# BAB 3
# ANALISIS DAN PERANCANGAN

## 3.1 Analisis Masalah

Permasalahan yang terjadi pada penelitian ini yaitu mengetahui tingkat kemiripan dokumen untuk menangani masalah plagiarisme setiap informasi yang didapatkan oleh pengguna bisa menjadi bentuk dari plagiat dikarenakan adanya perlakuan *copy-paste* dari suatu dokumen tanpa mencantumkan sumber dari kutipan yang diambil tersebut.

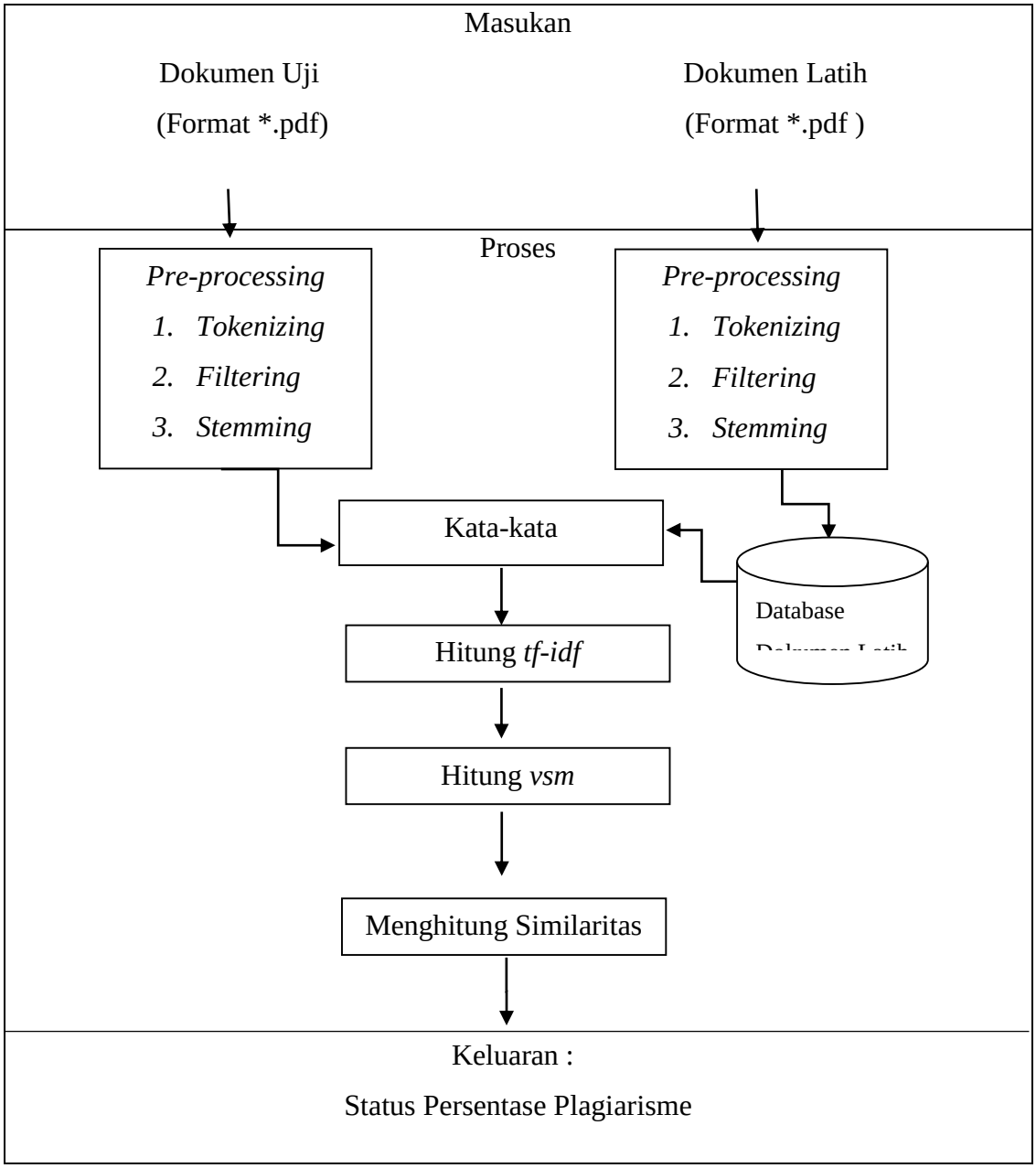
Pembangunan sistem ini diperlukan untuk menangani plagiarisme dengan ketentuan file yang sudah terintegrasi sebelumnya. Sistem ini dapat mengolah dokumen dengan tahap awal *preprocessing,* perhitungan TF-IDF, dan perhitungan *vsm*. Keluaran dari sistem ini memberikan informasi berupa persentase plagiarisme pada dokumen.

## 3.2 Analisis Sistem

Analisis sistem digunakan dengan tujuan untuk mengidentifikasi dan mengevaluasi seluruh komponen yang terkait dengan sistem yang dibangun. Sistem ini memiliki tahap proses di mulai dengan *input* dokumen, *preprocessing* hingga perhitungan similaritas, dan *output* berupa hasil persentase plagiarisme.

### 3.2.1 Analisis Metode

Penelitian yang akan dilakukan yaitu membuat sistem yang mampu melakukan pendeteksi plagiarisme antara dua dokumen. Inputan yang dimasukan adalah dokumen berupa paper yang memiliki format (*.pdf). Gambar 3.1 menunjukan gambaran mengenai sistem yang akan dibangun.




**Gambar 3.1 Gambaran Umum Sistem**

Pada sistem yang akan dibangun terdapat 3 tahapan yang akan dilakukan yaitu masukan, proses yang dilakukan oleh sistem, dan yang terakhir yaitu keluaran yang dihasilkan oleh sistem. berikut merupakan penjelasan dari masing-masing tahapan:

### 3.2.2 Analisis Masukan

Masukan dalam aplikasi pendeteksi plagiarisme disini menjelaskan tentang data yang digunakan. Masukan yang digunakan yaitu berupa dokumen *teks Bahasa Indonesia* dengan format *.pdf dimana file dalam bentuk tersebut akan diubah menjadi file .txt yang berisi teks dan sudah terintegrasi, nantinya teks tersebut dipecah menjadi kata-kata untuk menjadi pembanding dengan teks lainnya yang sudah tersimpan didalam sistem. Contoh dokumen masukan.

**Abstrak**

Problema plagiarisme merupakan salah satu persoalan klasik dalam dunia pendidikan sekarang ini karena seorang bisa dengan mudah mendapatkan informasi secara mudah. Tindakan plagiarisme tersebut dapat membuat seseorang menjadi malas berkreatifitas karena tidak mau berfikir.

### 3.2.3 Proses

Pada tahap ini dokumen yang dimasukkan oleh pengguna mengalami beberapa tahap proses. Tahap pertama yaitu *preprocessing*, menghitung bobot dengan *tf-idf*, dan metode *Vector Space Model*, dan yang terakhir yaitu menghitung nilai *similarity*.

#### 3.2.3.1 Preprocessing

Tahapan *preprocessing* adalah proses untuk mendapatkan pengetahuan dari data, dalam tahap *preprocessing* pengetahuan tersebut berarti diambil dari dokumen dengan mencari keterkaitan kata-kata yang terdapat pada isi setiap dokumen, tujuannya adalah untuk menganalisa informasi setiap kata yang saling berhubungan pada sebuah dokumen. Proses temu informasi pada *preprocessing* pada setiap dokumen melalui tahapan *tokenizing, filtering,* dan *stemming.*

##### 3.2.3.1.1 Tokenizing

Pada tahap *tokenizing* berfungsi untuk menghilangkan spasi pada dokumen, atau memisahkan sebuah kalimat menjadi token-token. Sebuah kalimat dipisahkan dan dibuat menjadi bagian-bagian kata yang dapat dilihat pada tabel 3.1 berikut.

**Tabel 3.1 Tabel *Tokenizing* Dokumen Latih**

| Dokumen | Sebelum | Sesudah |
|---|---|---|
| 1 | **Abstrak**<br>Problema plagiarisme merupakan salah satu persoalan klasik dalam dunia pendidikan sekarang ini. | Abstrak Problema<br>plagiarisme merupakan<br>salah satu<br>persoalan klasik<br>dalam dunia<br>pendidikan sekarang<br>ini . |
| 2 | **Abstrak**<br>Tindakan plagiarisme tersebut dapat membuat seseorang menjadi malas berkreatifitas karena tidak mau berfikir. | Abstrak Tindakan<br>plagiarisme tersebut<br>dapat membuat<br>seseorang menjadi<br>malas berkreatifitas<br>karena tidak<br>mau berfikir<br>. |
| 3 | **Abstrak**<br>Untuk memperkecil tindakan plagiarisme, diperlukan sistem yang mampu mendeteksi plagiarisme tanpa mengabaikan perubahan posisi kalimat. | Abstrak Untuk<br>memperkecil tindakan<br>plagiarisme ,<br>diperlukan sistem<br>yang mampu<br>mendeteksi plagiarisme<br>tanpa mengabaikan<br>perubahan posisi<br>kalimat . |
| 4 | **Abstrak**<br>Penelitian ini membuat sistem pendeteksi plagiarisme suatu dokumen berupa paper dengan menggunakan metode. | Abstrak Penelitian<br>ini membuat<br>sistem pendeteksi<br>plagiarisme suatu<br>dokumen berupa<br>paper dengan<br>menggunakan metode<br>. |

Setelah dilakukan proses Tokenizing dan didapat hasilnya maka selanjutnya akan masuk pada tahap Filtering.

