BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1. Tinjauan Pustaka

Pada penelitian ini terdapat beberapa kontribusi kajian ilmu dari penelitian terdahulu, terkait upaya menyelesaikan permasalah klasifikasi karakteristik *dataset*, dan juga metode yang dapat digunakan peneliti untuk membangun sistem pendeteksi model yang optimal. Dengan mengidentifikasi permasalahan, berbagai sudut pandang para peneliti terdahulu memberikan kontribusinya nyata dalam perkembangan dan memajukan sistem informasi dan teknologi, khususnya dalam upaya membangun sistem pendeteksi informasi *hoax* yang paling optimal dalam mengidentifikasi informasi *hoax*.

Penelitian yang dilakukan Aldwairi dkk. (2018) yang berjudul "*Detecting Fake News in Social Media Networks*" menjelaskan bahwa berita dan tipuan palsu sudah ada sebelum munculnya internet. Eksposisi menganalisis prevalensi berita palsu (*hoax* atau *fake news*) mengingat kemajuan dalam komunikasi dimungkinkan oleh munculnya situs jejaring sosial. Tujuan penelitian ini untuk menghasilkan solusi yang dapat digunakan oleh pengguna untuk mendeteksi dan menyaring situs yang mengandung *false* dan menyesatkan. Fitur sederhana dan *post* yang dipilih dengan baik untuk mengidentifikasi *posting* palsu secara akurat. Hasil penilitian ini menunjukkan akurasi 99,4% klasifikasi logistik (keterkaitan hubungan antara beberapa variabel, dimana variabel tersebut bersifat kategorik) (Mother & Alwahedi, 2018).

Penelitian yang berkaitan dengan proses suatu kompleksitas sistem dilakukan Saquete dkk. (2020) dalam penelitiannya yang berjudul "*Fighting post-trurh using natural language processing: A review and challenges*". *Post-truth* merupakan istilah yang menggambarkan fenomena menyimpang yang bertujuan untuk memanipulasi opini publik dan tingkah laku. Saat ini sebagaian besar berita tersebar dengan cepat, dalam Bahasa yang tertulis di media digital dan jejaring sosial. Penerapkan *Artificial Intelligence* (AI) dan *Natural Language Processing* (NLP) berguna untuk mendeteksi permasalahan *post-truth*. Penarapan AI sendiri, untuk menyelesaikan permasalahan yang kompleks secara otomatis, dengan

mempertimbangan kompleksitas sistem. Penerapan metode *divide and conquer* untuk mengidentifikasi serangkaian subtugas untuk mengatasi masalah dari perspektif komputasi. Hasil dari penelitiannya, dapat mengidentifikasi sub tugas seperti mendeteksi opini publik tentang penipuan, kontroversi dan polarisari (opini kelompok orang yang berkepentingan), pemeriksaan fakta secara otomatis, dan *clickbait* (konten di dalam tautan tersebut biasanya ditulis sesuai fakta namun judul yang dibuat berlebihan) (Saquete dkk., 2020).

Penggunaan dua algoritma dalam proses mengintegrasikan *dataset*, dilakukan Ahmed dkk. (2014) dalam penelitiannya yang berjudul "*SMS Classification Based on Naïve Bayes Classifier and Apriori Algoritm Frequent Itemset*". Penyaringan spam yang dilakukan peneliti, dengan menerapkan algoritma Apriori dan Klasifikasi Naïve Bayes. Dimana ia membangun sistem filter spam dengan menggunakan *dataset* yang ada pada pesan singkat SMS. Hasilnya dengan menggabungkan algoritma Apriori dan algoritma Naïve Bayes terdapat peningkatan dari pada menggunakan klasik Naïve Bayes *Classifier* pada data SMS bentuk teks, dan dapat menangkap (mengolah) penggunaan bahasa dalam bentuk tertulis atau lisan (Corpus) v.0.1 *Big* (Ahmed dkk., 2014).

Penelitian yang dilakukan Ahmed dkk. (2018), membangun sistem pendeteksi opini spam, dengan judul "*Detection Opinion Spams and Fake News using Text Classification*". Masalah spam dapat dirumuskan untuk pertama kalinya dengan banyaknya konten ulasan palsu atau bahkan menuliskan berita palsu di web yang telah dibuat pengguna. Tantangan terbesar adalah kurangnya cara yang efisien untuk membedakan antara ulasan nyata dan palsu. Dengan memperkenalkan model *n*-gram baru dan menggunakan teknik klasifikasi untuk mendeteksi secara otomatis dengan fokus khususnya pada ulasan palsu dan berita palsu. Hasil dari penelitian ini adalah penerapan metode empiris dan gap teoritik dengan akurasi model memiliki nilai yang lebih tinggi yaitu 50.000-100.000 dari *dataset* yang digunakan peneliti (Ahmed dkk., 2018).

