BAB II

TINJAUAN PUSTAKA

II.1. *Algoritma Latent Semantic Indexing* (LSI)

Algoritma *Latent Semantic Indexing* (LSI) merupakan salah satu bentuk teknik proses temu kembali dengan menggunakan *Vector Space Model* (VSM), untuk menemukan informasi yang relevan. Fungsi matematis di dalam LSI mampu menemukan hubungan semantik antar kata.

Selain itu menurut (Dhia Alkadri & Arum Sari, 2019) *Latent Semantic Indexing* (LSI) adalah algoritma yang melakukan prediksi sebuah kelas berdasarkan pola yang dihasilkan oleh proses data training yang diciptakan oleh Vladimir Vapnik dan merupakan salah satu metode klasifikasi dengan menggunakan metode *machine learning* (*supervised learning*). Pemberian garis batas (*Hyperlane)* dalam melakukan klasifikasi agar memisahkan antara sebuah opini yang berorientasi positif dan negatif. Secara intuitif, suatu garis pembatas yang baik adalah memiliki jarak terbesar ke titik data pelatihan terdekat dari setiap kelas karena pada umunya semakin besar *margin*, semakin redah *error* generalisasi dari pemilah. *Margin* merupakan jarak dari suatu titik vektor di suatu kelas terhadap *hyperplane*.

Representasi dari LSI pada persamaan 2.1

$$q' = q^T \cdot U_k \cdot S_k^{-1} \quad (2.1)$$

Keterangan:

q' = query vector representasi dari LSI

q^T = transpose TDM dari pembobotan ternormalisasi TFIDF query U_k= reduksi dimensi k dari matriks U

S_k^{-1} = inverse dari reduksi dimensi k matriks S.

II.2. Algoritma *Naive Bayes*

Naive Bayes adalah model klasifikasi dengan menggunakan metode probabilitas dan statistik yang dikemukakan oleh seorang ilmuwan Inggris yang bernama Thomas Bayes. Olson dan Delen (2008) dalam (Costa et al., 2017) menjelaskan bahwa *Naïve Bayes* untuk setiap class decision akan menghitung probabilitas dengan syarat bahwa decision class bernilai benar, mengingatvektor informasi objek. Algoritma naiver bayes akan melakukan asumsi bahwa object atrubut sifatnya adalah independen. Kemungkinan yang terlibat dalam menghasilkan prediksi dihitung sebagai jumlah frekuensi dari tabel keputusan master.

Akan tetapi, menurut Han dan Kamber (2011) (Ngai et al., 2011), proses dari metode Naïve Bayes simpel sebagai berikut:

1. Variable D akan menjadi tupple set training dan label yang terkait dengan kelas. Biasanya, setiap vektor atribut n-dimensi akan mewakili tuple, ini

mengilustrasikan bahwa pengukuran n dibuat pada tuple daripada atribut n, dimana masing-masing adalah $A_1, A_2, A_3, ... A_n$

2. Misalkan pada kasus tersebut ada kelas m, yaitu $C_1, C_2, C_3, ... C_m$, maka diberi sebuah tuple, yaitu X. Classifier akan melakukan prediksi X yang masuk kelompok memiliki probabilitas posterior tertinggi, dimana kondisi disebutkan pada X. Artinya, classifier naive bayes akan memprediksi bahwa X tuple milik kelas Ci jika dan hanya jika, terdapat pada persamaan 2.2:

$$P(C|X) > P(C|X) > for 1 \leq j \leq m, j \leq \neq i > \quad (2.2)$$

Jadi memaksimalkan P (Ci | X). Ci kelas P (Ci | X) dimaksimalkan disebut sebagai hipotesis posterior maksimal. Dengan teorema Bayes, terdapat pada persamaan 2.3:

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)} \quad (2.3)$$

Dimana:

P(C*i*|X), merupakankemungkinan hipotesis C*i*apabila diberikan fakta X.

P(X|C*i*), adalah proses mencari nilai parameter dengan kemungkinan terbesar.

