PENGEMBANGAN APLIKASI "LOST & FOUND" BERBASIS ANDROID DENGAN MENGGUNAKAN METODE TERM FREQUENCY – INVERSE DOCUMENT FREQUENCY (TF-IDF) DAN COSINE SIMILARITY

APPLICATION DEVELOPMENT "LOST & FOUND" ANDROID-BASED USING TERM FREQUENCY - INVERSE DOCUMENT FREQUENCY (TF-IDF) AND COSINE SIMILARITY METHOD

Luluk Suryani[1], Kasmi Edy[2]

[1,2] Politeknik Saint Paul Sorong
luluk.suryani@gmail.com[1], kasmi_edy@poltekstpaul.ac.id[2]

Abstrak

Proses pencarian umumnya akan dilakukan dengan cara menyebarkan informasi dari orang ke orang, menyebarkan kertas informasi di tempat umum, melaporkan kepada pihak berwajib dan juga melalui media informasi seperti koran, radio, televisi dan sosial media. Proses pencarian melalui media informasi tentu sangat efektif, karena semakin banyak orang yang tahu maka akan semakin besar kemungkinan untuk ditemukan. Tapi cara tersebut masih memiliki kelemahan karena barang yang belum ditemukan hingga memamakan waktu berbulan-bulan bahkan bertahun-tahun, informasi kehilangan tersebut akan tertutup dengan berita baru yang berkembang di masyarakat sehingga informasi kehilangan yang telah dibuat akan mulai dilupakan oleh banyak orang. Dilatarbelakangi hal tersebut, mendorong peneliti mengembangkan Aplikasi "Lost & Found". Aplikasi bisa diakses melalui handphone android secara online. Tujuan mengembangkan aplikasi "Lost & Found" adalah untuk menyiapkan wadah bagi individu dalam menyebarkan informasi kehilangan atau temuan. Aplikasi diimplementasikan menggunakan Metode Cosine Similarity yang berguna mencari seberapa besar tingkat kemiripan antara dokumen, semakin besar nilai cosinus maka semakin mirip dokumen yang dibandingkan. Nilai cosinus 1 menyatakan kemiripan 100%, sedangkan nol menyatakan ketidakmiripan 100% [1]. Aplikasi juga akan menerapkan metode TF-IDF dimana metode ini berfungsi memberi bobot hubungan suatu kata terhadap dokumen.

Kata kunci : Pencarian, Kehilangan, Temuan, Cosine Similarity, TF-IDF, Android.

Abstract

The search process will generally be carried out by spreading information from person to person, distributing information papers in public places, reporting to authorities and also through information media such as newspapers, radio, television and social media. The process of searching through information media is certainly very effective, because the more people who know it, the more likely it is to be found. But this method still has weaknesses because the goods that have not been found to even take months and even years, the loss of information will be covered with new news developing in the community so that the loss of information that has been made will begin to be forgotten by many people. Against this background, the researchers encouraged researchers to develop the "Lost & Found" Application. The application can be accessed via android mobile online. The purpose of developing the "Lost & Found" application is to prepare a forum for individuals to disseminate lost information or findings. The application is implemented using the Cosine Similarity Method, which is useful for finding how similar the level of similarity between documents, the greater the cosine value, the more similar the documents are compared. A cosine value of 1 represents 100% similarity, while zero represents 100% dissimilarity [1]. The application will also apply the TF-IDF method where this method serves to give weight to the relationship of a word to the document.

Keywords*: *Search, Loss, Findings, Cosine Similarity, TF-IDF, Android

1. PENDAHULUAN

1.1 Latar Belakang

Kehilangan adalah hal yang biasa terjadi dalam kehidupan sehari-hari, baik kehilangan barang pribadi, surat-surat penting, bahkan kehilangan anggota keluarga. Untuk penyebaran informasi kehilangan atau temuan, sebagian besar individu akan melakukan beberapa cara diantaranya adalah mengumumankannya di tempat ramai seperti di pasar. Selain itu masih banyak juga yang melakukan dengan cara menempelkan kertas informasi di tembok-tembok, papan pengumuman atau koran. Alternatif lain yang digunakan adalah dengan memanfaatkan sosial media, dimana cara ini lebih mudah dalam penyebaran informasinya, akan tetapi ketiga cara tersebut kurang efektif dimana cara pertama dan kedua sudah jarang dilakukan lagi pada zaman sekarang karena jadwal pekerjaan yang sangat padat, belum tentu mendapat perhatian dari pembaca, sedangkan cara ketiga informasi yang diberikan akan dengan cepat tertutupi oleh status ataupun berita-berita baru yang berkembang di masyarakat. Sama halnya jika ada seseorang yang menemukan barang, akan mengalami kesulitan untuk menemukan pemilik asli dari barang tersebut, dengan adanya masalah tersebut maka peneliti terdorong untuk membuat wadah berupa aplikasi *"Lost & Found"* yang dapat membantu masyarakat untuk menyebarkan informasi mengenai kehilangan atau temuan.

