

Nama:Kevin Christy Parinussa(1672064)

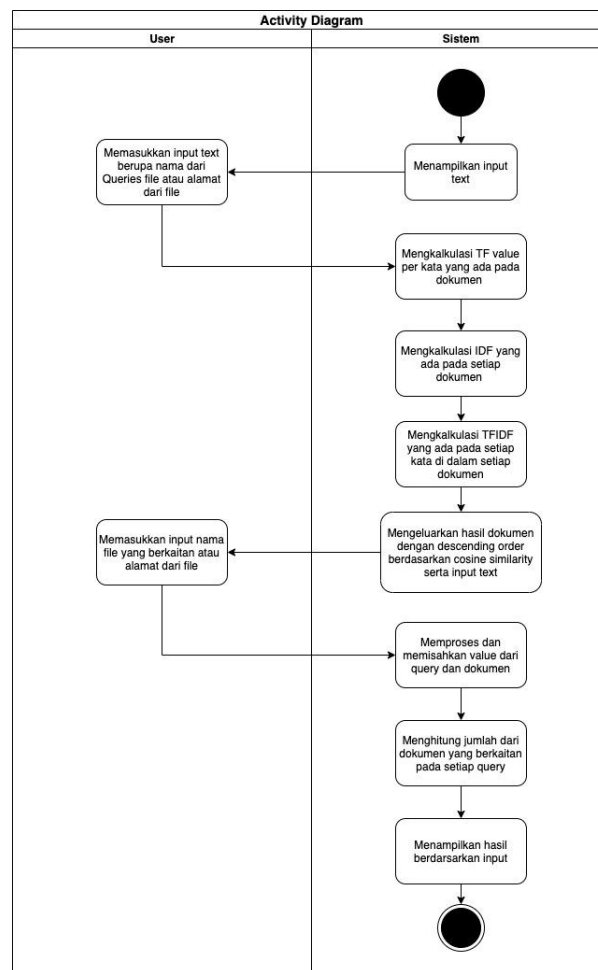
Reinhart Nehemia Suy(1672072)

Anang Adenan Sih(1472068)

David Hartono(1872031)

### C.Laporan Hasil Eksekusi IR System

1. Pada activity diagram dibawah ini, sistem menampilkan input text, lalu user memasukkan input lalu mengkalkulasi sampai hasilnya keluar.



2. Kode program tersebut dibagi menjadi tiga file python yang berbeda agar dapat dibaca dan di jalankan :
  - Prepro.py: file ini memiliki fungsi yang akan memproses input sebelumnya. Dalam file tersebut terdapat fungsi def preprocess, Fungsi ini melakukan preprocessing dan tokenize data yg didalamnya terdapat perintah-perintah seperti untuk menghapus

tag,tanda baca,semua digit,spasi,stopwords. Lalu terdapat for untuk menghapus semua kata yang panjangnya kurang dari 3, dan terakhir kembali mencari stopword serta menghapus dan mengembalikan data yang telah di proses sebelumnya.

- out.py: File ini memiliki fungsi output yang akan memberikan output seperti pada contoh output. Dalam file tersebut terdapat fungsi def output, Fungsi ini memberikan output yang diinginkan dengan menghitung presisi dan recall, menghitung jumlah presisi untuk setiap kueri, dan menghitung jumlah recall. Diakhir akan menampilkan Rata-rata presisi dan Rata-rata recall.
  - driver.py: File ini memiliki kode drive yang melakukan perhitungan untuk TF, IDF, TFIDF dan Cosine Similarity. Langkah pertamanya yaitu mengimport semua library yg diperlukan, lalu kita diminta untuk input nama file query, dan akan dihitung tfidf, tf, dan idf nya. Dan di akhir kode akan mengeluarkan output yang memberikan jawaban untuk semua pertanyaan yang ada.
3. Untuk query dan data, Preprocessing dan Tokenization dilakukan dan dictionary TF, IDF dan TFIDF dibuat. Untuk query dan data, Preprocessing dan Tokenization dilakukan dan dictionary TF, IDF dan TFIDF dibuat. Dalam fungsi output kami menghitung Presisi dan recall untuk setiap query dan presisi rata-rata dan recall untuk operasi ini. Berikut adalah perhitungan presisi dan recall yang terjadi :

```
FOR 10 MOST RELEVANT DOCUMENTS

Query 1
Precision 0.0 Recall 0.0

Query 2
Precision 0.2 Recall 0.13333333333333333

Query 3
Precision 0.2 Recall 0.13333333333333333

Query 4
Precision 0.1 Recall 0.05555555555555555

Query 5
Precision 0.2 Recall 0.10526315789473684
```

