

## PENERAPAN ALGORITMA TF-IDF DAN *COSINE SIMILARITY* DALAM SISTEM REKOMENDASI LOWONGAN PEKERJAAN

Venny Meida Hersianty, Eka Larasati Amalia, Dwi Puspitasari, Dimas Wahyu Wibowo

Teknik Informatika, Politeknik Negeri Malang  
Jalan Soekarno Hatta No.09, Malang, Indonesia  
vnymeida@gmail.com

### ABSTRAK

Penelitian ini berfokus pada pengembangan suatu sistem rekomendasi lowongan pekerjaan yang lebih relevan dan akurat. Dalam upaya mengatasi tantangan pencari kerja dalam menemukan peluang karir yang sesuai, sistem ini bekerja dengan cara menganalisis secara mendalam deskripsi lowongan kerja dan profil pencari kerja kemudian menyajikan rekomendasi yang paling relevan berdasarkan kesamaan antar keduanya. Sistem ini memanfaatkan algoritma TF – IDF untuk memberikan bobot terhadap kata – kata kunci yang memiliki frekuensi kemunculan tinggi namun jarang ditemukan pada dokumen lain akan dianggap lebih signifikan. Setelah itu, algoritma *Cosine Similarity* digunakan untuk menghitung tingkat kemiripan antara vektor kata kunci dari kedua dokumen tersebut. Semakin tinggi kemiripan, semakin relevan pula lowongan kerja yang direkomendasikan terhadap pencari kerja. Hasil evaluasi menunjukkan bahwa sistem ini mampu mencapai tingkat akurasi sebesar 80% yang mengindikasikan keberhasilan dalam menyandingkan pencari kerja dengan peluang karir yang sesuai dengan kompetensi dan minat pencari kerja

**Kata kunci :** Algoritma TF – IDF, *Cosine Similarity*, Pencari Kerja, Lowongan Pekerjaan

### 1. PENDAHULUAN

Dalam era digital yang semakin pesat, informasi lowongan pekerjaan begitu banyak dan mudah diakses [1]. Tetapi, di tengah banyaknya informasi ini, pencari kerja seringkali kesulitan menemukan lowongan yang benar – benar sesuai dengan kualifikasi dan minat [2]. Hal ini disebabkan oleh beberapa faktor, seperti jumlah lowongan yang sangat banyak, variasi format deskripsi pekerjaan, dan perbedaan penafsiran antara pencari kerja dan pemberi kerja[3]. Untuk mengatasi permasalahan tersebut, sistem rekomendasi lowongan pekerjaan ada sebagai solusi inovatif. Sistem ini memanfaatkan teknologi kecerdasan buatan untuk menganalisis data dan memberikan rekomendasi lowongan pekerjaan yang relevan kepada pengguna. Salah satu pendekatan yang populer dalam membangun sistem rekomendasi adalah dengan menggabungkan algoritma TF – IDF (*Term Frequency-Inverse Document Frequency*) dan *Cosine Similarity*.

Algoritma TF – IDF digunakan untuk memberikan bobot pada kata – kata dalam suatu dokumen (dalam hal ini, deskripsi lowongan pekerjaan dan profil pencari kerja)[4]. Kata – kata yang sering muncul dalam suatu dokumen akan diberikan bobot yang lebih tinggi, namun kata – kata yang umum muncul di banyak dokumen akan diberikan bobot yang lebih rendah[5]. Dengan demikian, algoritma TF – IDF dapat membantu mengidentifikasi kata – kata kunci yang paling relevan dalam setiap deskripsi pekerjaan dan profil pencari kerja. Metode TF – IDF ini digunakan karena merupakan metode pembobotan kata yang terkenal efisien, mudah dan memiliki hasil yang akurat [6]. Sedangkan *Cosine Similarity* digunakan untuk mengukur tingkat kemiripan antara dua vektor. *Cosine Similarity* mempunyai nilai akurasi yang tinggi dimana kelebihan utama dari metode cosine similarity adalah tidak terpengaruh pada panjang pendeknya suatu dokumen [7]. Dalam konteks sistem rekomendasi, vektor yang dimaksud adalah

representasi numerik dari deskripsi lowongan pekerjaan dan profil pencari kerja. Dengan menghitung *Cosine similarity* antara kedua vektor, sistem dapat menentukan seberapa mirip suatu lowongan pekerjaan dengan profil seorang pencari kerja.

