BAB II
TINJAUAN PUSTAKA

2.1 Penelitian Sebelumnya

Adapun beberapa penelitian yang dijadikan perbandingan sebagai bahan untuk penulisan dan tidak terlepas dari topik penelitian. Penelitian yang dijadikan bahan perbandingan mengenai topik *multi-label classification* dan penerapan model algoritma *K-NN*. Berikut merupakan penelitian mengenai topik tersebut.

Penelitian yang dilakukan oleh Erlangga P 2021, dkk pada [8]**.** *Multi-label Classification Pada Teks Hadits Dengan Mengintegrasikan Label Powerset Dan Convolutional Neural Network*. Data yang digunakan pada peneltian ini merupakan teks hadist yang sama yaitu hadist shahih bukari yang dijadikan *dataset*. Metode dan tujuan dari penelitian ini adalah menerapkan model alogritma *CNN-KIM* dan mengintegrasikan *label powerset* juga menghasilkan perbadingan nilai *hamming loss* berdasarkaan beberapa skenario dengan hasil total terbaik nilai *hamming loss* sebesar 0.173708 dan akurasi pada 88.95%.

Penelitian yang dilakukan oleh Kurnia Syuriadi I 2020, dkk pada[4]. *Klasifikasi Teks Multi Label pada Hadis dalam Terjemahan Bahasa Indonesia Berdasarkan Anjuran, Larangan dan Informasi menggunakan TF-IDF dan KNN* . Implementasi *multi-label* menggunakan model algoritma yang sama *KNN* juga *dataset* yang sama yaitu hadits shahih bukhari dengan menggunakan *feature extraction unigram* dan *problem transformation* yang digunakan adalah *binary relevance*. Dengan kombinasi pengolahan data dengan *unigram* dan *binary relevance* mendapatkan hasil akurasi *f1-score* sebesar 0.8539.

Penelitian yang dilakukan oleh Hanfi A 2020, dkk pada[10]**.** *Klasifikasi Multi Label pada Hadis Bukhari Terjemahan Bahasa Indonesia Menggunakan Mutual Information dan k-Nearest Neighbor*. Penelitian ini menggunakan *feature selection mutual information* sebagai salah satu fitur untuk membuat implementasi *multi-label* dengan model algoritma yang sama yaitu *K-NN* menjadi lebih baik pada *dataset* yang sama juga berdasarkan teks hadis shahih bukhari. Hasil akhir yang

didapatkan pada penelitian ini dengan nilai *hamming loss* terbaik di 0.0886 dengan parameter yang telah ditentukan peneliti untuk proses klasifikasi

Penelitian yang dilakukan oleh Hidayati D 2020, dkk pada [9]. *Klasifikasi Topik Multi Label pada Hadis Shahih Bukhari Menggunakan K-Nearest Neighbor dan Latent Semantic Analysis*. Data yang digunakan pada penelitian ini berupa *dataset* hadis shahih bukhari. Peneliti disini membuat model *K-NN* yang disatukan dengan metode *unsupervised* yaitu *latent semantic analysis* yang berfungsi untuk mereduksi fitur vector, sehingga mampu mengurangi kompleksitas saat dilakukan klasifikasi dengan model *K-NN.* Hasil akhir dari penggabungan model *K-NN* dan metode *latent semantic analysis* menunjukan performansi *f1-score* sebesar 90.28% dan waktu komputasi sebesar 19 menit 21 detik.

Penelitian yang dilakukan oleh Hendroprasetyo 2019, dkk pada[2]. *Klasifikasi Multi-label pada Hadis Bukhari dalam Terjemahan Bahasa Indonesia Menggunakan Mutual Information dan Backpropagation Neural Network*. Melakukan penelitian tentang membangun suatu sistem yang dapat mengklasifikasikan hadits shahih dari bukhari, dengan menggunakan metode *Backpropagation Neural Network* dan dikombinasikan dengan *mutual information* untuk melihat perolehan informasi yang berpengaruh pada setiap kelas *multi-label*. Penelitian ini memperoleh hasil dengan performansi *hamming loss* sebesar 0,0892 dan waktu komputasi sebanyak 5284,8s.