P(C*i*), adalah kemungkinan sebelumnya dari X

P(X), merupakan jumlah kemungkinan x yg muncul

1. Apabila Probability (X) adalah tetap untuk semua kelas, maka hanya Probabilitas ($X|C_i$) P (C_i) yang perlu dioptimalkan. Jika P(X|prior) tidak diketahui, maka biasanya akan dianggap masuk kedalam kelas yang sama, yaitu P (C_1) = P (C_2) = P (C_m), maka oleh karena itu akan dimaksimalkan P (X | Ci). Jika tidak, maka akan dimaksimalkan P (X | Ci) P (Ci). Selalu ingat

bahwa probabilitas class priori dapat prediksi oleh P (C_i) = |C_i,D | / |, dimana |C_i,D | / | merupakan jumlah tuple pelatihan kelas C_i di D.

2. Menimbang bahwa dataset memiliki bermacam-macam atribut, maka akan sangat sulit dalam melakukan komputasi dalam menghitung Probailitas (X|Ci). Supaya dapat mengurangi perhitungandalam mengevaluasiP(X|Ci), maka dibuatlah asumsi naïve independensikelasbersyarat. Hal ini apabila classifier menganggap bahwa nilaidari atribut merupakan kondisional independen satusama lain, maka akan diberikan label class dari tuple, dimana terdapat pada persamaan 2.4:

$$P(X \mid C_i = \prod_{k=1}^{n} P(x_k \mid C_i) = P(x_1 \mid C_i) \times P(x_2 \mid C_i) \times \ldots \times P(x_n \mid C_i)) \quad (2.4)$$

Oleh karena itu, dengan mudah dapat diperkirakan bahwa probabilitas $P(X_1|C_i)$, $P(X_2|C_i)$, ...,$P(X_n|C_i)$ dari pelatihan tuple. Selalu ingat bahwa nilai X_k mengacu pada nilai atribut A_k untuk X. Juga dilihat untuk semua atribut, apakah atribut itu bersifat kategorikal atau continuous-value, misalnya, dalam menentukan Probabilitas (X | C_i) harus dipertimbangkan beberapa hal seperti dibawah ini:

a. Apabila $A_k$bersifat kategorikal, maka Probailitas (X_k | C_i) merupakan jumlah class tupple C_i di D memiliki nilai X_k untuk atribut A_k, nilai ini kemudian dibagi dengan | C_i, D |jumlah class tupple C_i dalam D.
b. Apabila $A_k$bersifat continuousvalued, maka harus dilakukan lebih banyak pekerjaan, akan tetapi perhitungan nya cukup simpel. Suatu atribut yang

bersifat continuousvaluebiasanya diasumsikan memiliki distribusi Gaussian dengan rerata μ dan standardeviasi σ, didefinisikan dengan persamaan 2.5 :

$$P(Xk|Ci) = g(Xk, \mu C_i, \sigma C_i) \quad (2.5)$$

Selanjutnya, dilakukan perhitungan μC_i dan σC_i, yang mana itu merupakan deviasi mean dan standard masing-masing atribut k untuk tuple class training C_i. Selanjutnya, digunakan keduakuantitas dalam Persamaan, bersamaan dengan x_k, dalam memprediksikan P (x_k | C_i).

3. Dalam meramalkan class label x, P(X|C_i)P(C_i) dilakukan evaluasi untuk setiap kelas C_i. Classifier akan melakukan prediksi class label dari x tuple adalah kelas daripada C_i, terdapat pada persamaan 2.6:

$$P(Xi|P(Ci) > P(X|Cj)P(Cj) for\ i \leq j \leq m, j \neq i \quad (2.6)$$

Dengan kata lain, class label yang diprediksi merupakan $C_i$di mana P(X | C_i) P(C_i) adalah bernilai maksimal. Bayessian Classifier memiliki tingkat error minimal jika dibandingkan dengan classifier yang lainnya. Akan tetapi, dalam prakteknya hal seperti ini tidaklahsering terjadi, dikarenakan ketidakakuratan dugaan yang dibuat untuk penggunaannya, misalkan kondisi kelas yang berdiri sendiri, dan kurangnya data kemungkinan yang disediakan. Bayessian Classifier juga dapat berguna untuk memberikan pembenaran secara teoritis dalam mengklasifikasikan hal lain yang tidak menggunakan teorema Bayessian secara eksplisit.