Aplikasi yang dikembangkan akan bermanfaat terutama pada zaman sekarang dimana tindak kriminal semakin meningkat, seperti banyak terjadi aksi pencurian kendaraan bermotor. Pihak berwajib juga sudah sering menemukan barang-barang hasil curian tetapi permasalahanya akan memakan waktu yang cukup lama untuk menemukan pemilik asli yang merasa kehilangan, oleh karena itu diharapkan aplikasi ini dapat dimanfaatkan juga oleh pihak berwajib untuk memudahkan masyarakat mendapatkan kembali barang atau anggota keluarga yang hilang. Pada aplikasi *"Lost & Found"* akan menggunakan metode *Cosine Similarity* dan TF-IDF. Metode *Term Frequency Inverse Document Frequency* (TF-IDF) merupakan suatu cara untuk memberikan bobot hubungan suatu kata (*term*) terhadap dokumen. Sedangkan, Metode *cosine similarity* merupakan metode untuk menghitung kesamaan antara dua buah objek yang dinyatakan dalam dua buah vector dengan menggunakan *keywords* (kata kunci) dari sebuah dokumen sebagai ukuran [1]. Aplikasi *"Lost & Found"* dalam pengembangannya akan didukung oleh *tools* Android Studio dengan bahasa Pemprograman Kotlin.

Tahapan pada penelitian ini menggunakan metode *System Development Life Cycle* (SDLC) dengan model *Waterfall*. Dimana menurut [2], *waterfall* memiliki cara sekuensial yang sistematis untuk mengembangkan perangkat lunak yang diawali dengan spesifikasi kebutuhan, perancangan, pemodelan, pembuatan dan penyebaran yang berguna untuk mendukung dalam menyelesaikan perangkat lunak. Adapun tahapan pada metode *waterfall* yaitu analisis dan mendefinisikan kebutuhan, desain sistem, implementasi serta pengujian setiap unit, Integrasi dan juga Pengujian Sistem, Operasi dan Perawatan. Setelah pembuatan aplikasi selesai, maka akan dilakukan pengujian pada aplikasi dengan *metode Confusing Matrix.*

Penelitian mengenai pencarian yang menggunakan metode *Cosine Similarity* atau TF-IDF sudah pernah dilakukan sebelumnya. Beberapa penelitian terdahulu diantaranya yang dilakukan oleh Latif, dkk [3] penelitian ini bertujuan untuk membangun sistem yang dapat membantu masyarakat dalam melakukan pencarian dan pengumuman barang hilang atau orang hilang yang sesuai dengan aturan syariat. Pada penelitian ini aplikasi juga bisa dijalankan pada sistem operasi android. Proses pencarian menggunakan query. Hasil akhir dari penelitian ini bahwa aplikasi dapat berjalan dengan baik. Penelitian yang dilakukan oleh Abdul Aziz Maarif [4] bertujuan untuk menerapkan algoritma TF-IDF yang dapat digunakan mencari karya ilmiah sebagai pengukur tingkat similaritas antara dokumen dengan keyword yang didapat dari ekstraksi teks pada dokumen sehingga mendapatkan

data yang terurut dari yang kemiripannya (tingkat similaritas) paling tinggi sehingga pencarian karya ilmiah menjadi lebih efisien sebagai informasi yang relevan.

