****

**LAPORAN DESKRIPSI TUGAS BESAR APLIKASI PEMERIKSA PLAGIARISME**

**UNTUK MEMENUHI TUGAS MATA KULIAH STRUKTUR DATA DAN ALGORITMA**

Disusun oleh:

|  |  |  |
| --- | --- | --- |
| Ahmad Aji Naufal Ali | NIM 171524002 | Angkatan 2017 |
| Melina Pratiwi  Regawa Rama Prayoga | NIM 171524017  NIM 171524026 | Angkatan 2017  Angkatan 2017 |
|  |  |  |

**POLITEKNIK NEGERI BANDUNG**

**BANDUNG**

**2018**

**DESKRIPSI APLIKASI**

Aplikasi yang ingin kami buat adalah aplikasi yang dapat menentukan persentase plagiarisme antar dua teks berbahasa inggris.

**HAL YANG DIBANDINGKAN**

Aplikasi yang ingin kami buat akan membandingkan kata yang telah diolah dan mencari frekuensi jumlah kata yang sama.

**LANGKAH PENGOLAHAN KATA**

Setiap kata pada beberapa file yang telah diolah akan dimasukan ke dalam sebuah *AVL Tree*. Adapun dalam pengolahannya terdapat 4 tahap yang harus dilewati. Empat langkah tersebut ialah :

1. *Case Folding* (mengubah seluruh huruf cetak menjadi huruf kecil)
2. *Tokenizing* (mengelompokan kata)
3. *Stopword*
4. *Stemming*

*Case Folding* dan *Tokenizing* akan kami olah secara bersamaan, yaitu dengan mengubah huruf cetak menjadi huruf kecil dan tidak membiarkan karakter selain huruf masuk ke dalam array *temporary*.

Setelah itu kata yang berada dalam array akan kami bandingkan dengan *AVL Tree* *stopword* yang telah kami siapkan. Kata yang ditemukan pada *AVL* *stopword* akan tereleminasi dan yang tidak ditemukan akan di-*stemming.*

Proses *stemming* akan melibatkan tiga hal penting, yaitu algoritma pengecekan *prefix* dan *suffix,* AVL Tree yang berisi kata dasar dan *irregular verb* dan algorma penghilang *prefix* dan *suffix.*

Kata yang telah lolos dari proses *Stopword* akan diperiksa dengan AVL *Tree*  yang berisikan kata dasar dan *irregular verb.* Jika kata ditemukan pada AVL *Tree* kata dasar dan *irregular* verb, maka kata akan diubah menjadi kata dasar, apabila tidak ditemukan maka proses akan dilanjukan dengan pengecekan kombinasi *prefix* dan *suffix.*

Jika kata tidak memiliki kombinasi *prefix* atau *suffix* maka kata tersebut akan tereleminasi dan diinput ke AVL Utama, apabila memiliki kombinasi prefix dan suffix kata tersebut akan dihilangkan prefix dan suffixnya oleh algoritma stemming prefix suffix.*.*

Kata yang telah melaui keempat tahap ini akan di*-insert* ke dalam AVL Tree.

Program kemudian membandingkan statistik kata antar file dan menghitung persentase plagiarisme dengan rumus

Jumlah frekuensi kata sama pada setiap BST x Jumlah file yg dibandingkan

X 100%

Jumlah frekuensi kata pada setiap BST

Contoh:

Dibaca kata “Wing” pada file maka akan diubah menjadi “wing”, selanjutnya kata “wing” dicari katanya pada AVL Tree stopword, karena “wing” bukan merupakan *stopword*, proses dilanjut dengan mencari kata pada AVL Tree kata dasar dan irregular verb, karena “wing” merupakan kata dasar maka, proses selesai, kata “wing” akan diinput ke AVL Tree statistik kata.

**KELUARAN**

Aplikasi yang sudah selesai, akan memberikan beberapa keluaran sebagai berikut :

1. Aplikasi dapat menampilkan pada konsol berupa tabel kata per file teks
2. Aplikasi dapat mengeluarkan file output.txt yang berisikan persentase plagiarisme antar 2 file
3. Aplikasi dapat menambahkan stopword lewat konsol dan hasilnya akan disimpan pada file stopword.txt