BAB 2
LANDASAN TEORI

2.1 Natural Language Processing (NLP)

Natural Language Processing adalah serangkaian teknik komputasi secara teoretis untuk menganalisis dan mewakili teks yang terjadi secara alami pada satu atau lebih tingkat analisis linguistik untuk tujuan mencapai pengolahan bahasa mirip manusia untuk berbagai tugas atau aplikasi [11]. *Natural Language Processing* kemudian dapat secara akurat mengekstrak informasi dan wawasan yang terkandung dalam dokumen serta mengkategorikan dan mengatur dokumen itu sendiri. NLTK adalah salah satu *toolkit* NLP yang paling terkenal dan komprehensif yang ditulis dalam *Python* dan menyediakan sejumlah fasilitas pengolahan dasar[12].

Natural Language Processing adalah salah satu subbidang Kecerdasan Buatan dan linguistik, dikhususkan untuk membuat komputer memahami pernyataan atau kata-kata yang ditulis dalam bahasa manusia. Sebuah bahasa natural juga dikenal sebagai bahasa biasa yang diucapkan atau ditulis oleh manusia yang digunakan untuk berkomunikaso komunikasi. Bahasa alami muncul karena ketika pengguna ingin berkomunikasi dengan komputer kita tidak bisa memaksa pengguna untuk mempelajari mesin tertentu bahasa jadi ini pada dasarnya anak-anak yang tidak punya cukup waktu untuk mempelajari atau menjadi terampil di dalam sebuah bahasa[13].

2.2 TF-IDF

TF-IDF adalah statistik numerik yang menunjukkan relevansi kata kunci dengan beberapa dokumen tertentu atau dapat dikatakan menyediakan kata kunci tersebut, dengan beberapa dokumen tertentu dapat diidentifikasi atau dikategorikan. TF-IDF akan memberikan bobot kepada tiap kata yang ada di dokumen, bobot tersebut akan digunakan sebagai bagian dari kalkulasi kemiripan judul skripsi [14].Ada dua bagian dari TF-IDF, *Term Frequency* dan *Inverse Document Frequency*.

2.2.1 Term Frequency

Term Frequency digunakan untuk mengukur berapa kali istilah ada dalam dokumen. panjang total dokumen dapat bervariasi dari sangat kecil hingga besar,

sehingga ada kemungkinan istilah apapun dapat muncul lebih sering dalam dokumen besar dibandingkan dengan dokumen kecil [14].

$$tf(t,d) = \log(1 + \text{ freq } (t,d)) \tag{2.1}$$

Keterangan:

- t : Kata
- d : Dokumen yang berisikan kumpulan kata
- freq (t, d) : berapa kali t muncul dalam dokumen d

2.2.2 Inverse Document Frequency

Inverse Document Frequency Bekerja dengan sebaliknya, Dengan memberikan bobot yang lebih rendah untuk kata-kata yang sering dan memberikan bobot yang lebih besar untuk kata-kata yang jarang Keluar yang ada di dalam kumpulan dokumen.

$$\text{idf}(t,D) = \log\left(\frac{N}{\text{count}(d \in D : t \in d)}\right) \tag{2.2}$$

Keterangan:

- N : jumlah dokumen yang terdapat di dalam *corpus*
- t : Kata
- D : Kumpulan Dokumen
- d : Sebuah Dokumen
- $count(d \in D : t \in d)$: jumlah dokumen dimana t muncul. Jika tidak ada di dalam *corpus*, ini akan menghasilkan pembagian dengan nol.

Kalkulasi akhir dari TF-IDF adalah mengalikan hasil dari *Term Frequency* dan *Inverse Document Frequency* seperti yang ditunjukkan rumus 3.

$$tfidf(t,d,D) = tf(t,d) \cdot idf(t,D) \tag{2.3}$$

Keterangan :

- t : Kata
- d : Sebuah dokumen
- D : Kumpulan dokumen

Hasil yang dikeluarkan adalah bobot atau seberapa pentignya sebuah kata yang ada dalam kumpulan kalimat.