Penelitian yang dilakukan oleh Ria, dkk [5] yaitu membuat Sistem Temu Kembali Informasi yang dapat dimanfaatkan untuk mengetahui syarah hadits, karena dapat memberikan alternatif berupa metode similarity yang dapat digunakan untuk melakukan pencarian dokumen relevan dengan yang diinginkan. Metode *similiarity* yang digunakan adalah *cosine similarity* dengan pembobotan kata menggunakan metode TF-IDF dan menerapkan *teks preprocessing* terlebih dahulu untuk memperkecil *term* sehingga bisa mempercepat proses perhitungan *term. Teks preprocessing* tersebut *meliputi tokenizing, stopword removal* atau *filtering*, dan *stemming*. Hasil uji coba dengan pengujian *confusion matrix* didapatkan: *recall* 88.7%, *precision* 100%, *accuracy* 88,73 %, dan *error rate* 11,27 %.Pada penelitian Rahman [6] memiliki kasus yang kurang lebih sama yaitu menghasilkan sebuah website *"lost and found"* dengan fitur pencarian menggunakan metode *Cosine Similarity* yang dapat memudahkan pengguna dalam pencarian barang hilang. *Cosine similarity* adalah metode perhitungan kemiripan kata yang paling populer untuk diterapkan pada dokumen teks. Kelebihan utama dari metode *cosine similarity* adalah tidak terpengaruh pada panjang pendeknya suatu dokumen.

Pada penelitian yang dilakukan oleh Setyoko, dkk [7] bertujuan untuk melakukan pencarian pasal pada kitab undang-undang hukum pidana (KUHP) berdasarkan kasus menggunakan metode *cosine similarity* dan *latent semantic indexing* (LSI). Sistem yang dikembangkan dalam pencarian pasal KUHP berdasarkan kasus kejahatan pada penelitian ini menunjukkan hasil yang cukup baik, dimana nilai Mean Average Precision (MAP) yang dihasilkan mendekati nilai 1. Jurnal yang ditulis oleh David, dkk [8] menunjukkan bahwa kombinasi metode analisis sentimen menggunakan *SentiStrength* ke dalam metode peringkasan teks di Twitter (*Hybrid* TF-IDF), dan metode *Cosine Similarity* untuk mengeliminasi tweet hasil peringkasan yang memiliki kemiripan isi pesan. Dengan menggabungkan metode-metode tersebut dapat meningkatkan kemampuan Hybrid TF-IDF dalam menghasilkan peringkasan opini di Twitter, khususnya pada masalah peringkasan opini fans dan haters selebriti. Berdasarkan beberapa penelitian terdahulu yang pernah dilakukan dapat diketahui bahwa metode *Cosine Similarity* dan TF-IDF bisa digunakan untuk melakukan pencarian. Peneliti menggunakan 2 metode tersebut karena sudah terbukti bisa diterapkan untuk proses pencarian. Penelitian Aplikasi "*Lost and Found*" menggunakan sistem operasi android dalam penerapannya karena banyak sekali pengguna *smartphone* dari sistem operasi ini di Kota Sorong dan juga bahasa pemograman Kotlin akan mendukung kinerja aplikasi dengan baik.

2. DASAR TEORI /MATERIAL DAN METODOLOGI/PERANCANGAN

2.1 *Information Retrieval*

Information Retrieval (IR) adalah bagian dari *computer science* yang berkaitan dengan pengambilan informasi, dimana informasi berasalah dari dokumen yang didasarkan pada isi serta konteks dari dokumen itu sendiri. *Information Retrieval* berfungsi melakukan pencarian informasi melalui *query* (inputan pengguna) yang diharapkan dapat memenuhi keinginan pengguna dari kumpulan dokumen yang ada [9]

2.2 *Preprocessing*

Text Preprocessing adalah proses awal terhadap teks untuk mempersiapkan teks menjadi data yang akan diolah lebih lanjut. Teks tidak dapat diproses langsung oleh algoritma pencarian, oleh sebab itu dibutuhkan preprocessing text untuk mengubah teks menjadi data numeric [10]. Proses ini terdiri dari beberapa tahap pembersihan dokumen seperti berikut [11]:

a. Tokenizing

Tokenizing adalah proses memecah dokumen menjadi kumpulan kata. *Tokenization* berfungsi menghilangkan tanda baca dan memisahkan kata berdasarkan spasi. Tahapan ini juga menghilangkan karakter tertentu seperti tanda baca dan mengubah semua token ke bentuk huruf kecil *(lower case*).

b. *Stopword Removal* atau ***Filtering***

Stopwords removal adalah proses penghilangan kata yang tidak penting. Pada tahapan ini dilakukan pengecekan kata-kata hasil *parsing* deskripsi apakah termasuk di dalam daftar kata tidak penting (*stoplist*) atau tidak. Jika termasuk maka kata tersebut akan dihapus.

