

## **BAB II TINJAUAN PUSTAKA**

### **2.1 Implementasi**

Implementasi sebagai kegiatan mendistribusikan keluaran dari suatu kebijakan yang dijalankan oleh seorang pelaksana (untuk menyampaikan keluaran kebijakan) kepada suatu kelompok sasaran dalam upaya mencapai kebijakan tersebut [3].

### **2.2 Algoritma**

Algoritma adalah sekumpulan instruksi atau langkah-langkah yang dituliskan secara sistematis dan digunakan untuk menyelesaikan suatu persoalan atau suatu permasalahan matematika dan logika dengan bantuan komputer [4].

Algoritma sebagai suatu upaya dengan urutan operasi yang disusun secara logis dan sistematis untuk menyelesaikan suatu masalah untuk menghasilkan suatu output tertentu [5].

Dari beberapa pengertian sebelumnya, dapat disimpulkan bahwa algoritma adalah serangkaian langkah-langkah yang terdefinisi dengan baik yang digunakan untuk menyelesaikan suatu masalah. Algoritma haruslah jelas, terstruktur, dan terdefinisi dengan baik agar dapat dimengerti dan diterapkan dengan benar.

### **2.3 Deteksi**

Kata “deteksi” berasal dari bahasa latin *”Detectio”* yang berarti penemuan atau pengungkapan, dalam Bahasa Indonesia, deteksi merupakan penemuan, pengenalan, atau penentuan terhadap suatu objek [6]

## 2.4 Aplikasi Android

Aplikasi Android adalah satu unit perangkat lunak yang dibuat untuk melayani kebutuhan akan beberapa aktivitas seperti sistem perniagaan, game, pelayanan masyarakat, periklanan, atau semua proses yang hampir dilakukan manusia [7], sedangkan menurut Nazaruddin, Aplikasi Android adalah perangkat lunak yang berjalan pada platform Android. Platform Android adalah sistem operasi mobile yang dikembangkan oleh Google dan bersifat open source.

Aplikasi Android adalah program komputer yang dirancang untuk dijalankan pada perangkat mobile yang menggunakan sistem operasi Android. Aplikasi Android dapat digunakan untuk berbagai macam keperluan, seperti komunikasi, hiburan, edukasi, dan bisnis[8]. Sedangkan menurut Sibero Aplikasi Android adalah program yang dibuat dengan menggunakan bahasa pemrograman Java dan SDK (Software Development Kit) Android. Aplikasi Android dapat diunduh dan dipasang pada perangkat Android dari Google Play Store atau sumber pihak ketiga lainnya.

Sehingga dapat disimpulkan bahwa aplikasi Android adalah program komputer yang dirancang untuk dijalankan pada perangkat mobile yang menggunakan sistem operasi Android. Aplikasi Android dapat digunakan untuk berbagai macam keperluan, dan dapat diunduh dan dipasang pada perangkat Android dari berbagai sumber.

Beberapa ciri – ciri aplikasi android diantaranya:

1. Dibuat dengan menggunakan bahasa pemrograman Java dan SDK Android.
2. Dapat diunduh dan dipasang pada perangkat Android dari Google Play Store atau sumber pihak ketiga lainnya.
3. Berjalan pada sistem operasi Android.
4. Digunakan untuk berbagai macam keperluan, seperti komunikasi, hiburan, edukasi, dan bisnis.

## 2.5 Algoritma *TF-IDF*

Algoritma *Term Frequency–Inverse Document Frequency* merupakan metode yang digunakan untuk menghitung bobot atau nilai setiap kata kunci dalam sebuah dokumen. Bobot atau nilai ini mencerminkan seberapa penting kata kunci dalam dokumen dan seberapa jarang kata kunci muncul di dalam dokumen.

Algoritma TF-IDF banyak digunakan dalam berbagai aplikasi pengolahan teks, seperti:

- Pencarian informasi: Algoritma ini membantu mesin pencari menemukan dokumen yang relevan dengan kueri pengguna.
- Klasifikasi teks: Algoritma ini membantu mengklasifikasikan dokumen ke dalam kategori tertentu.
- Ringkasan teks: Algoritma ini membantu mengidentifikasi kata kunci penting dalam dokumen dan membuat ringkasan yang informatif.
- Analisis sentimen: Algoritma ini membantu menentukan opini atau perasaan yang terkandung dalam sebuah teks.

### 2.5.1 Term Frequency

*Term Frequency* merupakan metode yang bekerja dengan cara mengukur seberapa sering kata kunci muncul dalam dokumen. Semakin sering kata kunci muncul, semakin tinggi nilai TF-nya.

$$tf(t, d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}}$$

Keterangan:

$tf(t, d)$  = Nilai kata yang sering muncul dalam kumpulan dokumen.

