum_{l_1=1}^{L}\sum_{l_2=1}^{L} a_{l_1} a_{l_2} y_{l_1} y_{l_2} \left(\Phi\left(X_{l_1}\right).\Phi\left(X_{l_2}\right)\right) + \sum_{l=1}^{L} a_l \qquad (2.24)$$

dengan $\Phi(\mathbf{X}_{l_1})$ atau $\Phi(\mathbf{X}_{l_2})$ adalah data hasil transformasi. Namun, transformasi Φ tidak dapat diketahui dan sangat sulit dipahami, sehingga perhitungan *dot product* dapat secara implisit digantikan oleh fungsi kernel dan didapat,

$$L_d = -\frac{1}{2}\sum_{l_1=1}^{L}\sum_{l_2=1}^{L} a_{l_1} a_{l_2} y_{l_1} y_{l_2} K\left(\mathbf{X}_{l_1}.\mathbf{X}_{l_2}\right) + \sum_{l=1}^{L} a_l \qquad (2.25)$$

dengan $K\left(\mathbf{X}_{l_1}.\mathbf{X}_{l_1}\right)$ adalah fungsi kernel yang digunakan. Beberapa fungsi kernel yang akan digunakan pada penelitian ini adalah fungsi kernel *Radial Basis Function* (RBF) dan fungsi kernel *polynomial*. Rincian rumus dan parameter yang digunakan pada setiap fungsi kernel dapat dilihat pada Tabel 2.1.

Awalnya SVM dikembangkan untuk persoalan klasifikasi dua kelas, kemudian dikembangkan kembali untuk klasifikasi *multi-class*. Dalam klasifikasi *multi-class*, *hyperplane* yang terbentuk lebih dari satu.

Tabel 2.1 Fungsi Kernel *Support Vector Machine* (SVM)

Fungsi Kernel	Rumus $K(\mathbf{X}, \mathbf{X}_i)$
Polynomial	$(\gamma \mathbf{x}^T \mathbf{x}_i + r)^d,\ \gamma > 0$
RBF	$\exp\left(-\gamma \left\|\mathbf{x} - \mathbf{x}_i\right\|^2\right),\ \gamma > 0$

Keterangan :
C merupakan parameter *Cost,* γ adalah parameter gamma, *r* adalah *coefficient,* dan *d* adalah *degree*/derajat.

Salah satu metode pendekatannya adalah *One-Againts-All* (OAA, atau disebut juga sebagai *One-versus-Rest*). Penentuan kelas dari suatu data ditentukan berdasarkan nilai terbesar dari *hyperplane* seperti pada persamaan (2.26).

$$\hat{y} = \arg\max_{l=1,\ldots,L} \left(f_l(\mathbf{x}) \right) \tag{2.26}$$

dimana $f_l(\mathbf{x})$ adalah persamaan *hyperplane* atau pemisah sepeti persamaan (2.27) dan L merupakan jumlah kelas/label.

$$f_l(\mathrm{x}) = \sum_{i=1, \mathbf{X}_i \in SV}^{n} a_i y_i K\left(\mathbf{X}, \mathbf{X}_i\right) + b_l \tag{2.27}$$

dimana *SV* pada persamaan di atas merupakan subset dari data yang merupakan *support vector,* dengan kata lain data $\mathbf{X}_i$ yang berkorespondesi pada $a_l \geq 0$.

Software python yang digunakan dalam melakukan analisis klasifikasi pada data jurnal ScienceDirect memanfaatkan basis LibSVM untuk *Support Vector Classification* (SVC). LibSVM merupakan sebuah *library* untuk *Support Vector Machine* (SVM) yang paling banyak digunakan dan menerapkan algoritma *Sequential Minimal Optimization* (SMO) (Chang & Lin, 2011). SMO adalah sebuah algoritma yang mengatasi permasalahan optimasi *Quadratic Programming* (QP) pada *Support Vector Machine* (SVM). SMO mampu memperkecil permasalahan QP dan dapat memperkecil waktu optimasi (Wahyuni, 2016). Pada dasarnya penggunaan SVM hanya terbatas pada masalah yang kecil karena algoritma pelatihan SVM cenderung lambat, kompleks, dan sulit untuk diimplementa-

sikan. SMO menyelesaikan masalah optimasi seminimal mungkin untuk setiap tahapnya. Pada setiap tahap, SMO memilih dua *langrange multipliers* α_i untuk dioptimalkan bersama-sama, mencari nilai yang paling optimal untuk *langrange multiplier* tersebut, dan memperbaharui SVM dengan nilai yang baru. Algoritma SMO adalah sebagai berikut.

1. Inisiasi nilai *langrange multiplier* α dan *bias b.*
2. Melakukan iterasi pada seluruh data latih, cari α_1 yang melanggar sifat gradien. Jika α_1 diperoleh, maka bisa lanjut ke tahap 4. Jika belum, lakukan iterasi pada seluruh data latih hingga selesai dan lanjutkan pada data yang tidak terdapat pada batas secara bergantian untuk mencari α_1 yang melanggar sifat gradien sampai seluruh α memenuhi sifat gradien.
3. Mencari α_2 dari data yang tidak terdapat pada batas. Ambil α yang memberikan nilai $|E_1-E_2|$ terbesar sebagai α_2. E_1 dan E_2 merupakan *error cache* untuk α_1 dan α_2.
4. Membuang α_2 ini jika dua data identik dan ke tahap 7. Selanjutnya, menghitung nilai L dan H untuk α_2 :

$$L = \begin{cases} \max(0, \alpha_2 - \alpha_1), \text{ jika} & y_1 = y_2 \\ \max(0, \alpha_2 + \alpha_1 - c), \text{ jika} & y_1 \neq y_2 \end{cases} \quad (2.28)$$

$$H = \begin{cases} \min(\text{c}, \text{c} + \alpha_2 - \alpha_1), \text{ jika} & y_1 = y_2 \\ \min(\text{c}, \alpha_2 + \alpha_1), \text{ jika} & y_1 \neq y_2 \end{cases} \quad (2.29)$$

5. Jika $L=H$, maka perkembangan optimasi tidak dapat dibuat, sehingga langkah berikutnya adalah membuang α_2 ini dan ke tahap 7. Berikutnya menghitung nilai :

$$\eta = 2K(\vec{x}_1, \vec{x}_2) - K(\vec{x}_1, \vec{x}_1) - K(\vec{x}_2, \vec{x}_2) \quad (2.30)$$

Jika nilai η negatif, selanjutnya menghitung nilai α_2 yang baru. Kemudian menghitung fungsi objektif pada titik L dan H dan gunakan nilai α_2 yang memberikan fungsi objektif paling tinggi sebagai α_2 yang baru. Jika nilai $\left|\alpha_2^{baru} - \alpha_2^{lama}\right|$ lebih kecil dari nilai perubahan terkecil $\alpha(\varepsilon)$, sehingga perlu membuang nilai α_2 ini dan ke tahap 7. Selainnya ke tahap 9.

6. Melakukan iterasi pada data yang tidak terdapat pada batas sampai diperoleh α_2 yang dapat membuat perkembangan optimasi di tahap 4 sampai 6. Jika tidak diperoleh, maka lakukan iterasi pada seluruh data latih sampai diperoleh α_2 yang dapat membuat perkembangan optimasi di tahap 4 sampai 6.
7. Jika α_2 tidak diperoleh setelah dua iterasi tersebut, nilai α_1 yang sudah diperoleh tidak digunakan dan kembali ke tahap 3 untuk mencari α_1 baru yang melanggar sifat gradien.
8. Menghitung nilai α_2 yang baru dan perbarui nilai *b* dan *error cache.* Kemudian simpan α_1 dan α_2 yang baru. Kemudian kembali ke tahap 3.

2.1.8 Pengukuran Kinerja Klasifikasi *Multi-label*

Kinerja klasifikasi *multi-label classifier* dapat diklasifikasikan sebagai basis label dan basis sampel. Basis label dihitung untuk setiap label dan kemudian dirata-rata di semua label (mengabaikan hubungan di antara label), sedangkan basis sampel dihitung berdasarkan setiap sampel pengujian dan kemudian dirata-rata di set uji (Pushpa & Karpagavalli, 2017). Dalam penelitian ini, pengukuran kinerja *multi-label classifier* dilakukan dengan menggunakan ukuran basis sampel *hamming-loss.* Mengasumsikan *D* adalah sebuah kumpulan dokumen yang merupakan data *multi-label*, *L* merupakan kumpulan label yang digunakan dalam kumpulan data, Y_i merupakan subset dari label sebenarnya dari sampel ke-*i*, dan Z_i merupakan set label hasil prediksi. *Hamming-loss* yang merupakan evaluasi kinerja yang paling umum dalam literatur *multi-label*, dihitung sebagai banyaknya kesalahan klasifikasi dari subset label hasil prediksi terhadap label sebenarnya serta dibagi dengan jumlah total label dalam kumpulan data. Semakin kecil nilai *hamming-loss,* maka semakin baik pula kinerjanya. Performa dikatakan sempurna ketika *hamming loss* = 0. *Hamming loss* memiliki persamaan sebagai berikut.

$$hamming-loss = \frac{1}{|D|}\sum_{i=1}^{|D|}\frac{|Y_i \Delta Z_i|}{|L|} \tag{2.31}$$

Keterangan :
$|D|$: Jumlah dokumen
$|L|$: Banyak kategori
$|Y_i \Delta Z_i|$: Banyak kesalahan prediksi subset kategori Z_i terhadap Y_i

Contoh perhitungan kinerja klasifikasi dengan menggunakan *hamming loss* ditunjukkan berikut ini.

Tabel 2.2 Contoh Perhitungan *Hamming Loss*

	$\mathbf{Y}_i$	$\mathbf{Z}_i$
$\tilde{\mathbf{X}}_1$	[1 0 1 0]	[1 0 **0** **1**]
$\tilde{\mathbf{X}}_2$	[0 1 0 1]	[0 1 0 1]
$\tilde{\mathbf{X}}_3$	[1 0 0 1]	[1 0 0 1]
$\tilde{\mathbf{X}}_4$	[0 1 1 0]	[0 1 **0** 0]
$\tilde{\mathbf{X}}_5$	[1 0 0 0]	[1 0 0 **1**]

Tabel 2.2 di atas menunjukkan pada dokumen pertama terdapat dua subset kategori pada $\mathbf{Z}_1$ yang berbeda dengan subset kategori asli pada $\mathbf{Y}_1$. Kemudian pada dokumen keempat dan kelima masing-masing terdapat satu subset kategori yang salah diprediksi pada $\mathbf{Z}_4$ dan $\mathbf{Z}_5$ terhadap label aslinya. Selanjutnya dihitung nilai *hamming loss* yang didapatkan untuk klasifikasi tersebut seperti berikut.

$$hamming-loss = \frac{1}{5}\left(\frac{2+0+0+1+1}{4}\right) = \frac{4}{20}0.2$$

Didapatkan nilai *hamming loss* sebesar 0,2 yang menunjukkan bahwa tingkat kesalahan klasifikasi terhadap kategori dokumen di atas sebesar 20%.

2.1.9 *Grid Search*

Metode *Grid Search* merupakan salah satu metode yang sederhana untuk mengatasi masalah optimasi. Metode ini melibatkan penyusunan grid yang cocok dalam suatu ruang dimensi, mengevaluasi fungsi objek dari seluruh titik grid, dan menemukan titik grid yang sesuai dengan fungsi objektif yang memiliki nilai optimum. Algoritma *grid search* bekerja dengan mencoba seluruh nilai parameter yang ada dalam batasan nominal tertentu. Kemudian nilai

terbaik diambil berdasarkan hasil performa terbaik dari proses pemodelan data (Natan, Gunawan, & Dewantara, 2019).

Metode *grid search* memiliki cara kerja yang hampir serupa dengan percobaan secara manual menggunakan teknik *trial* dan *error*. Mencoba kombinasi parameter satu per satu dan membandingkan nilai terbaik yang diberikan oleh parameter tersebut. Namun perbedaan *grid search* terletak pada proses perbandingan nilai yang tidak dilakukan di awal saat terpilihnya pasangan kombinasi parameter. Pasangan kombinasi dari parameter terlebih dahulu disimpan dalam grid-grid. Selanjutnya perbandingan nilai *error* terkecil dilihat dari baris dan kolom pada grid tersebut. Baris ke-*i* dan kolom ke-*j* yang memiliki nilai terbaik merupakan kombinasi parameter yang terpilih.

Sebagai contoh, akan didapatkan nilai optimum suatu model atau fungsi dengan mencari kombinasi parameter-parameter yang memberikan nilai terbaik. Dalam penelitian ini, nilai terbaik ditunjukkan dengan *hamming loss* yang paling kecil. Ilustrasi metode *grid search* ditunjukkan pada Tabel 2.3 berikut.

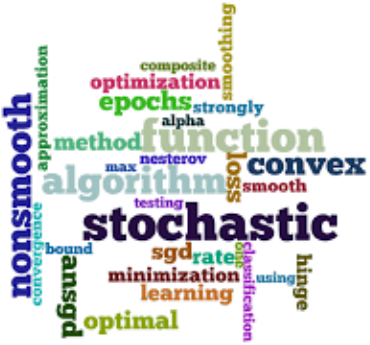
Tabel 2.3 Ilustrasi *Grid Search*

	A_1	**A_2**	**A_3**	**A_4**	**A_5**
B_1	g_{11}	g_{12}	g_{13}	g_{14}	g_{15}
B_2	g_{21}	g_{22}	g_{23}	g_{24}	g_{25}
B_3	g_{31}	g_{32}	g_{33}	g_{34}	g_{35}
B_4	g_{41}	g_{42}	**g_{43}**	g_{44}	g_{45}
B_5	g_{51}	g_{52}	g_{53}	g_{54}	g_{55}

Tabel 2.3 menunjukkan grid-grid yang menunjukkan nilai kinerja satu model hasil kombinasi dari parameter dari model tersebut. A dan B merupakan parameter dengan masing-masing parameter tersebut memiliki lima nilai yang dicobakan dan menghasilkan nilai kebaikan hasil kombinasi parameter sebanyak 25 nilai *hamming loss.* Didapatkan bahwa baris ke-4 dan kolom ke-3 sebagai nilai *hamming loss* yang terkecil sehingga terpilih menjadi kombinasi parameter yang optimal untuk model tersebut.