Query 6  
Precision 0.4 Recall 0.2222222222222222  
Query 7  
Precision 0.6 Recall 0.6666666666666666  
Query 8  
Precision 0.2 Recall 0.5  
Query 9  
Precision 0.2 Recall 0.25  
Query 10  
Precision 0.2 Recall 0.08333333333333333  
Average precision 0.23000000000000004  
Average recall 0.21497076023391815  
FOR 50 MOST RELEVANT DOCUMENTS  
Query 1  
Precision 0.0 Recall 0.0  
Query 2  
Precision 0.08 Recall 0.26666666666666666  
Query 3  
Precision 0.12 Recall 0.4  
Query 4  
Precision 0.04 Recall 0.11111111111111111  
Query 5  
Precision 0.2 Recall 0.5263157894736842  
Query 6  
Precision 0.12 Recall 0.3333333333333333  
Query 7  
Precision 0.16 Recall 0.8888888888888888  
Query 8  
Precision 0.06 Recall 0.75  
Query 9  
Precision 0.1 Recall 0.625

Query 10  
Precision 0.08 Recall 0.16666666666666666  
Average precision 0.096  
Average recall 0.4067982456140351

FOR 100 MOST RELEVANT DOCUMENTS

Query 1  
Precision 0.0 Recall 0.0  
Query 2  
Precision 0.1 Recall 0.6666666666666666  
Query 3  
Precision 0.09 Recall 0.6  
Query 4  
Precision 0.06 Recall 0.3333333333333333  
Query 5  
Precision 0.13 Recall 0.6842105263157895  
Query 6  
Precision 0.08 Recall 0.4444444444444444  
Query 7  
Precision 0.08 Recall 0.8888888888888888  
Query 8  
Precision 0.03 Recall 0.75  
Query 9  
Precision 0.06 Recall 0.75  
Query 10  
Precision 0.04 Recall 0.16666666666666666  
Average precision 0.06700000000000002  
Average recall 0.528421052631579

FOR 500 MOST RELEVANT DOCUMENTS

Query 1

```

Precision  0.002  Recall  1.0

Query  2

Precision  0.03  Recall  1.0

Query  3

Precision  0.03  Recall  1.0

Query  4

Precision  0.032  Recall  0.8888888888888888

Query  5

Precision  0.038  Recall  1.0

Query  6

Precision  0.032  Recall  0.8888888888888888

Query  7

Precision  0.018  Recall  1.0

Query  8

Precision  0.008  Recall  1.0

Query  9

Precision  0.016  Recall  1.0

Query  10

Precision  0.024  Recall  0.5

Average precision 0.023

Average recall 0.9277777777777778

```

## **D.Laporan dan Presentasi Studi Kasus Philoit.id**

### **1. Background**

#### **1.1 Latar Belakang**

Manusia adalah makhluk hidup yang memiliki banyak sifat yang berbeda. Akan tetapi, satu hal sifat yang dimiliki setiap manusia adalah rasa ingin tahu. Dengan adanya rasa ingin tahu ini menimbulkan keinginan manusia untuk bertanya. Setiap harinya, manusia sering melakukan kegiatan bertanya, seperti bertanya kepada keluarga, kerabat dan juga orang asing. Ketika manusia

telah bertanya dan mendapatkan jawaban yang dapat menjawab apa yang telah diinginkan, manusia akan merasa puas. Majunya teknologi dengan pesat, semakin mudah dan beragam informasi yang didapatkan manusia. Akibat dari banyaknya informasi yang diterima antara lain semakin besar juga rasa ingin tahu setiap manusia. Melihat dampak tersebut, maka mulai bermunculan aplikasi berbasis web maupun mobile yang menjadi sarana manusia untuk memuaskan rasa ingin tahu. Banyaknya manusia yang dapat mengakses aplikasi juga menimbulkan masalah-masalah baru yang dihadapi oleh user dan pihak aplikasi itu sendiri. Selain user mendapatkan jawaban yang memuaskan, user juga mendapatkan jawaban yang tidak memuaskan dan kurang berkualitas. Selain itu, user juga sering mendapatkan jawaban yang hanya spam, bahkan jawaban-jawaban duplikasi atau copy-paste. Salah satu aplikasi yang telah berjalan di bidang tersebut adalah Philoit. Philoit sendiri berupaya dalam mengatasi masalah-masalah yang terkait dan ingin memberikan pengalaman berbeda kepada user. Dengan adanya masalah ini, maka penulis akan menganalisa solusi untuk Philoit, antara lain berkaitan pada kualitas jawaban yang diberikan oleh user yang menjawab dan memisahkan jawaban yang tidak menjawab suatu pertanyaan demi menguntungkan kedua belah pihak.