Tujuan utama dari penerapan algoritma TF – IDF dan *Cosine Similarity* dalam sistem rekomendasi lowongan pekerjaan adalah untuk meningkatkan relevansi antara lowongan pekerjaan dengan profil pencari kerja. Dengan menggunakan algoritma TF – IDF, sistem dapat memberikan bobot yang tepat pada kata – kata kunci yang terdapat dalam deskripsi pekerjaan, sehingga sistem mampu mengidentifikasi lowongan yang paling sesuai dengan keahlian dan minat pencari kerja. Sementara itu, *Cosine Similarity* digunakan untuk menghitung tingkat kemiripan antara deskripsi pekerjaan dan profil pencari kerja, sehingga sistem dapat memberikan rekomendasi yang lebih akurat dan personal. Sistem ini juga akan memberikan manfaat bagi perusahaan dalam menemukan kandidat yang tepat untuk posisi pekerjaan yang tersedia.

### 2. TINJAUAN PUSTAKA

#### 2.1. Algoritma *Term Frequency-Inverse Document Frequency (TF-IDF)*

*Algoritma Term Frequency-Inverse Document Frequency (TF-IDF)* merupakan bidang *information retrieval* yang digunakan untuk menghitung seberapa relevan sebuah kata dalam rangkaian korpus dengan sebuah teks [4]. Algoritma ini menggabungkan dua konsep untuk perhitungan bobot, yaitu frekuensi kemunculan sebuah kata yang disebut *term frequency* (TF) di dalam sebuah dokumen tertentu dan *inverse frekuensi* dokumen yang mengandung kata yang disebut *inverse document frequency* (IDF) [8].

Algoritma ini akan menghitung bobot setiap term di dokumen dengan persamaan 1 sebagai berikut :

$$tf_{(t,d)} = 1 + \log_{tf} \quad (1)$$

Keterangan :

$tf$  = term frekuensi atau banyaknya kata pada dokumen  
 $tf_{(t,d)}$  = term frekuensi atau banyaknya kata  $t$  pada dokumen  $d$  atau pembobotan lokal

Berikut persamaan 2 untuk mencari nilai *inverse document frequency* :

$$idf_{(t)} = \log\left(\frac{N}{df_{(t)}}\right) \quad (2)$$

Keterangan :

$idf_{(t)}$  = *inverse document frequency*

$N$  = jumlah keseluruhan dokumen

$df_{(t)}$  = jumlah dokumen yang memuat *term*  $t$

Berdasarkan persamaan 1 dan persamaan 2 diatas, dapat ditentukan nilai bobotnya ( $W_{(t,d)}$ ) dengan mengalikan kedua persamaan sehingga menjadi persamaan 3 yaitu :

$$W_{(t,d)} = tf_{(t,d)} * idf_t \quad (3)$$

Keterangan :

$tf_{(t,d)}$  = jumlah  $t$  dalam  $d$  / jumlah kata pada  $d$

$df_{(t)}$  = jumlah dokumen yang memuat *term*

## 2.2. Algoritma Cosine Similarity

*Cosine Similarity* adalah salah satu ukuran kemiripan paling populer yang diterapkan pada dokumen teks, seperti dalam berbagai aplikasi pengambilan dalam pembentukan dan pengelompokan. Algoritma ini digunakan untuk menghitung *similarity* (tingkat kesamaan) antar dua buah objek. Secara umum perhitungan algoritma ini didasarkan pada *vector space similarity measure*. Algoritma *Cosine Similarity* ini menghitung *similarity* antara dua buah objek yang dinyatakan dalam dua buah *vector* dengan menggunakan kata kunci dari sebuah dokumen sebagai ukuran [8].

Rumus perhitungan *cosine similarity* :

$$\cos a = \frac{A \cdot B}{|A||B|} = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}} \quad (4)$$

Keterangan :

$A$  = Vektor A, yang akan dibandingkan kemiripan

$B$  = Vektor B, yang akan dibandingkan kemiripan

$A \cdot B$  = *dot product* antara vektor A dan B

$|A|$  = Panjang vektor A

$|B|$  = Panjang vektor B

$|A||B|$  = *cross product* antara  $|A|$  dan  $|B|$

Algoritma ini digunakan untuk membandingkan *similarity* antar dokumen, kemudian mencari nilai *similarity* tertinggi antar dokumen. Apabila total *similarity* yang didapatkan nol (0) maka dokumen yang diolah tidak memiliki kesamaan dan jika nilai yang didapatkan maksimal adalah 1 maka dokumen tersebut memiliki kemiripan.