2.1.10 *Word Cloud*

Word cloud merupakan salah satu metode visualisasi dokumen teks yang paling sering digunakan dimana merepresentasikan kata-kata pada dokumen secara grafis dua dimensi sesuai jumlah kemunculan kata tersebut. *Word cloud* merupakan gambaran visualisasi dari tabulasi frekuensi kata-kata dalam setiap bahan tertulis, seperti artikel jurnal dan berita *online* (Zuhri & Alamsyah, 2017).



Gambar 2.6 Visualisasi Data Teks dengan *Word Cloud*

Word cloud dapat memvisualisasikan besar frekuensi kata yang muncul melalui besar kecilnya ukuran huruf kata tersebut (Castella & Sutton, 2014). Semakin besar ukuran kata, semakin besar pula frekuensi kata tersebut muncul dalam dokumen. Contoh visualisasi dokumen teks dengan *word cloud* ditunjukkan pada Gambar 2.6.

2.2 ScienceDirect

ScienceDirect merupakan *platform* terkemuka dari Elsevier, penerbit *online* terkemuka yang berpusat di Belanda dalam literatur ilmiah hasil *peer-review* (Saputra, 2018). Sampai akhir tahun 2017, ScienceDirect masih menjadi *platform online* dengan jumlah terbitan jurnal terbanyak di dunia, terdiri dari 3.976 jurnal dan 48.124 buku. Saat ini, ScienceDirect menyediakan lebih dari 11 juta artikel, dengan pertumbuhan konten *database* mencapai 0,5 juta per tahun (Nashihuddin & Rahayu, 2013). Jurnal yang terdapat di ScienceDirect dikelompokkan ke dalam 4 kategori, yaitu *Physical Sciences and Engineering*, *Life Sciences*, *Health Sciences*, dan *Social*

Sciences and Humanities. Sub kategori dari setiap kategori dapat dilihat pada Tabel 2.4.

Tabel 2.4 Sub Kategori dari Kategori Jurnal pada ScienceDirect

Kategori	Subkategori
Physical Sciences and Enginering	- *Chemical Engineering* - *Chemistry* - *Computer Science* - *Earth and Planetary Sciences* - *Energy* - *Engineering* - *Material Sciences* - *Mathematics* - *Physics and Astronomy*
Life Sciences	- *Agricultural and Biological Science* - *Biochemistry, Genetics, dan Molecular Biology* - *Environmental Science* - *Immunology and Microbiology* - *Neuroscience*
Health Sciences	- *Medicine and Dentistry* - *Nursing and Health Profession* - *Pharmacology, Toxicology, and Pharmaceutical Science* - *Veterinary Science and Veterinary Medicine*
Social Sciences and Humanity	- *Arts and Humanities* - *Business, Management, and Finance* - *Decision Sciences* - *Economics, Econometrics, and Finance* - *Psychology* - *Social Sciences*

(*Halaman sengaja dikosongkan*)

BAB III
METODOLOGI PENELITIAN

3.1 Sumber Data

Sumber data yang digunakan pada penelitian ini adalah data sekunder yang diperoleh dari situs web ScienceDirect pada alamat https//www.sciencedirect.com. Data awal yang digunakan berupa abstrak dari setiap artikel jurnal yang terdapat di ScienceDirect yang dicari dengan menggunakan kata kunci "*Data Mining*" dan diterbitkan pada tahun 2019. Abstrak dari setiap jurnal tersebut diambil dengan menggunakan *software* JabRef.

3.2 Variabel Penelitian dan Struktur Data

Variabel-variabel yang digunakan dalam penelitian ini merupakan kategori/kelas/label yang terdapat dalam pangkalan data ScienceDirect dan *term*/kata yang terdapat pada abstrak dari artikel jurnal yang terpilih. Berikut ini merupakan variabel penelitian yang disajikan pada Tabel 3.1.

Tabel 3.1 Variabel Penelitian

Variabel	Nama Variabel	Skala
Y_1	Kategori *Physical Sciences and Engineering*	Nominal
Y_2	Kategori *Life Sciences*	Nominal
Y_3	Kategori *Health Sciences*	Nominal
Y_4	Kategori *Social Sciences and Humanities*	Nominal
X_j j = 1,2,...,m	Bobot *term*/kata ke-j yang muncul pada abstrak artikel jurnal	Rasio

Setelah data didapatkan dan kemudian dilakukan *preprocessing* terhadap data teks, akan didapatkan atribut berupa *term*/kata. Kemudian *term*/kata tersebut akan dibobotkan sesuai dengan jumlah kata tersebut pada dokumen abstrak jurnal sehingga didapatkan struktur data yang akan digunakan dalam penelitian ini seperti pada Tabel 3.2.

Tabel 3.2 Struktur Data Penelitian

i	X_1	X_2	X_3	⋯	X_m	Y_1	Y_2	Y_3	Y_4
1	$X_{1,1}$	$X_{1,2}$	$X_{1,3}$	⋯	$X_{1,m}$	$Y_{1,1}$	$Y_{1,2}$	$Y_{1,3}$	$Y_{1,4}$
2	$X_{2,1}$	$X_{2,2}$	$X_{2,3}$	⋯	$X_{2,m}$	$Y_{2,1}$	$Y_{2,2}$	$Y_{2,3}$	$Y_{2,4}$
3	$X_{3,1}$	$X_{3,2}$	$X_{3,3}$	⋯	$X_{3,m}$	$Y_{3,1}$	$Y_{3,2}$	$Y_{3,3}$	$Y_{3,4}$
⋮	⋮	⋮	⋮	⋱	⋮	⋮	⋮	⋮	⋮
n	$X_{n,1}$	$X_{n,2}$	$X_{n,3}$	⋯	$X_{n,m}$	$Y_{n,1}$	$Y_{n,2}$	$Y_{n,3}$	$Y_{n,4}$

keterangan :

$X_{i,j}$: bobot term/kata ke-*j* pada dokumen ke-*i*

$Y_{i,c}$: kategori *c* untuk dokumen ke-*i*

3.3 Langkah Analisis

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

1. Melakukan *web crawling* artikel jurnal pada situs web Science-Direct, yaitu https://www.sciencedirect.com/. Langkah-langkah dalam melakukan *web cralwing* sebagai berikut.
   a. Membuka situs web ScienceDierct, kemudian mencari artikel jurnal dengan kata kunci "*Data Mining*" dengan mesin pencarian dan memfilternya untuk artikel yang hanya diterbitkan pada tahun 2019. Akan muncul artikel jurnal dengan batasan yang sudah ditetapkan.
   b. Mengunduh sitasi artikel jurnal dengan format BibTeX.
   c. Membuka file dalam format BibTeX yang sudah diunduh dengan *software* JabRef. Kemudian mengekspor file tersebut ke dalam format *excel* dan memilih atribut yang akan digunakan dalam analisis, yaitu abstrak dari artikel jurnal yang sudah didapatkan.
2. Melakukan *preprocessing* data abstrak dari semua artikel jurnal dengan melakukan beberapa langkah berikut ini.
   a. Mengubah semua huruf kapital pada abstrak artikel jurnal menjadi huruf non-kapital. Proses ini disebut *case folding.*
   b. Menghapus karakter non-alpabet pada abstrak artikel jurnal. Proses ini disebut *delete punctuation.*

c. Menghapus karakter yang berupa angka pada abstrak artikel jurnal. Proses ini disebut *remove number*.
d. Mengubah setiap kalimat pada abstrak artikel jurnal menjadi sebuah token atau kata per kata. Proses ini disebut sebagai proses *tokenization.*
e. Menghapus kata-kata yang tidak memiliki makna, seperti kata "*the*", "*and*", "*or*", dan sejenisnya yang masuk dalam kategori *stop words.* Proses ini adalah *removing stopwords.*
f. Mengubah setiap kata menjadi kata dasarnya sesuai dengan kamus bahasa inggris. Proses ini disebut *lemmatization*.

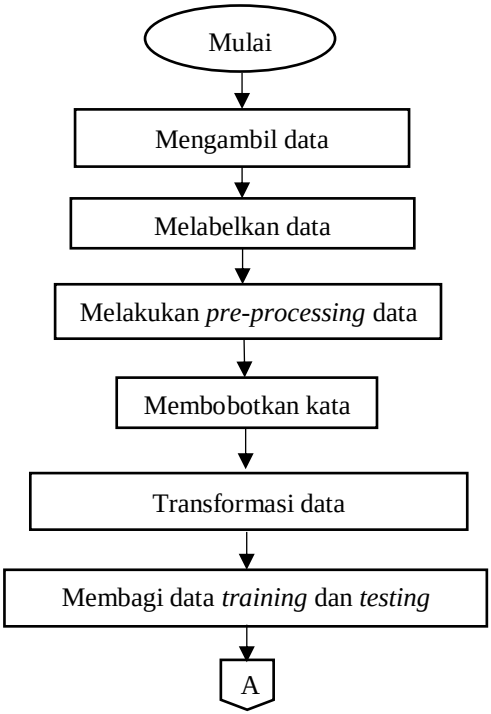
3. Melakukan pembobotan terhadap kata/*term* hasil *preprocessing* dengan menggunakan TF-IDF dengan rumus (2.4).
4. Melakukan transformasi data untuk diubah ke dalam pendekatan *single-label* dengan metode *Label Powerset* (LP).
5. Membagi data menjadi data *training* dan data *testing* dengan menggunakan *K-fold cross validation*.
6. Melakukan analisis klasifikasi *multi-label* dengan menggunakan metode KNN dan SVM dengan menggunakan data *training*. Adapun beberapa tahapan dalam metode KNN yakni sebagai berikut.
   a. Menentukan parameter *p* sebagai jumlah tetangga di sekitar objek. Parameter *p* tersebut ditentukan dengan menggunakan bilangan ganjil untuk menghindari mayoritas yang imbang dari tetangga terdekat.
   b. Menghitung jarak *euclidean* dari objek terhadap data *training* dengan rumus (2.5).
   c. Menentukan *p* data *training* yang terdekat dari objek.
   d. Mengklasifikasikan objek sesuai mayoritas label pada *p* data yang terdekat.

   Sedangkan tahapan dalam metode SVM yakni sebagai berikut.
   a. Menentukan nilai parameter *C* dan γ dari SVM serta parameter kernel.
   b. Mendapatkan nilai *Langrange Multiplier* yang optimum memanfaatkan rumus (2.25) dan mendapatkan *support vector*.
   c. Mendapatkan fungsi *hyperplane* untuk setiap kategori.

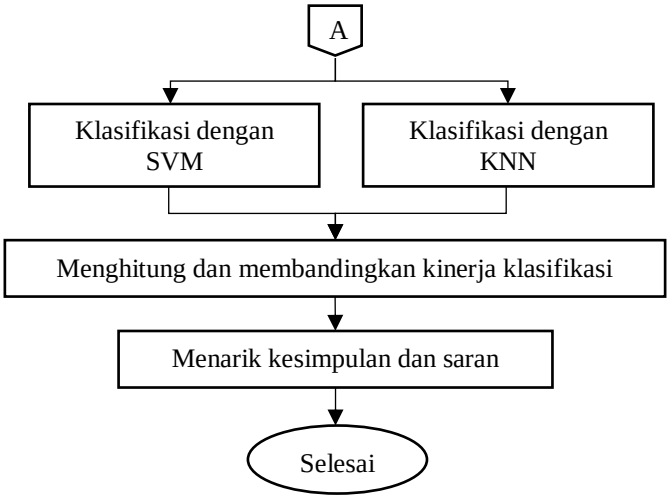
d. Mendapatkan hasil prediksi kategori artikel jurnal menggunakan persamaan (2.27).

7. Menghitung kinerja klasifikasi *multi-label* dengan metode *classifier* KNN dan SVM dengan pengukuran *nilai hamming-loss* dengan rumus (2.31). Kemudian membandingkan kebaikan kedua metode yang digunakan tersebut.
8. Melakukan visualisasi kata-kata yang paling sering muncul dalam setiap kategori hasil prediksi dengan model terbaik.
9. Menginterpretasikan hasil analisis, menarik kesimpulan dan saran berdasarkan hasil analisis.

