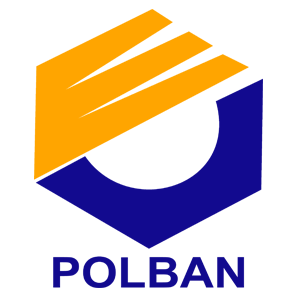
**LAPORAN STRUKTUR DATA DAN ALGORITMA**

**DESKRIPSI TUGAS BESAR “ *TEXT SIMILIARITY* “**

Untuk Memenuhi Salah Satu Tugas

Mata Kuliah Struktur Data dan Algoritma

Dosen : Ani Rahmani



Disusun oleh

Alifia Syalsabila (171524003)

Hamzah Prasetyo Utomo (171524011)

Muhammad Dena Adryan (171524015)

JURUSAN TEKNIK KOMPUTER DAN INFORMATIKA

PROGRAM STUDI D4 TEKNIK INFORMATIKA

POLITEKNIK NEGERI BANDUNG

2018

## Deskripsi Plagiarisme

Plagiarisme atau sering disebut plagiat adalah penjiplakan atau pengambilan karangan, pendapat, dan sebagainya dari orang lain dan menjadikannya seolah karangan dan pendapat sendiri. Dapat juga diartikan sebagai pengambilan [karangan](https://id.wikipedia.org/wiki/Karangan) ([pendapat](https://id.wikipedia.org/w/index.php?title=Pendapat&action=edit&redlink=1) dan sebagainya) orang lain yang kemudian dijadikan seolah-olah miliknya sendiri.

Silahkan ditambah keterangan bahwa pada pembuatan aplikasi/ program terdiri dari beberapa tahap…

1. Preprocessing

2. ….

## Text Processing

Dalam pengolahan suatu teks terdapat beberapa tahapan yang perlu dilakukan, diantaranya yaitu:

### Case Folding

**Case Folding** dibutuhkan dalam mengkonversi keseluruhan teks dalam dokumen menjadi suatu bentuk standar (biasanya huruf kecil atau lowercase). Sebagai contoh, user yang ingin mendapatkan informasi “**KOMPUTER**” dan mengetik “**KOMPOTER**”, “**KomPUter**”, atau “**komputer**”, tetap diberikan hasil retrieval yang sama yakni “**komputer**”. Case folding adalah mengubah semua huruf dalam dokumen menjadi huruf kecil. Hanya huruf ‘a’ sampai dengan ‘z’ yang diterima. Karakter selain huruf dihilangkan dan dianggap delimiter.

### Tokenizing

Tahap **Tokenizing** adalah tahap pemotongan string input berdasarkan tiap kata yang menyusunnya. Tokenisasi secara garis besar memecah sekumpulan karakter dalam suatu teks ke dalam satuan kata, bagaimana membedakan karakter-karakter tertentu yang dapat diperlakukan sebagai pemisah kata atau bukan.

### Stopword Removal / Filtering

Stopword adalah kata-kata umum dalam dokumen yang tidak memberikan informasi penting terkait dokumen tersebut (Vijayarani, et al., 2014). Proses ini dilakukan untuk mengurangi jumlah term yang harus diproses.

### Stemming

Pembuatan indeks dilakukan karena suatu dokumen tidak dapat dikenali langsung oleh suatu[**Sistem Temu Kembali Informasi**](https://informatikalogi.com/sistem-temu-kembali-informasi/) atau **Information Retrieval System (IRS).** Oleh karena itu, dokumen tersebut terlebih dahulu perlu dipetakan ke dalam suatu representasi dengan menggunakan teks yang berada di dalamnya.

# **Deskripsi Program**

Program ini dibuat untuk memeriksa potensi plagiarisme pada suatu teks dengan cara mengolah teks hingga menjadi kata dasar untuk selanjutnya dibandingkan dengan berbagai teks lain yang telah disediakan di perpustakaan sistem. Ruang lingkup program dalam memeriksa plagiarisme yaitu mengacu pada kesamaan pada kata (*word for word*). Adapun aturan yang diterapkan pada program ini mengacu pada Sastroasmoro, Sudigdo yaitu Suatu teks dapat dikatakan plagiat dengan aturan sebagai berikut : (Jelaskan spesifikasi program, mis. Berapa file yang mampu dibaca, kategori similarity, … ).

