1. Pendahuluan

1.1 Latar belakang

Klasifikasi merupakan proses mengelompokkan suatu data ke dalam kelompok data yang telah ditentukan berdasarkan tingkat kemiripannya. Klasifikasi ini pun dapat diterapkan dalam dokumen teks. Dengan tujuan mempermudah penentuan seluruh dokumen dengan kategori tertentu. Permasalahan ditemukan saat jumlah dokumen yang besar, tidak mungkin diklasifikasikan secara manual dengan dibaca satu persatu, maka harus dibuat sistem yang dapat mengklasifikasikan dokumen teks tersebut secara otomatis.

Metode pengklasifikasian teks yang umum digunakan adalah *K-Nearest Neighbor* regular [3]. Dan sudah banyak peneliti yang menemukan bahwa metode *K-Nearest Neighbor* regular memiliki performansi yang bagus dalam penelitiannya [2][4]. Ide dari *K-Nearest Neighbor* reguler untuk mengklasifikasikan dokumen baru, adalah dengan cara menemukan sejumlah k tetangga terdekat dari dokumen *training*, dan menggunakan kategori dari k tetangga terdekat untuk menentukan kategori dari data kandidat. Namun kelemahan *K-Nearest Neighbor* reguler memiliki kelemahan saat menentukan *class* dari data kandidat. Seperti *K-Nearest Neighbor* reguler akan salah menentukan *class* dari data kandidat saat k tetangga terdekat memiliki anggota *neighbor* lebih banyak yang memiliki nilai *similarity* yang kecil dari suatu *class* menjadi *class* pemenenang, sedangkan anggota *neighbor* yang memiliki nilai *similarity* lebih besar kalah dalam jumlah dimana yang seharusnya menjadi *class* dari data kandidat menjadi *class* yang kalah[1]. Untuk mengatasi kelemahan tersebut telah ditemukan metode *Improved K-Nearest Neighbor*[1]. Maka dari itu dalam tugas akhir ini akan digunakan metode *Improved K-Nearest Neighbor* untuk mengklasifikasikan dokumen teks.

Perbedaan algoritma *Improved K-Nearest Neighbor* dan algoritma *K-Nearest Neighbor* regular terletak pada jumlah *neighbor* yang digunakan [1]. Pada algortima *K-Nearest Neighbor* jumlah *neighbor* yang sama diterapkan pada setiap kasus walaupun jumlah anggota tiap *class* berbeda. Sedangkan pada algoritma *Improved K-Nearest Neighbor* jumlah *neighbor* yang digunakan adalah berbeda untuk *class* yang berbeda. Dengan kata lain pada algoritma *K-Nearest Neighbor* reguler, *nearest neighbor* yang digunakan adalah sejumlah K yang telah ditentukan. Sedangkan pada algoritma *Improved K-Nearest Neighbor*, hanya digunakan *top* n *nearest neighbor* yang mewakili dari tiap *class* yang ada.

1.2 Perumusan masalah

Untuk text classification dengan menggunakan algortima *Improved K-Nearest Neighbor*, terdapat beberapa masalah yang akan diselesaikan di Tugas Akhir ini, yaitu sebagai berikut:

1. Bagaimana mengubah suatu dokumen teks menjadi data yang siap untuk digunakan dalam proses klasifikasi?
2. Bagaimana membuat sistem *text classification* dengan menggunakan metode *Improved K-Nearest Neighbor*?

3. Bagaimana performansi *Improved K-Nearest Neighbor* dalam mengklasifikasikan dokumen teks dibandingkan dengan performansi *K-Nearest Neighbor Regular* berdasarkan confusion matrix dengan parameter F-measure?

Adapun batasan masalah Tugas Akhir ini adalah sebagai berikut :

1. *Document collection* yang digunakan adalah dokumen berbahasa Indonesia yang berasal dari situs www.okezone.com.
2. Dokumen yang diuji berupa dokumen teks berita dengan format *..txt*.

1.3 Tujuan

Tujuan yang ingin dicapai dalam penyusunan Tugas Akhir ini adalah sebagai berikut:

1. Menerapkan metode *Improved K-Nearest Neighbor* dalam klasifikasi dokumen teks.
2. Menganalisis hasil implementasi Algoritma *Improved K-Nearest Neighbor* dalam *text classification* dibandingkan dengan hasil implementasi Algoritma *K-Nearest Neighbor* regular dilihat dari *precision* dan *recall, F-measure,* dan standar deviasi.