Langkah-langkah 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)

(*Halaman sengaja dikosongkan*)

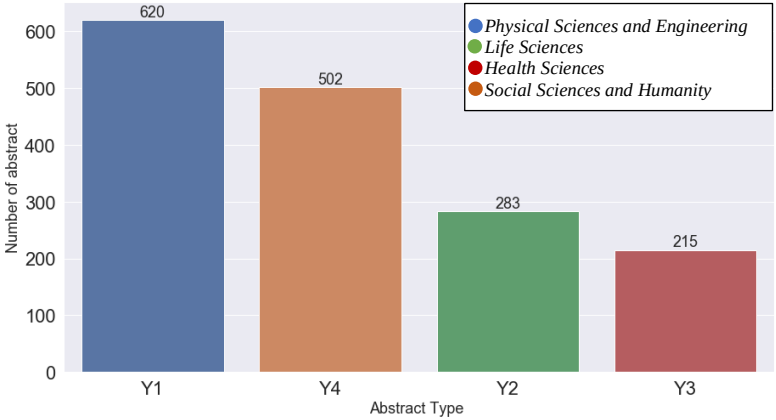
BAB IV
ANALISIS DAN PEMBAHASAN

Pada penelitian ini dilakukan klasifikasi *multi-label* artikel jurnal yang terdapat pada pangkalan data ScienceDirect berdasarkan abstrak setiap artikel jurnal menggunakan *K-Nearest Neighbors* (KNN) dan *Support Vector Machine* (SVM). Data yang digunakan adalah artikel jurnal yang dipublikasikan melalui ScienceDirect pada tahun 2019 menggunakan kata kunci "*Data Mining*" dengan jumlah artikel jurnal sebanyak 994 yang dipublikasi dan terbagi ke dalam empat kategori dimana jurnal-jurnal tersebut dapat masuk ke dalam dua atau lebih kategori. Kemudian akan dilakukan pengukuran ketepatan klasifikasi dengan menggunakan *Hamming loss.*

4.1 Karakteristik Artikel Jurnal ScienceDirect

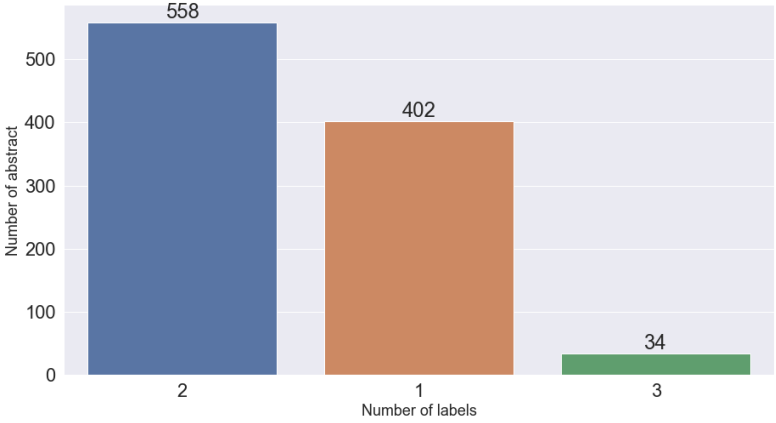
ScienceDirect merupakan salah satu *platform* terkemuka dan pangkalan data artikel jurnal ilmiah yang dikelola oleh Elsevier, penerbit *online* dalam hal literatur ilmiah terkemuka yang berpusat di Belanda. Data artikel jurnal tersebut terbagi ke dalam empat kategori dan setiap artikel dapat terbagi ke dalam dua atau lebih kategori berdasarkan abstrak masing-masing artikel. Dari data tersebut, dapat diketahui karakteristik data awal sebelum dilakukan analisis. Data tersebut memuat abstrak dari setiap artikel jurnal beserta kategorinya. Pertama-tama akan ditunjukkan terlebih dahulu sebaran artikel jurnal berdasarkan kategorinya pada Gambar 4.1.

Gambar 4.1 merupakan jumlah artikel jurnal di setiap kategori yang terdapat pada pangkalan data ScienceDirect dengan kata kunci "*Data Mining"* yang dipublikasikan pada tahun 2019. Dapat dilihat bahwa artikel yang paling banyak terdapat pada kategori *Physical Sciences and Engineering* dengan jumlah sebanyak 620 artikel. Hal ini bisa saja disebabkan karena kategori *Physical Sciences and Engineering* memiliki jumlah sub kategori terbanyak jika dibandingkan dengan kategori yang lain. Sementara itu, kategori *Health Sciences* menjadi kategori dengan jumlah artikel yang paling sedikit, yaitu sebanyak 215 artikel. Hal ini disebabkan karena



Gambar 4.1 Jumlah Artikel Jurnal Setiap Kategori

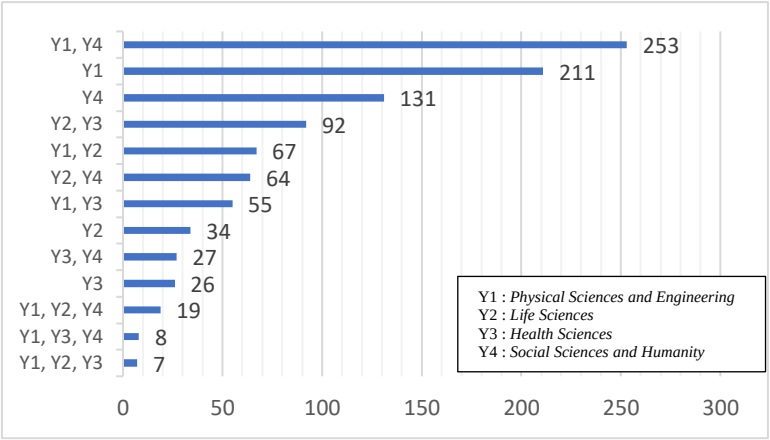
masih sedikit penelitian dalam bidang kesehatan yang memanfaatkan proses *data mining* yang dipublikasikan melalui *platform* ScienceDirect, selain karena kategori *Health Sciences* juga memiliki sub kategori yang lebih sedikit jika dibandingkan dengan kategori yang lain. Hal ini menunjukkan, *platform* ScienceDirect sebagian besar mempublikasikan artikel jurnal ilmiah dalam bidang *Physical Sciences* dan teknik serta bidang sosial sains yang memanfaatkan *data mining* pada tahun 2019.



Gambar 4.2 Jumlah Kategori yang Dimiliki oleh Setiap Artikel Jurnal

Selanjutnya ditunjukkan jumlah kategori yang dimiliki oleh setiap artikel jurnal pada Gambar 4.2. Dapat dilihat bahwa artikel jurnal dengan kata kunci "*Data Mining"* paling banyak dikategorikan ke dalam dua kategori sebanyak 558 artikel. Salah satu contoh artikel jurnal yang termasuk ke dalam dua kategori adalah artikel jurnal urutan pertama dengan judul "*An IoT based framework for energy monitoring and analysis of die casting workshop*" dan dapat dilihat pada Lampiran 1 dimana masuk ke dalam kategori *Physical Sciences and Engineering* dan kategori *Life Sciences* serta beberapa artikel jurnal yang lainnya yang terdapat pada Lampiran 1. Artikel jurnal yang hanya tergolong ke dalam satu kategori saja terdapat sebanyak 402 artikel. Salah satu contohnya adalah artikel jurnal urutan ke-989 dengan judul "*Learning-based network path planning for traffic engineering*" yang hanya masuk ke dalam kategori *Physical Sciences and Engineering* pada Lampiran 1. Sementara itu hanya terdapat sebanyak 34 artikel jurnal saja yang dikategorikan ke dalam 3 kategori. Salah satu contohnya adalah artikel jurnal urutan ke-143 dengan judul "*Screening of enhanced oil recovery techniques for Iranian oil reservoirs using TOPSIS algorithm*" yang masuk ke dalam kategori *Physical Sciences and Engineering*, kategori *Life Sciences*, dan kategori *Social Sciences and Humanity*. Persebaran artikel jurnal tersebut berdasarkan pa-sangan kategorinya akan ditunjukkan pada Gambar 4.3.

Gambar 4.3 menunjukkan sebaran dari setiap artikel jurnal yang lebih rinci. Dapat dilihat bahwa artikel jurnal yang termasuk ke dalam dua kategori *Physical Science and Engineering* dan *Social Sciences* dengan jumlah sebanyak 211 artikel jurnal. Hal ini bisa saja dikarenakan kedua kategori ini berkaitan dengan bidang teknik dan industri serta sosial sains yang notabene banyak penelitian yang dilakukan dengan memanfaatkan *data mining* yang mengarah kepada revolusi industri 4.0, jika dibandingkan dengan ilmu kesehatan dan juga ilmu lingkungan. Sedangkan yang paling sedikit adalah artikel jurnal yang termasuk dalam tiga kategori *Physical Sciences and Engineering*, *Life Sciences*, dan *Health Sciences* dengan jumlah hanya sebanyak 7 artikel jurnal.



Gambar 4.3 Jumlah Artikel Jurnal yang Termasuk ke dalam Satu atau Lebih Kategori

4.2 Klasifikasi Artikel Jurnal ScienceDirect Berdasarkan Abstrak

Selanjutnya akan dilakukan analisis klasifikasi artikel jurnal ScienceDirect bedasarkan abstrak dengan menggunakan *K-Nearest Neighbors* (KNN) dan *Support Vector Machine* (SVM). Tahap pertama yang harus dilakukan sebelum dilakukan klasifikasi adalah melakukan *pre-processing* data artikel jurnal sehingga siap untuk analisis lebih lanjut.

4.2.1 *Pre Processing* Data Artikel Jurnal

Pre-processing data dilakukan untuk menyiapkan data agar dapat dianalisis lebih lanjut dan memberikan hasil analisis yang lebih tepat. Dalam kasus data teks, tahapan-tahapan yang dilakukan adalah *case folding, delete punctuation, remove number, tokenization, remove stopwords,* dan *lemmatization.* Berikut ini akan ditunjukkan sebelum dan hasil *pre-processing* data artikel jurnal berdasarkan abstrak dengan *syntax* seperti pada Lampiran 3.

1. *Case Folding*

Case folding merupakan proses untuk mengubah semua karakter teks menjadi non kapital. Abstrak dari artikel jurnal tentu di-

tulis dengan aturan standar dan mengandung huruf kapital sehingga perlu ditransformasi menjadi huruf non kapital. Tabel 4.1 menunjukkan tahapan *case folding* terhadap data artikel jurnal.

Tabel 4.1 Contoh Data Sebelum dan Sesudah Proses *Case Folding*

Sebelum *Case Folding*	**Setelah *Case Folding***
In order to reduce the impact on resource shortage and environmental pollution caused by the massive use of energy, meeting the requirements...	in order to reduce the impact on resource shortage and environmental pollution caused by the massive use of energy, meeting the requirements...
⋮	⋮
Soft computing techniques are becoming even more popular and particularly amenable to model the complex behaviors of most geotechnical engineering systems...	soft computing techniques are becoming even more popular and particularly amenable to model the complex behaviors of most geotechnical engineering system...

Setelah proses *case foding* selesai dilakukan, selanjutnya adalah melakukan proses *delete punctuation.*

2. *Delete Punctuation*

Delete punctuation merupakan proses untuk menghilangkan tanda baca yang terdapat pada data teks. Tanda baca titik (.), koma (,), dan beberapa tanda baca lainnya sering sekali dijumpai dalam tulisan artikel ilmiah sehingga perlu dihapuskan terlebih dahulu. Contoh teks sebelum dan sesudah *delete removal* ditunjukkan pada Tabel 4.2.

Tabel 4.2 Contoh Data Sebelum dan Sesudah Proses *Delete Punctuation*

Sebelum *Delete Punctuation*	**Setelah *Delete Punctuation***
in order to reduce the impact on resource shortage and environmental pollution caused by the massive use of energy, meeting the requirements...	in order to reduce the impact on resource shortage and environmental pollution caused by the massive use of energy meeting the requirements...
⋮	⋮
... extreme gradient boosting (xgboost), multivariate adaptive regression splines (mars), artificial neural networks (ann), and ...	... extreme gradient boosting xgboost multivariate adaptive regression splines mars artificial neural networks ann and ...

Setalah proses *delete punctuation* selesai dilakukan, langkah selanjutnya adalah proses *remove number.*

3. *Remove Number*

Remove number merupakan proses untuk menghapus angka yang terdapat dalam teks. Dalam hal ini, angka pada teks dianggap sebuah karakter dan biasanya tidak dianggap sebagai kata yang penting sehingga perlu untuk dihapus. Pada artikel jurnal yang merupakan teks ilmiah tidak lepas dari penulisan angka, baik angka sebagai waktu, nilai dari sebuah hasil penelitian, nilai sebuah sampel, hingga nilai-nilai yang lain yang ditunjukkan dengan angka. Angka-angka tersebut perlu dihapus untuk dapat dianalisis lebih lanjut dan memberikan hasil yang lebih baik. Contoh artikel jurnal sebelum dan sudah melalui proses *remove number* ditunjukkan pada Tabel 4.3.

Tabel 4.3 Contoh Data Sebelum dan Sesudah Proses *Remove Number*

Sebelum *Remove Number*	**Setelah *Remove Number***
... from big data all ethiopian demographic and health survey datasets from 2000 to 2016 were used for this study ...	... from big data all ethiopian demographic and health survey datasets from to were used for this study ...
⋮	⋮
... applied to assess the generalization of the proposed system the results of the experiment show that the accuracy of the system is 81	... applied to assess the generalization of the proposed system the results of the experiment show that the accuracy of the system is

Setelah proses *remove number* telah dilakukan, langkah selanjutnya adalah *tokenization* dan *remove stopwords.*

4. *Tokenization* dan *Remove Stopwords*

Tokenization merupakan proses untuk memecah keseluruhan teks yang sebelumnya berupa kalimat menjadi kata per kata. Sedangkan *remove stopwords* merupakan proses untuk menghapus kata yang tidak memiliki arti penting atau yang dalam bahasa Inggris disebut *stopwords.* Kata-kata yang termasuk ke dalam *stopwords* adalah "*is*", "*have*", "*has",* dan beberapa kata sejenis. Con-

toh abstrak artikel jurnal sebelum dan sesudah proses *tokenization* dan *remove stopwords* ditunjukkan pada Tabel 4.4.

Tabel 4.4 Contoh Data Sebelum dan Sesudah *Tokenization* dan *Remove Stopwords*

Sebelum *Tokenization* dan *Remove Stopwords*	**Setelah *Tokenization* dan *Remove Stopwords***
in order to reduce the impact on resource shortage and environmental pollution caused by the massive use of energy meeting ...	'order', 'reduce', 'impact', 'resource', 'shortage', 'environmental', 'pollution', 'caused', 'massive', 'use', 'energy', 'meeting', ...
⋮	⋮
soft computing techniques are becoming even more popular and particularly amenable to model the complex behaviors of most geotechnical ...	'soft', 'computing', 'techniques', 'becoming', 'even', 'popular', 'particularly', 'amenable', 'model', 'complex', 'behaviors', 'geotechnical', ...