## **1.2 Rumusan Masalah**

Berdasarkan uraian latar belakang, terdapat beberapa permasalahan yang dapat dirumuskan sebagai berikut:

- a. Bagaimana menentukan tingkat kualitas jawaban dari setiap user?
- b. Bagaimana membuat aplikasi yang dapat memisahkan jawaban yang termasuk duplikasi atau spam?

## **1.3 Tujuan Pembahasan**

Berdasarkan latar belakang dan rumusan masalah dapat diambil tujuan pembahasan yaitu:

- a. Untuk mengetahui tingkat kualitas jawaban dari setiap user.
- b. Membuat aplikasi yang dapat memisahkan jawaban yang termasuk duplikasi atau spam.

## **1.4 Batasan Masalah**

Untuk menghindari pembahasan tidak menyimpang dari rumusan masalah yang ada, maka penelitian ini dibatasi yaitu hanya menganalisa dan merancang tool web yang digunakan untuk menentukan tingkat plagiarisme.

## **2. Kajian Teori**

### **2.1 Sistem Tanya Jawab**

Kebutuhan sistem yang ada hubungannya dengan pencarian informasi adalah mencari jawaban yang tepat untuk kebutuhan informasi yang bersifat spesifik dari sebuah dokumen yang kompleks. Hal tersebut dikenal sebagai sistem tanya jawab. Mekanisme dari sistem tanya jawab adalah dengan melakukan ekstraksi jawaban kemudian dilanjutkan dengan pencarian dokumen, sehingga potongan teks yang ditanyakan terdapat kesamaan yang bertujuan untuk digunakan menjawab pertanyaan. Teknologi sistem tanya jawab menerapkan ekstraksi suatu pertanyaan yang digunakan untuk mencari jawaban yang sesuai. Hasil ekstraksi pertanyaan bertujuan untuk menemukan pola dalam sistem. Secara umum sistem tanya jawab terdiri dari empat modul yaitu analisa dari proses pertanyaan, perbaikan dokumen.

### **2.2 Lexical Dan Semantic Similarity Measure Feature**

Tahapan yang dilakukan pada penelitian ini yaitu, dimulai dari preprocessing dataset berdasarkan SemEval 2016 question answering similarity, melakukan ekstraksi fitur untuk membantu proses klasifikasi dengan menggunakan lexical similarity feature, semantic similarity feature, non textual feature grup dan Heuristic. Penelitian ini memfokuskan pada penggunaan lexical similarity dan semantic similarity untuk mencari kemiripan antara pertanyaan dan jawaban. Hasil ekstraksi fitur ini akan dijadikan inputan untuk classifier untuk membuat model yang akan digunakan oleh data uji. Classifier yang digunakan yaitu Support Vector Machine (SVM) dan logistic regression untuk mendapatkan score kualifikasi dimana score ini yang menentukan peringkat sebuah jawaban untuk setiap pertanyaan. Hasil penelitian yang diperoleh menunjukkan pengaruh fitur terhadap kedekatan antara pertanyaan dan jawaban. Lexical similarity feature terutama sub feature Cosine similarity dan LCS menunjukkan semakin tinggi nilai feature pada jawaban semakin jawaban tersebut mendekati pertanyaan yang diajukan. Sedangkan nilai semantic similarity menggunakan Wu Palmer Algorithm, persebaran nilai antar kelasnya lebih merata, sehingga cukup sulit untuk membedakan ciri setiap kelasnya. Non Textual Feature Group membantu dalam melakukan klasifikasi jawaban dan meningkatkan akurasi sebanyak 4%.

### **2.3 Non-Textual Feature Non-Textual Feature**

Dibedakan menjadi dua tipe yaitu Monotomic Feature dan Non-Monotomic Feature. Monotonic feature adalah feature yang kualitas jawabannya dapat diperkirakan dengan melihat nilai atau frekuensi dari feature tersebut. Dengan kata lain, semakin besar nilai featurenya maka semakin baik pula kualitas jawabannya. Feature – feature yang merupakan monotonic adalah :

a. Answerer Acceptance Ratio

Persentase jawaban terbaik dibandingkan dengan seluruh jawaban yang dimiliki oleh penjawab.

b. Answer Evaluation

Nilai bintang yang didapat oleh jawaban.

c. Answerer Activity Level

Seberapa sering pengguna bertanya dan menjawab dalam sistem.

d. Answerer Category Specialty

Jumlah pertanyaan pada suatu kategori, dan jawaban terbaik pada suatu kategori.