#### 3.2.3.1.2 Filtering

Pada tahap *filtering* berfungsi untuk menghilangkan tanda baca, angka, serta menghilangkan kata-kata yang tidak penting seperti: abstrak, ada, yang, ke, dengan, kepada, oleh, dll. Lebih lengkapnya dapat dilihat pada tabel 3.2 berikut.

**Tabel 3.2 Tabel *Filtering* Dokumen Latih**

| Dokumen | Sebelum | Sesudah |
|---|---|---|
| 1 | Abstrak Problema<br>plagiarisme merupakan<br>salah satu<br>persoalan klasik<br>dalam dunia<br>pendidikan sekarang<br>ini . | problema plagiarisme<br>merupakan salah<br>satu persoalan<br>klasik dunia<br>pendidikan sekarang |
| 2 | Abstrak Tindakan<br>plagiarisme tersebut<br>dapat membuat<br>seseorang menjadi<br>malas berkreatifitas<br>karena tidak<br>mau berfikir<br>. | tindakan plagiarisme<br>membuat seseorang<br>menjadi malas<br>berkreatifitas karena<br>mau berfikir |
| 3 | Abstrak Untuk<br>memperkecil tindakan<br>plagiarisme ,<br>diperlukan sistem<br>yang mampu<br>mendeteksi plagiarisme<br>tanpa mengabaikan<br>perubahan posisi<br>kalimat . | memperkecil tindakan<br>plagiarisme diperlukan<br>sistem mampu<br>mendeteksi tanpa<br>mengabaikan perubahan<br>posisi kalimat |
| 4 | Abstrak Penelitian<br>ini membuat<br>sistem pendeteksi<br>plagiarisme suatu<br>dokumen berupa<br>paper dengan<br>menggunakan metode<br>. | penelitian membuat<br>sistem pendeteksi<br>plagiarisme suatu<br>dokumen berupa<br>paper menggunakan<br>metode |

Untuk proses *case folding* dilakukan pada proses ini, sebagai contoh untuk kata ”Sering” dengan huruf kapital diubah menjadi “sering”. Setelah dilakukan proses Filtering dan didapat hasilnya maka selanjutnya akan masuk pada tahap Stemming.

#### 3.2.3.1.3 Stemming

Tahap *stemming* berfungsi untuk menghilangkan semua imbuhan, baik imbuhan yang berada diawal, diakhir, atau berada diawal dan diakhir. Lebih lengkapnya dapat dilihat pada tabel 3.3 berikut.

**Tabel 3.3 Tabel *Stemming* Dokumen Latih**

| Dokumen | Sebelum | Sesudah |
|---|---|---|
| 1 | problema plagiarisme<br>merupakan salah<br>satu persoalan<br>klasik dunia<br>pendidikan sekarang | problema plagiarisme<br>rupa salah<br>satu soal<br>klasik dunia<br>didik sekarang |
| 2 | tindakan plagiarisme<br>membuat seseorang<br>menjadi malas<br>berkreatifitas karena<br>mau berfikir | tindak plagiarisme<br>buat orang<br>jadi malas<br>kreatif mau<br>fikir |
| 3 | memperkecil tindakan<br>plagiarisme diperlukan<br>sistem mampu<br>mendeteksi tanpa<br>mengabaikan perubahan<br>posisi kalimat | kecil tindak<br>plagiarisme sistem<br>mampu deteksi<br>ubah posisi<br>kalimat |
| 4 | penelitian membuat<br>sistem pendeteksi<br>plagiarisme suatu<br>dokumen berupa<br>paper menggunakan<br>metode | teliti buat<br>sistem deteksi<br>plagiarisme suatu<br>dokumen rupa<br>paper guna<br>metode |

Setelah dilakukan proses Stemming maka didapatkan kata-kata dari masing-masing dokumen latih.

#### 3.2.3.1.4 *Tokenizing* Dokumen Uji

Pada tahap *tokenizing* dokumen Uji ini menghilangkan spasi pada dokumen, atau memisahkan sebuah kalimat menjadi token-token. Sebuah kalimat dipisahkan dan dibuat menjadi bagian-bagian kata yang dapat dilihat pada tabel 3.4 berikut.

**Tabel 3.4 Proses *Tokenizing* Dokumen Uji**

| Sebelum | Sesudah |
|---|---|
| **Abstrak**<br>Mendeteksi plagiarisme dokumen sudah banyak dilakukan dengan berbagai macam metode yang digunakan. | Abstrak Mendeteksi<br>plagiarisme dokumen<br>sudah banyak<br>dilakukan dengan<br>berbagai macam<br>metode yang<br>digunakan . |

#### 3.2.3.1.5 *Filtering* Dokumen Uji

Pada tahap *filtering* dokumen Uji ini, menghilangkan tanda baca ",", menghilangkan kata "pada" dan kata "di" karena termasuk kata yang tidak penting. Untuk proses *case folding* dilakukan pada proses ini, sebagai contoh untuk huruf "M" pada kata "Mendeteksi" diubah menjadi huruf kecil.

**Tabel 3.5 Proses *Filtering* Dokumen Uji**

| Sebelum | Sesudah |
|---|---|
| Abstrak Mendeteksi<br>plagiarisme dokumen<br>sudah banyak<br>dilakukan dengan<br>berbagai macam<br>metode yang<br>digunakan . | mendeteksi plagiarisme<br>dokumen sudah<br>banyak dilakukan<br>berbagai macam<br>metode digunakan |

#### 3.2.3.1.6 *Stemming* Dokumen Uji

Pada tahap *stemming* dokumen uji ini kata "mendeteksi" dihilangkan imbuhannya sehingga menjadi kata "deteksi", kata "dilakukan" menjadi "laku", dan kata "berbagai" menjadi "bagai".

**Tabel 3.6 Proses *Stemming* Dokumen Uji**

| Sebelum | | Sesudah | |
|---|---|---|---|
| mendeteksi | plagiarisme | deteksi | plagiarisme |
| dokumen | sudah | dokumen | sudah |
| banyak | dilakukan | banyak | laku |
| berbagai | macam | bagai | macam |
| metode | digunakan | metode | guna |

#### 3.2.3.2 Proses Perhitungan Term Frequency-Inverse Document Frequency (TF-IDF)

Dari tabel 3.3 didapat kata-kata dari hasil *stemming* dokumen latih dan tabel 3.6 didapat kata-kata dari hasil *stemming* dokumen uji, kemudian kata-kata tersebut diambil dan dihitung bobot setiap katanya menggunakan *tf-idf* . Tahap perhitungan *tf-idf* yaitu sebagai berikut.

a) Menghitung nilai IDF dari query:

D = jumlah data latih yang terdapat dalam database, DF = jumlah query

D = 4 DF = 2

Maka nilai IDF dari query berdasarkan persamaan 2.2 adalah:

$$IDF = log\frac{D}{DF} \quad \rightarrow \quad IDF = log\frac{4}{2} \quad = 0{,}3010$$

b) Menghitung nilai *TF-IDF* dari query berdasarkan persamaan 2.3 adalah:

Wi = nilai perkalian dari Tf dan Idf setiap dokumen

W1 = 0 x 0,3010 = 0 W3 = 0 x 0,3010 = 0

W2 = 1 x 0,3010 = 0,3010 W4 = 1 x 0,3010 = 0,3010

Untuk perhitungan lebih lengkap dapat dilihat pada Tabel 3.7.