## 2.3. Black Box Testing

*Black Box Testing* merupakan metode pengujian yang berfokus pada persyaratan fungsional perangkat

lunak atau aplikasi. Pengujian ini bertujuan 13 untuk menemukan kesalahan dalam beberapa kategori [11], antara lain : Fungsi yang tidak tepat atau hilang, Kesalahan pada antarmuka, Ketidaksesuaian kinerja

## 2.4. User Usability Scale (SUS)

*System Usability Scale (SUS)* adalah sebuah kuisioner yang dapat digunakan untuk mengukur kegunaan dari sebuah sistem komputer menurut perspektif pengguna [12]. SUS merupakan suatu metode evaluasi *usability* yang melibatkan pengguna akhir dalam prosesnya.

SUS memiliki aturan yang berbeda pada proses perhitungan hasil jawaban responden. Perbedaan tersebut terletak pada nomor ganjil dan genap instrumen pengujian [13], berikut cara perhitungan hasil pengujian *System Usability Scale (SUS)* :

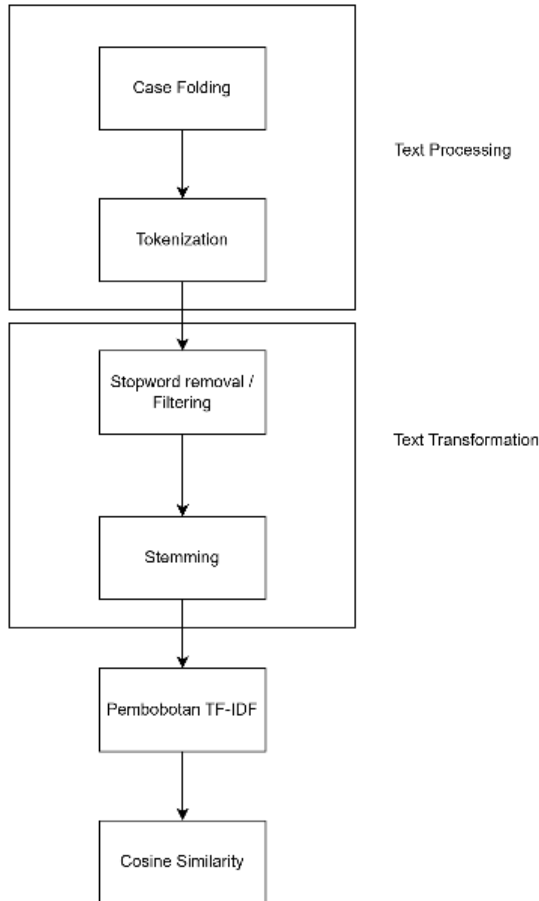
- Pernyataan instrumen nomor ganjil skala jawaban instrumen dikurangi 1.
- Pernyataan instrumen nomor genap maka 5 dikurangi skala jawaban instrumen.
- Hasil penilaian skala 0 – 4 (4 merupakan jawaban terbaik).
- Melakukan penjumlahan jawaban kemudian dikali dengan 2.5.
- Menentukan nilai rerata jawaban instrumen pengujian semua responden dalam menentukan hasil perhitungan.

Penilaian terdapat tiga sudut pandang yaitu *acceptability*, *grade scale*, dan *adjective rating*. *Acceptability* terdapat tiga tingkatan yang terdiri dari *not acceptable*, *marginal* (rendah dan tinggi), dan *acceptable*. Sedangkan *grade scale* terdiri dari A, B, C, D dan F. Untuk *adjective rating* lebih banyak tingkatan yaitu *worst imaginable*, *poor*, *ok*, *good*, *excellent* dan *best imaginable*. Dari ketiga penilaian *System Usability Scale (SUS)* bahwa *acceptability* digunakan untuk melihat tingkat penerimaan pengguna terhadap perangkat lunak, *grade scale* untuk melihat tingkatan (*grade*) *website* dan *adjective rating* untuk melihat *rating* dari *website* yang dihasilkan. Selain dari ketiga cara tersebut *System Usability Scale (SUS)* memiliki cara lain dalam melakukan penentuan hasil perhitungan penilaian yaitu dengan cara *SUS score percentile rank*. Penentuan hasil penilaian berdasarkan *SUS score percentile rank* dilakukan secara umum berdasarkan hasil perhitungan penilaian pengguna. *SUS score percentile rank* memiliki perbedaan dengan *acceptability*, *grade scale*, *adjective rating* yang dikelompokkan menjadi tiga kategori. Berikut adalah ketentuan penentuan penilaian pada *SUS score percentile rank* (Sauro, 2011)