e. User Recommendation Banyak

Merekomendasikan sebuah pertanyaan sebagai pertanyaan yang menarik.

f. User Dis-Recommendation

Banyak dis-merekomendasikan pertanyaan

g. Click Count

Jumlah akses pada sebuah pertanyaan.

h. Number of Answer

Jumlah jawaban yang diberikan pada sebuah pertanyaan.

i. Save Count

Jumlah pengguna yang menyimpan sebuah pertanyaan ke dalam foldernya. Sedangkan non-monotonic feature adalah feature yang kualitas jawabannya tidak dapat diperkirakan dengan hanya melihat nilai mentah dari feature tersebut, karena nilai feature yang tinggi tidak menjamin bahwa kualitas jawabannya akan semakin baik pula.

Feature – feature yang merupakan non-monotonic adalah :

a. Jumlah Kata (Word)

Semua kata yang ada dalam jawaban.



b. Jumlah Kalimat (Sent)

Semua kalimat dari jawaban.

## 2.4 Textual Feature

1. Question Mark Question Mark

digunakan untuk mengetahui apakah jawaban yang ada memiliki tanda/symbol tanya. Biasanya jawaban yang terdapat tanda tanya cenderung jawaban yang tidak memberikan suatu solusi, karena penjawab menanyakan hal kembali.

2. Emotikon Emoticon

Digunakan untuk mengetahui apakah jawaban yang ada memiliki emotikon seperti simbol tertawa, sedih, dan marah. Biasanya jawaban yang terdapat emotikon cenderung jawaban yang kurang tepat.

3. Special Word

Special word digunakan untuk mengetahui apakah pada jawaban terdapat kata yang spesifik menunjukkan kualitas jawaban pada kelas (bad). Contohnya untuk jawaban yang memiliki kata pada kelas bad jawaban biasanya terdapat banyak simbol/kata tawa.

4. Heuristic/Link

Heuristic/Link Feature digunakan untuk mengetahui apakah pada jawaban terdapat alamat website/link. Pada data latih yang ada, biasanya jawaban yang memberikan alamat website/link adalah jawaban yang termasuk dalam kelas good, karena jawaban tersebut memberikan sugesti untuk melihat informasi lebih lanjut.

5. Head Word

Head Word digunakan untuk mengetahui hubungan antara pertanyaan dan jawaban berdasarkan pasangan kata tanya-jawab yang biasanya digunakan. Contohnya, pada pertanyaan terdapat kata tanya where dan jawaban yang baik mengandung kata tempat. Fitur textual memberikan nilai biner nol dan satu, nol jika jawaban tidak teridentifikasi oleh fitur, dan satu jika jawaban teridentifikasi oleh fitur tersebut.

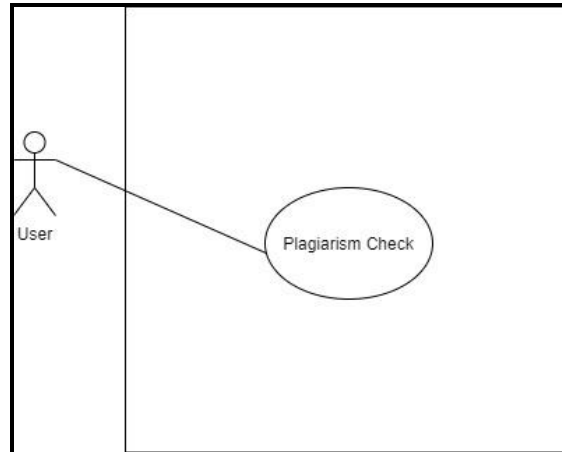
## **2.5 Evaluasi Performansi Sistem**

Terdapat beberapa cara untuk mengukur performansi metode klasifikasi diantaranya yaitu dengan menggunakan akurasi, precision, recall dan F-Measure (F1-score). Akurasi adalah tingkat kedekatan antara nilai prediksi dengan nilai aktual, precision adalah tingkat ketepatan antara informasi yang diminta dan jawaban yang diberikan oleh sistem, ISSN : 2355-9365 eProceeding of Engineering : Vol.3, No.2 Agustus 2016 | Page 3703 sedangkan recall adalah tingkat keberhasilan sistem dalam menemukan kembali informasinya. Namun terkadang perhitungan antara precision dan recall memiliki perbedaan yang cukup tinggi, untuk itu dilakukan penyetaraan nilai precision dan recall menggunakan F-measure (F1-Score).

