BAB II

LANDASAN TEORI

2.1 Analisis Sentimen

Sentimen analisis adalah riset komputasional dari opini sentimen dan emosi yang diekspresikan secara tekstual (Ira Zulfa, dan Edi Winarko 2017). Dalam analisis sentimen, teks data yang didapatkan akan diklasifikasikan menjadi beberapa jenis, seperti teks sentimen "positif", "negatif", dan "netral". Pada penerapannya, analisis sentimen dimanfaatkan untuk memberikan nilai reputasi pada pelayanan pelanggan, produk perusahaan, dan reputasi seorang tokoh publik.

2.2 Word Embedding

Word Embedding adalah istilah yang digunakan untuk teknik mengubah sebuah kata menjadi sebuah *vector* atau *array* yang terdiri dari kumpulan angka. *Word Embedding* adalah sebuah pendekatan yang digunakan untuk. merepresentasikan *vector* kata. *Word Embedding* merupakan pengembangan komputasi permodelan kata-kata yang sederhana, seperti perhitungan menggunakan jumlah dan frekuensi kemunculan kata dalam sebuah dokumen (Yulius Denny dkk, 2019).

Contoh cara tradisional untuk membaca teks dan mengubah menjadi vektor angka, misalnya terdapat sebuah kalimat yakni "sore ini merupakan sore yang indah". Langkah pertama adalah membuat sebuah *dictionary* yang berisi *list* dari seluruh kata yang *unique* atau tidak berulang, sehingga *dictionary* yang terbentuk adalah ["Sore", "ini", "merupakan", "yang", "indah"]. Langkah selanjutnya adalah

menggunakan metode *one-shot encoding* yang akan mengeluarkan *output* vektor berupa vektor ‘1’ merepresentasikan tempat kata tersebut pada *list*, dan vektor ‘0’ untuk merepresentasikan tempat kata lainnya. Contoh vektor representasi pada kata ‘merupakan’ mengacu pada metode *one-shot encoding* adalah [0, 0, 1, 0 ,0].

2.3 Teknik N-gram

Menurut Wahyu Candra Indhiarta (2017) N-gram merupakan penggabungan kata sifat yang sering muncul untuk menunjukkan suatu sentimen. Teknik N-gram memiliki jenis-jenisnya berupa *unigram* (n = 1), *bigram* (n = 2), *trigram* (n = 3), dan seterusnya. Pada dasarnya, model N-gram adalah model probabilistik yang awalnya dirancang oleh ahli matematika dari Rusia pada awal abad ke-20 dan kemudian dikembangkan untuk memprediksi *item* berikutnya dalam urutan *item* yang bisa berupa huruf atau karakter, kata, atau yang lain sesuai dengan aplikasi (Sendy Andrian dkk, 2013). Pada pengambilan karakter, N-gram terdiri dari potongan karakter sepanjang n-karakter dari sebuah teks. Contoh dari penerapan N-gram pengambilan karakter pada teks “HIMTI UMN” dapat diuraikan ke dalam beberapa n-gram berikut.

Tabel 2.1 N-Gram Karakter

N-Gram	Hasil Penerapan
unigram	‘H’, ‘I’, ‘M’, ‘T’, ‘I’,’ ‘, ‘U’, ‘M’, ‘N’
bigram	‘HI’, ‘IM’, ‘MT’, ‘TI’, ‘I ‘, ‘ U’, ‘UM’, ‘MN’.
trigram	‘HIM’, ‘IMT’, ‘MTI’, ‘TI ‘, ‘I U’, ‘ UM’, ‘UMN’

Pada pengambilan kata, N-gram terdiri dari potongan beberapa kata sepanjang n-kata dari sebuah teks atau kalimat. Contoh dari penerapan N-gram pengambilan kata pada teks “Pembelajaran mesin merupakan salah satu mata kuliah jurusan informatika” dapat diuraikan ke dalam beberapa N-gram berikut.

Tabel 2.2 N-Gram Kata

N-Gram	Hasil Penerapan
unigram	Pembelajaran, mesin, merupakan, salah, satu, mata, kuliah, jurusan, informatika
bigram	Pembelajaran mesin, mesin merupakan, merupakan salah, salah satu, satu mata, mata kuliah, kuliah jurusan, jurusan informatika
trigram	Pembelajaran mesin merupakan, mesin merupakan salah, merupakan salah satu, salah satu mata, satu mata kuliah, mata kuliah jurusan, kuliah jurusan informatika

2.4 Term Frequency-Inverse Document Frequency (TF-IDF)

Term Frequency-Inverse Document Frequency atau TF-IDF merupakan algoritma yang berguna untuk mengetahui bobot setiap kata atau seberapa sering kata tersebut. Musfiroh Nurjannah, dkk. (2013) menyatakan bahwa metode ini menggabungkan perhitungan bobot, yaitu frekuensi kemunculan sebuah kata di dalam sebuah dokumen (Term Frequency) tertentu dan *inverse* frekuensi dokumen yang mengandung kata tersebut (Inverse Document Frequency).