Tabel 2.1 Penelitian Sebelumnya

No	Judul	Penulis	Metode	Masalah	Hasil
1	*Multi-label* Classification Pada Teks Hadits Dengan Mengintegrasikan Label Powerset Dan Convolutional Neural Network	Erlangga Pratama P, Muhammad Zidny Naf'an, Rifki Adhitama 2021	Convolutional Neural Network	Belum diketahuinya apakah metode CNN-KIM dapat menghasilkan akurasi yang baik dalam *multi-label* classification pada teks hadis	Data yang digunakan pada peneltian ini merupakan teks hadist yang sama yaitu hadist shahih bukari yang dijadikan *dataset*. Metode dan tujuan dari penelitian ini adalah menerapkan model alogritma *CNN-KIM* dan mengintegrasikan *label powerset* juga menghasilkan perbadingan nilai *hamming loss* berdasarkaan beberapa skenario dengan hasil total terbaik nilai *hamming loss* sebesar 0.173708 dan akurasi pada 88.95%
2	Klasifikasi Teks Multi Label pada Hadis dalam Terjemahan Bahasa Indonesia Berdasarkan Anjuran, Larangan dan Informasi menggunakan TF-IDF dan KNN	Ilham Kurnia Syuriadi, Adiwijaya, Widi Astuti 2020	K-Nearest Neighbor	Mencari pengolahan data terbaik hingga data teks hadis bisa digunakan untuk klasifikasi multi label dengan metode KNN	Implementasi *multi-label* menggunakan model algoritma yang sama yaitu K-NN juga *dataset* yang sama yaitu hadits shahih bukhari dengan menggunakan *feature extraction unigram* dan *problem transformation* yang digunakan adalah *binary relevance*. Dengan kombinasi pengolahan data dengan *unigram* dan *binary relevance* mendapatkan hasil akurasi *f1-score* sebesar 0.8539
3	Klasifikasi Multi Label pada Hadis Bukhari Terjemahan Bahasa Indonesia Menggunakan Mutual Information dan k-Nearest Neighbor	Arfian Hanafi, Adiwijaya, Widi Astuti 2020	K-Nearest Neighbor	Banyaknya fitur yang digunakan pada klasifikasi membuat terjadinya penurunan akurasi pada klasifikasi multi label diteks hadis yang dilakukan dimodel KNN	Penelitian ini menggunakan *feature selection mutual information* sebagai salah satu fitur untuk membuat implementasi *multi-label* dengan model algoritma *K-NN* menjadi lebih baik pada *dataset* yang sama yaitu teks hadis bukhari. Hasil akhir yang didapatkan pada penelitian ini dengan nilai hamming loss terbaik di 0.0886 dengan parameter yang telah ditentukan peneliti untuk proses klasifikasi

No	Judul	Penulis	Metode	Masalah	Hasil
4	Klasifikasi Topik Multi Label pada Hadis Shahih Bukhari Menggunakan K-Nearest Neighbor dan Latent Semantic Analysis	Dian Chusnul Hidayati, Said Al Faraby, Adiwijaya 2020	K-Nearest Neighbor	Mencari metode yang dapat mengurangi waktu komputasi terhadap teks hadis pada metode K-NN dimulti label klasifikasi	Data yang digunakan pada penelitian ini berupa *dataset* hadis shahih bukhari. Peneliti disini membuat model K-NN yang disatukan dengan metode unsupervised yaitu latent semantic analysis yang berfungsi untuk mereduksi fitur vector, sehingga mampu mengurangi kompleksitas saat dilakukan klasifikasi dengan model K-NN. Hasil akhir dari penggabungan model K-NN dan metode latent semantic analysis menunjukan performansi f1-score sebesar 90.28% dan waktu komputasi sebesar 19 menit 21 detik
5	Klasifikasi *Multi-label* pada Hadis Bukhari dalam Terjemahan Bahasa Indonesia Menggunakan Mutual Information dan Backpropagation Neural Network	Hendro Prasetyo, Adiwijaya, Widi Astuti 2019	Backpropagation Nerual Network	Mencari performansi yang lebih baik dengan penggunaan seleksi fitur yang berbeda pada teks hadis dengan metode BNN	Melakukan penelitian tentang membangun suatu sistem yang dapat mengklasifikasikan hadits shahih dari bukhari, dengan menggunakan metode *Backpropagation Neural Network* dan dikombinasikan dengan *mutual information* untuk melihat perolehan informasi yang berpengaruh pada setiap kelas *multi-label*. Penelitian ini memperoleh hasil dengan performansi hamming loss sebesar 0,0892 dan waktu komputasi sebanyak 5284,8s.