### 3. Analisis Dan Desain

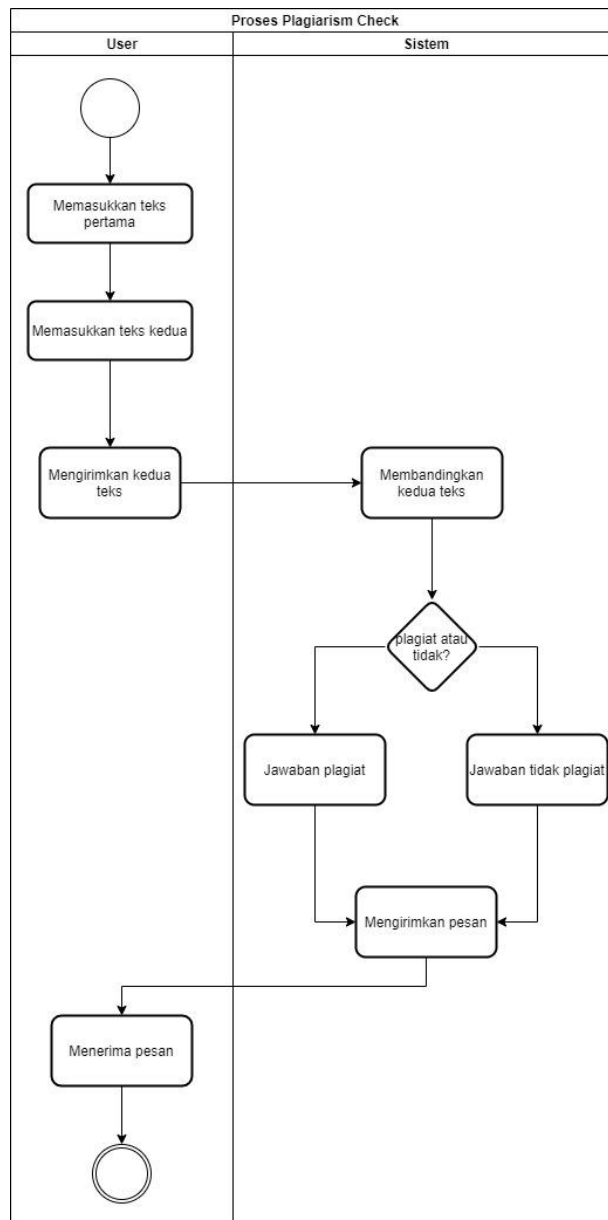
#### 3.1 Use Case

Ini adalah use case untuk perancangan aplikasi plagiarism and quality checker.



### 3.2 Activity Diagram

Ini adalah activity diagram aplikasi plagiarism and quality checker.



User mulai memasukkan kedua teks kedalam field yang sudah ada dalam aplikasi. Setelah selesai, user akan mengirimkannya ke server dimana system akan membandingkan kedua teks. Setelah selesai membandingkan, maka system akan mengirimkan pesan ke client/user dalam bentuk pesan pop-up bahwa teks pertama merupakan jawaban yang plagiat atau tidak.

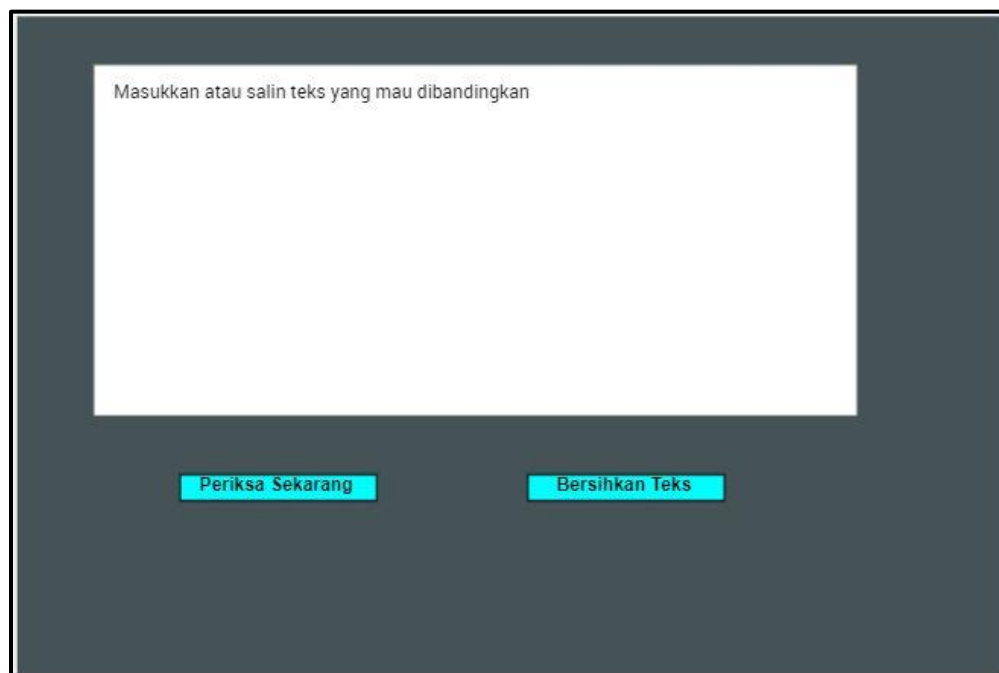
### 3.3 User Interface

Ini adalah user interface dari program yang akan kami buat untuk sekarang ini.