**Tabel 3.7 Perhitungan *TF-IDF***

| NO | Kata-kata | Q(Wiq) | TF | | | | DF | D/DF | IDF | W (term) = TF x IDF | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | D1 | D2 | D3 | D4 | | | | W1 | W2 | W3 | W4 |
| 1 | Problema | 0 | 1 | 0 | 0 | 0 | 1 | 4 | 0,6021 | 0,6021 | 0 | 0 | 0 |
| 2 | Plagiarisme | 1 | 1 | 1 | 1 | 1 | 4 | 1 | 0 | 0 | 0 | 0 | 0 |
| 3 | Rupa | 0 | 1 | 0 | 0 | 1 | 2 | 2 | 0,3010 | 0,3010 | 0 | 0 | 0,3010 |
| 4 | Salah | 0 | 1 | 0 | 0 | 0 | 1 | 4 | 0,6021 | 0,6021 | 0 | 0 | 0 |
| 5 | Satu | 0 | 1 | 0 | 0 | 0 | 1 | 4 | 0,6021 | 0,6021 | 0 | 0 | 0 |
| 6 | Soal | 0 | 1 | 0 | 0 | 0 | 1 | 4 | 0,6021 | 0,6021 | 0 | 0 | 0 |
| 7 | Klasik | 0 | 1 | 0 | 0 | 0 | 1 | 4 | 0,6021 | 0,6021 | 0 | 0 | 0 |
| 8 | Dunia | 0 | 1 | 0 | 0 | 0 | 1 | 4 | 0,6021 | 0,6021 | 0 | 0 | 0 |
| 9 | Didik | 0 | 1 | 0 | 0 | 0 | 1 | 4 | 0,6021 | 0,6021 | 0 | 0 | 0 |
| 10 | Sekarang | 0 | 1 | 0 | 0 | 0 | 1 | 4 | 0,6021 | 0,6021 | 0 | 0 | 0 |
| 11 | Tindak | 0 | 0 | 1 | 1 | 0 | 2 | 2 | 0,3010 | 0 | 0,3010 | 0,3010 | 0 |
| 12 | Buat | 0 | 0 | 1 | 0 | 1 | 2 | 2 | 0,3010 | 0 | 0,3010 | 0 | 0,3010 |
| 13 | Orang | 0 | 0 | 1 | 0 | 0 | 1 | 4 | 0,6021 | 0 | 0,6021 | 0 | 0 |
| 14 | Jadi | 0 | 0 | 1 | 0 | 0 | 1 | 4 | 0,6021 | 0 | 0,6021 | 0 | 0 |
| 15 | Malas | 0 | 0 | 1 | 0 | 0 | 1 | 4 | 0,6021 | 0 | 0,6021 | 0 | 0 |
| 16 | Kreatif | 0 | 0 | 1 | 0 | 0 | 1 | 4 | 0,6021 | 0 | 0,6021 | 0 | 0 |
| 17 | Mau | 0 | 0 | 1 | 0 | 0 | 1 | 4 | 0,6021 | 0 | 0,6021 | 0 | 0 |
| 18 | Fikir | 0 | 0 | 1 | 0 | 0 | 1 | 4 | 0,6021 | 0 | 0,6021 | 0 | 0 |
| 19 | Kecil | 0 | 0 | 0 | 1 | 0 | 1 | 4 | 0,6021 | 0 | 0 | 0,6021 | 0 |
| 20 | Sistem | 0 | 0 | 0 | 1 | 0 | 1 | 4 | 0,6021 | 0 | 0 | 0,6021 | 0 |
| 21 | Mampu | 0 | 0 | 0 | 1 | 0 | 1 | 4 | 0,6021 | 0 | 0 | 0,6021 | 0 |
| 22 | Deteksi | 1 | 0 | 0 | 1 | 1 | 2 | 2 | 0,3010 | 0 | 0,3010 | 0,3010 | 0 |
| 23 | Ubah | 0 | 0 | 0 | 1 | 0 | 1 | 4 | 0,6021 | 0 | 0 | 0,6021 | 0 |
| 24 | Posisi | 0 | 0 | 0 | 1 | 0 | 1 | 4 | 0,6021 | 0 | 0 | 0,6021 | 0 |
| 25 | Kalimat | 0 | 0 | 0 | 1 | 0 | 1 | 4 | 0,6021 | 0 | 0 | 0,6021 | 0 |
| 26 | Teliti | 0 | 0 | 0 | 0 | 1 | 1 | 4 | 0,6021 | 0 | 0 | 0 | 0,6021 |
| 27 | Buat | 0 | 0 | 0 | 0 | 1 | 1 | 4 | 0,6021 | 0 | 0 | 0 | 0,6021 |
| 28 | Suatu | 0 | 0 | 0 | 0 | 1 | 1 | 4 | 0,6021 | 0 | 0 | 0 | 0,6021 |
| 29 | Dokumen | 1 | 0 | 0 | 0 | 1 | 1 | 4 | 0,6021 | 0 | 0 | 0 | 0,6021 |
| 30 | Paper | 0 | 0 | 0 | 0 | 1 | 1 | 4 | 0,6021 | 0 | 0 | 0 | 0,6021 |
| 31 | Guna | 1 | 0 | 0 | 0 | 1 | 1 | 4 | 0,6021 | 0 | 0 | 0 | 0,6021 |
| 32 | Metode | 1 | 0 | 0 | 0 | 1 | 1 | 4 | 0,6021 | 0 | 0 | 0 | 0,6021 |

Dalam tabel 3.7 diambil contoh kata “deteksi” dengan nilai DF yaitu 2 dimana kata tersebut terdapat pada dokumen 3 dan dokumen 4, lalu nilai D/DF yaitu 2 didapat dari banyaknya dokumen dibagi dengan kemunculan kata tersebut di setiap

dokumennya, dan nilai IDF yaitu 0,3010 didapat dari log D/DF. Setelah semua nilai didapat maka selanjutnya menghitung nial bobot dokumen dengan rumus W=TF x IDF, diperoleh hasil dari kata “sering” di setiap masing-masing dokumen. Hasil perhitungan bobot tersebut nantinya akan digunakan dalam perhitungan *vector space model*(*vsm*).

#### 3.2.3.3 Proses Perhitungan Vector Space Model (VSM)

Dari tabel 3.7 didapatkan bobot *tf-idf* dari masing-masing dokumen, maka dapat dilanjutkan dengan menghitung menggunakan metode *vector space model (vsm).* Tahap perhitungan metode *vsm* yaitu sebagai berikut.

a) Menghitung nilai panjang vektor dari query sebagai berikut.

Nilai dari |Q| merupakan hasil dari penjumlahan seluruh query lalu diakarkan.

$|Q| = \sqrt{\sum_i w^2 q,j}$

$=\sqrt{1^2 + 1^2 + 1^2 + 1^2 + 1^2} = \sqrt{5} = 2{,}24$

Nilai dari $Wi^2$ merupakan hasil dari pembobotan yang dikuadratkan

$W1^2 = 0$

$W2^2 = 0{,}0906$

$W3^2 = 0$

$W4^2 = 0{,}0906$

b) Menghitung nilai *vector space model* dari query berdasarkan persamaan 2.4 adalah:

Q x W1 = 0

Q x W2 = 0,0906

Q x W3 = 0

Q x W4 = 0,0906

Perhitungan menggunakan metode *vector space model* lebih lengkap dapat dilihat pada Tabel 3.8.

**Tabel 3.8 Perhitungan *Vector Space Model***

| NO | Kata-kata | Q(Wiq) | TF-IDF | | | | VECTOR SPACE MODEL | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | W1 | W2 | W3 | W4 | $Q.W_1$ | $Q.W_2$ | $Q.W_3$ | $Q.W_4$ |
| 1 | Problema | 0 | 0,3625 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | Plagiarisme | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | Rupa | 0 | 0,0906 | 0 | 0 | 0,0906 | 0 | 0 | 0 | 0 |
| 4 | Salah | 0 | 0,3625 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | Satu | 0 | 0,3625 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | Soal | 0 | 0,3625 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | Klasik | 0 | 0,3625 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | Dunia | 0 | 0,3625 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 9 | Didik | 0 | 0,3625 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10 | Sekarang | 0 | 0,3625 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 11 | Tindak | 0 | 0 | 0,0906 | 0,0906 | 0 | 0 | 0 | 0 | 0 |
| 12 | Buat | 0 | 0 | 0,0906 | 0 | 0,0906 | 0 | 0 | 0 | 0 |
| 13 | Orang | 0 | 0 | 0,3625 | 0 | 0 | 0 | 0 | 0 | 0 |
| 14 | Jadi | 0 | 0 | 0,3625 | 0 | 0 | 0 | 0 | 0 | 0 |
| 15 | Malas | 0 | 0 | 0,3625 | 0 | 0 | 0 | 0 | 0 | 0 |
| 16 | Kreatif | 0 | 0 | 0,3625 | 0 | 0 | 0 | 0 | 0 | 0 |
| 17 | Mau | 0 | 0 | 0,3625 | 0 | 0 | 0 | 0 | 0 | 0 |
| 18 | Fikir | 0 | 0 | 0,3625 | 0 | 0 | 0 | 0 | 0 | 0 |
| 19 | Kecil | 0 | 0 | 0 | 0,3625 | 0 | 0 | 0 | 0 | 0 |
| 20 | Sistem | 0 | 0 | 0 | 0,3625 | 0 | 0 | 0 | 0 | 0 |
| 21 | Mampu | 0 | 0 | 0 | 0,3625 | 0 | 0 | 0 | 0 | 0 |
| 22 | Deteksi | 1 | 0 | 0,0906 | 0,0906 | 0 | 0 | 0,0906 | 0,0906 | 0 |
| 23 | Ubah | 0 | 0 | 0 | 0,3625 | 0 | 0 | 0 | 0 | 0 |
| 24 | Posisi | 0 | 0 | 0 | 0,3625 | 0 | 0 | 0 | 0 | 0 |
| 25 | Kalimat | 0 | 0 | 0 | 0,0906 | 0 | 0 | 0 | 0 | 0 |
| 26 | Teliti | 0 | 0 | 0 | 0 | 0,3625 | 0 | 0 | 0 | 0 |
| 27 | Buat | 0 | 0 | 0 | 0 | 0,3625 | 0 | 0 | 0 | 0 |
| 28 | Suatu | 0 | 0 | 0 | 0 | 0,3625 | 0 | 0 | 0 | 0 |
| 29 | Dokumen | 1 | 0 | 0 | 0 | 0,3625 | 0 | 0 | 0,3625 | 0 |
| 30 | Paper | 0 | 0 | 0 | 0 | 0,3625 | 0 | 0 | 0 | 0 |
| 31 | Guna | 1 | 0 | 0 | 0 | 0,3625 | 0 | 0 | 0 | 0,3625 |
| 32 | Metode | 1 | 0 | 0 | 0 | 0,3625 | 0 | 0 | 0 | 0,3625 |
| | | Sqrt($\sum_i w^2 q,j$) | Sqrt ($\sum_i w^2 i,j$) | | | | $\sum_i w_{Q,j} \times w_{i,j}$ | | | |
| | | 2,24 | 1.73 | 1,56 | 1.44 | 1.65 | 0 | 0.0906 | 0,4531 | 0,725 |

#### 3.2.3.4 Proses Perhitungan Cosine Similarity

Selanjutnya menghitung nilai kemiripan *similarity* dengan menggunakan nilai *cosine* antara *vector* kata kunci pada setiap dokumen berdasarkan persamaan 2.4 sebagai berikut.