2.2 Dasar Teori

Untuk mendukung pembuatan penelitian ini, maka perlu dikemukakan teori-teori yang berhubungan dengan permasalahan dan ruang lingkup pembahasan sebagai landasan dalam pembuatan penelitian ini.

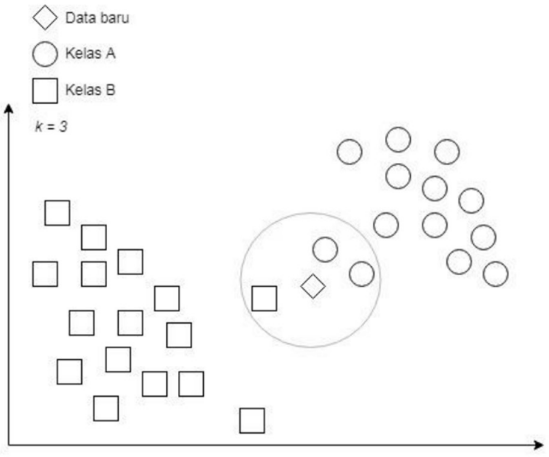
2.2.1 Multi-label Classification

Multi-label classification adalah sebuah teknologi prediktif dari data mining dengan berbagai macam layanan aplikasi yang nyata dari berbagai macam media seperti teks, music, bahkan video yang dapat dikelompokan menjadi beberapa kelompok atau label[11]. Pembelajaran dari *multi-label* dapat dilakukan dengan banyak cara yang berbeda-beda dari transformasi data, metode adaptasi, sampai penggunaan ensambel classifier.

Multi-label klasifikasi mempunyai dua kategori metode[12], yang pertama ada metode *transformation problem*, dan yang kedua ada metode *algorithm adaptation*. *Multi-label* klasifikasi menghasilkan beberapa label tidak seperti klasifikasi pada umumya yang hanya memiliki satu hasil klasifikasi label.

2.2.2 K-Nearest Neighbor

K-Nearest Neighbor merupakan algoritma yang digunakan untuk melakukan klasifikasi pada *machine learning*, algortia KNN melakukan klasifikasi dengan konsep dasar mencari "tetangga terdekat" dari suatu data[10]. Pada algorimta K-NN saat ingin memprediksi atau mengklasifikasikan data baru K-NN akan mencari titik data terdekat berdasarkan jarak sejumlah data yang memilki jarak paling dekat dengan data yang lain kemudian akan mengambil label yang paling banyak atau nilai dari target data terdekat sebagai prediksi untuk data baru.



Gambar 2. 1 Gambaran klasifikasi *k-nearest neighbor*[4]

Langkah-langkah algoritma K-NN secara umum dapat dilakukan sebagai berikut[13] :

1. Menentukan nilai atau jumlah dari *k.*
2. Menghitung jarak antara suatu data dengan seluruh data *training* .
3. Hasil perhitungan diurutkan berdasarkan nilai kemiripannya.
4. Menentukan beberapa objek berdasarkan jarak terdekat sebanyak dari nilai *k.*
5. Menentukan kelas berdasarkan hasil yang paling tinggi dari nilai k.

