

# **LAPORAN TUGAS BESAR SISTEM PARALEL TERDISTRIBUSI**

Disusun Untuk Memenuhi Tugas Sistem Paralel Terdistribusi



Oleh:

ADINDA ARWA SALSABIL (1301204011)  
GAGAH AJI GUNADI (1301204093)  
IHSANI HAWA ARSYTANIA (1301204105)  
MUHAMMAD RAFFIF HAZIQ (1301204146)  
SYAHDAN NAUFAL NUR IHSAN (1301204110)

**IF-44-01**  
**S1 INFORMATIKA**  
**FAKULTAS INFORMATIKA**  
**UNIVERSITAS TELKOM**

# DAFTAR ISI

DAFTAR ISI	2
PENDAHULUAN	3
1.1 Pengertian	3
1.2 Pembagian Tugas	3
ANALISIS	4
2.1 Penjelasan	4
2.2 Solusi yang Dipilih	4
2.3 Library NLTK	4
2.4 Model Sistem	4
PERANCANGAN	5
3.1 Arsitektur Sistem dan Jaringan	5
3.2 Alur Proses Aplikasi	5
IMPLEMENTASI	7
4.1 Screen Capture Saat Aplikasi Dijalankan	7
4.2 Keterbatasan / Permasalahan yang Muncul Saat Mengerjakan Tugas Besar (opt)	7
4.3 Url Link	8
PENUTUP	9
5.1 Kesimpulan	9
5.2 Daftar Pustaka	9

# **BAB 1**

## **PENDAHULUAN**

### **1.1 Pengertian**

Pada tugas besar ini setiap kelompok diminta memilih topik berbeda dengan kelompok lainnya dalam satu kelas. Topik yang kami pilih ialah Document Similarity. Program yang kami buat dengan menggunakan sistem paralel dengan menggunakan Thread.

### **1.2 Pembagian Tugas**

- Adinda Arwa Salsabil - 1301204011 : Penyusunan laporan
- Gagah Aji Gunadi - 13012040932 : Implementasi algoritma dan penerapan multi threading
- Ihsani Hawa Arsyntania - 1301204105 : Penyusunan laporan
- Muhammad Raffif Haziq - 1301204146 : Implementasi algoritma dan penerapan multi threading
- Syahdan Naufal Nur Ihsan - 1301204110 : Penyusunan laporan

## **BAB II**

### **ANALISIS**

#### **2.1 Penjelasan**

Document Similarity adalah menentukan seberapa mirip kedua atau lebih dokumen yang diberikan. Dengan "dokumen", yang kami maksud adalah kumpulan string. Misalnya, esai atau file .txt. Banyak orang menggunakan prinsip kesamaan dokumen ini untuk memeriksa plagiarisme. Kemiripan dokumen di proses dengan langkah awal membaca dokumen dan memisahkan kalimat yang ada pada dokumen. Setelah didapatkan kalimat pada dokumen, kalimat tersebut dipecah lagi untuk mendapatkan setiap kata yang ada pada kalimat. Kemudian setiap kata pada dokumen akan dicocokkan dengan kata pada dokumen lain untuk mencari kesamaan. Hasil akhir didapat dengan menghitung rata-rata kemiripan antar dokumen.

#### **2.2 Solusi yang Dipilih**

Untuk mengimplementasikan document similarity kami menggunakan sistem paralel dengan metode thread. Dalam pengimplementasian nanti setiap thread akan handle satu fungsi. Pada komputer menggunakan bantuan library threading pada python untuk menerapkan sistem multi-threading.

#### **2.3 Library NLTK**

NLTK (Natural Language Toolkit) merupakan pustaka perangkat lunak Python yang dirancang untuk membantu dalam pemrosesan bahasa alami. NLTK menyediakan berbagai alat dan sumber daya yang bisa digunakan untuk melakukan tugas pemrosesan bahasa alami seperti tokenisasi (pemecahan teks menjadi kata-kata atau token), stemming (menghilangkan imbuhan pada kata), tagging (memberikan label pada token sesuai dengan part of speech-nya), chunking (memisahkan teks menjadi bagian-bagian yang lebih kecil), dan parsing (menguraikan struktur gramatikal dari teks). NLTK juga menyediakan banyak korpus bahasa alami yang bisa digunakan sebagai data latihan dan uji untuk sistem pemrosesan bahasa alami.

#### **2.4 Model Sistem**

Model sistem yang kami gunakan adalah sistem paralel dengan menggunakan Thread. Thread adalah sebuah unit dari sebuah proses yang dapat dijalankan secara independen dalam sistem operasi. Thread dapat digunakan dalam sistem paralel untuk meningkatkan efisiensi dan kecepatan penyelesaian tugas. Dalam sistem ini, komputer dapat menjalankan secara bersamaan untuk menyelesaikan sebagian dari tugas tersebut, sehingga meningkatkan kecepatan dan efisiensi dalam menyelesaikan tugas secara keseluruhan.

