BAB II
TINJAUAN PUSTAKA

2.1. Plagiarisme

Pengertian plagiarisme menurut Kamus Besar Bahasa Indonesia (KBBI, n.d.) merupakan tindakan penjiplakan yang melanggar hak cipta. Sedangkan plagiat atau jiplakan berarti "pengambilan karangan (pendapat dan sebagainya) orang lain dan menjadikannya seolah-olah karangan (pendapat dan sebagainya) sendiri, misalnya menerbitkan karya tulis orang lain atas nama dirinya sendiri".

Ruang lingkup plagiarisme menurut (Sukaesih, 2018) yaitu tindakan mengutip kalimat atau kata-kata, gagasan, pandangan atau teori, mengggunakan fakta berupa data dan informasi dari orang lain tanpa memberikan tanda kutip dan menyebutkan identitas sumbernya. Tindakan selanjutnya adalah menyerahkan karya ilmiah hasil penjiplakan dengan berbagai kepentingan kepada pihak lain seolah-olah karya tersebut adalah karya milik sendiri.

2.2. Pencocokan String (String Matching)

Menurut *Dictionary of Algorithms and Data Structures, National Institute of Standards and Technology* (NIST, n.d.) *string* adalah susunan dari karakter-karakter (angka, alfabet atau karakter yang lain) dan biasanya diimplementasikan sebagai struktur data array. *String* dapat berupa kata, frasa, atau kalimat.

Pencocokan *string* menjadi bagian penting dari sebuah proses pencarian *string* (*string searching*) dalam sebuah database. Hasil dari pencarian sebuah string dalam database tergantung dari teknik atau cara pencocokan *string* yang digunakan.

Pencocokan *string (String Matching*) menurut *Dictionary of Algorithms and Data Structures, National Institute of Standards and Technology* (NIST, n.d.) diartikan sebagai sebuah permasalahan untuk menemukan pola susunan karakter *string* di dalam *string* lain atau bagian dari isi teks.

Pencocokan *string* secara garis besar dapat dibedakan menjadi dua(Syaroni & Munir, 2005) yaitu:

1. *Exact String Matching*, merupakan pencocokan *string* secara tepat dengan susunan karakter dalam *string* yang dicocokkan memiliki jumlah maupun urutan karakter dalam *string* yang sama. Contoh: *string* **entity** akan menunjukkan kecocokan hanya dengan *string* **entity**.

   Exact String Matching akan bermanfaat jika pengguna ingin mencari *string* dalam dokumen atau teks yang sama persis dengan *string* masukan. Salah satu algoritma *Exact String Matching* adalah algoritma knuth- morris pratt
2. *Inexact String Matching* atau *Fuzzy String Matching*, merupakan pencocokan string secara samar, maksudnya pencocokan *string* dimana *string* yang dicocokkan memiliki kemiripan dimana keduanya memiliki susunan karakter yang berbeda tetapi string tersebut memiliki kemiripan baik kemiripan tekstual/penulisan *(approximate string matching)* atau kemiripan ucapan (*phonetic string matching).*

2.3. Algoritma Knuth Morris Pratt

Algoritma Knuth Morris Pratt merupakan proses pencocokan *string*(Fau & Ginting, 2017). Algoritma ini dikembangkan secara terpisah oleh Donald E. Knuth pada tahun 1967 dan James H. Morris bersama Vaughan R. Pratt pada tahun 1966, namun keduanya mempublikasikannya secara bersamaan pada tahun 1977. Bila terjadi ketidak cocokan pada saat *pattern* sejajar dengan teks [i...i + n - 1], kita bisa menganggap ketidak cocokan pertama terjadi diantara teks [i+j] dan pattern [j], dengan 0<j<n. Berarti, teks [i...i + j] = pattern [0..j + 1] dan a = teks [i + j] tidak sama dengan b = pattern [j], ketika kita menggeser.

Dengan kata lain, pencocokan *string* akan berjalan secara efesien bila kita mempunyai tabel yang menentukan berapa panjang seharusnya menggeser seandainya terdeteksi ketidak cocokan di karakter ke-j dari pattern. Tabel itu harus memuat *next* [j] yang merupakan posisi karakter pattern setelah digeser, sehingga kita menggeser pattern secara besar j-next [j] relatif terhadap teks.