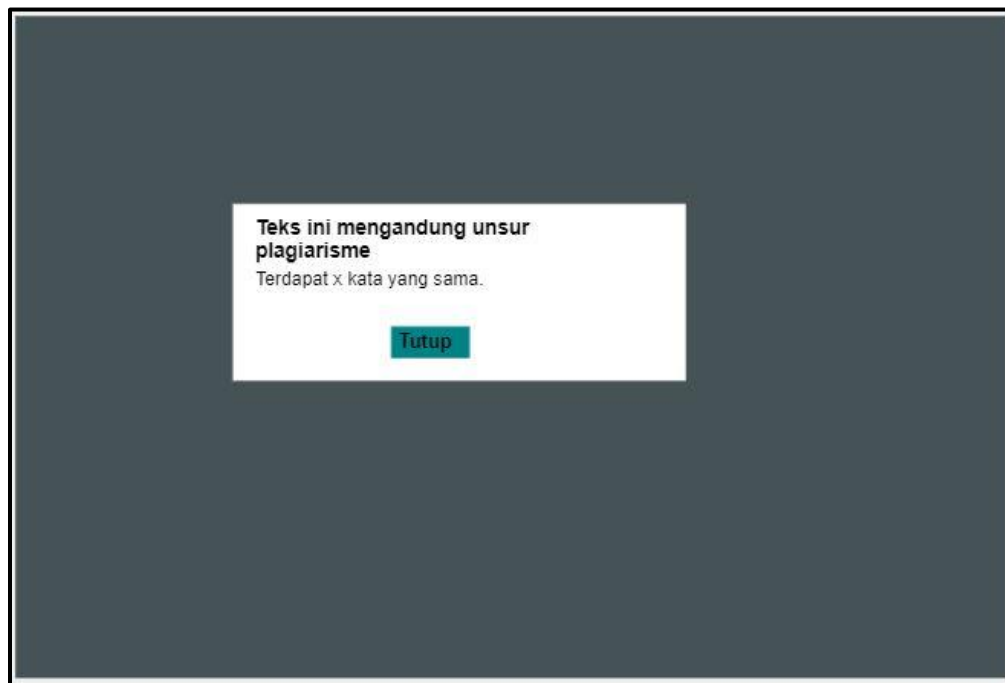
The screenshot displays a user interface with a dark gray background. At the top, there is a white rectangular text input field containing the placeholder text "Masukkan atau salin teks". Below this field, centered horizontally, is a cyan button with the text "Bersihkan Teks" in black.

Disini terdapat satu field textarea yang akan menerima teks pertama dan juga ada tombol “Bersihkan Teks” yang berguna untuk mengosongkan field.