Frekuensi kemunculan kata (Term Frequency) di dalam dokumen menunjukkan seberapa penting kata tersebut di dalam dokumen tersebut. Frekuensi dokumen yang mengandung kata tersebut (Inverse Document Frequency) 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 Musfiroh Nurjannah, dkk. (2013).

$$TFIDFt,d = TFt,d * IDFt \quad (2.1)$$

$$TFt,d = Nt,d \; / \; NTd \quad (2.2)$$

$$\mathrm{IDFt} = \log(\mathrm{N}/\mathrm{DFt}) \quad (2.3)$$

Keterangan :

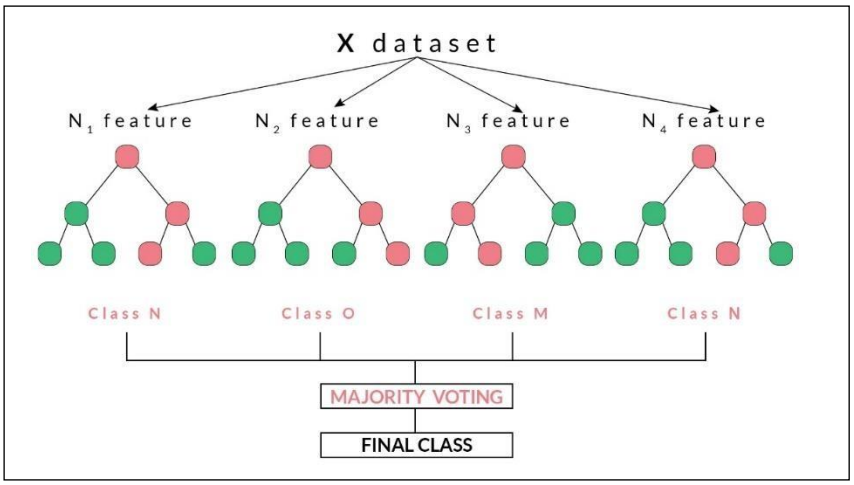
1. *TFIDFt,d* adalah Nilai TF-IDF *term t* pada suatu dokumen.
2. *TFt,d* adalah Nilai TF *term t* pada suatu dokumen.
3. *IDFt* adalah nilai IDF *term t.*
4. *Nt,d* adalah jumlah *term t* muncul pada suatu dokumen.
5. *NTd* adalah jumlah seluruh *term* pada suatu dokumen.

6. *N* adalah jumlah seluruh dokumen.
7. *DFt* adalah jumlah dokumen dengan *term t*.

2.5 Random Forest Classifier

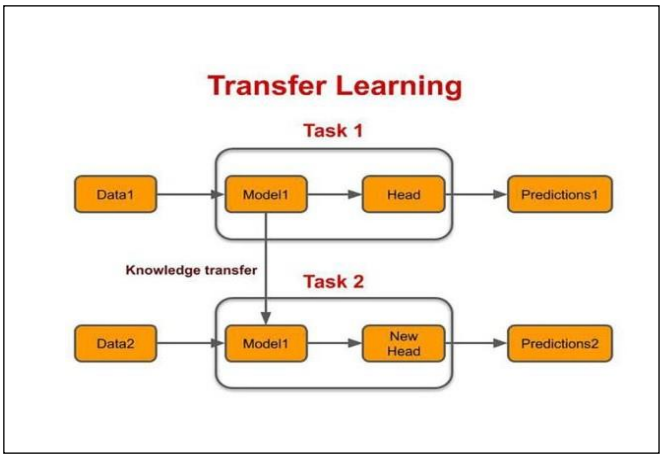
Random Forest adalah salah satu teknik *machine learning* yang dapat digunakan untuk melakukan klasifikasi. Random Forest merupakan salah satu metode dalam *decision tree*. Menurut (Aditya Yanuar, 2018) *decision tree* atau pohon pengambil keputusan adalah sebuah diagram alir yang berbentuk, seperti pohon yang memiliki sebuah *root node* yang digunakan untuk mengumpulkan data, Sebuah *inner node* yang berada pada *root node* yang berisi tentang pertanyaan tentang data dan sebuah *leaf node* yang digunakan untuk memecahkan masalah serta membuat keputusan. Random Forest memiliki beberapa *decision tree*, kemudian algoritma Random Forest mengambil keputusan berdasarkan hasil *voting* terbanyak dari semua *decision tree*.

Kelebihan dari Random Forest adalah jika terdapat data yang hilang dengan jumlah tertentu, Random Forest masih dapat melakukan klasifikasi dengan akurasi yang stabil karena tidak bergantung dengan satu *decision tree* saja melainkan membandingkan data *voting decision tree* lainnya. Dengan menggunakan *library* Random Forest Classifier dari scikit-learn, fitur yang dapat digunakan dari Random Forest Classifier tersebut adalah fitur untuk mendapatkan data *list term* apa saja yang dianggap penting untuk membangun model klasifikasi tersebut yang kemudian akan digunakan sebagai informasi untuk diterapkan pada *transfer learning*.



Gambar 2.1 Struktur Algoritma Random Forest
(Shagufta, 2019)

2.6 Transfer Learning



Gambar 2.2 Visualisasi Gambaran *Transfer Learning*

(Pratik, 2019)

Transfer learning adalah metode Deep Learning yang menerapkan pengetahuan atau *knowledge* dari domain yang berbeda namun terkait ke domain tujuan (Di Zhang dkk, 2019). *Transfer learning* dianalogikan dengan seseorang belajar untuk mengendarai sepeda, maka informasi berupa pengetahuan atau *knowledge* untuk menstabilkan sepeda tersebut bisa diterapkan untuk menstabilkan motor. Tujuan dari transfer learning adalah untuk meningkatkan proses pembelajaran sesuai target domain melalui transfer pengetahuan pada data *train* (Di Zhang dkk, 2019).