$t$  = *term*

$d$  = dokumen

### 2.5.2 Inverse Document Frequency

Sedangkan *Inverse Document Frequency*, Mengukur seberapa jarang kata kunci muncul di seluruh dokumen dalam kumpulan data. Semakin jarang kata kunci muncul, semakin tinggi nilai IDF-nya.

$$idf(t, D) = \log \frac{N}{|\{d:d \in D \text{ and } t \in d\}|}$$

Keterangan:

$idf(t, D)$  = Nilai kata yang jarang muncul dalam kumpulan dokumen.

$t$  = *term*

$N$  = Total nilai dokumen dalam sebuah list.

$d$  = Dokumen

$D$  = Total dokumen

Nilai *TF-IDF* dihitung dengan mengalikan nilai *TF* dan nilai *IDF*, sehingga rumusnya adalah:

$$tfidf(t, d, D) = tf(t, d) * idf(t, D)$$

Keterangan:

$tf(t, d)$  = Total nilai kata yang sering muncul dalam kumpulan dokumen

$idf(t, D)$  = Nilai kata yang jarang muncul dalam kumpulan dokumen.

$t$  = *term*

$d$  = Dokumen

$D$  = Total dokumen

### 2.5.3 Langkah Langkah Algoritma *TF – IDF*

Untuk menyusun langkah-langkah algoritma *TF-IDF* sebagai berikut beserta contoh dokumennya:

1. Dokumen 1: "Mobil berwarna merah itu melaju kencang di jalan tol."
2. Dokumen 2: "Sepeda motor adalah kendaraan roda dua yang banyak digunakan di Indonesia."

Selanjutnya hitung nilai *TF-IDF* untuk kata kunci "mobil" pada kedua dokumen:

1. Dokumen 1:

TF: Kata kunci "mobil" muncul 1 kali.

IDF: Asumsikan kata kunci "mobil" muncul dalam 10% dokumen dalam kumpulan data. Maka,  $IDF = \log(N/n)$ , di mana N adalah jumlah total dokumen dan n adalah jumlah dokumen yang mengandung kata kunci "mobil". Dalam kasus ini,  $IDF = \log(10/1) = 2.3025$ .

$$TF-IDF = 1 * 2.3025 = 2.3025$$

2. Dokumen 2:

TF: Kata kunci "mobil" tidak muncul.

IDF: IDF sama seperti pada Dokumen 1.

$$TF-IDF = 0 * 2.3025 = 0$$

Dari contoh ini, dapat terlihat bahwa nilai TF-IDF untuk kata kunci "mobil" pada Dokumen 1 lebih tinggi daripada Dokumen 2. Hal ini menunjukkan bahwa kata kunci "mobil" lebih penting bagi Dokumen 1 dibandingkan Dokumen 2.

## 2.6 Algoritma Fuzzy Matching

Algoritma Fuzzy Matching adalah teknik pencocokan string yang mempertimbangkan kemiripan antara dua string, bukan hanya persamaan kedua string. Hal ini memungkinkan algoritma untuk menemukan kecocokan bahkan ketika terdapat perbedaan kecil dalam ejaan, tata bahasa, atau struktur kalimat.

Algoritma Fuzzy Matching bekerja dengan menghitung jarak antara dua string. Jarak ini mewakili jumlah operasi yang diperlukan untuk mengubah satu string menjadi string lainnya. Operasi ini dapat berupa:

1. Penyisipan: Menambahkan karakter ke string
2. Penghapusan: Menghapus karakter dari string
3. Penggantian: Mengubah karakter dalam string

Setelah menghitung jarak string, algoritma Fuzzy Matching akan menentukan apakah dua string cukup mirip untuk dianggap sebagai kecocokan. Nilai ambang batas kesamaan biasanya digunakan untuk menentukan hal ini.

### **2.6.1 Cara Kerja Algoritma Fuzzy Matching**

Berikut adalah contoh bagaimana algoritma Fuzzy Matching dapat digunakan untuk menemukan kecocokan antara dua string:

String 1: "Sepatu lari"

String 2: "Sepatu lari merah"

Algoritma Levenshtein Distance akan menghitung jarak antara dua string ini sebagai satu. Hal ini karena hanya perlu satu operasi (penghapusan kata "merah") untuk mengubah String 1 menjadi String 2. Misalkan nilai ambang batas kesamaan untuk algoritma Fuzzy Matching adalah 0.8. Dalam hal ini, algoritma akan menganggap String 1 dan String 2 sebagai kecocokan karena jaraknya (0.8) lebih besar dari nilai ambang batas.

### **2.7 UML (Unified Modeling Language)**

UML adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek [9].