* Plagiarisme ringan < 30 %
* Plagiarisme sedang 30 – 70 %
* Plagiarisme berat atau total > 70 %

## Penjelasan Alur Sistem

Input – proses - output

### Preprocessing

Akan ada 2 file yang di proses. file pertama adalah file yang diperiksa kemiripannya terhadap file kedua .

Sebelum dilakukan perbandingan, kedua file akan dibuat menjadi AVL Tree dengan setiap kata pada file dijadikan node pada tree kecuali stopword dan imbuhan. Dengan tahapan sebagai berikut:

1. Case Folding, pada tahap ini teks dibaca melalui file lalu di konversi (menggunakan algoritma…) sehingga menjadi huruf keciltanpa karakter non-alphabet untuk memudahkan saat proses perbandingan.
2. Tokenizing, pada tahapan ini teks yang telah diolah sebelumnya dipisahkan perkata menggunakan fungsi split dimana pemisahan string dilakukan berdasarkan *white space* (spasi dan tab) untuk selanjutnya dimasukkan ke dalam tree file sementara.
3. Stopword Removal , dalam proses ini akan disiapkan sebuaharray *stopword* yang kemudian dibentuk menjadi sebuah *Binary Search Tree*(BST) untuk memudahkan saat melakukan pencarian , teks yang telah tersimpan dalam Tree dibandingkan dengan *stopword tree,* jika string yang ada pada tree file sama dengan string yang ada pada *stopword tree* maka string tersebut akan dieliminasi, jika tidak maka string tersebut akan diproses ke tahap stemming.
4. Stemming, proses ini dilakukan dengan menggunakan program stemming open source yang di dapat dari internet. Dengan mentraversal node pada AVLTreeText secara In Order, setiap kata pada node akan dilakukan proses stemming yang selanjutnya dimasukkan ke dalam tree AVL baru, saat ada kata yang sama maka akan dijumlahkan statistik kemunculannya dua node tersebut dengan tujuan untuk menggambungkan statistik kata polos dengan kata yang sebelumnya berimbuhan. Hasil proses file pertama dinamai tree utama dan file kedua dinamakan tree pembanding.

### Presentase Kemiripan Teks

Pada tahap ini , treeUtama akan dibandingkan dengan treePembanding .

1. Pertama akan dilakukan traversal pada treeUtama
2. Setiap Node hasil traversal treeUtama akan dicari di treePembanding
3. Jika terdapat kata yang sama maka akan dibandingkan frekuensi kata pada treeUtama dan treePembanding lalu diambil frekuensi paling kecil untuk selanjutnya ditampung di variable *temporary.(similiarWord)*
4. Kemudian dihitung seluruh statistik kata file kedua.
5. Setelah traversal selesai maka akan dihitung presentasenya. Rumus yang kami gunakan untuk menghitung presentase adalah

(Disesuaikan…)

Keterangan

similiarWord : kata yang sama pada dokumen utama

total : jumlah kata pada dokumen utama

## Jadwal Kegiatan

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No | Jenis Kegiatan | Waktu Pengerjaan | | | | | | | | | | | | | | |
| Minggu I | | | | | Minggu II | | | | | Minggu III | | | | |
| I | II | III | IV | V | I | II | III | IV | V | I | II | III | IV | V |
| Preprocess | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1 | Studi Literatur |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 | Pengumpulan Data |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Pembuatan Perangkat Lunak | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 | Case Folding |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 | Tokeninzing |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 | Stopword Removal |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 | Stemming |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 7 | Word Weigthing |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 8 | Presentase Kemiripan Teks |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Praprocess | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 9 | Uji Coba |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 10 | Maintenance |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 11 | Laporan Akhir |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

## *Uji Coba*

Scenario uji coba :

1. file yang sama

2…

3…

## *Jobdesk*

Pembagian tugas dalam mengerjakan program akan dilakukan bersama pada setiap prosesnya dan dibagikan per modul. ???