2.3 Cosine Similarity

Cosine Similarity adalah ukuran kemiripan antara dua vektor ruang hasil kali dalam yang mengukur cosinus sudut di antara keduanya [15]. Dengan menggunakan model ini, kemiripan antara dua dokumen dapat diturunkan dengan menghitung nilai cosinus antara vektor suku dua dokumen. Semakin tinggi skor kesamaan antara vektor istilah dokumen dan vektor *Query* berarti semakin banyak relevansi antara dokumen dan *Query*[16].

$$\text{similarity } (A,B) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}} \tag{2.4}$$

Keterangan:

- A : Vektor A
- B : Vektor B
- $\|A\|$: *Euclidian norm* dari Vektor A
- $\|B\|$: *Euclidian norm* dari Vektor B

Jika total similaritas yang didapatkan adalah 0 maka dokumen yang diolah sama sekali tidak memiliki kesamaan, dan nilai yang maksimal yang bisa didapatkan adalah 1, yang artinya dokumen tersebut memiliki kemiripan. Berikut adalah contoh sederhana penggunaan *cosine similarity*.

$$\begin{aligned} A &= [5,6,9,7,3,1] \\ B &= [7,1,3,4,2,4] \end{aligned} \tag{2.5}$$

Hasil *Cosine Similarity* untuk A dan B adalah.

$$\begin{aligned}\text{similarity } (A,B) &= \frac{5\cdot 7+6\cdot 1+9\cdot 3+7\cdot 4+3\cdot 2+1\cdot 4}{\sqrt{5^2+6^2+9^2+7^2+3^2+1^2}\times\sqrt{7^2+1^2+3^2+4^2+2^2+4^2}} \\ &= \frac{106}{14.177\times 9.746} \\ &= 0.7671\end{aligned} \tag{2.6}$$

2.4 Text Preprocessing

Text preprocessing artinya membersihkan teks dari *stop words*, tanda baca, istilah yang tidak membawa banyak bobot dalam konteks sebuah teks[17]. Terdapat beberapa tahap dalam *text preprocessing* sebagai berikut;

2.4.1 Lowercasing

Lowercasing adalah teknik *preprocessing* yang paling sederhana yang terdiri dari mengubah seluruh *token* kata menjadi huruf kecil. Karena kesederhanaannya, *Lowercasing* telah menjadi metode populer dalam pembelajaran mendalam *deep learning* dan *word embedding*[18].

2.4.2 Tokenization

Tokenization adalah fase pemotongan *string input* didasarkan pada setiap kata yang menyusunnya. Untuk pemotongan ini, biasanya koma, titik dan spasi yang digunakan sebagai penanda dalam memisahkan kata-kata[19].

2.4.3 Lemmatization

Lemmatization adalah teknik yang digunakan untuk mencari kata dasar dari kata yang dicari. Cara ini dilakukan dengan menggunakan kamus untuk mengubah kata dalam bentuk apapun ke kata dasarnya. Tidak seperti *Stemming*, *Lemmatization* tidak menghilangkan awalan dan sufiks kata untuk mendapatkan akar kata. Jadi *Lemmatization* lebih efisien daripada *Stemming*[6].

2.4.4 Stopword Removal

Stopword Removal adalah sebuah proses untuk menghapus kata-kata dari dokumen yang tidak berperan penting dalam memberi pola atau informasi cerdas seperti kata sambung yang sering keluar pada sebuah kalimat[17].

2.5 Knowldege Center Universitas Multimedia Nusantara

Knowldege Center Universitas Multimedia Nusantara adalah sebuah halaman yang dikelola oleh Universitas Multimedia Nusantara (UMN) yang berisikan kumpulan artikel, jurnal dan dokumen lainnya yang terbuka untuk mahasiswa UMN. Ada beberapa jenis cara yang dapat dilakukan untuk melakukan pencarian di *website ini*, yang termudah adalah menggunakan fitur *Quick Search* yang terletak pada halaman utama. Fitur ini dapat mencari berdasarkan judul, nama *author* atau nama kontributor dan akan melakukan pencarian ke semua jenis dokumen yang ada di dalam *website*. Lalu ada metode *Advanced Search*, yang dapat memberikan parameter atau batasan untuk pencarian, sehingga hasil yang didapatkan akan lebih spesifik. Yang terakhir adalah fitur *Browse Collection*, dimana *user* dapat mencari sebuah dokumen secara manual, untuk mempermudah pencarian telah diberikan kriteria pengelompokkan yang dapat dipilih oleh *user*.

2.6 Robotic Process Automation (RPA)

Robotic Process Automation (RPA) adalah pendekatan otomatisasi proses yang muncul dengan cepat, yang menggunakan robot perangkat lunak untuk mereplikasi tugas manusia. Setelah merekam alur kerja proses, robot virtual meniru tindakan yang dilakukan oleh manusia di antarmuka aplikasi dan melaksanakan perintah yang telah direkam secara langsung [20].