Penerapan algoritma *Random Forest* dalam membangun sistem pendeteksi spam pada pesan SMS pun dilakukan oleh Sjarif dkk. (2019), dengan menerapkan algoritma *Random Forest* dan metode *Term Frequency-Inverse Document Frequency*. Pada penelitiannya menyajikan sistem yang memiliki kinerja yang

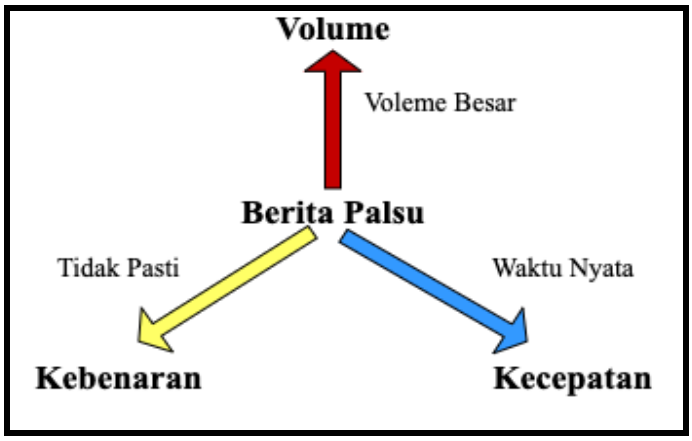
lebih baik dibandingkan algoritma lainnya dalam hal presentasi akurasi, yang mana algoritma *Random Forest* mampu memberikan presisi dan *f-meansure* yang baik dengan tingkat presisi 98% (Sjarif dkk., 2019).

2.2. Dasar Teori

2.2.1. Berita Palsu (*Hoax*)

Berita palsu (*hoax*) atau tipuan memiliki banyak definisi seperti masalah berulang yang digunakan sebagai senjata politik, senjata publik berdasarkan kebenaran yang tidak relevan (pasca-kebenaran) atau sengaja menyebarkan informasi kepalsuan (alt-fakta). Fakta-fakta alternatif (alt-fakta) adalah informasi tanpa dasar dalam realitas sedangkan *post*-kebenaran didefinisikan sebagai di luar kebenaran atau informasi yang tidak relevan. *Hoax* merupakan akses negatif yang bisa berupa video, pesan teks, audio maupun artikel. *Hoax* sendiri bisa membuat banyak pendapat publik (opini), menggiring pendapat tanpa asalan yang jelas, membentuk persepsi yang membuat tidak jelas, juga untuk bersenang-senang para oknum tidak bertanggung jawab untuk membuat kegaduhan pada pesan instan ataupun sosial media (Juditha, 2018).

Semakin berkembangnya teknologi, internet menjadi wadah besar untuk berkembang biaknya pertumbuhan berita palsu (*hoax*), seperti halnya ulasan palsu, iklan palsu, desas desus pernyataan politik palsu dan sebagainya. Selain itu, informasi *hoax* tersebut mengancam potensial komunitas khususnya pada kegiatan mengirim pesan dan memiliki dampak negatif paling berpengaruh dalam kehidupan, yang mana aktifitas tersebut sering kali menanyakan kebenaran dari berita palsu (*hoax*) yang tersebar dengan sangat cepat membuat berita palsu tersebut masuk dan mempengaruhi ke dalam sendi-sendi kehidupan manusia dalam mendapatkan berita, memuat lingkaran perkembangan berita palsu (*hoax*) pada gambar 2.1. (Zhang & Ghorbani, 2020).



Gambar 2.1 Volume, Kebenaran, dan Kecepatan Berita Palsu.

- Volume berita palsu: Tanpa prosedur verifikasi, semua orang dapat dengan mudah menulis berita palsu di internet. Oleh karena itu, sejumlah besar konten palsu didistribusikan melalui internet, bahkan tanpa kesadaran penggunanya (Zhang & Ghorbani, 2020).
- Kebenaran berita palsu: Ada beberapa definisi tentang berita palsu, seperti rumor, berita sindirian, ulasan palsu, informasi yang keliru, iklan palsu, teori konspirasi, pernyataan palsu atau lain sebagainya, yang mempengaruhi setiap aspek kehidupan manusia. Dengan seringnya manusia menggunakan internet, berita palsu dapat mendominasi opini, minat, dan keputusan publik. Selain itu, berita palsu mengubah cara manusia berinteraksi dengan berita nyata (Zhang & Ghorbani, 2020).
- Kecepatan berita palsu: Pembuatan berita palsu cenderung berumur pendek. Misalnya, banyak yang palsu aktif halaman web berita selama pemilihan presiden Amerika Serikat 2016 tidak ada lagi kampanye. Karena lebih banyak perhatian diberikan pada berita palsu baru-baru ini. Selain itu, sebagian berita palsu yang tersebar di internet berfokus pada peristiwa terkini dan hubungan panas untuk lebih menarik perhatian pengguna *online*, karena yang sifatnya *real-time* membuat mengidentifikasi berita palsu menjadi lebih sulit (Zhang & Ghorbani, 2020).

Ciri-ciri berita palsu (*hoax*), yang sering dijumpai adalah (Juditha, 2018):

1. Biasanya didistribusikan melalui media sosial, email ataupun pesan instan, karena efeknya lebih besar.
2. Berisi pesan yang membuat cemas dan panik para pembaca.

3. Bermuatan informasi yang provokatif, yang mana menyembunyikan kebenaran informasi yang ada.
4. Tidak adanya sitasi atau sumber yang jelas dari berita tersebut.
5. Memiliki tingkat kompleksitas yang sangat tinggi, dikarenakan berita palsu (*hoax*) dapat tumbuh sangat cepat jika didistribuskan melalui media elektronik, karena memungkinkan yang sifatnya *real-time* membuat mengidentifikasi berita palsu menjadi lebih sulit.