- Grade A: dengan skor  $\geq 80,3$
- Grade B: dengan skor  $\geq 74$  dan  $< 80,3$
- Grade C: dengan skor  $\geq 68$  dan  $< 74$
- Grade D: dengan skor  $\geq 51$  dan  $< 68$
- Grade F: dengan skor  $< 51$

### 3. METODE PENELITIAN

Metode penelitian ini secara umum meliputi pengumpulan data, *preprocessing* data, pemodelan sistem dan evaluasi kinerja. Data yang digunakan dalam penelitian ini adalah kumpulan data lowongan pekerjaan dan profil pencari kerja yang diperoleh dari PT. XYZ.



Gambar 1. Tahapan Perhitungan Algoritma

#### 3.1. Tahap Preprocessing Data

Tahap *text preprocessing* bertujuan untuk membersihkan dan menyederhanakan teks agar lebih mudah diproses oleh komputer[5]. Proses ini meliputi:

##### a. Case Folding

Merupakan proses mengubah semua huruf menjadi huruf kecil (*lowercase*) untuk menghindari duplikasi kata yang hanya berbeda pada huruf kapital.

##### b. Tokenization

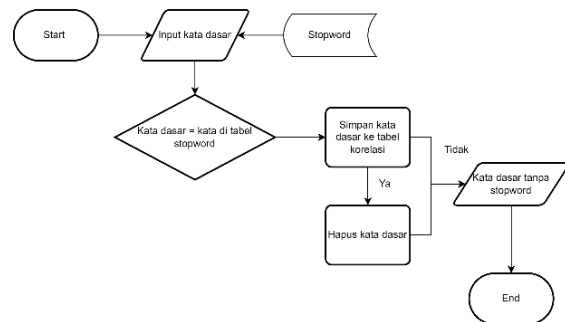
Proses pembagian teks menjadi unit – unit kata atau token. Token ini akan menjadi dasar untuk analisis selanjutnya. Karakter selain huruf akan dianggap *delimiter* dan akan dihilangkan atau dihapus untuk proses mendapatkan kata – kata penyusun teks. *Delimiter* yaitu seperti tanda (.), (,), (:), (;), (?), dan (!) [6].

#### 3.2. Text Transformation

##### a. Stopword Removal/Filtering

*Stop word* adalah kata – kata yang sering muncul dalam bahasa (misal : “dan”, “atau”, “yang”) namun tidak

memiliki makna signifikan dalam analisis. Kata – kata ini dihapus untuk mengurangi *noise* dalam data [6].



Gambar 2. Flowchart Stopword Removal

##### b. Stemming

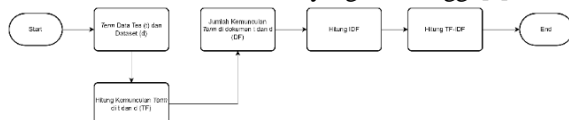
Proses mengubah kata menjadi bentuk dasarnya. Misalnya kata “menjalankan”, “jalankan”, dan “dijalankan” akan diubah menjadi “jalan”. Hal ini bertujuan untuk mengurangi variasi bentuk kata [6].



Gambar 3. Flowchart Stemming

### 3.3. Pembobotan TF - IDF

TF – IDF adalah metode untuk memberikan bobot pada kata – kata dalam sebuah dokumen. Kata yang sering muncul dalam sebuah dokumen akan memiliki bobot yang lebih tinggi, namun kata yang sering muncul di banyak dokumen akan memiliki bobot yang lebih rendah. Dengan demikian, kata – kata yang unik dan relevan dalam sebuah dokumen akan memiliki bobot yang lebih tinggi [6].



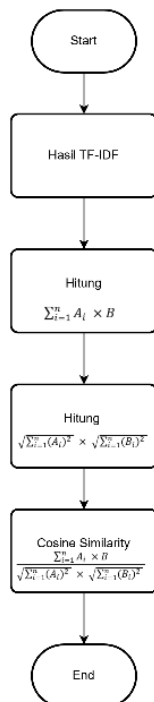
Gambar 4. Flowchart Perhitungan TF – IDF

Cara kerja TF – IDF :

- Term Frequency* (TF) : menghitung frekuensi kemunculan suatu kata dalam sebuah dokumen.
- Inverse Document Frequency* (IDF) : menghitung seberapa penting suatu kata dalam keseluruhan koleksi dokumen. Kata yang jarang muncul di banyak dokumen akan memiliki IDF yang tinggi.
- TF – IDF : menghitung perkalian antara TF dan IDF untuk mendapatkan bobot akhir suatu kata.