II.3. Analisis Sentimen Masyarakat

Analisis sentimen merupakan metode pembelajaran mesin untuk mengekstrak, mengidentifikasi, atau sebaliknya mencirikan isi sentiment dari sebuah teks. *Sentiment analysis* atau opinion mining mengacu pada bidang yang luas dari pengolahan bahasa alami, komputasi linguistik dan text mining yang bertujuan menganalisis pendapat, sentimen, evaluasi, sikap, penilaian dan emosi seseorang berkenaan dengan suatu topik, produk, layanan, organisasi, individu, ataupun kegiatan tertentu. (Munawar, 2018) Dampak dari analisis sentimen adalah perkembangan pada penelitian dan analisis sentimen berbasis aplikasi sangat pesat. Bahkan di Amerika ada sekitar 20-30 perusahaan yang berfokus pada layanan analisis sentimen.

Analisis sentimen dilakukan untuk menentukan apakah opini atau komentar terhadap suatu permasalahan, memiliki kecenderungan positif atau negatif dan dapat dijadikan sebagai acuan dalam meningkatkan suatu pelayanan, ataupun meningkatkan kualitas produk. Penggunaan analisis sentimen dapat diterapkan pada opini masyarakat terhadap COVID-19. Hal ini disebabkan oleh beberapa faktor seperti penulisan kata yang disingkat, penggunaan bahasa *modern* atau slang, salah dalam mengetik huruf dan tidak baku dalam penulisan opini calon maupun para pendukung. Teknik yang berkembang untuk penggalian dokumen teks saat ini adalah *text mining*. *Text mining* dapat diolah untuk berbagai macam keperluan diantaranya adalah untuk *summarization*, pencarian dokumen teks dan sentimen analisis.

Salah satu contoh sentimen terhadap COVID-19 adalah terdapat pada akun *twitter* dengan nama akun @khairaTM, @khairaTM membuat tweet pada halaman berandanya pada tanggal 31 Desember 2020 dengan status "*Kenyataan kayak gini, while banyak influencer, selebgram, artis, selebtweet masih bilang covid isnt real dan hanya konspirasi sambil posting foto ngumpul2 liburan tanpa masker*". Sehingga memberikan dampak respon 36 *Retweets*, 3 *Quote Tweets* dan 263 *Likes.*

Gambar II.1. Contoh Sentimen Masyarakat
(Sumber: Twitter, 2021)

II.4. Bahasa Pemrograman Python

Bahasa pemrograman yang digunakan dalam penelitian ini adalah bahasa pemrogrman Python. Menurut (Harrington, n.d.), bahasa pemrogrman python merupakan bahasa pemrograman yang memiliki banyak fitur, interaktif, object oriented dan juga bahasa python adalah bahasa pemrograman tingkat tinggi (High Level Language). Bahasa pemrograman python merupakan bahasa formal yang miliki aturandan format sendiri. Python adalah bahasa pemrograman tingkat tinggi yang diciptakan oleh Guido Van Rossum di Amsterdam Belandapada tahun 1989 (Syaikhuriza & Prasetyo, n.d.). Sebagai bahasa tingkat tinggi, python memberikan berbagai kemudahan dalam penulisan suatu kode program. Kemudahan bahasa pemrograman python terdapat pada sintaks nya yang terbilang sederhana sehingga memudahkan para pengembang untuk merancangan sebuah program. Beberapa Operating System yang berbasis pada IoT menggunakan python sebagai bahasa pemrograman.