Jenis-jenis informasi *hoax* yaitu di antaranya (Juditha, 2018):

1. *Fake news* (berita palsu): Berita yang sengaja dibuat tanpa sumber informasi yang jelas dengan menyembunyikan berita yang asli. Berita ini bertujuan untuk memalsukan dalam suatu berita. Oknum berita palsu biasanya menambahkan hal-hal yang tidak benar dan teori yang tidak relevan. Berita palsu bukanlah komentar atau persepsi terhadap suatu berita.
2. *Clickbait* (tautan jebakan): Tautan yang diletakkan secara stategis di dalam suatu penulisan dengan tujuan untuk menarik pembacanya. Konten di dalam tautan tersebut biasanya ditulis sesuai fakta namun judul yang dibuat berlebihan atau dipasang gambar yang menarik untuk memancing pembacanya.
3. *Confirmation bias*: Kecenderungan untuk menafsirkan kejadian yang baru saja terjadi sebagai bukti dari kejadian yang ada tanpa mengetahui kebenarannya.
4. *Mis information*: Informasi yang buat kurang atau tidak akurat, terutama yang ditujukan untuk menipu.
5. *Satire*: Sebuah tulisan yang menggunakan bahan lelucon, ironi, hal yang dibesar-besarkan untuk mengkomentari kejadian yang baru saja terjadi.
6. *Post-truth*: Istilah yang menggambarkan fenomena menyimpang yang bertujuan untuk memanipulasi opini publik dan tingkah laku.

2.2.2. Media Sosial

Media sosial merupakan sekumpulan atau sekelompok aplikasi berbasis internet sebagai teknologi informasi dari perkembangan dan kemajuan internet, yang mengizinkan para penggunanya dalam satu cakupan jaringan internet untuk

mengirim *posting* (pesan) atau sebagai wadah pertukaran *user-generated*, baik secara langsung pada saat yang bersamaan (*real time*) menggunakan teks, dokumen, gambar, dan video kepada sesama pengguna lainnya yang sedang terhubung ke jaringan yang sama. Perkembangan dari media sosial membawa manusia untuk biasa saling membagikan ide, gagasan dalam menciptakan kreasi, berfikir, berpendapat ataupun dapat menjadi wadah mencari teman baru. Kecepatan yang melengkapi aplikasi media sosial bisa diakses dalam hitungan detik, ini menjadi alasan utama media sosial sangat berkembang pesat, selain itu para pengguna dari setiap jejaring aplikasi bertumbuh dengan pesat dikarenakan sekarang manusia hidup tidak jauh dari penggunaan internet (Kaplan & Haenlein, 2010).

Ada beragam jenis media sosial yang perlu dibedakan lebih lanjut. Namun, sebagian besar orang mungkin akan setuju bahwa *Wikipedia*, aplikasi *Youtube*, *Facebook, Twitter*, *Instagram* dan banyak lainnya, yang mana memiliki perbedaan satu sama lain. Adapun kemajuan dari perkembangan media sosial tersebut dari perbedaan kegunaan aplikasi media sosial membawa banyak dampak (pengaruh besar) bagi kehidupan manusia, baik berdampak positif maupun berdampak negatif. Dampak positif media sosial memberikan manfaat di antaranya seseorang dapat memiliki akses cepat dalam mendapatkan informasi baru, wadah mengelolah jaringan pertemanan ataupun dapat menjadi wadah untuk saling belajar, berbisnis dan berdiskusi antar sesama pengguna sosial media. Selain itu, dampak negatif yang disajikan media sosial adalah dapat menjadi wadah penyebaran informasi palsu (*hoax*), wadah penipuan, wadah menebarkan kebencian dan membuat manusia menjadi malas bersosialisasi secara langsung (Khairuni, 2016).

2.2.3. *Twitter*

Twitter merupakan jejaring sosial media (*microblog*) yang mana memungkinkan penggunanya untuk membaca dan mengirim pesan berbasis teks, hingga 140 karakter teks, serta sering juga disebut kicauan (*tweet*). *Twitter* didirikan pada tahun 2006 tepatnya bulan Maret oleh Jack Dorsey, dan resmi diluncurkan pada bulan juli 2006. Fitur yang disuguhkan *Twitter* menjadi alat yang menarik diberbagai kalangan. Setiap pemilik akun dapat membuat kicauan

(*tweet*) sesuai keinginan dan dapat dilihat oleh banyak orang, yang mana menjadi sumber data potensial untuk digunakan oleh jutaan orang. Data *Twitter* dapat disebut juga sebagai data *real time* dan data dengan volume besar (*big data*) karena dapat diakses dalam satu jangkauan jaringan internet satu dunia (Nur, 2015).