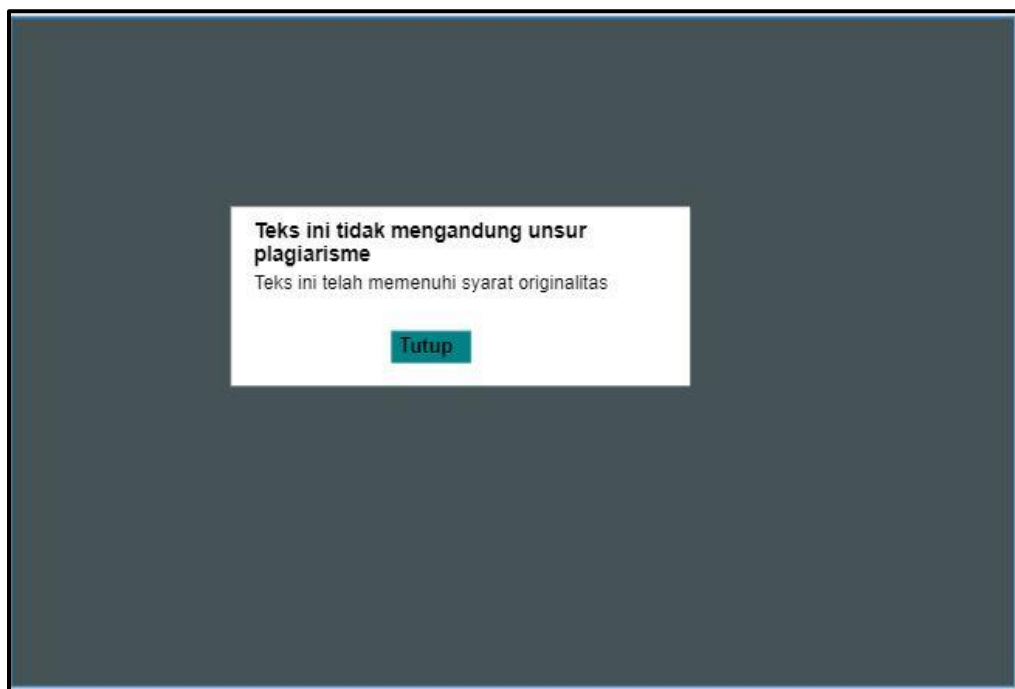


The screenshot displays a user interface with a dark gray background. At the top, there is a white rectangular text input field containing the placeholder text "Masukkan atau salin teks yang mau dibandingkan". Below this field, there are two cyan buttons: "Periksa Sekarang" on the left and "Bersihkan Teks" on the right, both in black text.

Berkutnya terdapat satu field textarea yang akan menerima teks kedua dan juga ada tombol “Periksa Sekarang” yang berguna untuk mengirimkan kedua teks ke system untuk dibandingkan.



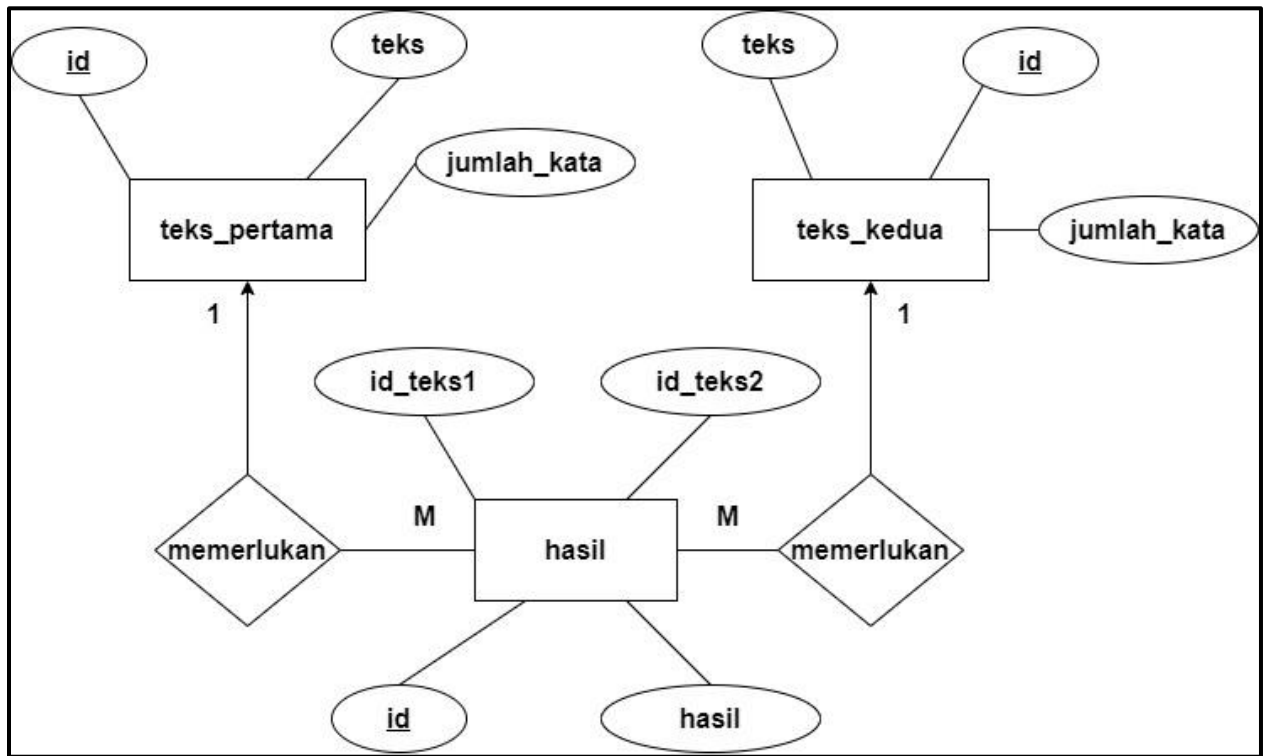
Jika teks pertama tidak memenuhi kualifikasi originalitas, maka akan muncul pesan pop-up yang menyatakan bahwa teks pertama mengandung unsur plagiarisme.