II.5. *Twitter*

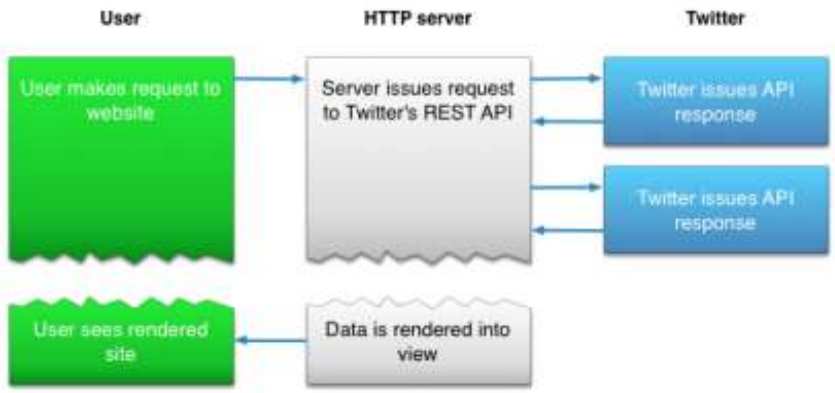
Twitter merupakan salah satu sosial media yang sudah populer selama beberapa tahun belakangan ini. *Twitter* mampu masuk sebagai salah satu media sosial yang paling banyak digunakan masyarakat dan bersaing dengan situs besar seperti Facebook. *Twitter* dimulai dengan satu pertanyaan: *"What's happening?"* dan dijawab hanya dalam 140 karakter saja.

Tweet entity atau disebut juga sebagai konten *tweet* adalah konten/ isi dari *tweet* itu sendiri. *Tweet entity* tersebut adalah: teks dari *tweet* itu sendiri, *hashtag* (#),

mention (@), *retweet* (RT), url, *emoticons*, media. *C* menyediakan *Application Programming Interface* (API) untuk mengakses datanya. Termasuk *update* status, operasi pencarian dan akses pengguna *timeline*

II.6. *Twitter API*

Data *twitter* merupakan sumber data yang kaya serta beragam dan dapat di gunakan untuk mengungkap informasi tentang topik yang kita inginkan. Data ini dapat digunakan dalam penggunaan yang berbeda-beda seperti menemukan kasus yang sedang popular, prediksi, kategorisasi berdasarkan tag atau kata kunci tertentu, mengukur sentimen merek maupun mengumpulkan umpan balik tentang produk layanan baru. Hal tersebut memicu para programmer atau developer untuk mengembangkan kreatifitas dalam membangun suatu sistem berdasarkan data melimpah yang dimiliki oleh *twitter* tersebut. Oleh karena itu, *twitter* menyediakan API programming yang dapat dikembangkan oleh pihak ketiga untuk membangun suatu aplikasi baru.



Gambar II.2. Arsitektur *Twitter* API

(Sumber: Hutagalung, 2018)

Twitter API tersedia untuk berbagai platform dan bahasa pemrograman. Untuk menggunakannya dibutuhkan pemasangan paket-paket tertentu serta melengkapi library dari *twitter* API itu sendiri. (Hutagalung, 2018) Pengembang yang ingin mengimplementasikan *twitter* API melalui bahasa pemrograman *Hypertext Prepocessor* (PHP) dapat menggunakan *twitter OAuth*, yaitu PHP *library* untuk berkomunikasi dengan *Twitter* API. Hasil yang didapat akan diekstrak dalam bentuk format data *JSON* yang bisa diolah sebelum disimpan ke dalam database. Ada 3 bagian penting dari platform *twitter,* yakni:

1. *OAuth Authentication*

   OAuth dapat memungkinkan pengguna untuk mengakses situs *twitter* yang dimiliki pengguna tanpa harus melalui dari situs *twitter*. Otentikasi dibutuhkan untuk sign in ke akun *twitter* yang dimiliki pengguna. Sign in adalah tahapan yang harus dilalui pengguna untuk dapat masuk ke akun *twitter*-nya. Otentikasi application-user diperlukan untuk mendapatkan user-specific API. Dengan kata lain, ketika kita mulai untuk mengakses API *twitter* menggunakan akun pengguna, pengguna akan diarahkan menuju *twitter* untuk mengotorisasi aplikasi kita. *Twitter* akan memberikan akses token yang akan berakhir sampai pengguna tidak lagi menggunakannya. Pengembang akan menggunakan token-token tersebut untuk otentikasi aplikasi yang akan dibuat atas nama akun pengembang sebagai pengguna.

2. *Representational state transfer* (*REST*) API

   Cara paling umum yang digunakan oleh pengembang untuk mendapatkan akses data *twitter* adalah melalui *REST* API. *REST* API menggunakan token yang akan diperoleh melalui OAuth sehingga sistem yang kita buat akan melakukan permintaan (*request*) kepada *twitter* untuk menarik data tertentu. Misalnya, kita akan mengakses data status pengguna. *REST* API dapat memenuhi kebutuhan para pengembang aplikasi *twitter*.