1. Dokumen Latih ke-1

   Q.W1 = jumlah nilai perkalian antara query dengan bobot dokumen 1

   |Q| = nilai akar dari jumlah query , |W1| = nilai akar dari jumlah bobot dokumen 1

Similaritas (Q,W1)= $\frac{Q.W1}{(|Q|.|W1|)}$

$= \frac{0}{(2{,}24 \times 1{,}73)}$

$= \frac{0}{3{,}875}$

$= 0$

2. Dokumen Latih ke-2

   Q.W2 = jumlah nilai perkalian antara query dengan bobot dokumen 2

   |Q| = nilai akar dari jumlah query , |W2| = nilai akar dari jumlah bobot dokumen 2

Similaritas (Q,W2)= $\frac{Q.W2}{(|Q|.|W2|)}$

$= \frac{0{,}0906}{(2{,}24 \times 1{,}56)}$

$= \frac{0{,}0906}{3{,}494}$

$= 0{,}0259$

3. Dokumen Latih ke-3

   Q.W3 = jumlah nilai perkalian antara query dengan bobot dokumen 3
   |Q| = nilai akar dari jumlah query , |W3| = nilai akar dari jumlah bobot dokumen 3

Similaritas (Q,W3)= $\frac{Q.W3}{(|Q|.|W3|)}$

$= \frac{0,4531}{(2,24 \times 1,44)}$

$= \frac{0,4531}{3,226}$

= 0,1405

4. Dokumen Latih ke-4

   Q.W4 = jumlah nilai perkalian antara query dengan bobot dokumen 4
   |Q| = nilai akar dari jumlah query , |W4| = nilai akar dari jumlah bobot dokumen 4

Similaritas (Q,W4)= $\frac{Q.W4}{(|Q|.|W4|)}$

$= \frac{0,725}{(2,24 \times 1,65)}$

$= \frac{0,725}{3,696}$

= 0,1962

### 3.2.4 Keluaran

Keluaran dalam aplikasi pendeteksi plagiarisme ini berupa status persentase plagiarisme. Sesuai perhitungan di atas maka nilai *cosinus* setiap dokumen telah didapat, seperti tabel di bawah ini.

**Tabel 3.9 Tabel Perhitungan *Similarity***

| | **D1** | **D2** | **D3** | **D4** |
|---|---|---|---|---|
| | **Similaritas x 100 %** | | | |
| **Nilai Persentase *Similarity*** | 0 x 100% = 0 % | 0,0259 x 100% = 2,59 % | 0,1405 x 100% = 14,05 % | 0,1962x 100% = 19,62 % |
| Ranking | **4** | **3** | **2** | **1** |

Setelah diurutkan menjadi:

| | **D4** | **D3** | **D2** | **D1** |
|---|---|---|---|---|
| | **Similaritas x 100 %** | | | |
| **Nilai Persentase *Similarity*** | 0,1962x 100% = 19,62 % | 0,1405 x 100% = 14,05 % | 0,0259 x 100% = 2,59 % | 0 x 100% = 0 % |
| Ranking | **1** | **2** | **3** | **4** |

Berdasarkan hasil perhitungan *vector space model* dan nilai *similarity* diatas dapat diketahui bahwa solusi yang terdapat pada D4 memiliki tingkat similaritas tertinggi terhadap dokumen uji (DU) yaitu 19,62%, disusul dengan D3 yaitu 14,05%, dan D2 yaitu 2,59%.

## 3.3 Analisis Pengguna

Adapun kriteria pengguna yang dapat menggunakan atau menjalankan aplikasi pendeteksi plagiarisme yang dibangun ini adalah sebagai berikut.

**Tabel 3.10 Deskripsi Pengguna**

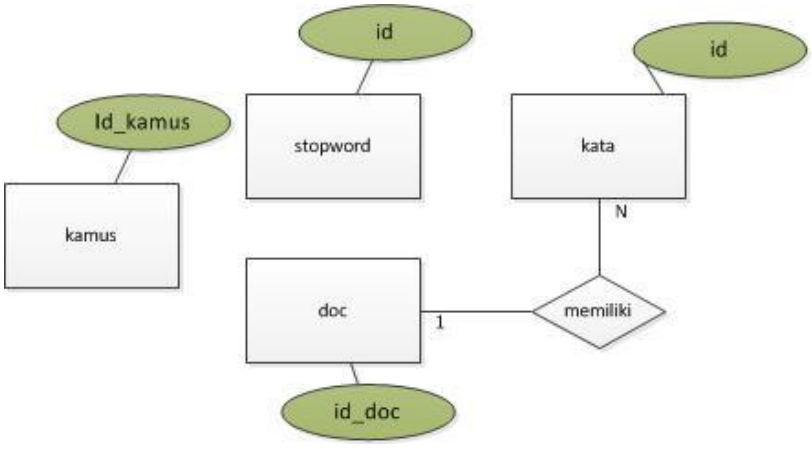
| No | Aktor | Deskripsi |
|---|---|---|
| 1 | User(Pengguna) | Merupakan aktor yang berperan melakukan input dokumen dalam bentuk *.pdf dan melihat hasil persentase similaritas dokumen, dan serta melakukan pengelolaan data pelatihan dan stopword. |

## 3.4 Perancangan Sistem

Perancangan sistem ini bertujuan untuk menggambarkan bagaimana sistem yang akan dibangun sebagai bentuk penyempurnaan dari sistem yang sedang berjalan, maka dibangunlah aplikasi plagiarisme yang dapat membantu pengguna dalam mencari solusi yang tepat. Tahapan pemodelan untuk pembuatan sistem pada penelitian ini menggunakan diagram perancangan yang terdapat pada Pemodelan Analisis Terstruktur.

### 3.4.1 Entity Relationship Diagram (ERD)

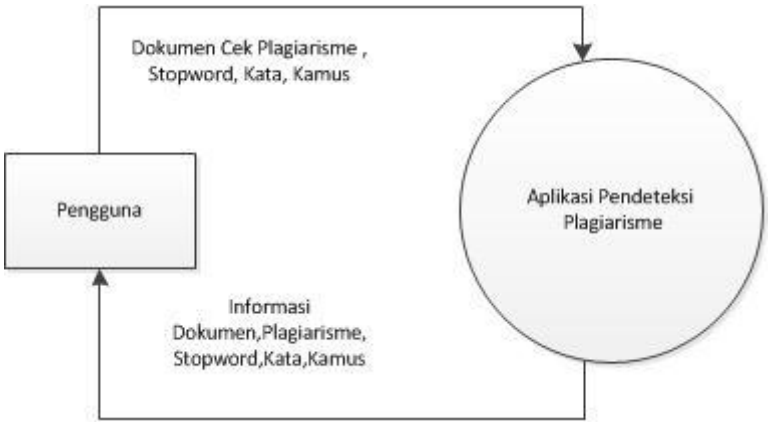
ERD merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. ERD untuk memodelkan struktur data dan hubungan antar data, untuk menggambarkannya digunakan beberapa notasi dan simbol.




**Gambar 3.2 Entity Relationship Diagram**

### 3.4.2 Diagram Konteks

Diagram yang terdiri dari suatu proses saja dan biasa diberi nomor proses 0. Proses ini mewakili dari dari seluruh sistem. Diagram konteks menggambarkan input atau output suatu sistem dengan dunia luar atau dunia kesatuan luar. Proses yang terjadi dalam aplikasi pendeteksian plagiarisme ini dapat dilihat pada gambar berikut.

