**LAPORAN STRUKTUR DATA DAN ALGORITMA**

**DESKRIPSI TUGAS BESAR “*TEXT SIMILARITY”***

Untuk Memenuhi Salah Satu Tugas

Mata Kuliah Struktur Data dan Algoritma

Dosen : Ani Rahmani

Disusun oleh :

Anggita Intan Fania (171524004)

Muhammad Alvie Irvansyah (171524018)

Muhammad Randi Noor (171524022)

**PRODI D4 TEKNIK INFORMATIKA**

**JURUSAN TEKNIK KOMPUTER DAN INFORMATIKA**

**POLITEKNIK NEGERI BANDUNG**

**2018**

**Deskripsi *Text Similarity* Secara Umum**

Secara umum, *text similarity* yang tidak menuliskan sumbernya dapat disebut sebagai plagiarisme. Menurut KBBI 2008, Plagiat adalah pengambilan karangan (pendapat, ide dsb) orang lain dan menjadikannya seolah-olah karangan (pendapat) sendiri. Dalam konteks ini, *text similarity* yang menuliskan sumber tidak disebut sebagai plagiat. Terdapat kategori dalam plagiarisme yaitu sebagai berikut. (Sastroasmoro, Sudigdo, 2007)

1. Persentase < 30% merupakan plagiat ringan
2. Persentase 30% - 70% merupakan plagiat sedang
3. Persentase > 70% merupakan plagiat berat

**Deskripsi *Text Similarity* Menurut Aplikasi**

Aplikasi yang akan kami buat yaitu menghitung persentase *text similarity* dari dua file teks. Namun, aplikasi ini lebih sederhana dibandingkan dengan plagiarisme. Pada aplikasi ini, kami menghitung presentasi dari jumlah kata yang sama baik itu menggunakan sumber ataupun tidak. Rumus yang kami gunakan untuk menghitung presentase kata yang sama yaitu :

(similiarWord / totalWord) x 100

\*Keterangan :

similarWord : jumlah kata yang sama pada kedua file

totalWord : jumlah seluruh kata pada file pertama

Adapun struktur data yang akan kami gunakan dalam program, yaitu.

* **struct Infotype:**

char word[MAX\_WORD\_LENGTH]

int frequency

* **Node list:**

Infotype info

list\* next

* **Node AVL:**

Infotype info

Node\* left

Node\* right

Berikut ini merupakan rancangan kerja yang telah kami rencanakan.

1. Membangun kamus stopword pada AVL *tree*
2. Membangun kamus kata dasar pada AVL *tree*
3. Program membaca 2 file teks dalam bahasa Indonesia
4. File pertama adalah teks yang akan dicek presentase kemiripannya dengan file kedua
5. Pada file pertama, struktur data untuk menyimpan statistik kata berupa *linkedlist*. Sedangkan untuk file kedua, struktur data untuk menyimpan statistik kata berupa AVL *tree*.
6. Melakukan *preprocessing. Preprocessing* dibagi menjadi beberapa tahap, diantaranya.

* *Case folding,* yaitu mengubah seluruh kata pada kedua file dari huruf kapital menjadi huruf kecil serta menghapus simbol (Lestari, et al., 2013).
* *Tokenizing,* yaitu memisahkan kalimat menjadi per kata (Lestari, et al., 2013). Tetapi kami tidak melakukan tahap ini karena kami akan menggunakan fscanf.
* *Stopword removal*, yaitu penghapusan kata-kata yang umum digunakan dan tidak memberikan informasi penting (Vijayarani, et al., 2014). Jika kata yang dipindai terdapat pada kamus *stopword* yang telah dibuat, kata tersebut tidak dimasukkan ke dalam masing-masing struktur data. Jika tidak ada, maka kata tersebut akan dilanjutkan ke tahap berikutnya.
* *Stemming,* yaitu proses untuk mengubah kata yang masih berimbuhan ke dalam kata dasar (Vijayarani, et al., 2014).

1. Setelah tahap *preprocessing* selesai, kata-kata tadi akan dimasukkan ke dalam masing-masing struktur data.
2. Inisialisasi variabel integer similarWord dengan 0.
3. Lalu melakukan traversal setiap node pada *linkedlist*. Jika kata pada *linkedlist* terdapat juga pada AVL *tree*, maka similarWord akan ditambah dengan frekuensi minimal dari kata yang sama pada *list* dan *tree*.
4. Lalu, hitung presentase teks yang sama pada file 1 dengan file 2 menggunakan rumus (similiarWord / totalWord) x 100.