Setelah proses *tokenization* dan *remove stopwords* selesai dilakukan, maka dilanjutkan ke proses yang terakhir, yaitu *lemmatization.*

5. *Lemmatization*

Lemmatization merupakan normalisasi pada sebuah kata dimana berbagai varian atau jenis morfologis suatu kata diubah menjadi kata dasar yang sama dengan memperhatikan kamus. Dalam artikel jurnal yang terdapat pada pangkalan data ScienceDirect, bahasa yang digunakan yaitu Bahasa Inggris sebagai bahasa internasional sehingga kata-kata yang terdapat dalam setiap abstrak artikel jurnal diubah menjadi kata dasarnya yang memiliki arti yang sama menurut kamus Bahasa Inggris. Contoh artikel abstrak sebelum dan sesudah proses *lemmatization* ditunjukkan pada Tabel 4.5.

Hasil dari proses *lemmatization* yang dalam bentuk kata per kata seperti pada Tabel 4.5 akan digunakan sebagai kata kunci dari abstrak artikel jurnal. Setelah itu dibentuk struktur data yang baru dengan masing-masing kata kunci yang sudah didapatkan tersebut menjadi variabel dan diketahui frekuensi kemunculannya di setiap abstrak artikel jurnal dengan *syntax* seperti pada Lampiran 4.

Tabel 4.5 Contoh Data Sebelum dan Sesudah Proses *Lemmatization*

Sebelum *Lemmatization*	**Setelah *Lemmatization***
'order', 'reduce', 'impact', 'resource', 'shortage', 'environmental', 'pollution', 'caused', 'massive', 'use', 'energy', 'meeting', 'requirements', 'fierce', 'market', ...	'order', 'reduce', 'impact', 'resource', 'shortage', 'environmental', 'pollution', 'cause', 'massive', 'use', 'energy', 'meet', 'requirement', 'fierce', 'market', ...
⋮	⋮
'soft', 'computing', 'techniques', 'becoming', 'even', 'popular', 'particularly', 'amenable', 'model', 'complex', 'behaviors', 'geotechnical', 'engineering', 'systems', ...	'soft', 'compute', 'technique', 'become', 'even', 'popular', 'particularly', 'amenable', 'model', 'complex', 'behavior', 'geotechnical', 'engineer', 'system', ...

Struktur data baru yang didapatkan hasil dari *pre-processing* data artikel jurnal ditampilkan pada Tabel 4.6.

Tabel 4.6 Struktur Data Baru Berdasarkan Kata Kunci Abstrak Artikel Jurnal

No	**aa**	⋯	**increase**	⋯	**order**	⋯	**process**	⋯	**zoonotic**
1	0	⋯	0	⋯	1	⋯	2	⋯	0
2	0	⋯	1	⋯	0	⋯	1	⋯	0
3	0	⋯	0	⋯	1	⋯	2	⋯	0
⋮	⋮	⋱	⋮	⋱	⋮	⋱	⋮	⋱	⋮
127	0	⋯	0	⋯	0	⋯	0	⋯	1
⋮	⋮	⋱	⋮	⋱	⋮	⋱	⋮	⋱	⋮
985	1	⋯	0	⋯	0	⋯	0	⋯	0
⋮	⋮	⋱	⋮	⋱	⋮	⋱	⋮	⋱	⋮
994	0	⋯	0	⋯	0	⋯	0	⋯	0

Tabel 4.6 adalah hasil perhitungan frekuensi kata kunci pada setiap dokumen artikel jurnal. Abstrak artikel jurnal pertama didapatkan bahwa kata “order” ditulis sebanyak 1 kali, tetapi tidak terdapat pada abstrak artikel jurnal ke-2, ke-127, dan ke-985. Selanjutnya kata “aa” terdapat pada abstrak artikel jurnal ke-985, tetapi tidak terdapat di abstrak artikel jurnal yang pertama dan seterusnya. Pada abstrak atikel jurnal ke-127 terdapat kata “zoonotic”, tetapi tidak terdapat kata “aa”, “increase”, “order”, dan “process”. Secara total didapatkan sebanyak 9421 kata kunci dari 994 abstrak artikel jurnal

yang terdapat pada pangkalan data ScienceDirect. Selanjutnya dari struktur data di atas, akan dilakukan pembobotan untuk masing-masing kata menggunakan *Term Frequency – Inverse Document Frequency* (TF-IDF) menggunakan persamaan (2.3) dan (2.4) dan hasil perhitungannya ditunjukkan pada Lampiran 2.

4.2.2 Transformasi Kategori Artikel Jurnal Menggunakan *Label Powerset*

Salah satu pendekatan yang dapat digunakan untuk melakukan klasifikasi data *multi-label* adalah dengan menggunakan pendekatan transformasi data. Transformasi data yang digunakan dalam klasifikasi artikel jurnal ScienceDirect pada penelitian ini adalah dengan transformasi *Label Powerset.* Transformasi *Label Powerset* pada kategori artikel jurnal ScienceDirect yang sebelumnya berupa *multi-label* diubah menjadi data kategori *multi-class.* Data awal dari kategori artikel jurnal sebelum dilakukan transformasi *Label Powerset* ditunjukkan pada Tabel 4.7 berikut.

Tabel 4.7 Data Kategori Artikel Jurnal Sebelum Ditransformasi

No.	aa	⋯	market	⋯	zoonotic	Y1	Y2	Y3	Y4
1	0	⋯	0.028729	⋯	0	1	1	0	0
2	0	⋯	0	⋯	0	0	1	0	1
3	0	⋯	0	⋯	0	0	1	1	0
4	0	⋯	0	⋯	0	1	0	0	1
5	0	⋯	0	⋯	0	1	0	0	1
⋮	⋮	⋱	⋮	⋱	⋮	⋮	⋮	⋮	⋮
989	0	⋯	0	⋯	0	0	1	1	0
990	0	⋯	0	⋯	0	0	0	1	1
991	0	⋯	0	⋯	0	1	0	1	0
992	0	⋯	0	⋯	0	1	0	0	1
994	0	⋯	0	⋯	0	1	1	0	0

Tabel 4.7 merupakan data yang belum dilakukan transformasi dimana kategori dari data hasil *pre-processing* dan masih dalam bentuk *multi-label*. Dapat dilihat pada artikel jurnal pertama termasuk

ke dalam kategori Y1 dan Y2 yang merupakan kategori *Physical Sciences and Engineering* dan *Life Scieces.* Begitu pula pada artikel jurnal urutan kedua dan seterusnya yang masih tergolong ke dalam jenis data *multi-label.* Data kategori tersebut kemudian ditransformasi ke dalam data *single-label* dengan transformasi *Label Powerset.* Hasil transformasi data kategori tersebut dapat dilihat pada Tabel 4.8 berikut.

Tabel 4.8 Data Kategori Artikel Jurnal yang Sudah Ditransformasi

No.	aa	⋯	market	⋯	zoonotic	Y_Transform
1	0	⋯	0,028729	⋯	0	0
2	0	⋯	0	⋯	0	1
3	0	⋯	0	⋯	0	2
4	0	⋯	0	⋯	0	3
5	0	⋯	0	⋯	0	3
⋮	⋮	⋱	⋮	⋱	⋮	⋮
989	0	⋯	0	⋯	0	2
990	0	⋯	0	⋯	0	6
991	0	⋯	0	⋯	0	5
992	0	⋯	0	⋯	0	3
994	0	⋯	0	⋯	0	0

Hasil dari transformasi data kategori artikel jurnal dengan menggunakan *Label Powerset* merupakan data *multi-class* seperti yang ditunjukkan pada Tabel 4.8. Artikel jurnal urutan pertama pada data yang sebelumnya masuk ke dalam kategori *Physical Sciences and Engineering* dan *Life Sciences*, setelah ditransformasi menjadi kategori dengan label 0. Begitu pula pada kategori urutan kedua pada data artikel jurnal yang ditransformasi menjadi kategori dengan label 1 yang sebelumnya masuk ke dalam kategori *Life Sciences* dan kategori *Social Sciences and Humanity.* Begitu pula pada artikel jurnal ketiga dan seterusnya, kombinasi dari setiap kategori artikel jurnal tersebut ditransformasi menjadi kategori baru dan terdapat 13 kategori hasil kombinasi dari setiap kategori artikel jurnal dan

diberikan label dengan angka 0 hingga 12. Hasil dari transformasi data kategori artikel jurnal tersebut kemudian menjadi data *multi-class* dengan 13 kategori baru. Artikel-artikel jurnal tersebut kemudian di kategorikan ke dalam 13 kategori baru. Selanjutnya data tersebut dapat digunakan untuk analisis lebih lanjut dengan metode *K-Nearest Neighbor* (KNN) dan *Support Vector Machine* (SVM) dan setelah itu data kategori dari artikel jurnal tersebut akan dikembalikan ke dalam bentuk *multi-label* seperti awal.

4.2.3 Klasifikasi Artikel Jurnal Menggunakan *K-Nearest Neighbor* (KNN)

K-Nearest Neighbor (KNN) merupakan metode klasifikasi yang paling umum untuk digunakan. Artikel jurnal yang terdapat pada pangkalan data ScienceDirect tergolong ke dalam data *multi-label* dimana setiap artikel jurnal memiliki satu atau lebih kategori. Selanjutnya akan dilakukan klasifikasi artikel jurnal ScienceDirect dengan menggunakan *classifier* KNN. Setelah data artikel jurnal *multi-label* ditransformasi ke dalam *multi-class* dengan metode *Label Powerset*, dilanjutkan dengan klasifikasi dengan KNN.

Data artikel jurnal yang sudah ditransformasi dibagi menjadi *training* dan *testing* dengan menggunakan metode *K-Fold Cross Validation* untuk meminimalkan hasil yang bias dimana dibagi menjadi 5 bagian. Jarak antar objek artikel jurnal dihitung menggunakan jarak *Euclidean* seperti pada persamaan (2.5) dan jumlah tetangga terdekat yang akan digunakan 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, dan 29. Parameter dengan angka ganjil untuk menghindari mayoritas yang imbang dari tetangga terdekat. Kemudian mendapatkan parameter *p* jumlah tetangga terdekat dengan menggunakan algoritma *Grid Search* untuk mendapatkan model yang paling optimal untuk digunakan pada model. Prinsip *grid search* yang digunakan mengacu dari penelitian yang telah dilakukan oleh Ranjan, Verma, & Radhika (2019). Penentuan parameter *p* dihitung dengan menggunakan *hamming loss.* Berikut hasil penentuan parameter *p* tetangga terdekat menggunakan *hamming loss* ditunjukkan pada Tabel 4.9.

Tabel 4.9 Penentuan Parameter KNN Menggunakan *Grid Search* dengan *Nilai Hamming Loss*

p	*Fold*					Rata-rata
	1	**2**	**3**	**4**	**5**	
3	0,25754	0,26256	0,25251	0,26131	0,26389	0,25956
5	0,25251	0,25126	0,24749	0,22864	0,23359	0,24271
7	0,26005	0,23869	0,24497	0,21985	0,21843	0,23642
9	0,23744	0,22739	0,24372	0,20980	0,20833	0,22535
11	0,23618	0,22362	0,23367	0,21106	0,19066	0,21906
13	0,23241	0,21859	0,23492	0,19849	0,18308	0,21353*
15	0,24121	0,22362	0,23618	0,20101	0,19571	0,21957
17	0,23869	0,23995	0,23367	0,21357	0,18687	0,22259
19	0,23367	0,25628	0,22111	0,21482	0,18813	0,22284
21	0,23744	0,24246	0,23116	0,20477	0,18561	0,22032
23	0,23618	0,25628	0,22739	0,20854	0,18056	0,22183
25	0,23116	0,25503	0,22613	0,21357	0,18434	0,22208
27	0,23618	0,25503	0,22111	0,21859	0,18687	0,22359
29	0,22864	0,25503	0,21608	0,20980	0,17677	0,21730

Keterangan : (*) Nilai rata-rata *hamming loss* terkecil

Berdasarkan Tabel 4.9, model yang paling optimum adalah model KNN dengan parameter 13 tetangga terdekat. Model KNN dengan parameter tersebut menghasilkan kebaikan model dengan nilai *hamming loss* sebesar 21,353%. Nilai tersebut menunjukkan bahwa tingkat kesalahan model tersebut dalam memprediksi label artikel jurnal atau tingkat misklasifikasinya sebesar 21,353%. Sehingga parameter yang akan digunakan dalam model KNN selanjutnya yaitu dengan parameter 13 tetangga terdekat dan hasil pemodela menggunakan *Cross Validation* dapat dilihat pada Tabel 4.10.

Tabel 4.10 Hasil Kebaikan Model Klasifikasi KNN

Fold	*Training*	*Testing*
1	0,18283	0,20488*
2	0,18782	0,21535
3	0,18122*	0,26010
4	0,18703	0,20918
5	0,18976	0,21373
Rata-rata	0,18573	0,22065

Keterangan : (*) Nilai *hamming loss* terkecil

Hasil kebaikan model KNN yang terdapat pada Tabel 4.10 menunjukkan bahwa dengan menggunakan parameter 13 tetangga terdekat hasil *hamming loss* yang paling optimal dari setiap *fold* pada data *training* adalah sebesar 18,122% dan rata-rata *hamming loss* sebesar 18,573%. Ketika model diterapkan pada data *testing* didapatkan nilai *hamming loss* paling optimal dari setiap *fold* yaitu sebesar 20,488%. Sementara itu, rata-rata nilai *hamming loss* yang didapatkan dari data *testing* sebesar 22,065%. Nilai tersebut menggambarkan bahwa tingkat kesalahan prediksi dari model atau misklasifikasi kategori untuk semua artikel jurnal dengan menggunakan KNN hanya sebesar 22,065%. *Syntax* yang digunakan untuk metode KNN dapat dilihat pada Lampiran 5.