### **2.8 Database Firebase Firestore**

*Firebase Firestore* adalah sebuah database *No-SQL serverless database* yang dikembangkan oleh *Google* yang di desain untuk menyederhanakan website dan pengembangan aplikasi *mobile* [10].

### **2.9 Pengujian Black Box**

Pengujian *Black Box* merupakan pengujian perangkat lunak dimana pengujian berasal dari persyaratan fungsional tanpa menguji desain dan kode program. Pengujian dilakukan untuk mengetahui setiap fungsi, masukan, dan keluaran dari program sesuai dengan persyaratan yang dibutuhkan. Pengujian *black box* dilakukan dengan membuat kasus – kasus yang bersifat menguji semua fungsi dengan alat uji apakah sesuai dengan spesifikasi yang dibutuhkan.

## 2.10 Hasil Penelitian Sebelumnya

Berikut ini adalah beberapa literatur yang digunakan dalam penelitian, dapat dilihat pada Tabel 2.1 :

**Tabel 2.1** Tinjauan Pustaka Jurnal

No	Nama, Tahun	Judul	Tujuan Penelitian	Hasil Penelitian
1	Herlina Sari, 2021	Penerapan Algoritma Text Mining Dan TF-IDF Untuk Pengelompokan Topik Skripsi Pada Aplikasi Repository STMIK Budi Darma	Tujuan dari penelitian yang dibahas di halaman ini adalah untuk mengembangkan aplikasi repository yang dapat mengelompokkan topik skripsi secara otomatis menggunakan algoritma Text Mining dan TF-IDF.	Studi ini menggali bagaimana teknik text mining dan algoritma TF-IDF dapat digunakan untuk mengklasifikasikan dan mengelompokkan topik skripsi yang ada dalam repositori digital.
2	Ni Putu Anggi Yuliani, 2017	Sistem Temu Balik Dokumen Teks Menggunakan Metodeboolean dan <i>Term Wight Tf.Idf</i>	Tujuan penelitian dari jurnal tersebut adalah untuk mengembangkan Sistem Temu Balik Dokumen Teks yang efisien menggunakan metode Boolean dan Term Weight TF.IDF.	Metode Boolean: Penelitian ini menggunakan metode Boolean untuk memproses pencarian dokumen teks. Term Weight TF.IDF: Penelitian ini juga mengimplementasikan pembobotan term menggunakan TF.IDF untuk meningkatkan akurasi hasil temu balik. Efisiensi Sistem: Hasil penelitian menunjukkan

				peningkatan efisiensi dalam sistem temu balik dokumen teks. Implementasi: Penelitian ini memberikan wawasan tentang implementasi sistem yang diusulkan dalam praktik nyata.
3	Nilawati	Penggunaan Metode Cosine Similarity Dan Tf-Idf Untuk Klasifikasi Judul Seminar Proposal Pada Fakultas Teknik Universitas Jabal Ghafur	Tujuan penelitian ini untuk mengklasifikasikan judul seminar proposal pada Fakultas Teknik Universitas Jabal Ghafur dengan metode Cosine Similarity dan TF-IDF, berfokus pada pengembangan sistem secara otomatis mengkategorikan judul berdasarkan kesamaan konten, sehingga membantu pengelolaan pengetahuan dan organisasi informasi seminar.	Penelitian ini menghasilkan sebuah sistem klasifikasi judul seminar proposal yang efektif. Sistem ini menggunakan metode Cosine Similarity dan TF-IDF (Term Frequency-Inverse Document Frequency) untuk mengukur kesamaan antara dokumen teks.
4	Dhebys Suryani Hormansyah, 2018	Penerapan Metode Tf-Idf Dan N-Gram Pada Pengembangan Aplikasi Chatbot Berbasis Line Untuk Layanan Publik Kesehatan	Tujuan dari penelitian ini adalah untuk mengembangkan aplikasi chatbot berbasis LINE yang menggunakan metode TF-IDF	Penelitian tersebut mengembangkan aplikasi chatbot berbasis LINE yang menggunakan metode TF-IDF dan N-gram untuk meningkatkan layanan publik

		Di Kota Malang	dan N-GRAM untuk meningkatkan layanan publik kesehatan di Kota Malang.	kesehatan di Kota Malang.
5	Kairunnisa, 2019	Penerapan Fuzzy String Matching Pada Aplikasi Pencarian Judul Skripsi Teknik Informatika Berbasis Android	Tujuan dari penelitian ini adalah untuk mengembangkan sebuah aplikasi repository berbasis Android yang memanfaatkan algoritma Fuzzy String Matching	Aplikasi ini dirancang untuk membantu mahasiswa dalam mencari referensi skripsi dengan cepat dan efisien. Dengan aplikasi ini, pengguna dapat melakukan pencarian judul skripsi pada STMIK Budi Darma dan mendapatkan saran kata yang relevan,