Untuk perhitungan antara jarak data, biasanya digunakan metode *euclidean distance* dengan rumus (2.1)

$$D(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)} \qquad (2.1)$$

Dimana :

D = nilai jarak antar data

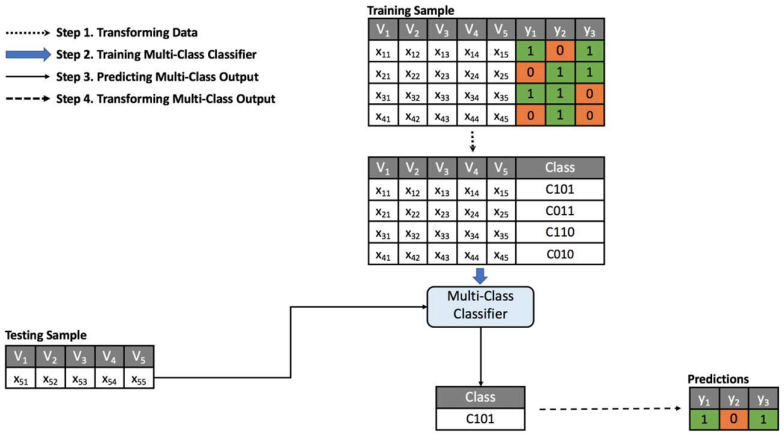
x = nilai data *training*

y = nilai data *testing*

n = dimensi data

2.2.3 *Label Powerset*

Label powerset merupakan salah satu bagian dari *problem transformation* dimana berfungsi untuk merubah data *multi-label* menjadi data *single-label* supaya data yang digunakan bisa diolah oleh model algoritma yang digunakan[13]. Cara kerja metode *label powerset* adalah dengan mengubah *dataset multi-label* menjadi *multi-class*, dimana setiap label yang berbeda akan dijadikan satu dan membuat label baru dari label yang sebelumnya terpisah.



Gambar 2. 2 Gambaran transformasi label powerset[14]

Pada *label powerset* secara tidak langsung sangat memperhitungkan kolerasi antar label dalam prosesnya. Menjadikannya akan sangat tidak efektif jika didalam *dataset* terdapat label yang sangat banyak, yang mengakibatkan kinerja dari hasil proses *multi-label* menjadi sangat buruk[14].

2.2.4 TF-IDF

Term Frequency (TF) merupakan salah satu dari ekstraksi fitur yang digunakan untuk mempre-procesing *dataset* yang telah ditentukan untuk indeksing suatu dokumen pada *dataset*. Untuk menghitung TF ada tiga cara yaitu[4] :

1. Menggunakan nilai biner pada kata - kata yang ada pada dokumen dan nilai 0 pada kata – kata yang tidak ada pada dokumen.

2. Menggunakan nilai frekuensi secara langsung pada kemunculan kata – kata untuk menjadi TF.
3. Menggunakan nilai pecahan term yang sudah dilakukan normalisasi, dengan rumus (2.2) .

$$TF\ (t,d) = f_{t,d} \tag{2.2}$$

Dimana :

(t,d) = dengan t adalah frekuensi kata yang muncul pada dokumen d.

Inverse Document Frequency (IDF) juga merupakan salah satu dari ekstarsi fitur pada dokumen yang di*pre-processing*, jika TF mencari banyaknya kata pada dokumen maka IDF mecari bobot dari setiap dokumen yang ada dengan mencari kata yang sama pada setiap dokumen. Dengan persamaannya menggunakan rumus (2.3) sebagai berikut :

$$IDF(t,d) = log\ \frac{N}{Df\,(t,d)} \tag{2.3}$$

Dimana :

N = jumlah banyaknya dokumen

$Df(t,D)$ = jumlah banyaknya dokumen dalam dokumen d yang terdapat term t.