4.2.4 Klasifikasi Artikel Jurnal Menggunakan *Support Vector Machine* (SVM)

Selanjutnya artikel jurnal akan ScienceDirect diklasifikasikan menggunakan *Support Vector Machine* (SVM). Seperti sebelumnya pada klasifikasi menggunakan KNN, data *multi-label* artikel jurnal ScienceDirect diubah menjadi *single-label* terlebih dahulu dengan menggunakan transformasi *Label Powerset.* Setelah ditransformasi dengan transformasi *Label Powerset* maka data yang semula *multi-label* menjadi data *multi-class* dan dapat dilanjutkan ke analisis berikutnya dengan SVM seperti biasa.

Parameter yang optimal untuk model SVM didapatkan dengan menggunakan algoritma *Grid Search* seperti yang dilakukan sebelumnya pada KNN. Prinsip Grid Search yang digunakan mengacu penelitian yang telah dilakukan oleh Hsu, Chang, dan Lin (2003), yaitu mencobakan beberapa kombinasi parameter dengan pertambahan eksponensial. Namun penambahan nilai parameter yang digunakan dalam menentukan model optimal dalam penelitian ini tidak semua secara eksponensial, untuk selain parameter *cost C* dilakukan secara berurutan dalam rentang tertentu. Kernel yang digunakan dalam penelitian ini, yaitu RBF dan *polynomial* dimana dapat menghasilkan nilai kinerja klasifikasi yang paling optimal untuk data *multi-class* (Octaviani, Wilandari, & Ispriyanti, 2014). Parameter yang digunakan untuk kernel RBF, yaitu *cost C* dan ga-

ma γ. Rentang nilai parameter C yang digunakan yaitu dari 10^{-3} hingga 1 dan parameter γ dari 0,1 hingga 0,4 dengan penambahan 0,1. Penentuan parameter yang paling optimal menggunakan nilai *hamming loss* yang terkecil. Parameter yang paling optimal yang didapatkan seperti pada Tabel 4.11.

Tabel 4.11 Penentuan Parameter SVM dengan Kernel RBF Menggunakan *Grid Search dengan Nilai Hamming Loss*

Rank	**Parameter**		**Rata-rata *Hamming Loss***
	C	***γ***	
1	1	0.4	0.27163*
2	1	0.3	0.28169
3	1	0.2	0.309356
4	1	0.1	0.342052
⋮	⋮	⋮	⋮
13	0.1	0.1	0.343058
14	0.1	0.2	0.343058
15	0.1	0.3	0.343058
16	0.1	0.4	0.343058

Keterangan :
(*) Nilai *hamming loss* terkecil

Tabel 4.11 menunjukkan hasil dari *tunning* parameter yang menghasilkan model SVM dengan kernel RBF yang terbaik dan diurutkan dari kombinasi parameter yang menghasilkan nilai *hamming loss* paling optimal. Didapatkan sebanyak 16 kombinasi parameter hasil *tuning* dengan *Grid Search* dan parameter yang digunakan untuk menghasilkan nilai kinerja klasifikasi yang terbaik yaitu menggunkan parameter $C = 1$ dan gamma $\gamma = 0{,}4$. Nilai rata-rata *hamming loss* yang didapatkan menggunakan parameter tersebut sebesar 21,13%.

Selanjutnya ditentukan parameter yang paling optimal untuk kernel *polynomial*. Parameter yang digunakan untuk kernel *polynomial* hampir sama seperti RBF namun terdapat parameter yang lain, yaitu *coefficient r* dan *degree d*. Penentuan parameter yang optimal dilakukan dengan menggunakan algoritma *Grid Search*, dengan mengombinasikan parameter C secara eksponensial rentang dari 10^{-3} hingga 10^{0}, parameter r dengan nilai -4, -2, 1, 2, dan

4, parameter *d* dengan nilai 4, 5, dan 6, serta parameter γ dengan rentang nilai 3 hingga 6 dengan penambahan 1. Hasil kombinasi dari parameter-parameter tersebut disajikan pada Tabel 4.12.

Tabel 4.12 Penentuan Parameter SVM dengan Kernel *Polynomial* Menggunakan *Grid Search dengan Nilai Hamming Loss*

Rank	**Parameter**				**Rata-rata *Hamming Loss***
	C	***r***	***d***	***γ***	
1	0.01	-2	5	4	0.21127*
2	0.001	-4	5	3	0.21353
3	0.001	-4	5	5	0.21806
4	0.001	-4	5	4	0.21831
5	0.01	-2	5	3	0.22108
⋮	⋮	⋮	⋮	⋮	⋮
236	0.01	-2	6	5	0.57897
237	0.1	-2	6	5	0.59658
238	1	-2	6	5	0.59733
239	0.1	-2	4	6	0.60312
240	1	-2	4	6	0.61142

Keterangan : (*) Nilai *hamming loss* terkecil

Terdapat sebanyak 240 kombinasi hasil dari *tuning* setiap parameter yang dicoba dan didapatkan parameter yang paling optimal adalah C = 0,01, r = -2, d = 5, dan γ = 4 bagi SVM dengan kernel *polynomial.* Kombinasi setiap parameter tersebut dapat dilihat di Tabel 4.12. Nilai rata-rata *hamming loss* yang didapatkan adalah sebesar 21,127%. Selanjutnya hasil perbandingan kernel RBF dan *Polynomial* ditunjukkan pada Tabel 4.13.

Tabel 4.13 Perbandingan Kernel RBF dan *Polynomial*

Fold	***Training***		***Testing***	
	RBF	*Polynomial*	RBF	*Polynomial*
1	0,18821	0,16065	0,27195	0,18659*
2	0,17393	0,14886*	0,25990	0,19802
3	0,18122	0,15861	0,29040	0,22475
4	0,18139	0,15257	0,27296	0,22194
5	0,17634	0,14981	0,24882	0,20207
Rata-rata	0,18022	0,15330*	0,26801	0,20667*

Keterangan : (*) Nilai *hamming loss* terkecil

Hasil perbandingan antara kernel RBF dan *polynomial* menunjukkan bahwa kernel yang paling terbaik digunakan dalam mengklasifikasikan artikel jurnal ScienceDirect adalah kernel *polynomial.* Dapat dilihat pada Tabel 4.13 bahwa baik dari data *training* maupun data *testing* bahwa nilai *hamming loss* kernel *polynomial* lebih kecil dari kernel RBF. Didapat nilai *hamming loss* terkecil dengan menggunakan kernel *polynomial* terdapat pada *fold* ke-2 pada data *training* sebesar 14,886%. Sedangkan pada data *testing* nilai *hamming loss* terkecil terdapat pada *fold* pertama sebesar 18,695%. Sementara itu rata-rata nilai *hamming loss* pada data *training* dengan kernel *polynomial* sebesar 15,33% dan ketika diterapkan pada data *testing* didapatkan rata-rata nilai *hamming loss* sebesar 20,667% yang menunjukkan bahwa tingkat kesalahan klasifikasi SVM dalam memprediksi kategori atau misklasifikasi label dari artikel jurnal ScienceDirect hanya sebesar 20,667%. Sehingga untuk selanjutnya yang digunakan adalah SVM dengan parameter kernel *polynomial, C* = 0,01, *r* = -2, *d* = 5, dan $\gamma = 4$.

Parameter yang sudah didapatkan tersebut kemudian disubstitusikan ke dalam persamaan kernel *polynomial* pada Tabel 2.1 sehingga didapatkan fungsi kernelnya sebagai berikut.

$$K(\mathbf{X}, \mathbf{X}_i) = (4\mathbf{x}^T \mathbf{x}_i - 2)^5$$

Kemudian fungsi kernel yang sudah didapatkan digunakan untuk mendapatkan persamaan *hyperplane* sehingga didapatkan fungsi seperti berikut.

$$f_l(\mathbf{x}) = \sum_{i=1}^{792} a_i y_i (4\mathbf{x}^T \mathbf{x}_i - 2)^5 + b_l$$

Fungsi tersebut dibangun dari data *training* sejumlah 795 artikel jurnal dengan b_l merupakan nilai *bias* pada fungsi *hyperplane* label ke-*l*, a_i merupakan matriks koefisien dari *support vector* dan y_i kelas dari *support vector*. Nilai dari vektor **x** di substitusi sebanyak 13 model yang terbentuk dan penentuan hasil prediksi kelas artikel jurnal dengan menggunakan persamaan berikut.

$$\hat{y} = \arg \max_{l=1,\ldots,13} (f_l(\mathbf{x}))$$

dimana jumlah kelas yang terdapat pada data artikel jurnal ScienceDirect adalah 13 kelas. Setelah didapatkan kelas prediksi hasil dari SVM dalam bentuk *multi-class,* data kategori tersebut akan ditransformasi kembali ke dalam data kategori *multi-label* awal. *Syntax* yang digunakan dalam melakukan klasifikasi menggunakan SVM dapat dilihat pada Lampiran 6.

4.2.5 Pebandingan Klasifikasi Artikel Jurnal Menggunakan *K-Nearest Neighbor* (KNN) dan *Support Vector Machine* (SVM)

Setelah didapatkan model klasifikasi artikel jurnal ScienceDirect dengan menggunakan KNN dan SVM, kinerja model klasifikasi kedua *classifier* tersebut dibandingkan. Perlu diketahui metode yang lebih sesuai untuk diterapkan dalam mengategorikan artikel jurnal yang terdapat pada ScienceDirect menggunakan nilai *hamming loss* dari kedua *classifier* tersebut. Model KNN dan SVM dengan masing-masing parameter yang memberikan hasil paling optimal dibandingkan seperti pada Tabel 4.14.

Tabel 4.14 Perbandingan Klasifikasi Artikel Jurnal Antara KNN dan SVM

Classifier	*Training*	*Testing*
KNN	0,18573	0,22065
SVM	0,15330*	0,20667*

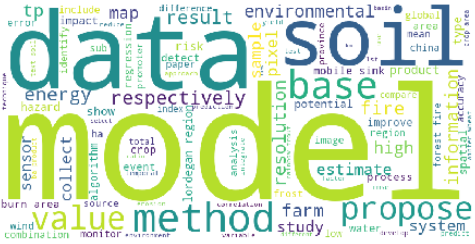
Keterangan :
(*) Nilai *hamming loss* terkecil

Hasil klasifikasi menggunakan SVM memiliki prediksi yang lebih baik dimana memiliki nilai *hamming loss* yang paling kecil seperti pada Tabel 4.14. Nilai *hamming loss* pada data *training* yang dihasilkan model SVM sebesar 15,33%, lebih kecil jika dibandingkan dengan yang dihasilkan model KNN, yaitu sebesar 18,573%. Begitu pula pada data *testing,* model SVM menghasilkan nilai *hamming loss* sebesar 20,667% yang lebih kecil dari nilai *hamming loss* yang dihasilkan oleh model KNN, yaitu sebesar 22,065%. Sehingga model klasifikasi SVM memberikan hasil yang lebih baik pada data artikel jurnal ScienceDirect. Hasil prediksi kategori yang diberikan oleh model SVM dengan parameter kernel *polynomial, C* = 0,01, r = -2, d = 5, dan γ = 4 menggunakan pembagian data *training*

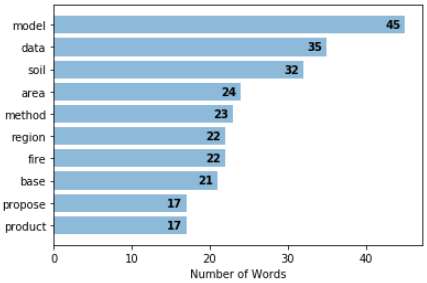
dan *testing* dari *fold* terbaik dari *Cross Validation* dapat dilihat pada Lampiran 8.

4.2.6 Visualisasi *Word Cloud* Hasil Prediksi Kategori Artikel Jurnal

Selanjutnya dilakukan visualisasi *word cloud* untuk setiap kategori hasil prediksi data *training* untuk mendapatkan kata kunci dari setiap kombinasi label/kategori. Artikel jurnal ScienceDirect yang termasuk ke data *training* tersebut diprediksi ke dalam 9 kombinasi kategori dan terdapat 4 kombinasi kategori pada data aktual yang tidak ada pada kategori hasil prediksi, contohnya seperti kategori *Health Sciences* dan kategori *Physical Sciences and Engineering*, *Life Sciences*, dan *Health Sciences* karena frekuensi kategori tersebut sangat sedikit jika dibandingkan dengan kombinasi kategori yang lainnya.



(a)

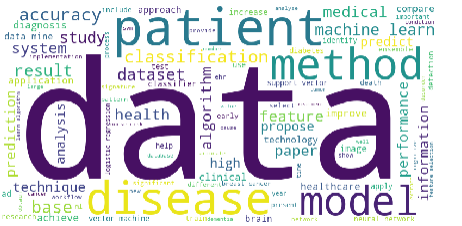


(b)

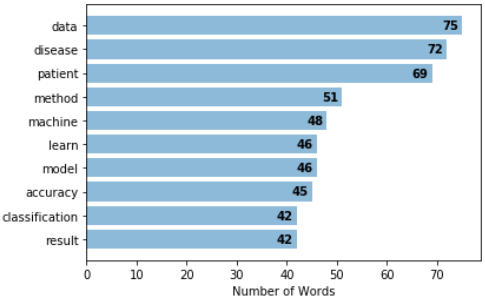
Gambar 4.4 Kriteria Artikel Jurnal dengan Kategori *Physical Sciences and Engineering* dan *Life Sciences* (a) Kata Kunci dan (b) Frekuensi Kata Terbanyak

Kata kunci yang didapatkan *word cloud* menjadi kriteria kategori tertentu pada artikel jurnal ScienceDirect. Kata kunci didapatkan dengan melihat kata yang paling banyak muncul dalam setiap kategori yang ditandai dengan ukuran kata yang paling besar. Berikut ini disajikan kombinasi kategori artikel jurnal yang hasil prediksi dari data *training* dan didapatkan 6 kombinasi kategori *multi-label*. Hasi dari visualisasi *word cloud* untuk kategori *Physical Sciences and Engineering* dan *Life Sciences* ditunjukkan pada Gambar 4.4.