Indonesia saat ini menempati peringkat 5 pengguna *Twitter* di dunia, menurut laporan Kementerian Komunikasi dan Informatika Republik Indonesia pada tahun 2019 (KOMINFO), dengan 19.5 juta pengguna di Indonesia dari total 500 juta pengguna global. Tingginya popularitas dari penggunaan *Twitter* menyebabkan aplikasi *Twitter* menjadi wadah pemanfaatan dalam berbagai keperluan di segala bidang aspek kehidupan, misalnya untuk promosi produk bisnis, sarana penyebaran informasi, sarana protes (beragumen atau beropini), ataupun sebagai media sarana pembelajaran. Pemanfaatan tersebut tergantung dengan kesesuaian dan pemahaman penggunaannya, apakah media sosial *Twitter* dapat membawa dampak baik atau negatif (Nur, 2015).

2.2.4. *Data Mining*

Data mining adalah suatu istilah yang telah digunakan untuk menentukan, menemukan pengetahuan yang tersembunyi di dalam suatu *database* atau basis data. *Data mining* merupakan proses semi otomatik yang dapat menggunakan tenik matematik, kecerdasan buatan atau *Artificial Intellegent* (AI), statistik dan *Machine Learning* untuk menjalankan, mengindentifikasi dan mengekstraksi informasi-informasi pengetahuan yang berkaitan dengan *database* yang memiliki volume yang besar. *Data mining* mewariskan banyak aspek dan teknik diberbagai ilmu pengetahuan yang sudah mapan pengetahuan (Putria, 2018).

Data mining, dapat disebut juga sebagai *Knowledge Discovery in Database* (KDD), yang merupakan kegiatan yang meliputi proses pengumpulan, pemakaian data historis untuk menentukan pola atau hubungan dalam *dataset* yang bervolume besar (*big data*). *Output* dari proses *data mining* dapat digunakan untuk memperbaiki pengambilan keputusan dimasa yang akan datang, banyak contoh bidang penelitian, *project* yang telah menerapkan *data mining* sebagai metode penyelesaian masalahnya. Dengan adanya *data mining* maka akan mempermudah dalam mendapatkan pengetahuan informasi dari kumpulan data-

data yang ada. Salah satu pengembangan dari metode *data mining* dapat menyelesaikan permasalahan *big data* pada pemproses *big data* dan menjadi ilmu penting bagi pekerja yang berkecimpung di dunia *data analytic* (Han dkk., 2012).

Sebagai salah sarana teknologi, *data mining* dapat diterapkan ke semua jenis data selama data bermakna untuk membangun sebuah aplikasi. Bentuk data yang paling dasar untuk pendukung metode dasar *data mining* adalah *database*, data *warehouse*, data transaksional. *Data mining* juga dapat diterapkan ke bentuk data lainnya misalnya aliran data (*data streams*), grafik atau data jaringan, data spasial, data teks, data multimedia dan lainnya. Garis besar metode *data mining* dapat mempermudah proses pekerjaan sistem dalam mengolah berbagai bentuk teks (Han dkk., 2012).

2.2.5. Algoritma Apriori

Algoritma Apriori adalah salah satu metode yang paling banyak digunakan untuk mencari basis *big data*. Algoritma Apriori menggunakan teori sederhana yang berdasarkan kenyataan dari semua subjek dari poin pokok. Komponen nilai penunjang atau dapat diartikan sebagai transaksi yang memuat seluruh *itemset* disebut *support* Apriori, digunakan untuk menambang *itemset* yang sering muncul dari basis data. *Support itemset* didefinisikan sebagai ukuran frekuensi *itemset* dalam *database*. Dengan menunjuk nilai *support* minimum yang ditentukan dalam percobaan dianggap sebagai poin pokok dari proses. Perhitungan nilai *support itemset X* dihitung seperti yang ditunjukkan pada formula persamaan 2.5 (John & shaiba, 2019). *Support (X)* adalah ukuran frekuensi *itemset* dalam *database* dan *N* total ukuran frekuensi *itemset* dalam *database*. *Itemset* dapat diartikan sebagai sekumpulan unit-unit yang dimiliki dalam sebuah wadah, dengan contoh keranjang belanja pada *platform* penjualan, barang-barang yang sudah dipilih pada keranjang tersebut, dapat dikatakankan sebagai *itemset*.

$$Support\ (X) = \frac{Count\ (X)}{N} \tag{2.5}$$

Algoritma Apriori ini pertama kali disampaikan oleh ilmuwan penelitian yang sekarang menjadi anggota dari perusahaan *Google*, Raskesh Agrawal dan R. Srikant pada tahun 1994. Penemuannya adalah tentang frekuensi *itemset* untuk aturan asosiasi boolean. Analisa asosiasi atau *association rule mining* adalah algoritma klasik *data mining* untuk menemukan pola hubungan antar suatu

kombinasi *itemset* dalam suatu *dataset*. Salah satu tahap analisa asosiasi yang menarik perhatian banyak peneliti adalah kemampuannya dalam menghasilkan algoritma yang efisien melalui analisa terhadap pola frekuensi tinggi atau *frequent pattern mining*. Penting tidaknya suatu assosiatif dapat diketahui melalui dua tolak ukur (parameter), yaitu nilai penunjang (*support*) dan nilai kepastian (*confidence*). Nilai penunjang (*support*) adalah presentase dari kombinasi *itemset* dalam *database*, sedangkan nilai kepastian (*confidence*) adalah kuatnya hubungan antar *item* dalam aturan assosiatif. Contoh dari aturan assosiatif biasanya dapat dinyatakan dalam bentuk: {Kopi, susu}->{gula}{*support* = 45%, *confidence* = 50%}, yang dapat diartikan "Seorang yang meminum kopi dan susu mempunyai kemungkinan 50% akan menambahkan gula" (Ahmed dkk., 2014).