## BAB III PERANCANGAN

### 3.1 Arsitektur Sistem dan Jaringan

T1
read(docCheck)
read(dbDoc1)
calculate(docCheck, dbDoc1)
store(calculate result)

T2
read(docCheck)
read(dbDoc2)
calculate(docCheck, dbDoc2)
store(calculate result)

T3
read(docCheck)
read(dbDoc3)
calculate(docCheck, dbDoc3)
store(calculate result)

T4
read(docCheck)
read(dbDoc4)
calculate(docCheck, dbDoc4)
store(calculate result)

T5
read(docCheck)
read(dbDoc5)
calculate(docCheck, dbDoc5)
store(calculate result)

T6
read(docCheck)
read(dbDoc6)
calculate(docCheck, dbDoc6)
store(calculate result)

T7
read(docCheck)
read(dbDoc7)
calculate(docCheck, dbDoc7)
store(calculate result)

T8
read(docCheck)
read(dbDoc8)
calculate(docCheck, dbDoc8)
store(calculate result)

T9
read(docCheck)
read(dbDoc9)
calculate(docCheck, dbDoc9)
store(calculate result)

### 3.2 Alur Proses Aplikasi

1. Membaca data yang berisikan kalimat-kalimat yang akan dibandingkan
2. Isi dari setiap data akan di bagi per kalimat
3. Dari setiap kalimat akan dipecah menjadi beberapa kata

4. Setiap thread menjalankan satu fungsi yang bernama calculate
5. Fungsi calculate membutuhkan 3 parameter yaitu document yang akan dicek, document dari database, dan list untuk menyimpan hasil perhitungan kemiripan
6. Menghitung *similarity* setiap perbandingan yang dilakukan oleh fungsi
7. Hasil similarity pada fungsi akan ditambahkan ke dalam list results
8. Menghitung rata2 keseluruhan *similarity* dari perbandingan yang sudah dilakukan yang telah disimpan pada list results

## BAB IV

### IMPLEMENTASI

#### 4.1 Screen Capture Saat Aplikasi Dijalankan

##### 4.1.1 Install Library nltk

```
C:\Users\Raffifhzq\Documents\KULIAH\Semester 5\Sister\Tubes sister\tubes>pip install nltk
Requirement already satisfied: nltk in c:\users\raffifhzq\appdata\local\programs\python\python310\lib\site-packages (3.8)
Requirement already satisfied: joblib in c:\users\raffifhzq\appdata\local\programs\python\python310\lib\site-packages (from nltk) (1.2.0)
Requirement already satisfied: regex>=2021.8.3 in c:\users\raffifhzq\appdata\local\programs\python\python310\lib\site-packages (from nltk) (2022.10.31)
Requirement already satisfied: tqdm in c:\users\raffifhzq\appdata\local\programs\python\python310\lib\site-packages (from nltk) (4.64.1)
Requirement already satisfied: click in c:\users\raffifhzq\appdata\local\programs\python\python310\lib\site-packages (from nltk) (8.1.3)
Requirement already satisfied: colorama in c:\users\raffifhzq\appdata\local\programs\python\python310\lib\site-packages (from click->nltk) (0.4.6)

[notice] A new release of pip available: 22.2.2 -> 22.3.1
[notice] To update, run: python.exe -m pip install --upgrade pip
```

##### 4.1.2 Menjalankan program

```
C:\Users\Raffifhzq\Documents\KULIAH\Semester 5\Sister\Tubes sister\tubes>py main.py
[nltk_data] Downloading package punkt to
[nltk_data] C:\Users\Raffifhzq\AppData\Roaming\nltk_data...
[nltk_data] Package punkt is already up-to-date!
Number of sentences : 5
Persentase kemiripan dokumen : 12.587412587412588%

Persentase kemiripan dokumen : 30.701754385964914%

Persentase kemiripan dokumen : 34.92063492063492%

Persentase kemiripan dokumen : 17.307692307692307%

Persentase kemiripan dokumen : 33.33333333333333%

Persentase kemiripan dokumen : 19.28571428571429%

Persentase kemiripan dokumen : 12.068965517241379%

Persentase kemiripan dokumen : 22.5%

Persentase kemiripan dokumen : 52.77777777777778%

Semua thread telah dieksekusi
Kemiripan dokumen keseluruhan : 26.164809457307943%
```

#### 4.2 Keterbatasan / Permasalahan yang Muncul Saat Mengerjakan Tugas Besar (opt)

- Membandingkan dokumen hanya dengan data di dalam database yang ada
- Jenis dokumen yang dapat dibandingkan atau dicek kesamaannya hanya file dengan ekstensi .txt

### 4.3 Url Link

Code

[TUBES-SISTER-HANSIP TRUMP-DOCUMENT SIMILARITY.zip](#)

Video presentasi/demo program

[tubes-demo.mp4](#)



## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Proses pengecekan dokumen dapat dilakukan secara parallel dengan menggunakan library Threading dari python. Karena menggunakan konsep plagiarisme, kami hanya melakukan pengecekan isi dari dokumen berupa setiap kata yang ada. Jika ingin dikembangkan lebih lanjut lagi, dapat ditambahkan dengan menghitung vektor.

#### **5.2 Daftar Pustaka**

- [Multithreading in Python: Running Functions in Parallel - wellsr.com](https://www.wellsr.com/blog/multithreading-in-python-running-functions-in-parallel/)