Kata kunci yang didapatkan pada kategori *Physical Sciences and Engineering* dan *Life Sciences* adalah "*model*", dimana kata tersebut memiliki nilai frekuensi yang paling banyak seperti pada Gambar 4.4 (b). Selain itu, kata "*data*", "*soil*", "*area*", dan "*method*" menjadi kata-kata yang cenderung menjadi kriteria artikel jurnal yang masuk ke dalam kategori *Physical Sciences and Engineering* dan *Life Sciences* karena memiliki frekuensi yang paling banyak setelah kata "*model*".



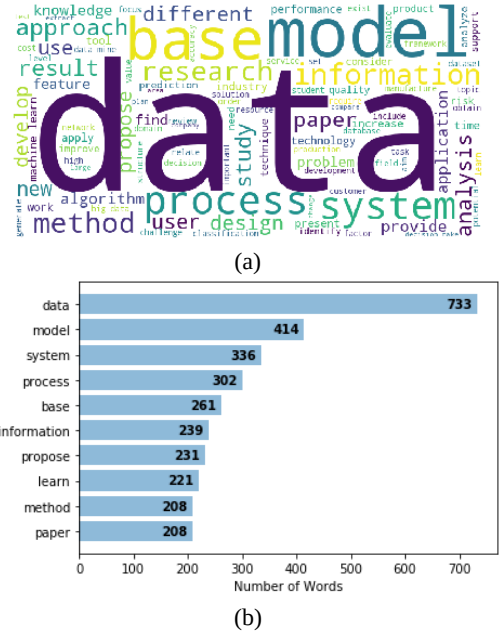
(a)



(b)

Gambar 4.5 Kriteria Artikel Jurnal Kategori *Physical Sciences and Engineering* dan *Health Sciences* (a) Kata Kunci dan (b) Frekuensi Kata Terbanyak

Kata kunci yang didapatkan untuk kategori *Physical Sciences and Engineering* dan *Health Sciences* seperti pada Gambar 4.5(a) adalah "*data*". Hal ini mengindikasikan bahwa kata "*data*" memiliki frekuensi yang paling banyak seperti pada Gambar 4.5(b). Selain kata tersebut, kata yang memiliki frekuensi yang tergolong banyak adalah "*disease*", "*patient*", "*method*", dan "*machine*" yang menjadi kriteria kata kunci yang cenderung terdapat pada artikel jurnal yang tergolong ke dalam kategori *Physical Sciences and Engineering* dan *Health Sciences.* Berikutnya didapatkan kata kunci yang terdapat pada kategori *Physical Sciences and Engineering* dan *Social Sciences and Humanity* seperti pada Gambar 4.6.



(a)

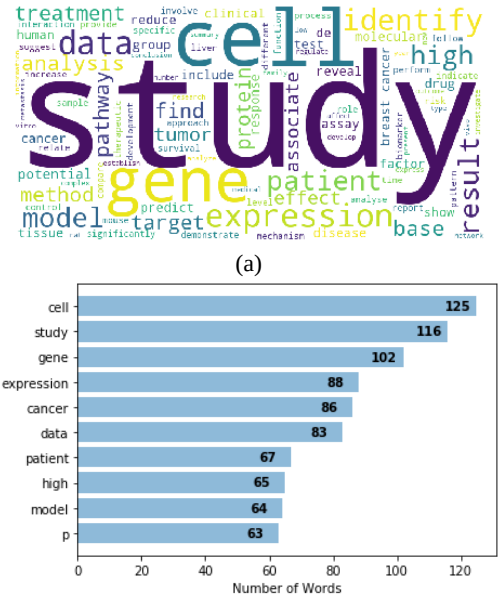
(b)

Gambar 4.6 Kriteria Artikel Jurnal dengan Kategori *Physical Sciences and Engineering* dan *Social Sciences and Humanity* (a) Kata Kunci dan (b) Frekuensi Kata Terbanyak

Kata kunci yang didapatkan adalah kata "*data*" karena kata memiliki frekuensi yang paling banyak seperti yang ditunjukkan Gambar

4.6 (b). Selain kata "*data*", terdapat kata "*model*", "*system*", "*process*", dan "*base*" memiliki frekuensi yang lebih banyak jika dibandingkan dengan kata-kata yang lain. Kata-kata tersebut menjadi kriteria dari artikel jurnal yang cenderung tergolong ke dalam kategori *Physical Sciences and Engineering* dan *Social Sciences and Humanity.*

Jika dilihat dari hasil visualisasi setiap kategori artikel jurnal yang setidaknya masuk ke dalam kategori *Physical Sciences and Engineering* menunjukkan kata kunci dengan frekuensi yang paling banyak adalah "*data*" dikarenakan artikel jurnal yang digunakan dalam penelitian ini berkaitan *data mining.*



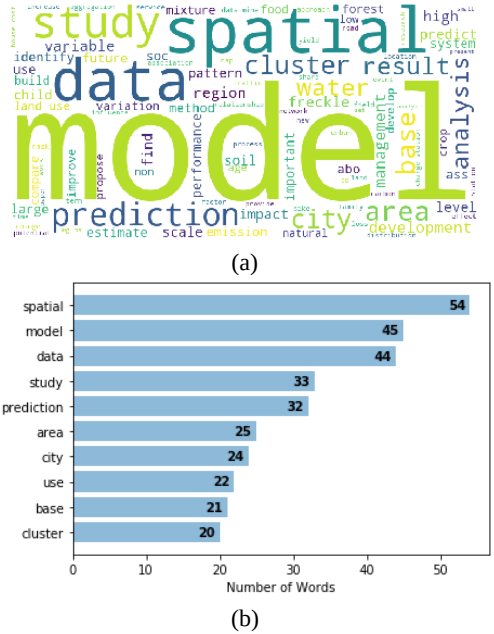
(a)

(b)

Gambar 4.7 Kriteria Artikel Jurnal dengan Kategori *Life Sciences* dan *Health Sciences* (a) Kata Kunci dan (b) Frekuensi Kata Terbanyak

Selanjutnya ditunjukkan kata kunci yang paling banyak muncul pada kategori *Life Sciences* dan *Health Sciences* pada Gambar 4.7. Kata yang memiliki frekuensi yang paling banyak pada kategori *Life Sciences* dan *Health Sciences* adalah kata "*cell*" seperti yang ter-

lihat pada Gambar 4.7(b). Kata "*cell*" menjadi kata kunci seperti pada Gambar 4.7(a). Selain kata tersebut, juga terdapat kata "*study*", "*gene*", "*expression*", dan "*cancer*" yang memiliki frekuensi kemunculan lebih banyak jika dibandingkan dengan kata yang lain. Kata-kata tersebut menjadi kriteria kata kunci yang terdapat pada artikel jurnal yang tergolong ke dalam kategori *Life Sciences* dan *Health Sciences.* Berikutnya didapatkan kata kunci yang terdapat pada kategori *Life Sciences* dan *Social Sciences and Humanity* pada Gambar 4.8.



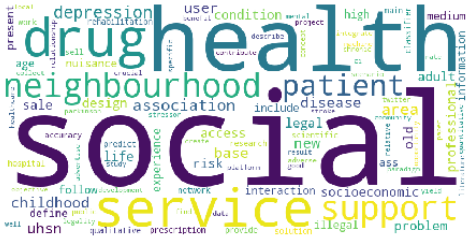
(a)

(b)

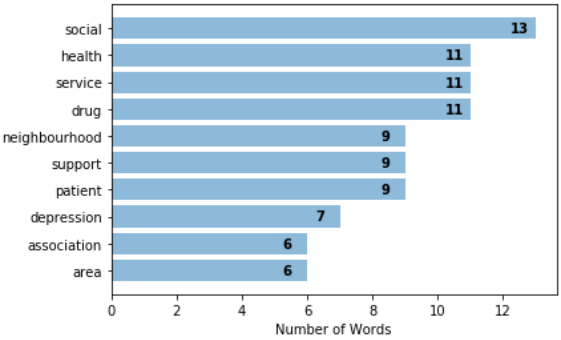
Gambar 4.8 Kriteria Artikel Jurnal dengan Kategori *Life Sciences* dan *Social Sciences and Humanity* (a) Kata Kunci dan (b) Frekuensi Kata Terbanyak

Kata kunci yang didapatkan untuk kategori *Life Sciences* dan *Social Sciences and Humanity* adalah kata "*spatial",* dapat dilihat pada Gambar 4.8(a). Hal ini menunjukkan bahwa kata "*spatial*" pada kategori *Life Sciences* dan *Social Sciences and Humanity* memiliki frekuensi yang paling banyak dan menjadi kriteria pada kategori tersebut seperti pada Gambar 4.8(b). Selain kata "*spatial*", tedapat

kata-kata "*model*", "*data*", "*study*", "*prediction* ", dan "*area*" sehingga kata-kata tersebut menjadi kriteria kata kunci dari artikel jurnal yang cenderung masuk ke dalam kategori *Life Sciences* dan *Social Sciences and Humanity*. Kategori selanjutnya menjadi kategori yang terakhir, yaitu *Health Science* dan *Social Sciences and Humanity* untuk didapatkan kata kuncinya dan ditunjukkan pada Gambar 4.9.



(a)



(b)

Gambar 4.9 Kriteria Artikel Jurnal dengan Kategori *Health Sciences* dan *Social Sciences and Humanity* (a) Kata Kunci dan (b) Frekuensi Kata Terbanyak

Kata "*social*" menjadi kata yang paling banyak muncul dalam artikel jurnal kategori *Health Science* dan *Social Sciences and Humanity* seperti yang ditunjukkan pada Gambar 4.9(b). Selain kata "*social*", terdapat kata "*health*", "*service*", "*drug*", dan "*neighbourhood*" yang memiliki frekuensi yang banyak juga pada kategori ini. Hal ini menunjukkan bahwa kata-kata tersebut merupakan kata kunci dan menjadi kriteria artikel jurnal yang tergolong ke dalam

kategori *Health Science* dan *Social Sciences and Humanity* seperti pada Gambar 4.9 (b). *Syntax* yang digunakan untuk mendapatkan kata kunci menggunakan *word cloud* dapat dilihat pada Lampiran 7.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Kesimpulan pada penelitian ini mengenai klasifikasi pada artikel jurnal ScienceDirect seperti berikut.

1. Artikel Jurnal dengan kategori *Physical Sciences and Engineering* yang berkaitan dengan "*Data Mining"* merupakan artikel jurnal yang paling banyak diterbitkan pada tahun 2019 oleh ScienceDirect, yaitu sebanyak 620 artikel jurnal, sedangkan artikel jurnal dengan kategori *Health Sciences* menjadi kategori artikel jurnal yang paling sedikit diterbitkan pada tahun tersebut. Selain itu sebagian besar artikel jurnal yang diterbitkan di tahun 2019 masuk ke dalam 2 kategori dengan jumlah 558 artikel jurnal dan artikel jurnal dengan 3 kategori memiliki jumlah yang paling sedikit, yaitu hanya sebanyak 34 artikel jurnal. Artikel jurnal yang memiliki kategori lebih dari satu tersebut jika dirinci yang paling banyak tergolong ke dalam kategori *Physical Sciences and Engineering* dan *Social Sciences* dikarenakan kedua kategori tersebut memiliki sub kategori yang lebih banyak jika dibandingkan dengan yang lain sehingga cakupan artikel jurnal penelitian yang dikategorikan ke dua kategori tersebut lebih banyak.
2. Model klasifikasi yang paling baik digunakan untuk mengategorikan artikel jurnal yang terdapat pada ScienceDirect adalah dengan menggunakan *Support Vector Machine* (SVM) yang memiliki kesalahan klasifikasi (misklasifikasi) yang lebih kecil jika dibandingkan dengan *K-Nearest Neighbor* (KNN).

5.2 Saran

Saran yang dapat diberikan bagi penerbit *online* Elsevier selaku pemilik *platform* pangkalan data artikel jurnal ScienceDirect adalah jika melakukan pemberian kategori secara *multi-label* terhadap artikel jurnal yang terdapat dalam pangkalan data ScienceDirect dapat menggunakan metode SVM yang mampu mengategorikan artikel jurnal cukup baik. Selain itu, pemberian kategori *multi-*

label juga dapat memudahkan para pengunjung pangkalan data untuk mencari sebuah artikel jurnal.

Saran yang dapat diberikan untuk penelitian selanjutnya adalah *pre-processing* data yang dilakukan dengan cermat lagi dan menggunakan metode yang lebih *advance*, seperti menggunakan *Natural Language Processing* (NLP) dengan menggunakan SpaCy pada *software* python. Hal tersebut memungkinkan untuk menghasilkan nilai kebaikan model klasifikasi yang lebih baik. Selain itu juga dapat dilakukan penelitian dengan membandingkan kebaikan klasifikasi antara menggunakan abstrak artikel jurnal dan dengan menggunakan judul dari artikel jurnal untuk menentukan atribut yang lebih baik digunakan dalam klasifikasi *multi-label.* Selain itu, jika data artikel jurnal tidak *balance* atau datanya yang digunakan *imbalance,* dapat digunakan metode untuk mengatasi data *imbalance* untuk menghasilkan nilai kinerja klasifikasi yang lebih baik. Metode yang bisa digunakan salah satunya adalah *Synthetic Minority Oversampling Technique* (SMOTE).

DAFTAR PUSTAKA

Arianto, M. S. (2010). Membangun Database E-Journal (Penguatan Local Content dan Peningkatan Akses Jurnal-jurnal Kampus). *Al-Maktabah, Vol. 10, No. 1*, 63-81.