1.4 Metodologi penyelesaian masalah

Metodologi yang digunakan dalam memecahkan masalah di atas adalah dengan menggunakan langkah-langkah berikut:

1. Studi literatur
   Pencarian referensi dan sumber-sumber yang berhubungan dengan text classification.
2. Pengumpulan data
   Mengumpulkan data *document collection* yang nantinya akan digunakan sebagai *data training* dan *data testing*.
3. Analisis dan perancangan sistem
   Melakukan analisis dan perancangan terhadap sistem yang dibangun, menganalisis metode yang akan digunakan untuk menyelesaikan permasalahan, termasuk menentukan bahasa pemrograman yang digunakan, arsitektur, fungsionalitas, dan antarmuka sistem. Menyiapkan data yang siap untuk dilakukan *data mining* dengan melewati proses *preprocessing*. Input sistem berupa data latih, data validasi, dan data uji. Data latih dan data validasi digunakan untuk membangun fungsi prediksi optimal sedangkan data uji digunakan untuk menguji akurasi sistem prediksi.
4. Implementasi dan pembangunan sistem
   Pada tahap ini, akan dilakukan implementasi sistem yang mampu mengklasifikasikan dokumen teks secara otomatis dengan menggunakan metode *Improved K-Nearest Neighbor* dan *K-Nearest Neighbor* regular.
5. Pengujian dan analisis
   Setelah sistem telah sempurna maka akan dianalisis hasil dari klasifikasi dan kestabilan dari sistem.
6. Pengambilan kesimpulan dan penyusunan laporan Tugas Akhir.

2. Dasar Teori

2.1 *Text Mining*

Text Mining memiliki definisi menambang data yang berupa teks dimana sumber data biasanya didapatkan dari dokumen, dan tujuannya adalah mencari kata-kata yang dapat mewakili isi dari dokumen shingga dapat dilakukan analisa keterhubungan antar dokumen[6].

Text Mining berbeda dengan *searching* pada dokumen seperti biasanya. *Searching* biasanya dilakukan untuk mencari sesuatu infromasi yang diinginkan, namun informasi itu sebelumnya sudah ada dalam dokumen. Sedangkan *Text Mining* adalah mencari informasi baru yang diinginkan dengan mengolah informasi yang ada. Informasi yang telah ada sebelumnya diproses dengan suatu cara khusus untuk menghasilkan informasi yang lebih berguna.

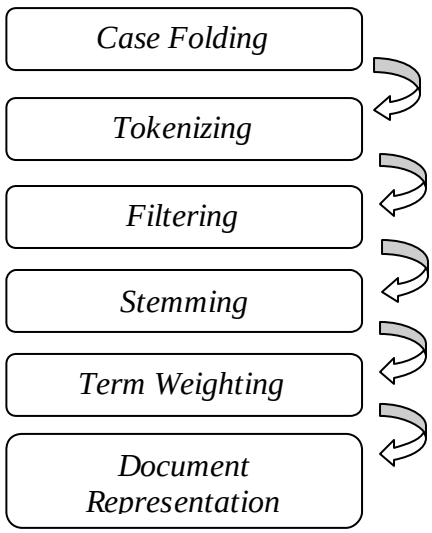
Dalam *Text Mining* terdapat dua tahapan yang akan dilakukan. Yang pertama adalah *Preprocessing*. *Preprocessing* bertujuan untuk mendapatkan data yang siap untuk dilakukan proses pada tahapan berikutnya, yang pada tugas akhir kali ini adalah proses klasifikasi.

2.2 *Preprocessing*

Dalam *preprocessing* dokumen yang ada sebelumnya akan dibuat menjadi dokumen yang terstruktur sehingga akan tercipta dokumen yang berkualitas untuk dilakukan proses klasifikasi. Tujuannya *preprocessing* dalam *text mining* adalah mentransformasi data ke suatu format yang prosesnya lebih mudah dan efektif untuk kebutuhan proses berikutnya, dengan indikator sebagai berikut :

a. Mendapatkan hasil yang lebih akurat.
b. Pengurangan waktu komputasi untuk *large scale problem*.
c. Membuat nilai menjadi lebih kecil tanpa merubah informasi yang dikandungnya.

Dalam Text Mining terdapat beberapa tahap dalam preprocessing [6], yaitu :



Gambar 2 - 1 Preprocessing