Algoritma Apriori menggunakan pemahaman mengenai frekuensi *itemset*, yang telah didapatkan sebelumnya. Pada algoritma Apriori untuk menentukan kandidat-kandidat yang akan muncul, dapat dilakukan dengan cara memperhatikan nilai *minimum support* (Ahmed dkk., 2014). Ada dua proses utama yang dilakukan dalam algoritma Apriori, yaitu (Ahmed dkk., 2014) :

1. *Join* (penggabungan): Pada proses ini setiap *item* dikombinasikan dengan *item* lainnya sampai tidak terbentuk pola kombinasi lagi.
2. *Prune* (pemangkasan): Pada proses ini, hasil dari *item* yang telah dikombinasikan, lalu dipangkas dengan nilai *minimum support* yang telah ditentukan oleh *user*.

Pada proses algoritma Apriori dibagi menjadi beberapa tahap iterasi. Setiap iterasi menghasilkan pola frekuensi tinggi dengan panjang yang sama, dimulai dari iterasi pertama yang menghasilkan pola frekuensi tinggi dengan panjang satu. Pada iterasi pertama, nilai *support* dari setiap *item* dihitung dengan meninjau *database* yang digunakan. Setelah nilai *support* dari setiap *item* diperoleh, *item* yang memiliki nilai *support* di atas nilai *minimum support*, dipilih sebagai pola frekuensi tinggi dengan panjang 1 atau sering disebut 1-*itemset*. *K* -*itemset* berarti 1 set yang terdiri dari *k* item. Selanjutnya, iterasi kedua akan menghasilkan 2-item set yang tiap setnya memiliki 2 *item*. Pertama dibuat kandidat 2-*itemset* kombinasi dari semua 1-*itemset*. Lalu untuk tiap kandidat 2-*itemset* ini dihitung nilai *support*-nya dengan meninjau *database*. *Support* disini

diartikan sebagai jumlah penyelesaian dari *database* yang mengandung kedua *item* dalam kandidat 2-*itemset*. Setelah nilai *support* dari semua kandidat 2-*itemset*, kandidat 2-*itemset* yang memenuhi syarat *minimum support*, dapat ditetapkan sebagai 2-*itemset* yang merupakan pola frekuensi tinggi dengan panjang 2 (Elektronik, 2015).

Untuk selanjutnya pada iterasi ke-*k* dapat dibagi lagi menjadi beberapa bagian (Elektronik, 2015), di antaranya:

1. Pembentukan kandidat *itemset*

   Kandidat *k*-*itemset* dibentuk dari kombinasi (*k* -1) *itemset*, yang didapat pada iterasi sebelumnya. Ini merupakan salah satu ciri dari algoritma Apriori yaitu adanya pemangkasan kandidat *k* -*itemset* yang subsetnya berisikan *k* -1 *item*, yang tidak termasuk dalam pola frekuensi.

2. Perhitungan nilai *support* dari setiap kandidat *k* -*itemset*

   Support dari setiap kandidat *k* -*itemset* didapatkan dengan meninjau *database* untuk menghitung jumlah penyelesaian (transaksi) yang berisi semua *item* di dalam kandidat *k* -*itemset*. Ini juga termasuk ke dalam salah satu ciri dari algoritma Apriori dimana diperlukan perhitungan dengan *scan* seluruh *database* sebanyak *k* -*itemset* terpanjang.

3. Pola frekuensi tinggi

   Pola frekunsi tinggi yang memuat *k* atau *k* -*itemset* ditetapkan dari kandidat *k* -*itemset* yang nilai *support*-nya lebih besar dari nilai *minimum support*.

4. Pilihan lain jika tidak dapatkan pola frekuensi tinggi

   Bila tidak didapat pola frekuensi tinggi yang baru, maka seluruh proses dihentikan. Bila tidak, maka *k* ditambah satu dan kembali pada prose pada point 1.

Penerapan dari kegunaan dan manfaat algoritma Apriori dalam kehidupan nyata, sering terjadi dalam bidang bisnis dan juga penelitian, sebagai contoh meneliti jumlah pelanggan dari perusahaan telekomunikasi seluler yang diharapkan untuk memberikan respon positif terhadap penawaran *upraged* layanan yang diberikan. Penerapan lainnya yaitu dalam menentukan barang dalam supermarket yang dibeli secara bersamaan dan yang tidak dibeli secaara

bersamaan, akan dipisahkan dalam satu kelas, yang biasanya ditampilkan pada tabel kejadian, agar mudah dilihat karena telah tersusun dengan baik (Putria, 2018).