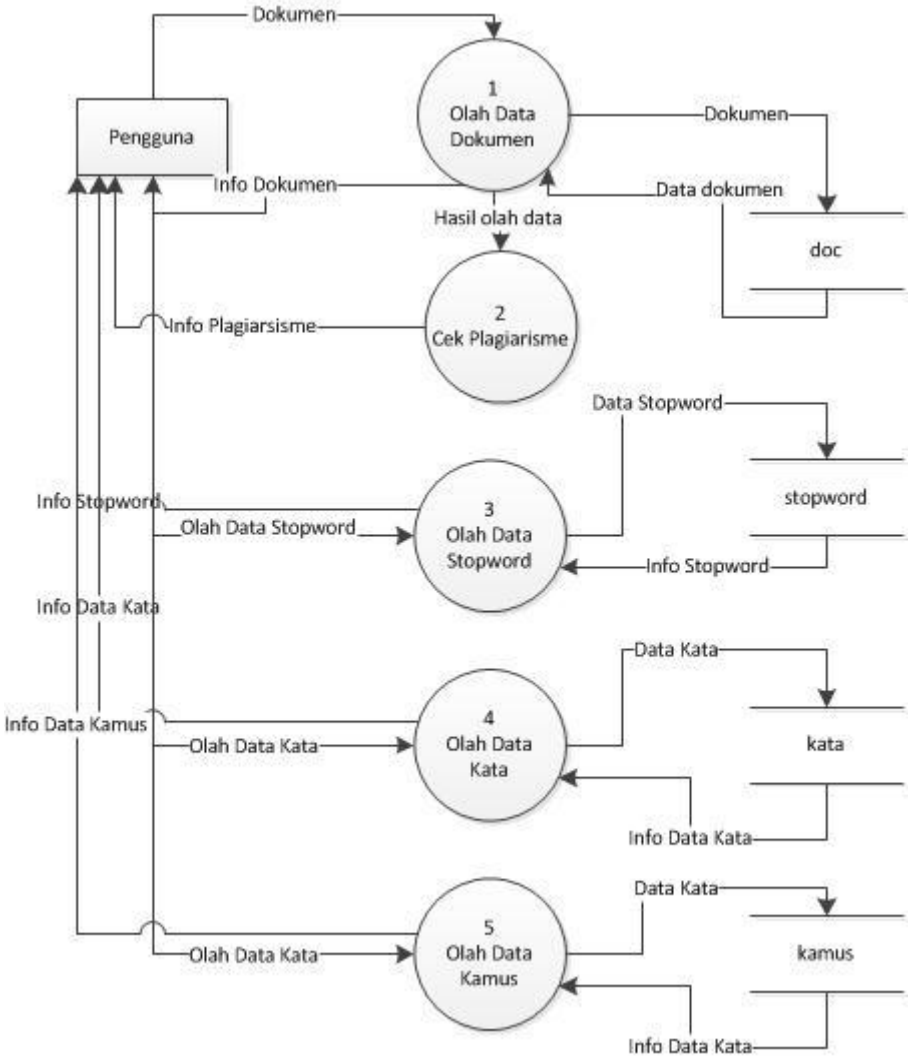



**Gambar 3.3 Diagram Konteks**

### 3.4.3 Data Flow Diagram (DFD)

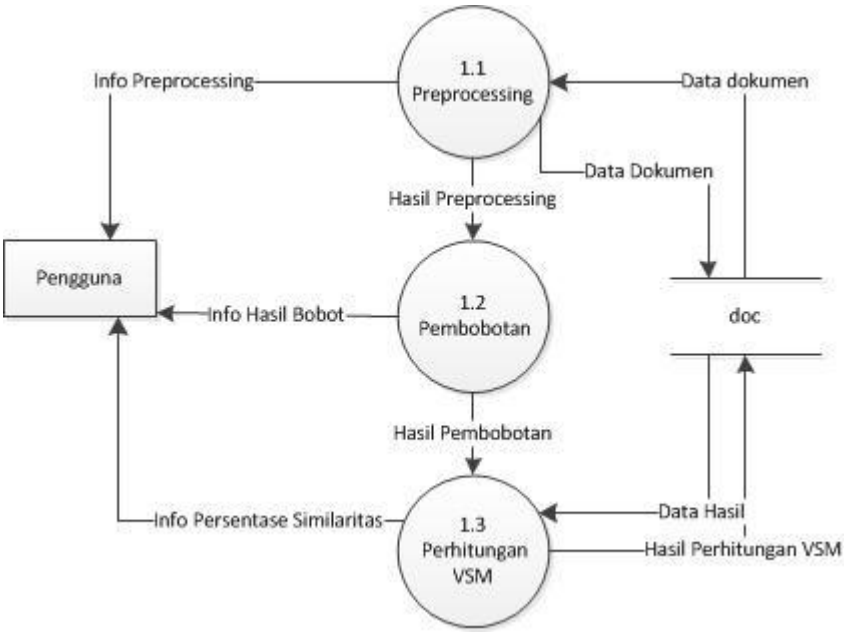
Model logika data atau proses yang dibuat untuk menggambarkan darimana asal data, kemana tujuan data yang keluar dari sistem, dimana data disimpan, proses yang dihasilkan data tersebut dan interaksi antara data yang tersimpan dan proses yang dikenakan pada data tersebut. Berikut merupakan alur dari aplikasi pendeteksian plagiarisme yang dapat dilihat pada gambar berikut.

### 3.4.3.1 DFD level 1 Pengolahan Data



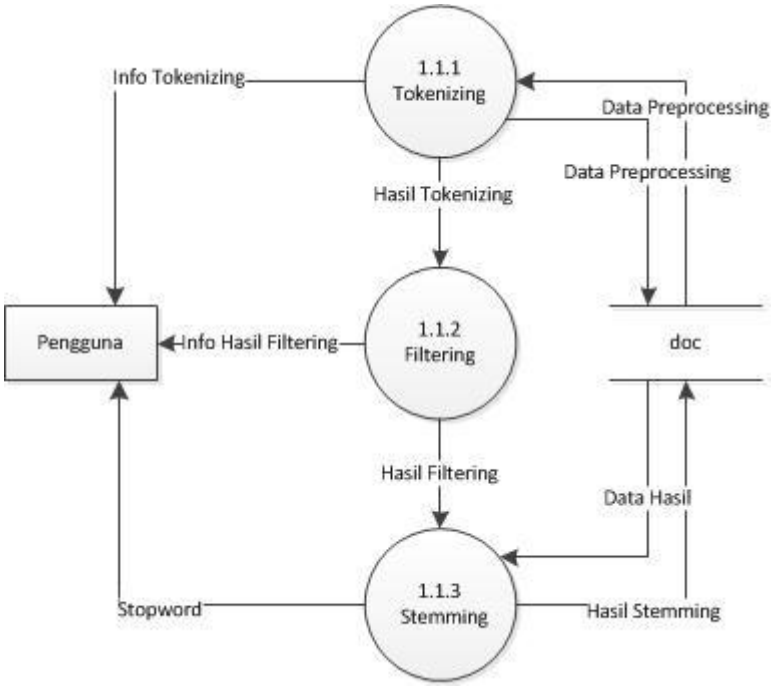

**Gambar 3.4 DFD level 1**

### 3.4.3.2 DFD level 2 Proses




**Gambar 3.5 DFD level 2**

### 3.4.3.3 DFD level 3 Preprocessing Dokumen




**Gambar 3.6 DFD level 3**

### 3.4.4 Kamus Data

Kamus data merupakan Katalog Fakta tentang data dan kebutuhan informasi dari suatu sistem dengan mendefinisikan data yang mengalir pada sistem secara lengkap. Dan berikut merupakan kamus data dari data yang mengalir pada sistem.

#### 3.4.4.1 Kamus Data Data Dokumen

**Tabel 3.11 Kamus Data Data Dokumen**

| | |
|---|---|
| Nama Arus Data | Data Dokumen |
| Alias | - |
| Bentuk Data | Dokumen |
| Arus Data | Proses 1 |
| Penjelasan | Data Dokumen diinputkan kedalam aplikasi |
| Periode | Setiap ada inputan |
| Volume | volume rata-rata adalah 1 dan volume puncak adalah 2 |
| Struktur Data | Id_doc = [0-9]<br>Doc = [A-Z \| a-z \| 0-9] |

#### 3.4.4.2 Kamus Data Data Stopword

**Tabel 3.12 Kamus Data Data Dokumen**

| | |
|---|---|
| Nama Arus Data | Data Stopword |
| Alias | - |
| Bentuk Data | Label |
| Arus Data | Proses 4 |
| Penjelasan | Admin memasukkan kata tambahan pada stopword |
| Periode | Saat login aplikasi dan mengolah stopword |
| Volume | Volume rata-rata adalah 1 dan volume puncak adalah 2 |
| Struktur Data | Id = [0-9]<br>Kata = [A-Z \| a-z] |

#### 3.4.4.3 Kamus Data Data Kata

**Tabel 3.13 Kamus Data Data Kata**

| | |
|---|---|
| Nama Arus Data | Data Kata |
| Alias | - |
| Bentuk Data | Kata |
| Arus Data | Proses 6 |
| Penjelasan | Kata kunci yang terdapat pada dokumen uji |
| Periode | Saat login admin dari aplikasi |
| Volume | Volume rata-rata adalah 1 dan volume puncak adalah 2 |
| Struktur Data | Id = [0-9]<br>Id_doc = [0-9]<br>Kata = [A-Z \| a-z] |

### 3.4.5 Spesifikasi Proses

Spesifikasi proses merupakan gambaran deskripsi dan spesifikasi dari setiap proses pada pemodelan DFD sesuai kebutuhan sistem.