3. *Streaming* API *OAuth Authentication*

   Streaming API memungkinkan pengguna untuk menerima postingan-postingan dan pemberitahuan secara *real-time* dari *twitter*. Namun, membutuhkan kinerja yang tinggi dan koneksi yang harus selalu ada antara server dengan *twitter*. Ada tiga variasi pada *twitter streaming* API, yakni :

   a. *The Public Stream*, hal ini memungkinkan sistem untuk memonitor data publik di *twitter*, seperti *tweet* yang dibagikan secara publik, filter hashtag dan sebagainya.

   b. *The User Stream*, hal ini memungkinkan untuk melacak aliran tweet pengguna secara *real-time*.

Site Stream, site stream membutuhkan persetujuan terlebih dahulu dari pihak *twitter* yang memungkinkan sistem untuk memonitor *real-time twitter feeds* untuk sejumlah besar pengguna. Tujuan dari implementasi *streaming* adalah untuk melacak peristiwa yang masuk secepat mungkin dan mengolahnya pada suatu aplikasi tertentu menggunakan *REST* API untuk mendapatkan data yang lebih dalam. Penggunaan *REST* API memiliki berbagai batasan-batasan yang diberikan

oleh twitter. Penting bagi pengembang untuk menggunakan dan bertanggung jawab atas penggunaan *twitter* API dengan merencanakan batasan-batasan aktivitas dalam sistem yang dibuat dan memantau respon-respon yang ada.

II.7. Penelitian Terkait

(Tuhuteru & Kristen Indonesia Maluku Jl Ot Pattimaipauw, n.d.) melakukan penelitian dengan mengangkat judul Analisis Sentimen Masyarakat Terhadap Pembatasan Sosial Berksala Besar Menggunakan Algoritma Support Vector Machine, hasil dari penelitian tersebut adalah Dengan memanfaatkan media sosial Twitter, dengan data 1075 tweet dan komentar, data training sebanyak 350 dan data testing sebanyak 725, dan memperoleh hasil sentimen positif sebesar 28%, sentimen negatif sebesar 27%, dan sentimen netral sebesar 45%.

Selanjutnya (Hikmawan et al., 2020) melakukan penelitian dengan judul Sentimen Analisis Publik Terhadap Joko Widodo Terhadap Wabah COVID-19 Menggunakan Metode Machine Learning. Penelitian ini menjadikan kata kunci "Jokowi" dan "Covid" untuk mencari sesering apa kata kunci ini dipakai dengan membandingkan tiga metode machine learning, yaitu metode Naive Bayes, Support Vector Machine dan KNN. Penelitian ini menghasilkan accuracy 84.58%, precision 82.14% dan recall 85.82%.

Penelitian (Dhia Alkadri & Arum Sari, 2019) mengangkat judul Analisis Sentimen Ulasan Video Animasi Menggunakan Metode Latent Semantic Indexing. Pada penelitian ini, pengujian dilakukan sebanyak 19 kali dengan menggunakan

masukkan k-rank yang berbeda-beda. Berdasarkan hasil pengujian, sistem ini menghasilkan akurasi optimal di k-rank = 10 yaitu sebesar 86%.

Penelitian yang dilakukan oleh (Rachmi et al., 2017) dengan judul Penerapan Principal Component Analysis Dan Genetic Algorithm Pada Analisis Sentimen Review Pengiriman Barang Menggunakan Algoritma Support Vector Machine. Pada penelitian ini, keakuratan yang dihasilkan dari algoritma Support Vector Machine sebesar 86.00%, setelah dioptimalkan dengan menggunakan Principal Component Analysis dan Genetic Algorithm accuracy telah meningkat menjadi 97%.

Penelitian oleh (Abidin, n.d.) dengan judul Pembangunan Kamus Bahasa Indonesia Kata Tidak Baku Menggunakan Algoritma Latent Semantic Indexing Dan Damerau levenshtein Distance. Hasil dari penelitian ini dengan penggabungan algoritma Algoritma Latent Semantic Indexing dan Damerau levenshtein Distance terbukti menghasilkan sebuah modul preprocessing teks yang efektif dalam membangun kamus Bahasa Indonesia kata tidak baku.