### 3.4. Cosine Similarity

Algoritma *Cosine Similarity* ini menghitung *similarity* antara dua buah objek yang dinyatakan dalam dua buah vector dengan menggunakan kata kunci dari sebuah dokumen sebagai ukuran.



Gambar 5. Flowchart Perhitungan Cosine Similarity

## 4. HASIL DAN PEMBAHASAN

### 4.1. Implementasi Algoritma

Pada proses perhitungan dilakukan pengambilan data profil pencari kerja dan lowongan pekerjaan dari database, lalu dilakukan tahap *preprocessing* yaitu *case folding* dan *tokenization*. Berikut adalah data keahlian yang di ambil dari profil pencari kerja dan data yang ada pada lowongan pekerjaan

Data kualifikasi di profil pencari kerja :

**Keahlian**  
CSS, HTML, JavaScript, Framework Laravel, Database MySQL

Data lowongan pekerjaan :

**Data Loker 1**  
Web Developer

Keahlian yang dibutuhkan :  
Laravel, HTML, CSS, Javascript, Web Programming, Web Desain, UI/UX

**Data Loker 2**  
Mobile Developer

Keahlian yang dibutuhkan :  
Mobile Programming, Kotlin

Setelah data yang akan dilakukan perhitungan telah tersedia maka tahap selanjutnya akan dilakukan *text preprocessing* pada data tersebut. Berikut pada Tabel 1. dan Tabel 2. merupakan hasil *case folding* dan *tokenization* dari kedua data profil pencari kerja dan lowongan pekerjaan.

Tabel 1. Tabel Hasil *Case Folding*

Nama Data	Teks Input
Data Keahlian Pencari Kerja	css, html, javascript, framework laravel, database mysql
Data Loker 1	web developer html, css, javascript, laravel full time, html, css, javascript, laravel, web programming, web desain, ui/ux
Data Loker 2	mobile developer, mobile programming, kotlin

Tabel 2. Tabel Hasil *Tokenization*

Nama Data	Teks Input
Data Keahlian Pencari Kerja	css html javascript framework larave database mysql
Data Loker 1	web developer html css javascript laravel full time html css javascript laravel web programming web desain ui ux
Data Loker 2	mobile developer mobile programming kotlin

### 4.2. Hasil Transformation

Pada Tabel 3. dan 4. menyajikan hasil data setelah dilakukan tahap *text transformation* berupa *stopword removal* dan *stemming*. Data tersebut yang akan digunakan untuk melakukan perhitungan selanjutnya menggunakan algoritma TF – IDF untuk mendapatkan nilai bobot.

Tabel 3. Tabel Hasil *Stopword Removal*

Nama Data	Teks Input
Data Keahlian Pencari Kerja	css html javascript framework laravel database mysql
Data Loker 1	web developer html css javascript laravel full time html css javascript laravel web programming web desain ui ux
Data Loker 2	mobile developer mobile programming kotlin

Tabel 4. Tabel Hasil *Stemming*

Nama Data	Teks Input
Data Keahlian Pencari Kerja	css html javascript framework laravel database mysql
Data Loker 1	web developer html css javascript laravel full time html css javascript laravel web programming web desain ui ux
Data Loker 2	mobile developer mobile programming kotlin

### 4.3. Hasil TF – IDF

Setelah data keahlian pencari kerja dan lowongan pekerjaan dilakukan tahap *text preprocessing* dan *text transformation*, hasil dari data tersebut selanjutnya adalah dilakukan perhitungan dengan Algoritma TF – IDF. Pada Tabel 5. menyajikan hasil perhitungan data pencari kerja dan lowongan pekerjaan dengan

Algoritma TF – IDF. Hasil tersebut didapatkan nilai bobot pada masing – masing *term* yang tersedia.

Tabel 5. Tabel Perhitungan TF - IDF

term	tf			df	log(idf)	tf-idf	
	q	d1	d2			d1	d2
web	1	1	0	1	0,477	0,477	0
css	1	1	0	1	0,477	0,477	0
html	1	1	0	1	0,477	0,477	0
javascript	1	1	0	1	0,477	0,477	0
laravel	1	1	0	1	0,477	0,477	0
programming	0	1	1	2	0,176	0,176	0,176
desain	0	1	0	1	0,477	0,477	0
ui	0	1	0	1	0,477	0,477	0
ux	0	1	0	1	0,477	0,477	0
mobile	0	0	1	1	0,477	0	0,477
kotlin	0	0	1	1	0,477	0	0,477

#### 4.4. Hasil Cosine Similarity

Pada Tabel 6. merupakan hasil dari kuadrat nilai bobot dari perhitungan TF – IDF, hasil kuadrat tersebut yang akan digunakan untuk menghitung rumus dari *Cosine Similarity* dan mendapatkan hasil perangkangan dari setiap data yang dihitung.