Secara sistematis, langkah-langkah yang dilakukan dalam Knuth Morris Pratt pada saat mencocokan *string*, sebagai berikut :

1. Algoritma Knuth Morris Pratt mulai mencocokan *pattern* pada awal teks.
2. Dari kiri ke kanan, algoritma ini akan mencocokan karakter per karakter *pattern*, dengan karakter di teks yang bersesuaian sampai salah satu kondisi berikut tepenuhi :
    a. Karakter di pattern dan teks yang dibandingkan tidak cocok (*mismatch*).
    b. Semua karakter di *pattern* cocok. Kemudian algortitma akan memberitahukan penemuan di posisi ini.
3. Algoritma kemudian menggeser *pattern* berdasarkan *table* next, lalu menghitung langkah 2 sampai pattern berada di ujung teks.

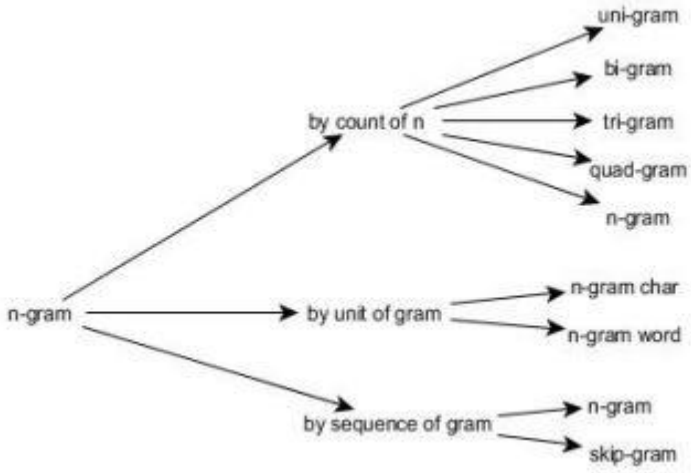
2.4. Python

Python adalah salah satu bahasa pemrograman tingkat tinggi yang bersifat *interpreter*, interaktif, *object-oriented* dan dapat beroprasi di hamper semua *platform*, seperti keluarga UNIX, Mac, Windows, dan lainnya. sebagai bahasa tingkat tinggi, Python termasuk salah satu bahasa pemrograman yang mudah untuk dipelajari karena sintaks yang jelas dan elegan, dikombinasikan dengan penggunaan module-module siap pakai dan struktur data tingkat tinggi yang efisien(Rosmala & L, 2012).

2.5. N-gram

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. Item bisa berupa huruf / karakter, kata, atau yang lain sesuai dengan aplikasi. Salah satunya, model n-gram yang berbasis kata digunakan untuk memprediksi kata berikutnya dalam urutan kata tertentu. Dalam arti bahwa sebuah n-gram hanyalah sebuah wadah kumpulan kata dengan masing-masing memiliki panjang n kata(Sugianto et al., 2013).

Berdasarkan dari jumlah potongan gram *substring*, n-gram dibedakan menjadi *uni-gram, bi-gram, tri-gram, quad-gram,* 5-gram dan seterusnya sejumlah nilai n dalam n-gram. Jika berdasarkan dari satuan unit yang diambil dalam proses parsing, n-gram dapat dibedakan menjadi n-gram karakter dan n-gram kata. Gambar klasifikasi jenis n-gram(Pratama et al., 2017).



Gambar 2.5.1 Klasifikasi Jenis N-gram

Berikut contoh pemenggalan kalimat untuk sebagian jenis n-gram diatas, jika input berupa : "saya senang meneliti dan mengembangkan metode n-gram". Maka:

1. *Uni-gram char*: {'s','a','y','a',' ','s','e','n','a','n','g',…,'r','a','m'}
2. *Bi-gram char*:{'sa','ay','ya','a ',' s','se','en','na','an','ng',…,'gr','ra','am'}
3. *6-gram char*: {'saya s','aya se','ya sen','a sena',…,' n-gra','n-gram'}
4. *Uni-gram word*: {'saya','senang','meneliti','dan',…,'metode','n-gram'}
5. *Bi-gram word*: {'saya senang','senang meneliti', 'dan mengembangkan','mengembangkan metode','metode n-gram'}
6. *1-Skip-bi-gram*: {'saya senang','saya meneliti','senang meneliti','senang dan', 'meneliti dan' , 'dan mengembangkan', 'dan metode', 'mengembangkan metode', 'mengembangkan n-gram', 'metode n-gram'}.