Jika tidak maka akan muncul pesan yang sebaliknya

### 3.4 ERD

Ini adalah ERD program plagiarism checker.



Penjelasan:

- **teks\_pertama**: entitas yang menyimpan teks pertama yang akan dibandingkan dengan teks kedua pada entitas **teks\_kedua**.
- **teks\_kedua**: entitas yang menyimpan teks kedua yang akan dibandingkan dengan teks pertama pada entitas **teks\_pertama**.
- **hasil**: sebuah entitas yang menyimpan hasil jika perbandingan teks pertama dan kedua mengandung unsur plagiarisme atau tidak.

## 4. Implementasi

Gambar dibawah ini menjelaskan sebuah kode program yang menjalankan fitur yang memeriksa persentase originalitas teks original dengan teks yang dibandingkan.

```
public function compare(Request $request)
{
    $similarity=0;
    similar_text($request->teks1,$request->teks2,$similarity);
    $request->session()->put('exist', true);
    if($similarity<=30){
        $request->session()->flash('alert-success', 'Persentase kemiripan data adalah '.number_format((f
    }
    else if (30<$similarity && $similarity<=70){
        $request->session()->flash('alert-warning', 'Persentase kemiripan data adalah '.number_format((f
    }
    else if(70<$similarity){
        $request->session()->flash('alert-danger', 'Persentase kemiripan data adalah '.number_format((f
    }
    return redirect('/');
}
```

Function compare memiliki sebuah parameter yang menerima masukan dari form web plagiarism checker. Masukkan tersebut akan diproses melalui syntax similar\_text(), dimana kedua input akan dibandingkan. Hasil persentase tersebut akan dimasukkan pada variable \$similarity. Kemudian variable tersebut akan dijadikan sebuah penentu melalui kondisi yang menentukan apakah plagiarisme tersebut berupa ringan, sedang atau berat.

**Selamat datang di web plagiarism checker, masukkan teks original yang akan dibandingkan dengan teks lainnya untuk mengetahui originalitas teks anda**

Teks Original

Teks Yang Dibandingkan

Periksa Sekarang

Bersihkan Teks

Activate Windows  
Go to Settings to activate Windows.



## 5. Uji Coba Sederhana

Dalam bab ini, kita akan mencoba membandingkan kedua teks untuk menentukan tingkat plagiarisme yang didapat.

**Teks Original**  
baik atau tidak semua tergantung pada sudut pandangnya,  
baik jika HP digunakan sebagai media belajar, memberikan anak pengetahuan yang  
kita tidak tau/ tidak menngerti.  
buruk jika hp hanya diberikan hanya agar anak diam dan bisa di tinggal untuk  
melakukan pekerjaan rumah tangga. ingat orang tua kita dulu tak pernah

**Teks Yang Dibandingkan**  
Menurut pengalaman pribadi saya sendiri yang sudah berkeluarga juga sebenarnya  
tidak salah juga kita memberikan smartphone kepada anak, tetapi kita  
rekomendasikan kepada hal hal yg positif untuk anak kita juga, contohnya dia ingin  
melihat serial kereta api atau kartun favorite nya, kita Carikan di menu pencarian

Periksa Sekarang

Bersihkan Teks

Setelah kita memasukkan kedua teks kedalam kolom yang sudah disediakan, system akan memeriksa kemiripan setiap karakter dalam kedua teks tersebut. Hasilnya akan memunculkan teks yang memberikan informasi persentase kemiripan data yang disertai warna balok. Balok berwarna hijau jika tingkat kemiripan dibawah 30%, berwarna kuning jika diantara 30% dan 70%, dan berwarna merah jika diatas 70%.

Persentase kemiripan data adalah 14.53%



Persentase kemiripan data adalah 54.46%



Persentase kemiripan data adalah 95.35%



## **6. Kesimpulan dan Saran**

Ini adalah kesimpulan dan saran dalam pencarian solusi plagiarisme dalam Philoit

### **6.1 Kesimpulan**

Berdasarkan apa yang telah dikerjakan selama ini dapat disimpulkan bahwa tool web yang dibuat berhasil menentukan apakah suatu jawaban dari Philoit.id merupakan hasil plagiarisme atau tidak

### **6.2 Saran**

Saran untuk apa yang sudah diteliti selama ini adalah perlu ditambahkan fitur waktu supaya pada saat pemeriksaan teks akan jauh lebih jelas dan efisien.