Tabel 6. Hasil Kuadrat TF - IDF

term	q	d1	d2
web	0,277	0,277	0
css	0,277	0,277	0
html	0,277	0,277	0
javascript	0,277	0,277	0
laravel	0,277	0,277	0
programming	0	0,030	0,030
desain	0	0,277	0
ui	0	0,277	0
ux	0	0,277	0
mobile	0	0	0,277
kotlin	0	0	0,277
Total Kuadrat	4,195	2,853	2,576
Akar Kuadrat	2,048	1,689	1,604

Setelah mendapatkan nilai kuadrat dan akar kuadrat dari masing – masing data maka selanjutnya akan dilakukan proses perhitungan menggunakan rumus cosine similarity sebagai berikut :

$$\text{Cos } d1 = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}} = \frac{1,445}{(2,048 \times 1,689)} = \frac{1,445}{3,459} = 0,417$$

$$\text{Cos } d2 = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}} = \frac{0,090}{(2,048 \times 1,604)} = \frac{0,090}{3,284} = 0,027$$

Berdasarkan hasil perhitungan cosine similarity di atas didapatkan hasil untuk persamaan data lowongan pekerjaan 1 (d1) dengan data keahlian pencari kerja (q) menghasilkan nilai 0,417 sedangkan hasil persamaan untuk data lowongan pekerjaan 2 (d2) dengan data keahlian pencari kerja (q) adalah 0,027. Maka dapat disimpulkan bahwa data yang paling dekat dengan data keahlian pencari kerja adalah data

lowongan pekerjaan 1 (d1), karena semakin besar nilai cosinus maka semakin mirip data tersebut. Hasil dari proses cosine similarity akan ditampilkan di dalam sistem dengan menampilkan kurang lebih lima data lowongan pekerjaan dengan nilai cosinus tertinggi. Lima data lowongan pekerjaan yang ditampilkan itu akan menjadi rekomendasi lowongan pekerjaan yang sesuai dengan kualifikasi bagi pencari kerja.

#### 4.5. Implementasi Sistem

Pada gambar 6. dan gambar 7. merupakan implementasi dari halaman hasil perhitungan TF – IDF dari data lowongan pekerjaan dan keahlian yang dihasilkan dari perhitungan pada algoritma TF - IDF untuk melakukan perhitungan rekomendasi lowongan pekerjaan. Ketiga halaman diatas akan menampilkan hasil masing – masing term, nilai TF dan TF – IDF dari masing – masing term yang dihitung. Pada halaman tersebut akan menampilkan bobot dari setiap kata yang dilakukan perhitungan.

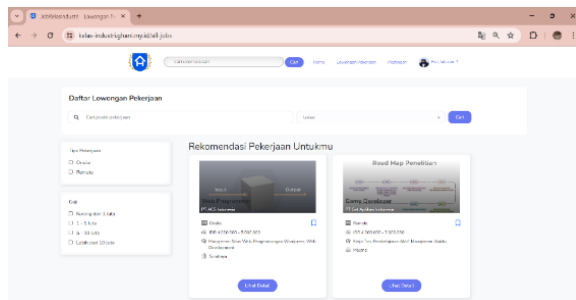
Gambar 6. Tampilan Halaman Perhitungan TF - IDF Loker

Gambar 7. Tampilan Halaman Perhitungan TF - IDF Keahlian Pencari Kerja

Pada gambar 8. merupakan implementasi dari halaman hasil perhitungan *cosine similarity* yang digunakan untuk menentukan hasil rekomendasi lowongan pekerjaan sesuai dengan kecocokan keahlian dari pencari kerja. Pada halaman ini akan menampilkan *score similarity* dan presentase kemiripan dari data lowongan pekerjaan dan profil pencari kerja yang dilakukan perhitungan. Dari hasil skor tersebut maka sistem akan menampilkan rekomendasi lowongan pekerjaan bagi masing – masing pencari kerja.