2.2.6. *Machine Learning*

Metode klasifikasi berbasis metode *machine learning* telah menarik banyak perhatian khusus sebagai salah satu sarana penyelesaian dalam membangun sebuah teknologi informasi ataupun sebuah aplikasi, pendekatan dengan metode *machine learning* memiliki kelebihan yang mana dapat menggali aturan dan korelasi yang melekat berdasarkan sejumlah analisis data (Zhang & Ghorbani, 2020), dalam industri perkembangan teknologi dengan tujuan untuk membangun sebuah sistem yang lebih dinamis, optimal dan secara luas dalam karakterisasi reservoir optimisasi. Berbagai algoritma *machine learning* yang dapat digunakan untuk proses klasifikasi yang sangat poluler untuk digunakan, di antaranya termasuk *Random Forest* (RF), *Support Vector Machine* (SVM), *Artificial Neuron Networks* (ANN), *K-Nearest Neighbor* (KNN) dan *Logistic Regression* (LR) (Li & Wang, 2020).

2.2.7. Algoritma *Random Forest*

Istilah *Random Forest* pertama kali disampaikan oleh ilmuwan komputer Amerika Tin Kam Ho pada tahun 1995, lalu algoritma *Random Forest* dikembangkan oleh anggota Akademis Sains California Breiman pada tahun 2001 (Breiman, 2001). *Random Forest* merupakan sebuah metode ensemble, yang mana metode ensemble merupakan cara untuk meningkatkan akurasi metode klasifikasi dengan mengkombinasikan metode klasifikasi dari sebuah pemilah tunggal yang tidak stabil melalui banyak kombinasi penyaringan dari suatu metode yang sama dengan proses keputusan (*voting*) untuk memperoleh prediksi klasifikasi akhir (Van & Potharst, 2007).

Random Forest diawali dengan teknik dasar dari *data mining* yaitu *decision tree*. Pada proses *decision tree*, dimana *input* berupa data akan dimasukkan pada bagian atas proses *tree* berupa akar pohon (*root*) kemudian akan dibawa turun ke bagian bawah berupa daun pohon (*leaf*) pada proses, untuk menentukan data *input*-an tersebut termasuk ke dalam kelas apa pada proses.

Random forest adalah pengklasifikasi yang terdiri dari kumpulan pengklasifikasian *tree* terstruktur di mana masing-masing *tree* melemparkan unit suara untuk kelas populer yang di-*input*, dengan kata lain *Random Forest* terdiri dari sekumpulan *decision tree* (pohon keputusan), yang mana kumpulan *decision tree* tersebut digunakan untuk mengklasifikasi data ke suatu kelas (Ahmed dkk., 2018).

Pada *decision tree* menggunakan *information gain* dan *gini index* untuk perhitungan dalam menentukan *root node* dan *rule.* Sama halnya dengan *Random Forest* yang akan menggunakan *information gain* dan *gini index* untuk perhitungan dalam membangun *tree*, hanya saja *Random Forest* akan membangun lebih dari satu *tree*. Masing-masing *tree* dibangun menggunakan *dataset* dengan atribut atau variabel yang diambil secara acak (*random*) dari data *training*, yang mana, setiap *tree* akan bergantung pada nilai dari sampel vektor yang bebas (*independent*) dengan pendistribusian yang sama pada setiap *tree* yang dibangun. Selama proses klasifikasi setiap *tree* akan memberikan *voting* kelas yang paling populer (Han dkk., 2012).

Operator *Random Forest* menghasilkan satu set *tree* acak, kelas yang dihasilkan dari proses klasifikasi dipilih dari kelas yang paling populer (modus) yang dihasilkan oleh *tree* acak yang ada. Pada gugus data yang terdiri atas jumlah pengamatan *n* dan *p* sebagai variabel penjelas, *Random Forest* dapat dilakukan dengan beberapa cara (Breiman, 2001):

1. Melakukan penarikan *random sampel* berukuran *n* dengan pemulihan pada gugus data di mana tahapan ini merupakan tahapan *bootstrap.*
2. Dengan menggunakan contoh *bootstrap, tree* dibangun sampai mencapai ukuran maksimum (tanpa pemangkasan). Pada setiap simpul, penyaringan dilakukan dengan memilih *m* variabel penjelas secara *random*, dimana *m* << *p*, lalu pemilah terbaik dipilih berdasarkan *m* variabel penjelas, dimana pada tahapan ini disebut dengan tahapan *random feature selection.*

Algoritma *Random Forest* diperlukan untuk menentukan nilai dari variabel *m*, jumlah variabel prediktor yang diambil secara *random* dan dari nilai *k tree* akan diproses, agar menghasilkan hasil yang optimal. Nilai dari *k* yang disarankan untuk digunakan pada metode *bagging.* Metode *bagging* sendiri adalah istilah

untuk *bootstrap* agregasi: mengambil sampel *L(θ) bootstrap* yang berbeda dari ukuran *n* dari set pembelajaran *(L)* ukuran *N* sebagai pembelajaran yang dimodifikasi diatur untuk setiap *tree* baru. Setiap *tree prediktor TL (θ)* tergantung pada vektor acak *θ*, yang menunjukkan sampel pada kantong dari *L*. Hasil prediksi yaitu merupakan suara mayoritas atau rata-rata dari semua *decision tree*, yaitu dengan persamaan rumus (Breiman, 2001):

$$y' = \{T_{L(\theta k)}\}_1^k \quad (2.1)$$

Ukuran *sampel* variabel penjelas *m* saat menggunakan metode *Random Forest*, akan sangat mempengaruhi korelasi dan kekuatan dari masing-masing *tree.* Untuk menentukan nilai *m* yaitu dengan menentukan jumlah variabel prediktor yang diambil secara *random* dengan nilai *p* adalah banyak variabel bebas (*independent),* seperti berikut ini (Breiman, 2001):

1. Untuk proses klasifikasi, penentuan nilai *m* adalah dengan cara $|\sqrt{p}|$ dengan nilai dari *node-nya* atau simpul terkecil adalah 1.
2. Untuk proses regresi, penentuan nilai *m* adalah dengan cara $|\frac{p}{3}|$ dengan nilai dari *node-nya* atau simpul terkecil adalah 5.