c. *Stemming Nazief & Adriani*

Stemming adalah bagian yang tidak terpisahkan dalam *Information Retrieval (IR).* Tidak banyak algoritma yang dikhususkan untuk *stemming* Bahasa Indonesia dengan berbagai keterbatasan didalamnya. Algoritma Nazief dan Adriani sebagai algoritma stemming untuk teks berbahasa Indonesia yang memiliki kemampuan persentase keakuratan lebih baik dari algoritma lainnya [12]

2.3 *Term Frequency Inverse Document Frequency* (Tf-Idf)

Metode *Term Frequency Inverse Document Frequency* (TF-IDF) adalah cara pemberian bobot hubungan suatu kata (*term*) terhadap dokumen. TF-IDF merupakan ukuran statistik yang berguna untuk mengevaluasi seberapa penting sebuah kata didalam sebuah dokumen atau dalam sekelompok kata. Pada dokumen tunggal setiap kalimat akann dianggap sebagai dokumen. Frekuensi kemunculan kata didalam dokumen yang diberikan menunjukkan seberapa penting kata itu didalam dokumen tersebut. Frekuensi dokumen yang mengandung kata tersebut menunjukkan seberapa umum sebuah kata. Bobot kata semakin besar jika sering muncul dalam sebuah dokumen dan semakin kecil jika muncul dalam banyak dokumen. [13] Pada algoritma TF-IDF menggunakan rumus untuk menghitung bobot (W) masing masing dokumen terhadap kata kunci dengan rumus sebagai berikut :

$$Wdt = tfdt \times Idft \qquad (1)$$

Keterangan:
Wdt = Bobot dokumen ke-d terhadap kata ke-t
tfdt = Banyaknya kata yang dicari pada sebuah dokumen
Idft = Inversed Document Frequency (log(N/df))
N = Total dokumen
df = Banyak dokumen yang mengandung katayang dicari.

2.4 *Cosine Similarity*

Cosine Similarity merupakan ukuran kesamaan antara dua buah vektor dalam sebuah ruang dimensi yang didapat dari nilai cosinus sudut dari perkalian dua buah *vector* yang dibandingkan karena cosinus dari 0 adalah 1 dan kurang dari 1 untuk nilai sudut yang lain, maka nilai similarity dari dua buah vektor dikatakan mirip ketika nilai dari cosine similarity adalah 1 [14]. Berikut adalah rumus *cosine similarity.*

$$Cos\ \alpha\ = \frac{Q \cdot D}{|Q||D|} = \frac{\sum_{i=1}^{n}(wqi\ \times wdij)}{\sqrt{\sum_{i=1}^{n}(wqi)^2} \times \sqrt{\sum_{i=1}^{n}(wdij)^2}} \qquad (2)$$

Keterangan :
Q = Vektor Q, dibandingkan kemiripannya
D = Vektor D, dibandingkan kemiripannya.
Q • D = dot product antara vektor Q dan D
|Q| = panjang vektor Q
|D| = panjang vektor D
|Q||D| = cross product antara |Q| dan |D|
wqi = bobot term pada query ke-i=tf x idf
wdij = bobot term pada dokumen ke-I istilah ke-j= tf x idf
i = jumlah term dalam kalimat.
N = jumlah vektor.

Pada Penelitian ini peneliti memilih menggunakan metode *cosine similarity* karena hasilnya memiliki nilai akurasi yang tinggi dimana menurut [15] kelebihan utama dari metode *cosine similarity* adalah tidak terpengaruh pada panjang pendeknya suatu dokumen. Sehingga, dengan melakukan perbandingan *keyword* yang dihasilkan, maka kedekatan antara dokumen dapat dipastikan.