2.6. Penelitian Terdahulu

Bagian penelitian sebelumnya digunakan untuk mempelajari dan melihat hasil dari beberapa penelitian yang dilakukan oleh peneliti sebelumnya yang relevan dengan topik penelitian yang dilakukan peneliti sekarang. Beberapa penelitian tentang algoritma Knuth Morris Pratt telah dilakukan oleh peneliti sebelumnya diantaranya:

1. (Indriyono, 2018) dalam jurnal yang berjudul "Memberdayakan Algoritma Knuth Morris Pratt untuk Pencarian dan Pemformatan Istilah Bahasa Inggris". Dalam penelitian tersebut menjelaskan tentang cara pencarian istilah bahasa inggris dalam sebuah naskah menggunakan algoritma Knuth Morris Pratt. Kesamaan dengan penelitian yang dilakukan penulis sekarang adalah memanfaatkan database untuk menampung data, menggunakan algoritma Knuth Morris Pratt untuk melakukan pencarian atau pencocokan data dan objek yang dicari adalah bertipe string. Perbedaannya terletak pada database untuk pengujian dan format ekstensi pada dokumen yang diuji. Pada penelitian sebelumnya, database berisi kumpulan istilah bahasa inggris, sedangkan pada penelitian ini database berisi kumpulan karya ilmiah.
2. (Lestari & Djaya, 2011) dalam penelitiannya yang berjudul "Aplikasi Search Engine Menggunakan Algoritma Knuth Morris Pratt", menjelaskan tentang cara pencarian berkas dalam komputer menggunakan algoritma Knuth Morris Pratt. Penelitian terdahulu dengan penelitian yang dilakukan sekarang memiliki kesamaan dalam hal pemanfaatan database untuk menampung data/berkas dan menggunakan algoritma Knuth Morris Pratt untuk melakukan pencarian. Perbedaan terletak pada objek yang dicari, jika pada penelitian sebelumnya, pencarian dilakukan terhadap file/folder, sedangkan pada penelitian ini, objek yang dicari adalah string yang terdapat dalam sebuah dokumen.
3. (Ginting et al., 2018) dalam jurnal berjudul "Perancangan Aplikasi Pendeteksi Kesalahan Perintah SQL Query Menggunakan Algoritma

Knuth Morris Pratt", menjelaskan tentang cara pencocokan query dalam database dengan query yang diinputkan mahasiswa. Persamaan dengan penelitian sekarang adalah pencarian dan pencocokan string menggunakan algoritma Knuth Morris Pratt. Perbedaan terletak pada hasil hasil akhir. Hasil akhir pada penelitian sebelumnya berupa pembobotan pada hasil pencarian query, sedangkan hasil akhir pada penelitian sekarang berupa string yang cocok dengan dokumen database akan dibedakan dengan memberi warna merah pada string yang terdeteksi sama.

4. (Nursobah et al., 2018) dalam jurnal yang berjudul "Penerapan Algoritma Pencarian Knuth-Morris-Pratt(KMP) Dalam Sistem Informasi Perpustakaan SMK TI Pratama", menjelaskan tentang penerapan algoritma Knuth Morris Pratt untuk sistem pencarian pada sistem informasi perpustakaan. Perbedaan dengan penelitian yang dilakukan sekarang adalah pada penerapan algoritma Knuth Morris Pratt. Penelitian sebelumnya, menerapkan algoritma Knuth Morris Pratt dalam sebuah aplikasi untuk penelusuran katalog perpustakaan sekolah, sedangkan pada penelitian sekarang diterapkan dalam sebuah aplikasi untuk mendeteksi plagiasi dokumen.
5. (Fau & Ginting, 2017) dalam jurnal berjudul "Analisa Perbandingan Boyer Moore dan Knuth Morris Pratt Dalam Pencarian Judul Buku Menerapkan Metode Perbandingan Eksponensial", menjelaskan analisa dalam perbandingan dari kedua algoritma Boyer Moore dan algoritma Knuth Morris Pratt dilakukan untuk mengetahui algoritma mana yang proses pencarian dan cara kerjanya lebih cepat dengan memanfaatkan metode perbandingan eksponensial (MPE) sebagai metode pengambilan keputusan dalam mementukan hasil perbandingannya. Perbedaan utama penelitian sebelumnya dengan penelitian yang sekarang terdapat pada hasil implementasi algoritma Knuth Morris Pratt. Pada penelitian terdahulu implementasi algoritma digunakan untuk menemukan tingkat efisiensi penggunaan algoritma pencarian string antara algoritma Knuth Morris Pratt dengan Boyer Moore, sedangkan pada penelitian sekarang hasil

implementasi algoritma Knuth Morris Pratt diterapkan untuk mendeteksi plagiasi pada dokumen.