No.	Rekomendasi Pekerjaan	Nilai Similarity	Peringkat Rekomendasi
1.	Fullstack Developer	0.95	1
2.	Web Developer	0.85	2
3.	Backend Developer	0.75	3
4.	Frontend Developer	0.65	4
5.	UI/UX Designer	0.55	5

Gambar 8. Tampilan Hasil Cosine Similarity



Gambar 9. Tampilan Halaman Rekomendasi Lowongan Pekerjaan

Pada gambar 9. merupakan implementasi dari halaman rekomendasi lowongan pekerjaan. Fitur

rekomendasi lowongan pekerjaan hanya dapat diakses oleh pencari kerja yang login pada sistem. Fitur ini akan menampilkan lowongan pekerjaan yang sesuai dengan keahlian yang dimiliki oleh pencari kerja tersebut. Halaman rekomendasi ini merupakan hasil implementasi dari perhitungan dengan algoritma TF – IDF dan *Cosine Similarity* dalam penentuan hasil rekomendasi lowongan pekerjaan yang sesuai.

#### 4.6. Hasil Pengujian Akurasi

Pengujian akurasi pada perhitungan rekomendasi lowongan pekerjaan dengan menggunakan algoritma TF – IDF dan *Cosine Similarity* didapatkan dengan membandingkan hasil keluaran rekomendasi dari sistem dengan data hasil rekomendasi yang dilakukan oleh perusahaan mitra. Pengujian yang dilakukan menggunakan 5 profil pencari kerja yang terdaftar dan 15 lowongan pekerjaan yang tersedia pada sistem. Dari data pencari kerja dan lowongan pekerjaan tersebut, akan dilakukan perbandingan hasil rekomendasi yang dilakukan oleh evaluator dan yang dihasilkan oleh sistem. Tabel 7. merupakan data yang dihasilkan dari rekomendasi lowongan pekerjaan untuk pencari kerja yang dilakukan oleh evaluator.

Tabel 7. Tabel Hasil Rekomendasi Loker oleh Sistem dan Perusahaan

No.	Data Keahlian Pencari Kerja	Rekomendasi Loker Sistem	Rekomendasi Loker Evaluator
1.	Pemrograman dasar, web development	Fullstack Developer Web Programmer Backend Developer Wordpress Developer	Fullstack Developer Wordpress Developer Web Programmer Backend Developer
2.	HTML, CSS, Javascript, PHP	Web Developer Web Admin Frontend Web Developer PHP Web Developer	PHP Web Developer Web Developer Backend Developer Web Admin
3.	Adobe Photoshop, Figma, Design Thinking, Desain Grafis	Graphic Design Desain Grafis UI / UX Design	UI / UX Designer Graphic Design Desain Grafis
4.	HTML, CSS, Javascript, Adobe Photoshop, Adobe Illustrator, Web Desain	UI UX Designer Graptic Design Desain Grafis Web Developer	Desain Grafis UI/IX Designer Graptic Design Web Developer
5.	Laravel, PHP, React JS, Vue	PHP Web Developer Frontend Web Developer Backend Developer Web Admin	PHP Web Programmer Frontend Web Developer Backend Developer Web Admin

Hasil akurasi di dapatkan dengan membandingkan antara hasil rekomendasi dari sistem dengan rekomendasi dari evaluator yang secara manual memberikan rekomendasi lowongan pekerjaan sesuai dengan profil pencari kerja dan lowongan pekerjaan yang tersedia. Adapun hasil akurasi sistem dapat dilihat pada perhitungan berikut :

$$\begin{aligned}
 \text{Akurasi} &= \frac{4}{5} \times 100\% \\
 &= 80\%
 \end{aligned}$$

Dari hasil perhitungan diatas didapatkan hasil pengujian dengan data yang diuji didapatkan akurasi sebesar 80%.

## 5. KESIMPULAN DAN SARAN

Penelitian ini berhasil mengimplementasikan algoritma TF-IDF dan *Cosine Similarity* dalam sistem rekomendasi lowongan kerja. Akurasi sistem diukur dengan membandingkan hasil rekomendasi sistem dengan hasil pencocokan yang dilakukan secara manual oleh evaluator. Hasil perbandingan menunjukkan bahwa sistem mampu mencapai akurasi sebesar 80%. Angka akurasi ini didapatkan melalui perhitungan presentase kesesuaian antara rekomendasi sistem dengan penilaian manual. Dengan memboboti kata kunci dalam deskripsi pekerjaan dan profil pencari kerja, sistem mampu memberikan rekomendasi yang lebih relevan. Penggunaan *Cosine Similarity* memungkinkan perhitungan kemiripan yang lebih akurat. Hasil ini menunjukkan bahwa pendekatan yang diusulkan efektif dalam meningkatkan efisiensi pasar kerja, mempercepat pencari kerja menemukan pekerjaan yang sesuai dan membantu perusahaan menemukan kandidat yang tepat.