RPA dapat digunakan untuk mengotomatisasi berbagai tugas seperti tugas *front office*, *back office*, proses *end to end*, mengirim detail harian atau *update* dan sebagainya. Jika tugas berulang seperti itu ditugaskan ke tenaga kerja digital daripada manusia, tenaga kerja manusia dapat dimanfaatkan untuk melakukan tugas yang lebih cerdas dan bernilai tambah. Ketika AI digabung dengan RPA, AI tersebut mampu melakukan pekerjaan seperti manusia yang memiliki kemampuan kognitif tingkat tinggi[21].

Salah satu keuntungan besar dari RPA adalah, robot dapat dilatih oleh pengguna dalam waktu yang singkat tanpa perlu campur tangan konsultan eksternal yang

memerlukan biaya yang signifikan. Karena RPA tidak memerlukan pengguna yang mempunyai latar belakang khusus. Pengawasan semua robot dan proyek masa depan atau perubahan yang membutuhkan robot yang ada dapat ditangani dan memastikan bahwa kinerja organisasi tidak terganggu[22].

RPA menjanjikan penghematan biaya yang besar, diperkirakan sekitar 20%-40% dengan lebih cepat, lebih efisien, lebih akurat, proses hemat tenaga kerja operasi dan, untuk penyedia layanan seperti Xchanging, lebih banyak nilai bisnis dan lebih tepat waktu dan kualitas layanan yang lebih tinggi yang diberikan kepada pelanggan. Robot akan melakukan tugas berulang dan sesuai dengan operasi yang ada. Bekerja di lingkungan virtual dari platform yang aman, diaudit, dan dikelola. Robot akan berjalan di lingkungan dengan pengawasan sehingga dapat disesuaikan dengan cepat sambil bekerja di yurisdiksi di berbagai tempat [4].

2.7 UIPath

UIPath atau sering juga dikenal dengan istilah UIPath RPA (*Robotic Process Automation*) merupakan sebuah perangkat yang berfungsi untuk membantu manusia dalam memasukkan segala jenis data secara berulang-ulang dengan tingkat kecepatan dan ketelitian sangat tinggi. Pada dasarnya, UIPath dibentuk oleh sekelompok insinyur yang didorong dengan ambisi untuk membangun teknologi yang terbaik. Mereka dengan sepenuh hati menjadikan UiPath sebagai platform RPA yang paling banyak digunakan di dunia saat ini, menyatukan perusahaan, mitra global berkomitmen untuk memberikan keunggulan dalam implementasi dan inovasi produk, dan komunitas pengembang RPA terbesar yang siap memberikan dampak bagi dunia[23]. Uipath memiliki tiga bagian penting untuk platform RPA-nya:

- Uipath Studio - alat pemodelan proses visual berbasis *flowchart*
- UiPath Orchestrator - konsol manajemen berbasis web, yang digunakan untuk mengelola robot dengan menawarkan gambaran yang jelas tentang apa yang sedang berjalan dan apa yang masing-masing robot itu lakukan.
- UiPath Robot - robot yang dirancang di UiPath Studio, yang dijalankan dan dikelola melalui UiPath Orchestrator

UiPath menyediakan lima jenis perekaman, diantaranya *basic recording* yang dapat digunakan untuk aktivitas tunggal, *desktop recording* yang digunakan untuk merekam beberapa tindakan yang dapat dilakukan di antara berbagai aplikasi,

web recording yang digunakan untuk merekam web dan browser aktivitas, *image recording* dan *citrix recording* digunakan untuk memberikan lingkungan virtual dan mampu merekam gambar, teks dan otomatisasi keyboard[21].

2.8 Flask

Flask adalah sebuah *micro framework* berbasis Python yang menyediakan fungsionalitas dasar kerangka kerja web dan memungkinkan lebih banyak lagi plug-in yang akan ditambahkan sehingga fungsionalitas dan set fitur dapat diperluas ke tingkat yang baru. *Flask* disebut sebagai m*micro framework* Python karena secara fungsionalitas sederhana tetapi memiliki potensial yang tinggi berdasarkan pengembangan[24]. Karena Flask adalah sebuah *micro framework*, Flask tidak membutuhkan library atau tools khusus dalam penggunaannya, tetapi flask sudah menyiapkan library dan kode yang telah dikumpulkan untuk membentuk sebuah website[25].

Flask menggunakan Jinja Template Engine dan Toolkit Werkzeug WSGI [24]. Struktur Flask dikategorikan menjadi dua bagian ”File statis” dan ”file Template”, Static File yang menyimpan kode status yang dibutuhkan untuk sebuah website seperti CSS, JavaScript dan file gambar, dan Template File yang beriskan Jinja Template Engine termasuk halaman HTML[25].