2.5 *Confusing Matrix*

Metode *confusing matrix* berguna untuk melakukan perhitungan tingkat akurasi. Pada saat pengujian tingkat akurasi hasil dari pencarian maka akan dievaluasi nilai *recall, precission, accuracy dan error rate*. Precission bertugas untuk evaluasi kemampuan dari sistem dalam menentukan peringkat yang paling relevan. *Recall* bertugas untuk evaluasi kemampuan sistem dalam menemukan semua item yang relevan didalam koleksi dokumen. *Accuracy* melakukan perbandingan kasus yang dianggap benar dalam semua kasus uji coba yang dilakukan. Terakhir *error rate* merupakan kebalikan dari *accuracy* dimana membandingkan kasus yang dianggap salah dengan jumlah seluruh kasus. [16]
Berikut rumus nilai-nilai dari metode *confusing matrix* :

$$Precission = \frac{tp}{tp+fp} \qquad (3)$$

$$Recall = \frac{tp}{tp+fn} \qquad (4)$$

$$Accuracy = \frac{tp+tn}{tp+fp+tn+fn} \qquad (5)$$

$$Error\ Rate = \frac{fn+tn}{tp+fp+tn+fn} \qquad (6)$$

Keterangan :
TP (True Positive) = Jumlah prediksi yang benar dari data yang relevant.
FP (False Positive) = Jumlah prediksi yang salah dari data yang tidak relevant.
FN (False Negative) = Jumlah prediksi yang salah dari data yang tidak relevant.
TN (True Negative) = Jumlah prediksi yang benar dari data yang relevant.

Tabel 1. Rumus Confusing Matrix

Dokumen	Nilai Sebenarnya	
	Relevant	Non Relevant
Retrieved	True Positive (tp)	False Positive (fp)
Not Retrieved	False negative (fn)	True Negative (tn)

2.6. Metodologi

Pada penelitian aplikasi "*lost & found*" peneliti melakukan beberapa metode dalam pengumpulan data maupun pengembangan sistem.

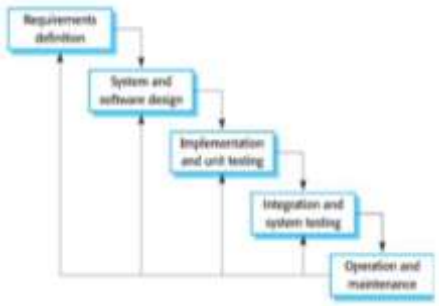
2.6.1 Metode Pengumpulan Data

Pengumpulan data dilakukan untuk memperoleh informasi yang dibutuhkan dalam rangka mencapai tujuan penelitian. Peneliti melakukan pengumpulan data dengan mempelajari berbagai buku, pencarian referensi dari google, jurnal, serta hasil penelitian sebelumnya yang berguna untuk mendapatkan landasan teori mengenai masalah yang akan diteliti.

Metode pengumpulan data berikutnya yaitu melakukan observasi dan wawancara kepada beberapa orang yang pernah mengalami kehilangan atau menemukan barang terkhusus di Kota Sorong. Peneliti mencari tahu bagaimana proses menemukan barang yang hilang, kendala apa yang dialami serta harapan kedepannya. Pertanyaan yang serupa juga peneliti ajukan jika seseorang pernah menemukan barang. Hal tersebut peneliti lakukan agar aplikasi yang akan dikembangkan bisa sesuai dengan kebutuhan pengguna.

2.2 Metode Pengembangan Sistem

Pada aplikasi "*Lost & Found*", peneliti mengembangkan sistem dengan menggunakan model *waterfall*, dimana model *waterfall* merupakan model pengembangan sistem informasi yang sistematik dan sekuensial [17] Model *waterfall* memiliki tahapan-tahapan sebagai berikut :



Gambar 1. Model *Waterfall*

1. *Requirements analysis and definition* adalah menggali kebutuhan pengguna yang didefinisikan secara rinci yang kemudian akan dijadikan sebagai spesifikasi sistem.
2. *System and software design* merancang sistem dengan mengalokasikan kebutuhan baik dari segi *hardware* maupun *software* membentuk arsitektur sistem secara keseluruhan. Perancangan *software* melibatkan identifikasi dan penggambaran abstrak sistem dasar.
3. *Implementation and unit testing* Pada tahap ini, perancangan *software* direalisasikan sebagai serangkaian unit program. Pengujian melibatkan verifikasi bahwa setiap unit memenuhi spesifikasinya.

4. *Integration and system testing* setiap unit program akan digabungkan menjadi satu kesatuan yang sudah lengkap untuk di uji coba kesesuaiannya dengan spesifikasi sistem.

Operation and maintenance merupakan tahapan yang paling panjang. Sistem dipasang dan digunakan secara nyata oleh pengguna kemudian akan di *maintenance* secara berkala sesuai kebutuhan.