Saran yang perlu dilakukan adalah evaluasi dan perbaikan lebih lanjut pada algoritma rekomendasi untuk meningkatkan akurasi dan relevansi rekomendasi pekerjaan. Penggunaan data yang lebih bervariasi dan pembaruan terhadap algoritma dapat membantu meningkatkan kualitas rekomendasi.

## DAFTAR PUSTAKA

- [1] R. B. Miko, O. Wijaya, and E. Diah Utami, "Determinan Pengangguran Lulusan SMK di Indonesia Tahun 2020 Analisis Data Sakernas Februari 2020 (Determinants of Unemployment of Vocational High School Graduates in Indonesia in 2020)," in *Seminar Nasional Official Statistics*, 2021, pp. 801–810.
- [2] S. Suyitno, Y. Kamin, D. Jatmoko, M. Nurtanto, and E. Sunjayanto, "Industrial Apprenticeship Model Based on Work-Based Learning for Pre-service Teachers in Automotive Engineering," *Front Educ (Lausanne)*, vol. 7, Jul. 2022, doi: 10.3389/educ.2022.865064.
- [3] M. Fawaid *et al.*, "Entrepreneurial Intentions of Vocational Education Students in Indonesia: PLS-SEM Approach," *Journal of Technical Education and Training*, vol. 14, no. 2 SPECIAL ISSUE, pp. 91–105, Sep. 2022, doi: 10.30880/jtet.2022.14.02.009.
- [4] G. Salton and C. Buckley, "Term-Weighting Approaches In Automatic Text Retrieval," *Inf Process Manag*, vol. 24, no. 5, pp. 513–523, 1988.
- [5] V. Amrizal, "Penerapan Metode Term Frequency Inverse Document Frequency (Tf-Idf) Dan Cosine Similarity Pada Sistem Temu Kembali Informasi Untuk Mengetahui Syarah Hadits Berbasis Web (Studi Kasus: Hadits Shahih Bukhari-Muslim)," *Jurnal Teknik Informatika*, vol. 11, no. 2, pp. 149–164, Nov. 2018, doi: 10.15408/jti.v11i2.8623.
- [6] D. Septiani and I. Isabela, "Analisis Term Frequency Inverse Document Frequency (Tf-Idf) Dalam Temu Kembali Informasi Pada Dokumen Teks".
- [7] D. Kurniadi, S. Farisa, C. Haviana, and A. Novianto, "Implementasi Algoritma Cosine Similarity pada sistem arsip dokumen di Universitas Islam Sultan Agung," *TRANSFORMATIKA*, vol. 17, no. 2, pp. 124–132, 2020.
- [8] Sintia, S. Defit, and G. Widi Nurcahyo, "Product Codification Accuracy With Cosine Similarity And Weighted Term Frequency And Inverse Document Frequency (Tf-Idf)," *Journal of Applied Engineering and Technological Science*, vol. 2, no. 2, pp. 14–21, 2021.
- [9] S. Lumbansiantar, S. Dwiasnati, and N. S. Fatolah, "Penerapan Metode Cosine Similarity dalam Mendeteksi Plagiarisme pada Jurnal," *Jurnal Format*, vol. 12, 2023.
- [10] M. Habibi, U. Jenderal, A. Yani, U. Islam, and N. Sunan Kalijaga, "Implementation of Cosine Similarity in an automatic classifier for comments," *JISKA*, vol. 3, no. 2, pp. 110–118, 2018.
- [11] H. Harijanto, I. Kusumaning Putri, and H. N. Javier, "Penerapan Metode User Centered Design (UCD) Katalog Online Berbasis Web di Rumah Makan Alfa Lesehan Kabupaten Blitar," *Seminar Informatika Aplikatif Polinema*, pp. 147–156, 2021.
- [12] G. W. Sasmito, L. O. M. Zulfikar, and M. Nishom, "Usability Testing based on System Usability Scale and Net Promoter Score," in *2019 2nd International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, 2019, pp. 540–545. doi: 10.1109/ISRITI48646.2019.9034666.
- [13] P. Laurentinus, F. P. Juniawan, and D. Y. Sylfania, "Evaluasi Usability Sistem Pelaporan Publikasi Penelitian Dosen Berbasis Android," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 4, no. 1, pp. 123–134, 2020.