Sedangkan untuk mendapatkan nilai *m* untuk mengamati dari *error out-of-bag* (oob), terdapat tiga cara yaitu seperti persamaan di bawah ini (Breiman, 2001):

$$m = \frac{1}{2}\,|p| \quad (2.2)$$

$$m = |\sqrt{p}| \quad (2.3)$$

$$m = 2 \times |\sqrt{p}| \quad (2.4)$$

Nilai p adalah total variabel. Penggunaan *m* yang tepat akan menghasilkan *Random Forest* dengan korelasi antar *tree* cukup kecil, namun kekuatan setiap *tree* yang dibangun cukup besar yang ditunjukan dengan perolehan *error* oob yang bernilai kecil. *Error* oob bergantung pada korelasi antar *tree* dan kekuatan dari masing-masing *tree* dalam proses *Random Forest*, dimana peningkatan korelasi dapat meningkatkan juga nilai *error* oob, sedangkan peningkatan *tree* dapat menurunkan nilai *error* oob. *Error* oob dihitung dari perbandingan dari

klasifikasi yang merupakan hasil prediksi dari algoritma *Random Forest* (Breiman, 2001).

Pengaplikasian algoritma *Random Forest* telah banyak dilakukan oleh banyak penelitian, salah satu contoh yaitu pada permasalahan prediksi nilai akurasi kecelakaan untuk penyebrangan jalan raya kelas rel dengan membandingkan hasil dari model *decision tree* yang dibangun (Zhou dkk., 2020). Contoh lain dari penerapan algoritma *Random Forest* yaitu dalam pendeteksi sumber kebisingan dengan mengklasifikasi data penginderaan (pendengaran) berdasarkan peta sebagai data *training* (Maas dkk., 2019).

2.2.8. *Text Preprocessing*

Pada *dataset* umumnya akan digunakan untuk mendapatkan data yang baik dimana sudah ada pengurangan volume dari kosa kata dan membuat data lebih terstruktur, sehingga data lebih mudah diolah sistem. Namun sebelum mendapatkan data siap diolah sistem data teks informasi memiliki teks mentah yang mengandung bagian-bagian yang tidak berarti pada proses yang akan dilakukan. Agar teks mentah tersebut dapat diubah menjadi suatu representasi dengan format yang sesuai untuk algoritma *learning*, maka perlu dilakukan proses pengolahan *dataset* agar siap digunakan sebagai masukkan pada tahapan selanjutnya dari alur proses. Berikut merupakan beberapa proses agar mendapatkan data baik dan lebih terstruktur (Analisis-data.com, 2017).

1. *Case Folding*

Case folding adalah proses penyesuaian *case* dalam sebuah dokumen. Ini dilakukan untuk mempermudah pencarian. Pada penulisan sebuah dokumen tidak semua dokumen teks konsisten dalam penggunaan pada huruf kapital, dan hanya huruf alfabet yaitu dari huruf ‘a’ sampai ‘z’. Oleh karena itu, peran *case folding* dibutuhkan dalam mengkonversi keseluruhan teks dalam dokumen menjadi suatu bentuk standar. Misalnya huruf kapital menjadi huruf kecil dengan contoh *user* memasukukan kata “INISIAL”, dan keluar hasil retrieval yaitu “inisial” (Analisis-data.com, 2017).

2. *Tokenizing*

Tokenizing adalah proses untuk membagi teks yang berasal dari kalimat atau paragraf menjadi bagian-bagian tertentu. Sebagai contoh, dari kalimat

“Mailia mahasiswa Magister Sistem Informasi”, maka menghasilkan lima token dari proses *tokenizing*, yakni: “Mailia”, “mahasiswa”, “Magister”, “Sistem”, “Informasi”. Biasanya yang menjadi acuan pemisah antar token adalah spasi dan tanda baca. *Tokenizing* sering kali dipakai dalam ilmu linguistik dan hasil tokenisasi berguna untuk analisis teks lebih lanjut (Analisis-data.com, 2017).

3. *Filtering*

Filtering adalah suatu proses untuk menghilangkan bagian-bagian dari dokumen mentah yang tidak mempunyai relavansi atau arti bagi proses klasifikasi. Sebagai contoh, tanggal yang mungkin terdapat dalam dokumen mentah, label, topik, atau elemen-elemen klasifikasi lainnya yang disertakan dalam dokumen, akan dihilangkan karena elemen-elemen tersebut menspesifikasikan nilai atau kategori yang sebenarnya didapatkan melalui cara lain, yaitu penerapan algoritma *learning.* Contoh proses *steamming* dari proses *stopword* yaitu, kata-kata penghubung seperti “yang”, “di”, “dan”, “itu”, dan lainnya, langkah ini bermanfaat untuk mengurangi jumlah indeks dari suatu dokumen, hal lain seperti adanya duplikasi atau kesamaan kata yang memiliki arti yang sama akan dikurangi dan diambil satu sampel saja untuk di bawa ke proses *training* (Analisis-data.com, 2017).