3. PEMBAHASAN

Pada bab pembahasan yang akan dijelaskan lebih dahulu adalah fase perencanaan sesuai dengan metode pengembangan sistem yang dilakukan. Perencanaan adalah pada saat proses *preprocessing* terhadap Dokumen (D) dan Query (Q) yang berguna untuk menemukan dokumen yang sesuai dengan query yang dimasukkan. Berikut detail proses dari *Preprocessing.*

3.1 *Text Preprocessing*

Proses Preprocessing bisa dilakukan jika semua dokumen yang menjadi dataset telah dimasukkan kedalam koleksi dokumen. Koleksi dokumen berisi daftar barang hilang dan juga barang temuan. Selanjutnya bisa dilakukan proses pencarian sebagai berikut :

1. *Tokenizing*

Pada saat proses *Tokenizing* dilakukan penghilangan tanda baca, pemecahan text dimana yang menjadi dalimeter ada spasi dan mengubah semua token terbentuk menjadi huruf kecil. Tokenizing dilakukan pada query dan juga koleksi dokumen. Berikut hasil Tokenizing dari query yang dimasukkan.

Tabel 2. Hasil *Tokenizing*

Sebelum Tokenizing	Sesudah Tokenizing
Telah ditemukan sebuah dompet pria warna biru tua di atm toko thio, isi berupa beberapa ktp dengan nama meily siyaniaoessy	telah
	ditemukan
	sebuah
	dompet
	pria
	berwarna
	biru
	tua
	di
	atm
	toko
	thio
	isi
	berupa
	beberapa
	ktp
	dengan
	nama
	meily
	siyaniaoessy

2. *Stopword Removel*

Pada saat proses *stopword removel* maka hasil dari query dan koleksi dokumen yang sudah melalui proses tokenizing akan dicocokan dengan *array stoplist* yang ada. Apabila token yang sedang di cek merupakan *stoplist* maka token tersebut akan dihapus. Sebaliknya jika token yang diproses tidak termasuk dalam stoplist maka akan dibiar tetap ada. Contoh penerapan hasil *stopword removel* adalah sebagai berikut :

Tabel 3. Hasil *Stopword Removel*

Sesudah *tokenizing*	**Sesudah proses *stopword removal***
telah	-
ditemukan	ditemukan
sebuah	-
dompet	dompet
pria	pria
berwarna	berwarna
biru	biru
tua	tua
di	-
atm	atm
toko	toko
thio	thio
isi	-
berupa	-
beberapa	-
ktp	ktp
dengan	-
nama	-
meily	meily
siyaniaoessy	sinaniaoessy

3. *Stemming*

Pada penelitian ini proses *stemming* menggunakan algoritma *stemming* Nazief Adriani dimana hasilnya memiliki tingkat akurasi lebih tinggi dibandingkan jenis *stemming* yang lain. Proses yang akan dilakukan yaitu token setelah proses *stopword removel* sebelumnya diperiksa kedalam database kata dasar atau kata berimbuhan. Jika token merupakan kata yang memiliki imbuhan maka akan melewati beberapa tahapan stemming yaitu menghapus *Inflection Suffix* (seperti -ku, -mu, -kah, dsb), menghapus *derivation suffix* (seperti -i, -an,atau -kan), dan menghapus *derivation prefix* (seperti di-, ke-,se-, dsb). Berikut hasil *stemming* menggunakan algoritma Nazief Adriani :

Tabel 4. Hasil *Stemming*

sesudah proses *stopword*	**sesudah proses *stemming***
ditemukan	temu
dompet	dompet
pria	pria
berwarna	warna

biru	biru
tua	tua
atm	atm
toko	toko
thio	thio
ktp	ktp
meily	meily
sinaniaoessy	sinaniaoessy

3.2 Pembobotan Kata dengan TF-IDF

Pada saat *text preprocessing* selesai, selanjutnya yaitu melakukan pembobotan kata (*term)*. Pembobotan dilakukan dengan *parsing* semua data hasil *preprocessing* kedalam *database* kemudian dihitung jumlah kemunculannya. Berikut adalah tabel dokumen yang status barangnya belum ditemukan, kemudian melewati hasil *preprocessing*.