Term Frequency-Inverse Document Frequency (TF-IDF) adalah kombinasi dari TF dan IDF untuk menentukan indeks suatu dokumen dengan mengkombinasikan kedua situasi dari TF dan IDF hanya dengan menkalikan hasil dari TF dan IDF setiap *term*. Dengan persamaan rumus (2.4) sebagai berikut :

$$TF - IDF(t,d,D) = tf(t,d)\ x\ idf(t,D) \tag{2.4}$$

2.2.5 Preprocessing

Preprocessing merupakan proses penghilangan *noise* atau data yang mungkin tidak konsisten pada *dataset*. Perolehan informasi yang berlebihan pada *dataset* dapat membuat waktu pemrosesan model menjadi memakan waktu yang cukup lama, dengan demikian representasi dari kualitas data menjadi faktor yang penting[15].

Pada tahapan *preprocessing* ini *dataset* yang diterima merupakan data mentah, yang ada didalamnya masih memiliki banyak *noise*, maka dari itu *dataset* yang diterima perlu dilakukan *preprocessing* agar dapat diproses kedalam tahap selanjutnya. Dimana tahapan - tahapan pada *preprocessing* yang dilakukan adalah kombinasi dari metode :

1. *Case Folding*

   Pada tahapan case folding ini, huruf pada teks yang ada di *dataset*, diubah menjadi huruf kapital semua atau menjadi huruf kecil semua, agar menghindari ketidaksaaman value dari kata yang didapat jika masih menggunakan huruf besar dan kecil pada suatu kata atau kalimat[16].

2. *Stopword Removal*

   Setelah menjalani tahap *case folding*, tahap *stopword removal* ini yang akan membuang kata - kata sambung yang dianggap kurang penting, bertujuan untuk mempercepat pengeksekusian program pada *dataset*[16]. Dengan contoh kata sambung seperti “yang, dan” untuk mempercepat proses *stopword removal* dapat menggunakan bantuan dari *library* SASTRAWI.

3. *Tokenization*

   Tahap *tokenization* dimana teks - teks kalimat diubah menjadi kumpulan – kumpulan kata per kata. Bertujuan untuk memudahkan model untuk melakukan pengecekan kemunculan kata pada *dataset*[16].

4. *Stemming*

Proses yang terakhir untuk menghilangkan *noise* yaitu *stemming* dimana proses untuk menghilangkan kata – kata yang berimbuhan menjadi kata dasar[16], seperti “bermain” menjadi “main” dan sebagainya, agar siap untuk diklasifikasi. Untuk mempercepat proses stemming dapat menggunakan bantuan dari *library* SASTRAWI.

2.2.6 K-fold Cross Validation

K-fold merupakan salah satu teknik *cross validation*[2] yang sering digunakan untuk memisahkan *dataset* menjadi data *training* dan data *testing* yang nantinya data tersebut akan dilakukan proses evaluasi ataupun diproses oleh modul algoritma yang dipakai. *K-fold cross validation* membagi *dataset* dengan mengacak menjadi bagian dari *k* yang berukuran sama setiap *k* nya, dengan alur proses secara bertahap akan terus mengulang proses dilakukannya *training* dan *testing* sebanyak nilai *k* yang telah diatur.

Data ke 1-266	Data ke 267-532	Data ke 533-798	Data ke 799-1064
Testing	Training	Training	Training

Data ke 1-266	Data ke 267-532	Data ke 533-798	Data ke 799-1064
Training	Testing	Training	Training

Data ke 1-266	Data ke 267-532	Data ke 533-798	Data ke 799-1064
Training	Training	Testing	Training

Data ke 1-266	Data ke 267-532	Data ke 533-798	Data ke 799-1064
Training	Training	Training	Testing

Gambar 2. 3 Gambaran proses *k-fold cross validation*[4]

Pada setiap perulangan *k* , *dataset* yang telah dibagi menjadi *training* dan *testing* akan secara bergantian dijadikan data *training* dan *testing* sampai perulangan nilai *k* terakhir[2].