4. *Stemming*

Stemming adalah proses pengubah kata ke dalam bentuk kata dasar, sehingga berfungsi mengurangi jumlah indeks yang berbeda dari suatu dokumen. Dalam proses *stemming* bisa terjadi proses *stopword,* yang mana didefinisikan sebagai sebuah kata yang sering muncul dalam suatu dokumen teks yang kurang memberikan arti penting terhadap isi dokumen. Pada teks Bahasa Indonesia semua kata imbuhan baik sufiks (akhiran kata) dan prefiks (awalan kata) juga dihilangkan. (Analisis-data.com, 2017).

2.2.9. Indeks Evaluasi Kerja

Sistem pendeteksi membutuhkan tahapan dimana, sistem tersebut di evaluasi kerja sistem berdasarkan kearutannya, terkhusus ketika menggunakan

metode klasifikasi. Salah satu teknik evaluasi kerja (salah satu sistem pendukung keputusan) yaitu *confusion matrix* (Han dkk., 2012).

2.2.9.1. *Confusion Matrix*

Confusion matrix adalah metode yang digunakan untuk melihat seberapa baik atau seberapa besar nilai dari performasi sistem yang dibangun, baik untuk menghitung tingkat akurasi, presisi dan *recall* yang dihasilkan dari model sistem khususnya dari model sistem klasifikasi, berupa prediksi atau berdasarkan kelompok (kelas) diilustrasikan pada Tabel 2.1 (Xu dkk., 2020).

Tabel 2.1 Matriks Klasifikasi Model Dua Kelas.

	Kelas Prediksi		
Kelas Aktual		Kelas Positif	Kelas Negatif
	Kelas positif	TP (*True Positive*)	FP (*False Positive*)
	Kelas Negatif	FN (*False Negative*)	TN (*True Negative*)

Nilai akurasi dapat dihitung dengan menggunakan formula persamaan 2.6.

$$Akurasi = \frac{Tp+Tn}{Tp+Tn+Fp+Fn} \times 100\% \quad (2.6)$$

$$Presisi = \frac{Tp}{Fp+Tp} \times 100\% \quad (2.7)$$

$$Recall = \frac{Tp}{Fn+Tp} \times 100\% \quad (2.8)$$

Tingkat akurasi sistem dapat dihitung dengan menggunakan persamaan 2.6, dan untuk menggambarkan jumlah data berkategori positif yang diklasifikasikan secara benar dibagi dengan total data yang diklasifikasi positif, dihitung dengan persamaan presisi 2.7. Sementara itu, persamaan *recall* menunjukan seberapa besar (persen) data kategori positif yang terklasifikasi dengan benar oleh sistem ditunjukan pada persamaan 2.8 (Xu dkk., 2020).

Dimana:

Tp : Jumlah kelompok (kelas) positif yang diklasifikasi sebagai positif.

Fp : Jumlah kelompok (kelas) negatif yang diklasifikasi sebagai positif.

Tn : Jumlah kelompok (kelas) positif yang diklasifikasi sebagai negatif.

Fn : Jumlah kelompok (kelas) negatif yang diklasifikasi sebagai negatif.

2.2.10. Metode *Undersampling*

Teknik *resampling* merupakan salah satu teknik dari *preprocessing*, dimana terdapat pendistribusian *database* yang tidak seimbang dalam pelabelan atau dari proses pembelajaran. Metode dari teknik *resampling* di antaranya ada metode *oversampling* dan *undersampling* (Jian dkk., 2016). *Undersampling* adalah cara pengambilan sampel *database* yang sedemikian rupa menjadi proporsi *class* mayoritas menjadi lebih kecil, yang bertujuan untuk menyeimbangkan *class* yang minoritas, dengan asumsi hasil bobot proporsinya menjadi 50:50. Cara *undersampling* ini dapat menghilangkan ketimpangan proporsi mayoritas dan minoritas menjadi berkurang, serta membuat data lebih presisi (Yen & Lee, 2009).

2.2.11. *Mean Absolute Error*

Mean absolute error merupakan salah satu cara paling sederhana untuk mengevaluasi keberhasilan dan bergantung pada perbedaan rata-rata antara pengamatan dan nilai nyata. *Mean absolute error* dijadikan jalan tengah peneliti dalam mengetahui dan menghasilkan bobot dari akurasi proses pengklasifikasian dari setiap data berdasarkan *absolute error* program. *Mean absolute error* (MAE) adalah rata-rata *error* dalam suatu data yang dilatih. Berikut ini merupakan formula dari rumus *Mean absolute error* (Akdemir & Çetinkaya, 2012):

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |f_i - y_i| \quad (2.9)$$

Dimana:

F_i : Nilai hasil peramalan atau perkiraan dari indeks ke-i.

y_i : Nilai sebenarnya dari indeks ke-i.

n : Jumlah keseluruhan data yang digunakan.