Tabel 5. Dokumen Setelah *Preprocessing*

D	Term
Q	temu dompet pria warna biru tua atm toko thio ktp meily sinaniaoessy
D1	hilang dompet pria pasar warna hitam temu mohon eko
D2	hilang atm bri toko thio kampung baru temu tolong biya
D3	hilang dompet warna biru tua toko thio ktp meily sinaniaoessy temu mohon

1. *Term Frequency* (TF)

Term Frequency adalah frekuensi kemunculan term/kata yang terdapat pada dokumen. Berikut adalah *term frequency (TF)* dari tabel 4.

Tabel 6. *Term Frequency*

Term	Key (Q)	D1	D2	D3
atm	1	0	1	0
biru	1	0	0	1
dompet	1	1	0	1
tua	1	0	0	1
ktp	1	0	0	1
meily	1	0	0	1
pria	1	0	1	0
sinaniaoessy	1	0	0	1
temu	1	0	1	1
thio	1	0	1	1
toko	1	0	1	1
warna	1	1	0	1

2. *Document Frequency* (DF)

Document Frequency adalah jumlah dari dokumen yang mengandung kata (*term)* yang terdapat pada *keyword* (Q).

Tabel 7. *Document Frequency*

Term	DF
atm	1
biru	1
dompet	2
tua	1
ktp	1
meily	1
pria	1
sinaniaoessy	1
temu	2
thio	2
toko	2
warna	2

3. *Inverse Document Frequency* (IDF)

Inverse Document Frequency adalah *inverse* dari jumlah semua koleksi dokumen dibagi dengan jumlah dokumen yang mengandung kata (*term*) yang terdapat pada *keyword*.

Tabel 8. *Inverse Document Frequency*

Term	Df	Idf
atm	1	0,60205999
biru	1	0,60205999
dompet	2	0,30102999
tua	1	0,60205999
ktp	1	0,60205999
meily	1	0,60205999
pria	1	0,60205999
sinaniaoessy	1	0,60205999
temu	2	0,30102999
thio	2	0,30102999
toko	2	0,30102999
warna	2	0,30102999

4. *Term Frequency - Inverse Document Frequency (Tf-Idf)*

TF-IDF yaitu perhitungan yang menggambarkan seberapa pentingnya kata (*term*) dalam sebuah dokumen. Proses ini digunakan untuk menilai bobot relevansi *term* dari sebuah dokumen terhadap seluruh dokumen.

Tabel 9. *Term Frequency - Inverse Document Frequency*

Term	Tf . Idf			
	Q	D1	D2	D3
atm	0,60205999	0	0,60205999	0
biru	0,60205999	0	0	0,60205999
dompet	0,30102999	0,30102999	0	0,30102999

tua	0,60205999	0	0	0,60205999
ktp	0,60205999	0	0	0,60205999
meily	0,60205999	0	0	0,60205999
pria	0,60205999	0	0,60205999	0
sinaniaoessy	0,60205999	0	0	0,60205999
temu	0,30102999	0	0,30102999	0,30102999
thio	0,30102999	0	0,30102999	0,30102999
toko	0,30102999	0	0,30102999	0,30102999
warna	0,30102999	0,30102999	0	0,30102999

3.3 *Cosine Similarity*

Pembobotan kata dengan *Term Frequency – Inverse Document Frequency* (TF-IDF) telah dilakukan, langkah selanjutnya dalam pencarian dokumen yang sesuai dengan query yang dimasukkan adalah menghitung kemiripan antara dokumen dengan menggunakan rumus *cosine similarity.* Berikut langkah-langkahnya :

1. Langkah Pertama

Menghitung perkalian scalar antara *term* query dengan *term* dokumen menggunakan rumus :

$$\sum_{i=1}^{t}(Wqi \times Wdij) \qquad (7)$$

Dimana :

$$\sum_{i=1}^{t} = Mewakili\ total\ nilai\ TF\ IDF$$

W_{qi} = Nilai query ($TF\text{-}IDF_q$)
W_{dij} = Nilai Dokumen ($TF\text{-}IDF_d$)

Tabel 10. *Cosine Similarity Langkah Pertama*

Term	**Tf .Idf(q) x Tf.idf(d)**		
	D1	**D2**	**D3**
atm	0	0,362476	0
biru	0	0	0,362476
dompet	0,090619	0	0,090619
dongker	0	0	0,362476
ktp	0	0	0,362476
meily	0	0	0,362476
pria	0	0,362476	0
sinaniaoessy	0	0	0,362476
temu	0	0,090619	0,090619
thio	0	0,090619	0,090619
toko	0	0,090619	0,090619
warna	0,090619	0	0,090619
jumlah	**0,181238**	**0,996809**	**2,265475**