2.2.7 Hamming loss

Hamming loss merupakan perhitungan performance metric yang sering digunakan pada klasifikasi label[12]. Hamming loss memprediksi errors dan miss dalam perhitungan klasifikasi. Semakin kecil nilai hamming loss maka model klasifikasi yang digunakan semakin baik[17]. Dengan persamaan (2.5) sebagi berikut :

$$HL = \frac{1}{NL} \sum_{i=1}^{N} \sum_{j=1}^{N} \left[\hat{Y}_{j(i)} \neq Y_{j(i)} \right] \qquad (2.5)$$

Dimana :

N = Jumlah data.

L = Panjang output *multi-label*.

$\hat{y}j(i)$ = adalah target untuk klasifikasi *multi-label*.

Yj(i) = adalah output dari klasifikasi *multi-label*.

2.2.8 Phyton

Python merupakan bahasa pemrograman tingkat tinggi yang sangat populer dan serbaguna. Ia dikenal dengan sintaks yang sederhana, mudah dibaca, *open source plugin* atau program alternatif untuk mempermudah pemograman[18]. Python memiliki ekosistem yang kaya dengan berbagai Pustaka atau *plugin* dan alat yang mendukung diberbagai bidang pengembangan seperti pengembangan web, ilmu data, kecerdasan buatan, pemrosesan bahasa alami.

2.2.9 Natural Language Processing

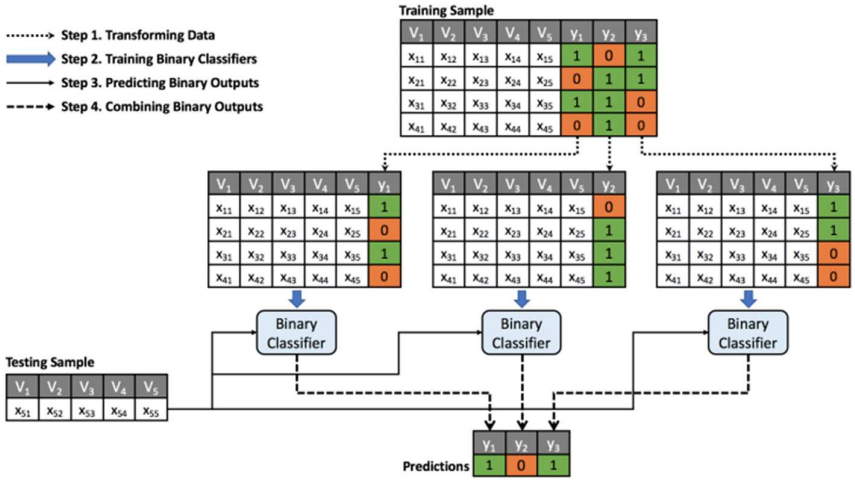
Natural Language Processing merupakan bagian dari kecerdasan buatan yang berfokus pada interaksi antara manusia dan komputer melalui bahasa manusia baik lisan maupun tulisan[19], dimana proses yang dilakukan yaitu mengizinkan komputer untuk memahami, memproses, dan menghasilkan teks atau bahasa manusia seperti halnya manusia. Proses komputasi dari *natural language processing* merubah bahasa manusia menjadi rangakaian simbol atau angka[19] yang memenuhi aturan tertentu untuk menduplikat informasi yang ada dari bahasa manusia.

Natural Language Processing mempelajari kemampuan bahasa alami manusia untuk menghasilkan tanggapan dari komputer, untuk dilakukan komunikasi atau pencarian makna teks dalam suattu konteks yang diproses.

2.2.10 Binary Relevance

Binary Relevance merupakan salah satu bagian dari *problem transformation* sama dengan *label powerset,* dimana berfungsi untuk

merubah data *multi-label* menjadi data *single-label* [13]. Cara kerja metode *binary relevance* dengan mengubah *dataset multi-label* dengan memisahkan label menjadi satu per satu, dengan menduplikatkan *dataset* menjadi satu label dengan *dataset* yang sama, dimana setiap label yang berbeda akan dijadikan satu dan membuat label baru dari label yang sebelumnya terpisah.



Gambar 2.4 Gambaran transformasi *binary relevance*[14]