Penelitian oleh (Marcus & Maletic, n.d.) dengan judul *Recovering Documentation to Source-Code Traceability Links using Latent Semantic Indexing.* Pencarian informasi dengan *Latent Semantic Indexing* digunakan untuk mengidentifikasi secara otomatis tautan dari dokumentasi sistem. Hasil ini dibandingkan dengan hasil eksperimen sejenis lainnya dari identifikasi tautan keterlacakan menggunakan berbagai jenis teknik pencarian informasi. Metode yang disajikan terbukti memberikan hasil yang baik dengan perbandingan dan selain itu, ini adalah metode yang berbiaya rendah dan sangat fleksibel untuk berlaku

sehubungan dengan pra-pemrosesan dan/atau penguraian kode sumber dan dokumentasi.

Penelitian oleh (Rish, n.d.) dengan judul *An empirical study of the naive Bayes classifier*. Klasifikasi dengan Naive Bayes sangat memudahkan pembelajaran dengan mengasumsikan bahwa fitur diberikan secara independen kelas. Pendekatan peneliti dengan menggunakan simulasi Monte Carlo yang memungkinkan studi sistematis tentang akurasi klasifikasi untuk beberapa kelas masalah yang dihasilkan secara acak. Hasil adalah akurasi Naive Bayes tidak berkorelasi langsung dengan derajat dependensi fitur diukur sebagai informasi mutual classconditional antara fitur.

Tabel II. 1. Penelitian Terkait

No	Judul	Penulis	Metode	Hasil
1	Analisis Sentimen Masyarakat Terhadap Pembatasan Sosial Berksala Besar Menggunakan Algoritma Support Vector Machine	Hennie Tuhuteru	Metode Algoritma Svm (Support Vector Machine)	Hasil sentimen positif sebesar 28%, sentimen negatif sebesar 27%, dan sentimen netral sebesar 45%.
2	Sentimen Analisis Publik Terhadap Joko Widodo Terhadap Wabah COVID-19 Menggunakan Metode Machine Learning	Sisferi Hikmawan, dkk	Dengan menggunakan Metode Machine Learning	Hasil accuracy 84.58%, precision 82.14% dan recall 85.82%.
3	Analisis Sentimen Ulasan Video Animasi Menggunakan	Sutrisno, dkk	Metode Latent Semantic Indexing	Menghasilkan akurasi optimal di k-rank = 10 yaitu sebesar 86%.

	Metode Latent Semantic Indexing			
4	Penerapan Principal Component Analysis Dan Genetic Algorithm Pada Analisis Sentimen Review Pengiriman Barang Menggunakan Algoritma Support Vector Machine	Hilda Rachmi	Principal Component Analysis Dan Genetic Algorithm	Keakuratan yang dihasilkan dari algoritma SVM sebesar 86.00%, setelah dioptimalkan dengan menggunakan PCA dan Genetic Algorithm accuracy telah meningkat menjadi 97%.
5	Pembangunan Kamus Bahasa Indonesia Kata Tidak Baku Menggunakan Algoritma Latent Semantic Indexing Dan Damerau levenshtein Distance	Jenal Abidin	Algoritma Latent Semantic Indexing Dan Damerau levenshtein Distance	Menghasilkan sebuah modul preprocessing teks yang efektif dalam membangun kamus Bahasa Indonesia kata tidak baku.
6	Recovering Documentation to Source-Code Traceability Links using Latent Semantic Indexing	Andrian Marcus, Jonathan I. Maletic	Latent Semantic Indexing	Metode yang disajikan terbukti memberikan hasil yang baik dengan perbandingan dan selain itu, ini adalah metode yang berbiaya rendah dan sangat fleksibel untuk berlaku sehubungan dengan pra-pemrosesan dan/atau penguraian kode sumber dan dokumentasi.
7	An empirical study of the naive Bayes classifier	I Rish	Naive Bayes Classifier	Hasil adalah akurasi Naive Bayes tidak berkorelasi langsung dengan derajat dependensi fitur diukur sebagai informasi mutual classconditional antara fitur.