**Tabel 3.14 Spesifikasi Proses**

| No | Proses | Keterangan |
|---|---|---|
| 1 | No. Proses | 1 |
| | Nama Proses | Olah Data Dokumen |
| | Sumber | - |
| | *Masukan* | Data Dokumen |
| | *Keluaran* | Info Dokumen |
| | Tujuan | Pengguna |
| | Logika Proses | ***Begin***<br>- Pengguna memasukkan dokumen.<br>- Jika data *valid* maka data akan proses.<br>-Menampilkan info persentase plagiat dokumen, info tabel perhitungan.<br>***End.*** |

| No | Proses | Keterangan |
|---|---|---|
| 2 | No. Proses | 2 |
| | Nama Proses | Cek Plagiarisme |
| | Sumber | - |
| | *Masukan* | Hasil olah data |
| | *Keluaran* | Info plagiarisme |
| | Tujuan | Pengguna |
| | Logika Proses | ***Begin***<br>- Sistem mengambil hasil olah data dokumen.<br>- Jika data *valid*, data akan diproses.<br>- Menampilkan info plagiarisme.<br>***End.*** |
| **No** | **Proses** | **Keterangan** |
| 3 | No. Proses | 3 |
| | Nama Proses | Olah Data Stopword |
| | Sumber | - |
| | *Masukan* | Data Stopword |
| | *Keluaran* | Info Data Stopword |
| | Tujuan | Pengguna |
| | Logika Proses | ***Begin***<br>- Sistem menampilkan fitur Halaman depan.<br>- Pengguna memilih fitur stopword.<br>- Masukan data yang sesuai.<br>- Data disimpan dalam data *store.*<br>- Menampilkan info data stopword.<br>***End.*** |
| **No** | **Proses** | **Keterangan** |
| 4 | No. Proses | 4 |
| | Nama Proses | Olah Data Kata |
| | *Source* (Sumber) | - |

| | | |
|---|---|---|
| | *Input* | Data Kata |
| | *Output* | Info Data Kata |
| | *Destination* (Tujuan) | Pengguna |
| | Logika Proses | ***Begin***<br>- Sistem menampilkan fitur Halaman depan.<br>- Pengguna memilih fitur Pelatihan.<br>- Masukan dokumen yang sudah terintegrasi.<br>- Jika *valid* maka sistem akan menyimpan data dalam data store<br>- Menampilkan info data kata.<br>***End.*** |
| **No** | **Proses** | **Keterangan** |
| 5 | No. Proses | 5 |
| | Nama Proses | Olah Data Kamus |
| | Sumber | - |
| | *Masukan* | Data Kamus |
| | *Keluaran* | Info Data Kamus |
| | Tujuan | Pengguna |
| | Logika Proses | ***Begin***<br>- Sistem menampilkan fitur Halaman depan.<br>- Pengguna memilih fitur kamus.<br>- Masukan data yang sesuai.<br>- Data disimpan dalam data *store.*<br>- Menampilkan info data kamus.<br>***End.*** |
| **No** | **Proses** | **Keterangan** |
| 6 | No. Proses | 1.1 |
| | Nama Proses | Preprocessing |
| | *Source* (Sumber) | Olah Data Dokumen |
| | *Input* | Data Dokumen |

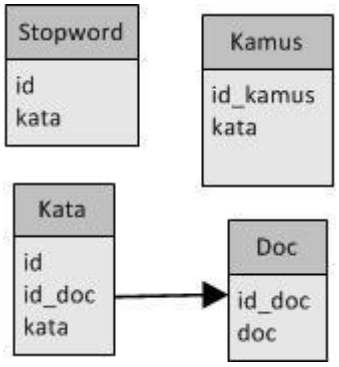
| | | |
|---|---|---|
| | *Output* | Info Preprocessing |
| | *Destination* (Tujuan) | Pengguna |
| | Logika Proses | ***Begin***<br>- Sistem mengolah data dokumen yang sudah dimasukkan sebelumnya.<br>- Melakukan proses *preprocessing* dengan tahapan *tokenizing*, *filtering*, dan *stemming*.<br>- Jika data *valid* maka data disimpan dalam data store.<br>- Menghasilkan info data preprocessing.<br>***End.*** |
| **No** | **Proses** | **Keterangan** |
| 7 | No. Proses | 1.2 |
| | Nama Proses | Pembobotan |
| | *Source* (Sumber) | Olah Data Dokumen |
| | *Input* | Hasil dari preprocessing |
| | *Output* | Info Hasil Bobot |
| | *Destination* (Tujuan) | Pengguna |
| | Logika Proses | ***Begin***<br>- Sistem mengolah data dokumen yang sudah dimasukkan sebelumnya.<br>- Melakukan proses dari hasil *preprocessing.*<br>- Sistem melakukan perhitungan bobot.<br>- Jika data *valid* maka data disimpan dalam data store.<br>- Menghasilkan info data bobot.<br>***End.*** |
| **No** | **Proses** | **Keterangan** |
| 8 | No. Proses | 1.3 |
| | Nama Proses | Perhitungan VSM |

| | | |
|---|---|---|
| | *Source* (Sumber) | Olah Data Dokumen |
| | *Input* | Hasil dari pembobotan |
| | *Output* | Info persentase similaritas |
| | *Destination* (Tujuan) | Pengguna |
| | Logika Proses | ***Begin***<br>- Sistem mengolah data dokumen yang sudah dimasukkan sebelumnya.<br>- Melakukan proses dari hasil bobot.<br>- Sistem melakukan perhitungan *vector space model (vsm)*.<br>- Jika data *valid* maka data akan disimpan dalam data store.<br>-Menghasilkan info data persentase plagiarisme.<br>***End.*** |
| **No** | **Proses** | **Keterangan** |
| 9 | No. Proses | 1.1.1 |
| | Nama Proses | Tokenizing |
| | *Source* (Sumber) | Preprocessing |
| | *Input* | Data Dokumen |
| | *Output* | Info Tokenizing |
| | *Destination* (Tujuan) | Pengguna |
| | Logika Proses | ***Begin***<br>- Sistem mengolah data dokumen yang sudah dimasukkan sebelumnya.<br>-Melakukan proses *tokenizing*, membuat kalimat-kalimat menjadi token-token.<br>- Menghasilkan info data *tokenizing*.<br>***End.*** |
| **No** | **Proses** | **Keterangan** |
| 10 | No. Proses | 1.1.2 |

| | | |
|---|---|---|
| | Nama Proses | Filtering |
| | *Source* (Sumber) | Preprocessing |
| | *Input* | Hasil dari tokenizing |
| | *Output* | Info Hasil Filtering |
| | *Destination* (Tujuan) | Pengguna |
| | Logika Proses | ***Begin***<br>- Sistem mengolah data dokumen yang sudah dimasukkan sebelumnya.<br>-Melakukan proses *filtering* dengan menghilangkan stopword.<br>- Menghasilkan info data *filtering*.<br>***End.*** |
| **No** | **Proses** | **Keterangan** |
| 11 | No. Proses | 1.1.3 |
| | Nama Proses | Stemming |
| | *Source* (Sumber) | Preprocessing |
| | *Input* | Hasil dari filtering |
| | *Output* | Kata-kata |
| | *Destination* (Tujuan) | Pengguna |
| | Logika Proses | ***Begin***<br>- Sistem mengolah data dokumen yang sudah dimasukkan sebelumnya.<br>- Melakukan proses *stemming* dengan membuat kata-kata yang ada menjadi kata dasar.<br>- Menghasilkan info data *stemming*.<br>***End.*** |

### 3.4.6 Perancangan Basis Data

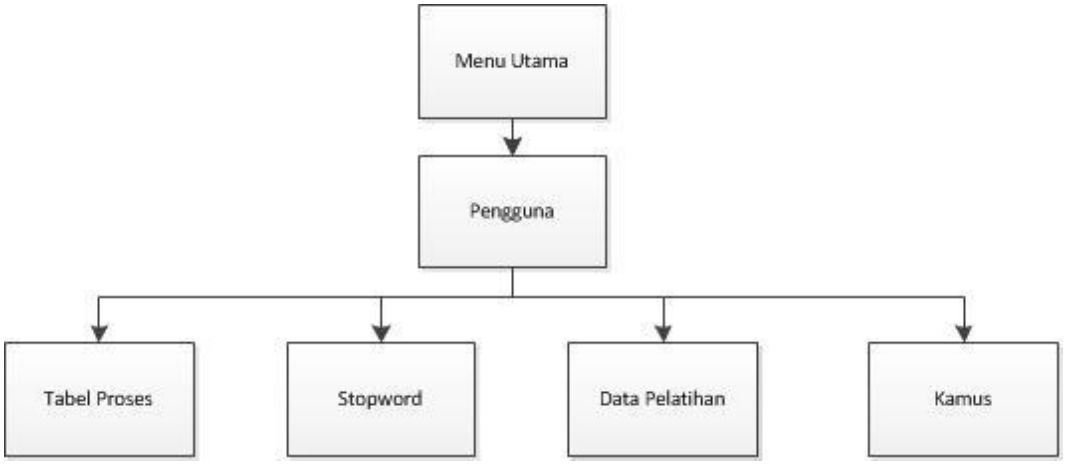
Perancangan Basis Data adalah proses untuk menentukan isi dan pengaturan data yang dibutuhkan untuk mendukung berbagai rancangan sistem.