Castella, Q., & Sutton, C. (2014). Word Storms : Multiples of Word Clouds for Visual Comparison of Document. *International World Wide Web Conference Committee (IW3C2)* (hal. 665-675). Seoul: IW3C2.

Chandani, V., Satria, R., Wahono, & Purwanto. (2015). Komparasi Algoritma Klasifikasi Machine Learning dan Feature Selection pada Analisis Sentimen Review Film. *Journal of Intelegent Systems, Vol. 1, No. 1*, 56-60.

Chang, C.-C., & Lin, C.-J. (2011). LIBSVM : A Library for Support Vector Machine. *ACM Tansactian on Intelligent Sytems and Technlogy, Vol. 2, No. 3*, 1-27.

Darujati, C., & Gumelar, A. B. (2012). Pemanfaatan Teknik Supervised untuk Klasifikasi Teks Bahasa Indonesia. *Jurnal Link, Vol. 16, No. 1*, 1-8.

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

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

Hurtado, J. L., Agarwal, A., & Zu, X. (2016). Topic Discovery and Future Trend Forecasting for Texts. *Journal of Big Data, Vol. 3, No. 7*, 1-21.

Hsu, C.W., Chang, C.C., & Lin, C.J. (2003). A Practicl Guide to Support Vector Classification, 1-16.

Isnaini, N., Adiwijaya, Mubarok, M. S., & Bakar, Y. A. (2019). A Multi-label Classsification on Topics of Indonesian News Using K-Nearest Neighbor. *The 2nd International Conference on Data and Information Science* (hal. 1-11). Bandung: IOP Publishing.

Jamaluddin. (2015). Mengenal Elektronik Jurnal dan Manfaatnya Bagi Pengembangan Karier Pustakawan. *Jupiter, Vol. XIV, No. 2*, 38-44.

Khan, A., Baharudin, B., Lee, L. H., & Khan, K. (2010). A Review of Machine Learning Algorithms for Text-Documents Classification. *Journal of Adavnces in Information Tehnology, Vol. 1, No. 1*, 4-20.

Kurniawan, B., Effendi, S., & Sitompul, O. S. (2012). Klasifikasi Konten Berita dengan Metode Text Mining. *Jurnal Dunia Teknologi Informasi, Vol. 1, No. 1*, 14-19.

Kusmayadi, E. (2008). Akses dan Pemanfaatan Pangkalan Data Jurnal Imiah. *Jurnal Perpustakaan Pertanian, Vol. 17, No. 1*, 1-9.

Liu, H., Christiansen, T., Baumgartner Jr., W. A., & Verspoor, K. (2012). BioLemmatizer : A Lemmatization Tool for Morphological Processing of Biomedical Text. *Journal of Biomedical Semantics, Vol. 3, No. 1*, 1-29.

Miswan. (2002). Jurnal ELektronik Sebagai Sarana Komunikasi Ilmiah. *Al-Maktabah, Vol. 4, N0. 1*, 1-12.

Mujilahwati, S. (2016). Pre-Processing Text Mining Pada Data Tiwtter. *Seminar Nasional Teknologi Informasi dan Komunikasi* (hal. 49-56). Yogyakarta: Universitas Atma Jaya.

Nashihuddin, W., & Rahayu, R. N. (2013). Aksesibilitas Informasi ilmiah ScienceDirect Pustaka Ristek di Lingkungan Ristek dan LPNK. *Jurnal Pustakawan Indonesia, Vol. 12, No. 12*, 1-9.

Natan, O., Gunawan, A. I., & Dewantara, N. S. (2019). Grid SVM: Aplikasi Machine Learning dalam Pengolahan Data Akuakultur. *Jurnal Rekayasa Elektrika Vol. 15 No.1*, 7-17.

Nurjannah, M., Hamdani, & Astuti, I. F. (2013). Penerapan Algoritma Term Frequency-Inverse Document Frequncy

(TF-IDF) Untuk Text Mining. *Jurnal Informatika Mulawarman, Vol. 8, No. 3*, 110-113.

Octaviani, P. A., Wilandari, Y., & Ispriyanti, D. (2014). Penerapan Metode Klasifikasi Support Vector Machine (SVM) pada Data Akreditas Sekolah Dasar (SD) di Kabupaten Magelang. *Jurnal Gaussian, Volume 3, No. 4*, 811-820.

Prihatin, P. M. (2016). Implementasi Ekstraksi Fitur pada Pengolahan Dokumen Bebahasa Indonesia. *Jurnal Matrix, Vol. 6, No. 3*, 174 - 178.

Pushpa, M., & Karpagavalli, S. (2017). Multi-label Classification: Problem Transformation methods in Tamil Phoneme classification. *7th International Conference on Advances in Computing & Communication* (hal. 572-579). Cochin: Elsevier.

Ranjan, G. S., Verma, A. K., & Radhika, S. (2019). K-Nearest Neighbor and Grid Search CV Based Real Time Fault Monitoring System for Industries. *5th International Conference for Convergence in Technology (I2CT)* (hal. 1-5). Bombay: IEEE.

Rozi, I. F., Pramono, S. H., & Dahlan, E. A. (2012). Implementasi Opinion Mining (Analisis Sentimen) untuk Ekstraksi Data Opini Publik pada Perguruan Tinggi. *Jurnal EECCIS, Vol. 6, No. 1*, 37-43.

Saputra, A. (2018). Mengukur Kontribusi Langganan E-Jurnal ScienceDirect Terhadap Produktivitas Perguruan Tinggu Menggunakan Studi Bibliometrik : Studi Kasus Universitas Andalas. *Jurnal Dokumentasi dan Informasi, Vol. 39, No. 2*, 91-99.

Soemantri, O., & Apriliani, D. (2018). Support Vector Machine Berbasis Feature Selection untuk Sentiment Analysis Kepuasan Pelanggan Terhadap Pelayanan Warung dan Restoran Kuliner Kota Tegal. *Jurnal Teknologi Informasi dan Ilmu Komputer*, 537-547.

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

Sreemathy, J., & Balamurugan, P. S. (2012). An Efficient Text Classification Using KNN and Naive Bayesian. *International Journal on Computer Science and Engineering, Vol. 4, No. 3*, 392-396.

Suprayitno, A. (2019). *Pedoman Penyusunan Jurnal Ilmiah bagi Guru.* Sleman: Deepublish.

Suryoputro, G., Riadi, S., & Sya'ban, A. (2012). *Menulis Artikel Untuk Jurnal Ilmiah.* Jakarta Selatan: Uhamka Press.

Tempola, F., Muhammad, M., & Khairan, A. (2018). Perbandingan Klasifikasi antara k-NN dan Naive Bayes pada Penentuan Status Gunung Berapi dengan K-fold Cross Validation. *Jurnal Teknologi Informasi dan Ilmu Komputer, Vol. 5, No. 5*, 577-584.

Wahyuni, E. S. (2016). Penerapan Metode Seleksi Fitur untuk Meningkatkan Hasil Diagnosis Kanker Payudara. *Jurnal Simetris, Vo. 7, No. 1*, 283-294.

Williams, G. (2011). *Data Mining With Rattle and R : The At of Excavating Data for Knowledge Discovery.* New York: Springer.

Zuhri, F. N., & Alamsyah, A. (2017). Analisis Sentimen Masyarakat Terhadap Brand Smartfren Menggunakan Naive Bayes Classifier di Forum Kaskus. *e-Proceeding of Management*, 242-251.

LAMPIRAN

Lampiran 1. Data Artikel Jurnal pada Pangkalan Data ScienceDirect

No.	Abstrak	Jurnal	Y1	Y2	Y3	Y4
1	In order to reduce the impact on resource shortage and environmental pollution ...	Procedia CIRP, Volume 80, 2019, Pages 693-698	1	1	0	0
2	The healthcare sector is paying attention to pregnancy and antenatal care (ANC) ...	Scientific African, Volume 3, May 2019, Article e00063	0	1	0	1
3	In order to deeply explore the interaction between prostate cancer (PCa)-related ...	Journal of Infection and Public Health	0	1	1	0
4	Research studies on educational data mining are on the increase due to the ...	Heliyon, Volume 5, Issue 2, February 2019, Article e01250	1	0	0	1
⋮	⋮		⋮	⋮	⋮	⋮
991	This paper presents a clinical decision support system using Artificial Neural ...	Applied Computing and Informatics, Vol/ 15, Issue 1, January 2019, 12-18	0	0	1	1
992	The non-stationary nature of electroencephalography (EEG) signals ...	Neurocomputing, Volume 343, 28 May 2019, Pages 154-166	1	0	1	0
993	Makeup face verification in the wild is an important research problem for its ...	Neurocomputing, Volume 333, 14 March 2019, Pages 339-350	1	0	0	1
994	Soft computing techniques are becoming even more popular and particularly ...	Geoscience Frontiers, Vo1.1, Issue 4, July 2020, Pages 1095-1106	1	1	0	0

Lampiran 2. Hasil Pembobotan TF-IDF

No.	aa	⋯	engine	engineer	⋯	network	⋯	zoonotic
1	0	⋯	0	0	⋯	0	⋯	0
2	0	⋯	0	0	⋯	0	⋯	0
3	0	⋯	0	0	⋯	0,2306	⋯	0
4	0	⋯	0	0,0848	⋯	0	⋯	0
5	0	⋯	0	0	⋯	0	⋯	0
6	0	⋯	0,0591	0	⋯	0	⋯	0
7	0	⋯	0	0	⋯	0,0544	⋯	0
8	0	⋯	0,0754	0	⋯	0	⋯	0
9	0	⋯	0	0	⋯	0	⋯	0
10	0	⋯	0	0	⋯	0,1094	⋯	0
⋮	⋮	⋱	⋮	⋮	⋱	⋮	⋱	⋮
491	0	⋯	0	0	⋯	0,0271	⋯	0
492	0	⋯	0	0	⋯	0	⋯	0
493	0	⋯	0,2048	0	⋯	0	⋯	0
494	0	⋯	0	0	⋯	0	⋯	0
495	0	⋯	0	0	⋯	0	⋯	0
496	0	⋯	0	0	⋯	0	⋯	0
497	0	⋯	0	0	⋯	0	⋯	0
498	0	⋯	0	0	⋯	0	⋯	0
499	0		0	0		0		0
500	0		0	0		0		0
⋮	⋮	⋱	⋮	⋮	⋱	⋮	⋱	⋮
984	0		0	0		0,0758047		0
985	0,059	⋯	0	0	⋯	0	⋯	0
986	0	⋯	0	0	⋯	0,0400	⋯	0
987	0	⋯	0	0	⋯	0,0342	⋯	0
988	0	⋯	0	0	⋯	0	⋯	0
989	0	⋯	0	0,2000	⋯	0,2359	⋯	0
990	0	⋯	0	0	⋯	0	⋯	0
991	0	⋯	0	0	⋯	0,0871	⋯	0
992	0	⋯	0	0	⋯	0	⋯	0
993	0	⋯	0	0	⋯	0	⋯	0
994	0	⋯	0	0,0774	⋯	0,0521	⋯	0

Lampiran 3. *Syntax Pre-Processing*

```
import numpy as np
import pandas as pd
import nltk
import seaborn as sns

data = pd.read_csv('Data Abstak ScienceDirect Final.csv')
data = data.drop(['Title','Keyword'],1)
print("Dimensi Data : ",np.shape(data))
data.head()

#Case Folding
abstract = data['Abstract']
abstract_lower = []
for line in abstract:
    result = line.lower()
    abstract_lower.append(result)
print(np.shape(abstract_lower))
#Delete Punctuation
import string, nltk, re
abstract_no_punct = []
for line in abstract_lower:
    result = re.sub(r"[^\w\s]"," ",line)
    abstract_no_punct.append(result)
print(np.shape(abstract_no_punct))
# Remove Number
abstract_no_number = []
for line in abstract_no_punct:
    result = re.sub("\d"," ",line)
    abstract_no_number.append(result)
print(abstract_no_number[0])
# Clear Space Enter
abstract_clear_space = []
for line in abstract_no_number:
    result = re.sub(r"\s+"," ",line)
    abstract_clear_space.append(result)
np.shape(abstract_clear_space)
```

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

```
#Removing Stopwords
from nltk.corpus import stopwords
stop_words = set(stopwords.words('english'))
abstract_no_stopwords = []
for line in abstract_clear_space:
    temp = [word for word in str(line).split() if word not in
stop_words]
    abstract_no_stopwords.append(temp)
print(np.shape(abstract_no_stopwords))
#Lemmatization
from nltk.stem import WordNetLemmatizer
lemmatizer = WordNetLemmatizer()
lemmatized_verb=[]
for line in abstract_no_stopwords:
    temp3 = []
    for word in line:
        temp3.append(lemmatizer.lemmatize(word, pos="v"))
    lemmatized_verb.append(temp3)
lemmatized_noun=[]
for line in lemmatized_verb:
    temp4 = []
    for word in line:
        temp4.append(lemmatizer.lemmatize(word, pos="n"))
    lemmatized_noun.append(temp4)
lemmatized_adv=[]
for line in lemmatized_noun:
    temp5 = []
    for word in line:
        temp5.append(lemmatizer.lemmatize(word, pos="r"))
    lemmatized_adv.append(temp5)
abstract_lemmatized=[]
for line in lemmatized_adv:
    temp6 = []
    for word in line:
        temp6.append(lemmatizer.lemmatize(word, pos="a"))
    abstract_lemmatized.append(temp6)
print(np.shape(abstract_lemmatized))
```