**Gambar 3.7 Perancangan Basis Data**

### 3.4.7 Perancangan Struktur Menu

Struktur menu adalah bentuk umum dari suatu rancangan sistem untuk memudahkan pemakai dalam menjalankan sistem. Sehingga saat menjalankan sistem, pengguna tidak mengalami kesulitan dalam memilih menu yang diinginkan. Pada perancangan perangkat lunak ini dibuat menu seperti dibawah ini.

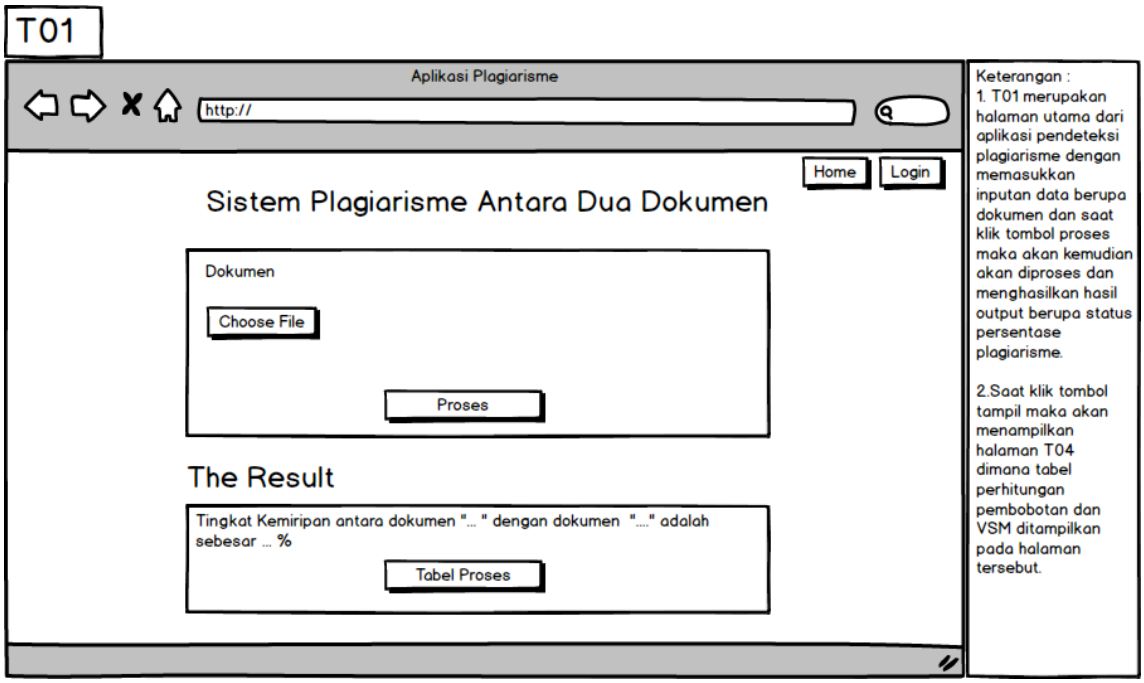



**Gambar 3.8 Struktur Menu**

### 3.4.8 Perancangan Antarmuka Sistem

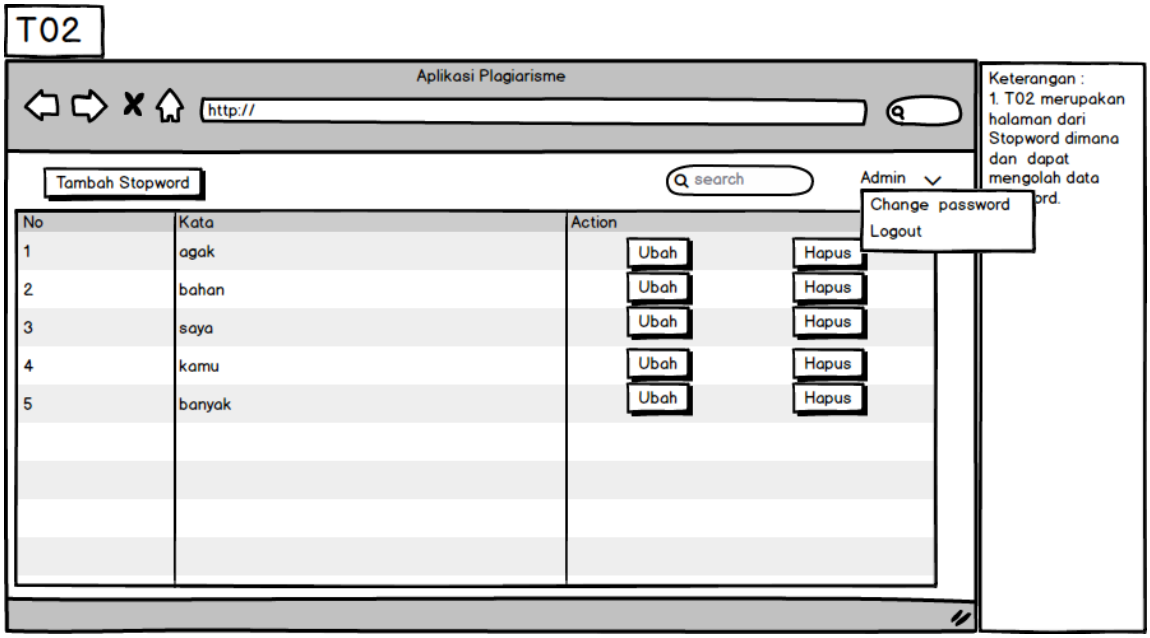
Tahap perancangan antarmuka dilakukan untuk merancang antarmuka agar dapat digunakan oleh pengguna. Berikut ini adalah tampilan antarmuka yang dirancang pada sistem.

### 3.4.8.1 Tampilan Halaman Utama



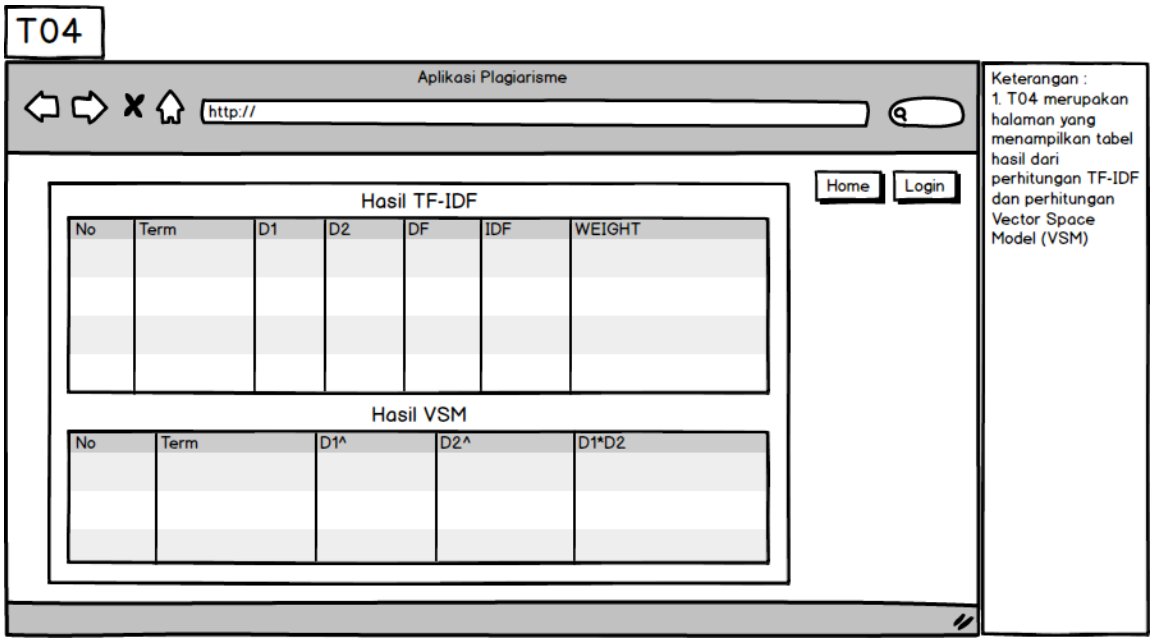

**Gambar 3.9 *Antarmuka* Halaman Utama**

### 3.4.8.2 Tampilan Halaman Stopword



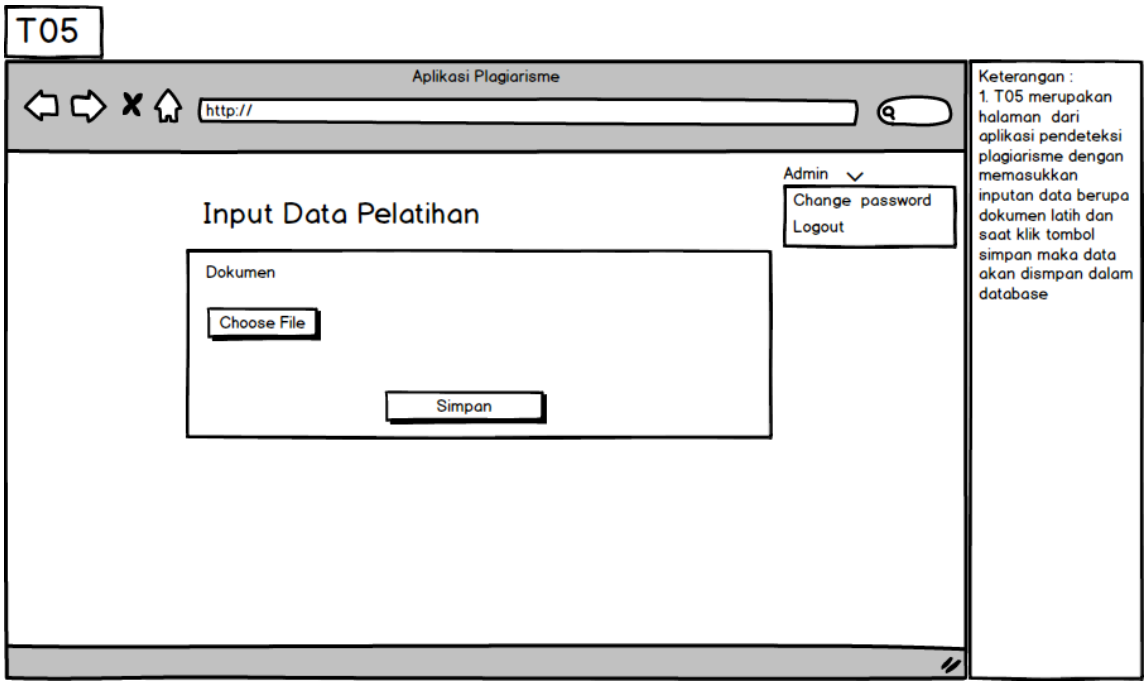

**Gambar 3.10 *Antarmuka* Halaman Stopword**

### 3.4.8.3 Tampilan Halaman Tabel Proses




**Gambar 3.11 *Antarmuka* Halaman Tabel Proses**

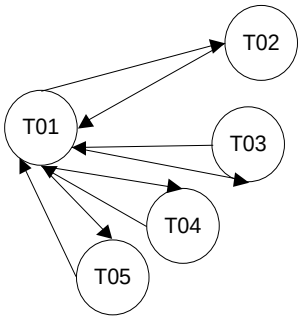
### 3.4.8.4 Tampilan Halaman Input Data Pelatihan




**Gambar 3.12 *Antarmuka* Halaman Data Pelatihan**

### 3.4.9 Perancangan Semantik

Jaringan Semantik merupakan gambaran pengetahuan grafis yang menunjukkan hubungan antar berbagai objek. Jaringan semantik terdiri dari lingkaran-lingkaran yang menunjukkan objek dan informasi tentang objek-objek tertentu. Jaringan semantik untuk sistem klasifikasi ini adalah sebagai berikut.




**Gambar 3.13 Jaringan Semantik**

**Keterangan:**

T01: Antarmuka Halaman Utama

T02: Antarmuka Stopword

T03: Antarmuka Pelatihan

T04: Antarmuka Perhitungan

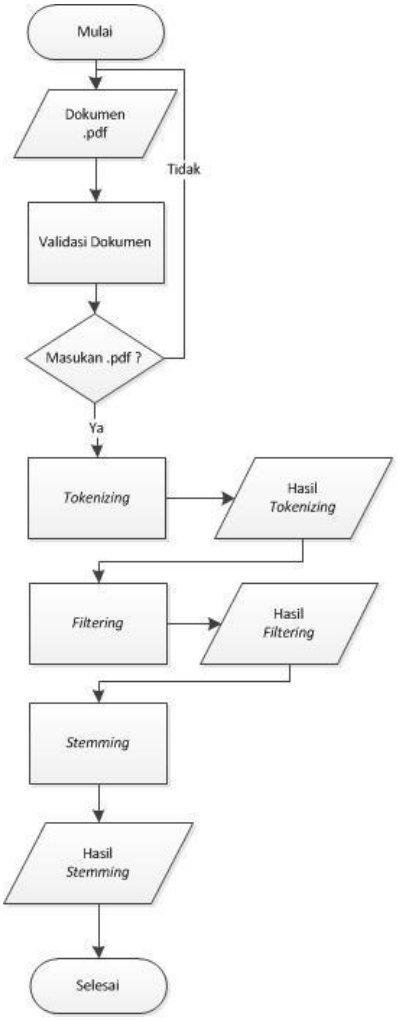
T05: Antarmuka Kamus

### 3.4.10 Perancangan Prosedural

Perancangan prosedural adalah tata cara atau urutan langkah-langkah untuk melakukan suatu proses. Prosedural ini dugunakan sebagai algoritma dasar dalam mengkodekan prosedur yang ada. Alat yang digunakan adalah *flowchart*. Membuat *flowchart* tujuannya untuk menggambarkan suatu tahapan penyelesaian masalah, secara sederhana, terurai, rapi, dan jelas menggunakan simbol-simbol standar. Adapun perancangan prosedural untuk pembangunan sistem klasifikasi yang akan dibangun adalah sebagai berikut.

#### 3.4.10.1 Prosedur Preprocessing

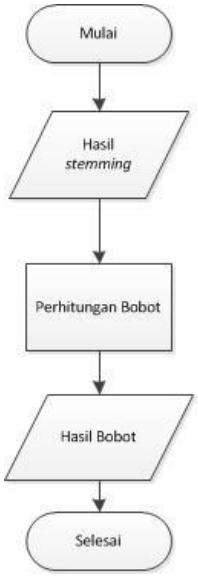
Pada prosedur ini dilakukan proses *preprocessing* dimana tahapan tersebut dilakukan oleh sistem sebelum masuk pada tahap proses utama.




**Gambar 3.14 *Flowchart* Preprocessing**

#### 3.4.10.2 Prosedur Perhitungan TF-IDF

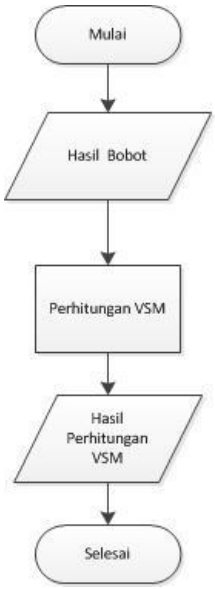
Pada prosedur ini dilakukan proses perhitungan TF-IDF dimana tahapan tersebut dilakukan oleh sistem setelah melakukan proses *preprocessing*.




**Gambar 3.15 *Flowchart* Perhitungan Bobot**

#### 3.4.10.3 Prosedur Perhitungan VSM

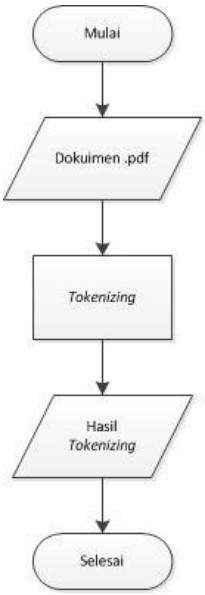
Pada prosedur ini dilakukan proses perhitungan *vsm* dimana tahapan tersebut dilakukan oleh sistem setelah melakukan proses perhitungan TF-IDF.




**Gambar 3.16 *Flowchart* Perhitungan VSM**

#### 3.4.10.4 Prosedur *Tokenizing*

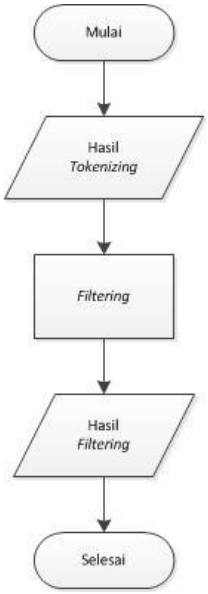
Pada prosedur ini dilakukan proses *tokenizing* dimana dalam proses ini dokumen dibuat menjadi token-token, tahapan tersebut dilakukan oleh sistem dan terjadi dalam proses *preprocessing*.




**Gambar 3.17 *Flowchart* Tokenizing**

#### 3.4.10.5 Prosedur *Filtering*

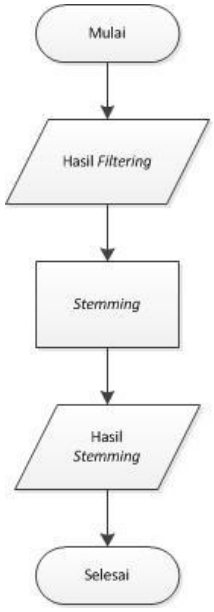
Pada prosedur ini dilakukan proses *filtering* dimana dalam proses ini token-token dihilangkan kata-kata tidak pentingnya, tahapan tersebut dilakukan oleh sistem dan terjadi dalam proses *preprocessing*.




**Gambar 3.18 *Flowchart* Filtering**

#### 3.4.10.6 Prosedur *Stemming*

Pada prosedur ini dilakukan proses *tokenizing* dimana dalam proses ini didapat kata-kata kuncinya, tahapan tersebut dilakukan oleh sistem dan terjadi dalam proses *preprocessing*.




**Gambar 3.19 *Flowchart* Stemming**