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

```
#Count Vectorizer
from sklearn.feature_extraction.text import CountVectorizer
df_token=[]
for line in abstract_lemmatized:
    df_token.append(" ".join(line))
count_vect = CountVectorizer(min_df = 1)
df_counts = count_vect.fit_transform(df_token)
df_counts = df_counts.toarray()
features_counts = pd.DataFrame(df_counts, columns =
count_vect.get_feature_names())
features_names = count_vect.get_feature_names()
np.shape(features_counts)

#TF-IDF
from sklearn.feature_extraction.text import TfidfTransformer
tfidf_transformer = TfidfTransformer(use_idf = True)
df_tfidf = tfidf_transformer.fit_transform(df_counts)
feature_tfidf = pd.DataFrame(df_tfidf.A, columns =
count_vect.get_feature_names())
print(np.shape(feature_tfidf))
```

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

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.model_selection import train_test_split,
StratifiedKFold, GridSearchCV, cross_val_score
from skmultilearn.problem_transform import LabelPowerset
from sklearn.metrics import accuracy_score, hamming_loss,
zero_one_loss, make_scorer
from scipy import sparse

x = feature_tfidf
y = data.drop('Abstract', 1)
LP_transformer = LabelPowerset(classifier = None, require_dense =
None)
Y_LP = pd.DataFrame(LP_transformer.transform(y))

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

#Tuning Parameter for KNN with Hamming Loss

scorer_KNN = make_scorer(hamming_loss, greater_is_better =
False)
param_KNN = [
   {
      'classifier' : [KNeighborsClassifier()],
      'classifier__n_neighbors' : list(range(1,31))
   }
]
knn_gscv = GridSearchCV(LabelPowerset(), param_KNN, scoring
= scorer_KNN, cv = 5, return_train_score = True)

knn_gscv.fit(x,y)
CV_Score_KNN = pd.DataFrame(knn_gscv.cv_results_)
CV_Score_KNN.to_csv('Tuning KNN with Hamming Loss.csv')
print ('Best Parameter :','\n',knn_gscv.best_params_)
print ('Best Score :','\n',knn_gscv.best_score_)
```

Lampiran 5. *Syntax K-Fold Cross Validation* KNN (Lanjutan)

```
# KNN with K-Fold Cross Validation

accuracy_score_KNN_train = []
accuracy_score_KNN_test = []
hamming_loss_KNN_train = []
hamming_loss_KNN_test = []
zero_one_loss_KNN_train = []
zero_one_loss_KNN_test = []

for train_index, test_index in kf.split(x, Y_LP):
    x_train, x_test = x.iloc[train_index], x.iloc[test_index]
    y_train, y_test = y.iloc[train_index], y.iloc[test_index]
    knn = LabelPowerset(KNeighborsClassifier(n_neighbors = 23,
p=2))
    knn.fit(x_train, y_train)
    y_train_pred = knn.predict(x_train)
    y_test_pred = knn.predict(x_test)

    accuracy_score_train = accuracy_score(y_train, y_train_pred)
    hamming_loss_train = hamming_loss(y_train, y_train_pred)
    zero_one_loss_train = zero_one_loss(y_train, y_train_pred)
    accuracy_score_test = accuracy_score(y_test, y_test_pred)
    hamming_loss_test = hamming_loss(y_test, y_test_pred)
    zero_one_loss_test = zero_one_loss(y_test, y_test_pred)

    accuracy_score_KNN_train.append(accuracy_score_train)
    hamming_loss_KNN_train.append(hamming_loss_train)
    zero_one_loss_KNN_train.append(zero_one_loss_train)
    accuracy_score_KNN_test.append(accuracy_score_test)
    hamming_loss_KNN_test.append(hamming_loss_test)
    zero_one_loss_KNN_test.append(zero_one_loss_test)

print("Average Accuracy Train = ",
np.mean(accuracy_score_KNN_train))
print("Average Accuracy Test = ",
np.mean(accuracy_score_KNN_test))
```

Lampiran 5. *Syntax K-Fold Cross Validation* KNN (Lanjutan)

```
print("Average Hamming Loss Train = ",
np.mean(hamming_loss_KNN_train))
print("Average Hamming Loss Test = ",
np.mean(hamming_loss_KNN_test))
print("Average Zero One Loss Train = ",
np.mean(zero_one_loss_KNN_train))
print("Average Zero One Loss Test = ",
np.mean(zero_one_loss_KNN_test))

print("Accuracy Train :\n", accuracy_score_KNN_train)
print("Accuracy Test :\n", accuracy_score_KNN_test)
print("Hamming Loss Train :\n", hamming_loss_KNN_train)
print("Hamming Loss Test :\n", hamming_loss_KNN_test)
print("One Loss Train :\n", zero_one_loss_KNN_train)
print("One Loss Test :\n", zero_one_loss_KNN_test)
```

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

```
#Tuning Parameter for SVC

scorer_SVC = make_scorer(accuracy_score, greater_is_better =
True)
param_SVC = [
    {
        'classifier' : [SVC()],
        'classifier__C' : [1, 1.1, 1.2, 1.3, 1.4, 1.5],
        'classifier__coef0' : [0.35, 0.36, 0.37, 0.38, 0.39, 0.40],
        'classifier__kernel' : ['sigmoid', 'linear'],
        'classifier__gamma' : [3.5, 3.6, 3.7, 3.8, 3.9, 4],
        'classifier__decision_function_shape' : ['ovo']
    }
]
SVC_gscv = GridSearchCV(LabelPowerset(), param_SVC, scoring
= scorer_SVC, cv = 5, return_train_score = True)

SVC_gscv.fit(x,y)
CV_Score_SVC = pd.DataFrame(SVC_gscv.cv_results_)
CV_Score_SVC.to_csv('CV Score for Tuning SVC.csv')
print ('Best Parameter :','\n',SVC_gscv.best_params_)
print ('Best Score :','\n',SVC_gscv.best_score_)

# SVC with K-Fold Cross Validation

accuracy_score_SVC_train = []
accuracy_score_SVC_test = []
hamming_loss_SVC_train = []
hamming_loss_SVC_test = []
zero_one_loss_SVC_train = []
zero_one_loss_SVC_test = []

for train_index, test_index in kf.split(x,Y_LP):
    x_train, x_test = x.iloc[train_index], x.iloc[test_index]
    y_train, y_test = y.iloc[train_index], y.iloc[test_index]
```

Lampiran 6. *Syntax K-Fold Cross Validation* SVM (Lanjutan)

```
    SVCLP = LabelPowerset(classifier = SVC(C = 1,
decision_function_shape = 'ovo', kernel = 'sigmoid', gamma = 4,
max_iter = -1))
    SVCLP.fit(x_train, y_train)
    y_train_pred = SVCLP.predict(x_train)
    y_test_pred = SVCLP.predict(x_test)

    accuracy_score_train = accuracy_score(y_train, y_train_pred)
    accuracy_score_test = accuracy_score(y_test, y_test_pred)
    hamming_loss_train = hamming_loss(y_train, y_train_pred)
    hamming_loss_test = hamming_loss(y_test, y_test_pred)
    zero_one_loss_train = zero_one_loss(y_train, y_train_pred)
    zero_one_loss_test = zero_one_loss(y_test, y_test_pred)

    accuracy_score_SVC_train.append(accuracy_score_train)
    accuracy_score_SVC_test.append(accuracy_score_test)
    hamming_loss_SVC_train.append(hamming_loss_train)
    hamming_loss_SVC_test.append(hamming_loss_test)
    zero_one_loss_SVC_train.append(zero_one_loss_train)
    zero_one_loss_SVC_test.append(zero_one_loss_test)

print("Average Accuracy Train = ",
np.mean(accuracy_score_SVC_train))
print("Average Accuracy Test = ",
np.mean(accuracy_score_SVC_test))
print("Average Hamming Loss Train = ",
np.mean(hamming_loss_SVC_train))
print("Average Hamming Loss Test = ",
np.mean(hamming_loss_SVC_test))
print("Average Zero One Loss Train = ",
np.mean(zero_one_loss_SVC_train))
print("Average Zero One Loss Test = ",
np.mean(zero_one_loss_SVC_test))
```

Lampiran 7. *Syntax Word Cloud* Kategori Jurnal Hasil Prediksi dari Data Training dengan SVM Fold Terbaik

```
from wordcloud import WordCloud
#Word Cloud for Physical Sciences and Engineering
data_wordcloud = abstract_Y1[0]
result = ("")
for line in data_wordcloud:
    result = result + " " + line
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 WordCloud image
plt.figure(figsize = (8, 4), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.show()

#Word Cloud for Life Sciences
data_wordcloud = abstract_Y2[0]
result = ("")
for line in data_wordcloud:
    result = result + " " + line
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 WordCloud image
plt.figure(figsize = (8, 4), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.show()
```

Lampiran 7. *Syntax Word Cloud* Kategori Jurnal Hasil Prediksi dari Data Training dengan SVM Fold Terbaik (Lanjutan)

```
#Word Cloud for Social Sciences and Humanity
data_wordcloud = abstract_Y4[0]
result = ("")
for line in data_wordcloud:
    result = result + " " + line
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 WordCloud image
plt.figure(figsize = (8, 4), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.show()

#Word Cloud for Physical Sciences and Engineering x Life
Sciences
data_wordcloud = abstract_Y1Y2[0]
result = ("")
for line in data_wordcloud:
    result = result + " " + line
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 WordCloud image
plt.figure(figsize = (8, 4), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.show()
```

Lampiran 7. *Syntax Word Cloud* Kategori Jurnal Hasil Prediksi dari Data Training dengan SVM Fold Terbaik (Lanjutan)

```
#Word Cloud for Physical Sciences and Engineering x Health
Sciences
data_wordcloud = abstract_Y1Y3[0]
result = ("")
for line in data_wordcloud:
    result = result + " " + line
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 WordCloud image
plt.figure(figsize = (8, 4), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.show()

#Word Cloud for Physical Sciences and Engineering x Social
Sciences and Humanity
data_wordcloud = abstract_Y1Y4[0]
result = ("")
for line in data_wordcloud:
    result = result + " " + line
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 WordCloud image
plt.figure(figsize = (8, 4), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.show()
```

Lampiran 7. *Syntax Word Cloud* Kategori Jurnal Hasil Prediksi dari Data Training dengan SVM Fold Terbaik (Lanjutan)

```
#Word Cloud for Life Sciences x Health Sciences
data_wordcloud = abstract_Y2Y3[0]
result = ("")
for line in data_wordcloud:
    result = result + " " + line
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 WordCloud image
plt.figure(figsize = (8, 4), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.show()

#Word Cloud for Life Sciences x Social Sciences and Humanity
data_wordcloud = abstract_Y2Y4[0]
result = ("")
for line in data_wordcloud:
    result = result + " " + line
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 WordCloud image
plt.figure(figsize = (8, 4), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.show()
```

Lampiran 7. *Syntax Word Cloud* Kategori Jurnal Hasil Prediksi dari Data Training dengan SVM Fold Terbaik (Lanjutan)

```
#Word Cloud for Health Sciences x Social Sciences and Humanity
data_wordcloud = abstract_Y3Y4[0]
result = ("")
for line in data_wordcloud:
    result = result + " " + line

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 WordCloud image
plt.figure(figsize = (8, 4), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.show()
```

Lampiran 8. Hasil Klasifikasi Data Abstrak Jurnal dengan SVM

No.	Abstrak	Kategori	
		Aktual	Prediksi
1	In order to reduce the impact on resource shortage and environmental pollution ...	Y1,Y2	Y1
2	The healthcare sector is paying attention to pregnancy and antenatal care (ANC) ...	Y2,Y4	Y1,Y4
3	In order to deeply explore the interaction between prostate cancer (PCa)-related ...	Y2,Y3	Y2,Y3
4	Research studies on educational data mining are on the increase due to the benefits ...	Y1,Y4	Y1,Y4
5	Construction industry generates lots of data due to the constant construction activities ...	Y1,Y4	Y1,Y4
6	Data extraction is one of the most prominent areas in data mining analysis that is been ...	Y1,Y4	Y1,Y4
7	Seoul National University has conducted a considerable number of six degree-of-...	Y1,Y4	Y1,Y4
⋮	⋮		
989	Recent advances in traffic engineering offer a series of techniques to address the ...	Y1	Y1,Y4
990	Visualization and detection of early-stage gynecological malignancies represents a ...	Y2,Y3	Y2,Y3
991	This paper presents a clinical decision support system using Artificial Neural ...	Y3,Y4	Y1,Y4
992	The non-stationary nature of electroencephalography (EEG) signals ...	Y1,Y3	Y1
993	Makeup face verification in the wild is an important research problem for its ...	Y1,Y4	Y1,Y4
994	Soft computing techniques are becoming even more popular and particularly ...	Y1,Y2	Y1

BIODATA PENULIS

Penulis dilahirkan di Kota Tarakan, 28 Juli 1998 dengan nama lengkap Rifqi Rabbanie dan sering dipanggil Rifqi. Penulis menempuh pendidikan formal di SD Negeri 001 Tanjung Redeb, kemudian dilanjutkan ke SMP Negeri 1 Berau, dan SMA Negeri 1 Berau, lalu melanjutkan pendidikan formal yakni jenjang perguruan tinggi di Departemen Statistika ITS tahun 2016. Selama masa perkuliahan, penulis aktif dalam kegiatan organisasi di jurusan maupun di institut, seperti menjadi Staf Departemen PSDM HIMASTA-ITS pada periode 2017/2018, Staf Magang Kementerian Komunikasi dan Informa-tika BEM ITS, Ketua HIMASTA-ITS pada periode 2018/2019, dan menjadi SC BCS pada periode 2019/2020. Selain itu penulis berkesempatan aktif dalam kegiatan kepanitiaan seperti menjadi Koordinator Sie *Sponsorship* Pekan Raya Statistika (PRS) 2018. Bagi pembaca yang ingin berdiskusi, memberikan saran, maupun kritik mengenai Tugas Akhir ini dapat langsung menghubungi penulis melalui *e-mail* langsung ke rabbanie.rifqi@gmail.com atau dapat dihubungi melalui nomor *handphone* 082264799470.