2. Langkah Kedua

Langkah kedua yaitu menghitung nilai panjang setiap dokumen termasuk *query* dengan mengkuadratkan bobot *query* dan bobot dokumen, kemudian jumlahkan nilai kuadrat kemudian hitung akarnya. Rumus yang digunakan adalah ... :

Tabel 11**.** Cosine Similarity Langkah Kedua

Term	**Tf . Idf**			
	Q	**D1**	**D2**	**D3**
atm	0,36247	0	0,362476	0
biru	0,36247	0	0	0,362476
dompet	0,09061	0,090619	0	0,090619
tua	0,36247	0	0	0,362476
ktp	0,36247	0	0	0,362476
meily	0,36247	0	0	0,362476
pria	0,36247	0	0,362476	0
sinaniaoessy	0,36247	0	0	0,362476
temu	0,090619	0	0,090619	0,090619
thio	0,09061	0	0,090619	0,090619
toko	0,09061	0	0,090619	0,090619
warna	0,090619	0,090619	0	0,090619
Jumlah	**2,99042**	**0,18124**	**0,99681**	**2,26548**
Akar	**1,7292**	**0,42572**	**0,9984**	**1,50515**

3. Langkah Ketiga

Langkah ketiga adalah membagi hasil dari perkalian skalar dan hasil panjang vektor yang sudah dihitung untuk menemukan hasil kemiripan antara query dengan dokumen, lalu sistem akan menampilkan dokumen yang relevan dengan query berdasarkan hasil perhitungan kemiripan dengan cosine similarity.

$$Similarity\ (D1, Q) = \frac{0{,}181238}{1{,}7292851 \times 0{,}42572} = 0{,}246183$$

$$Similarity\ (D2, Q) = \frac{0{,}996809}{1{,}7292851 \times 0{,}9984} = 0{,}577352$$

$$Similarity\ (D3, Q) = \frac{2{,}265475}{1{,}7292851 \times 1{,}50515} = 0{,}870388$$

Berdasarkan hasil perhitungan cosine similarity dimana persamaan dari query dan D1 bernilai 0,246183, untuk D2 bernilai 0,577352, dan untuk D3 bernilai 0,870388, dapat disimpulkan bahwa dokumen yang terdekat dengan query yang diinputkan adalah D3 karna hukum *cosine similarity* adalah semakin besar nilai cosinus (maksimal 1) berarti semakin dekat juga dengan dokumen yang dibandingkan. Sehingga yang menjadi *output* dari pencarian *query* adalah D3.

3.4 Hasil Penelitian

Gambar 2. Pencarian Dokumen

Pada gambar 2, terdapat 3 tab yaitu yang pertama tab fitur *search* yang berguna untuk mencari barang yang hilang atau ditemukan. Tab *Lost* hanya menampilkan barang hilang. Tab *Found* menampilkan barang yang telah ditemukan. Pada bagian atas terdapat tab home untuk melihat profil dari pemilik akun. Kemudian disebelah kanannya terdapat tombol untuk logout. Tombol untuk menginput kasus baru terdapat disebelah kanan dengan simbol (+). Contoh diatas yaitu *user* menginputkan query barang temuan seperti ini : (telah hilang dompet berwarna biru tua di Toko Thio dengan KTP atas nama meily siyaniaoessy). Kemudian sistem akan menampilkan hasil pencarian yang relevan ditengah *layout* seperti terlihat pada gambar 2 diatas.

3.5 Pengujian Sistem

Pada saat pengujian sistem dengan algoritma *stemming* ada beberapa data kata yang gagal di *stemming*. Hal ini terjadi karena proses *stemming* mendahulukan penghapusan imbuhan diakhir kata kemudian diawal kata. Sehingga terjadi *stemming* yang berlebihan. Selain itu ada beberapa kata yang gagal karena ada aturan imbuhan dari kata yang tidak terdapat pada aturan dari *stemming*, sehingga kata dasar tidak berhasil didapatkan.

Pada tahap pengujian dilakukan dengan menggunakan rumus dari *confusing matrix.* Dimana rumus ini biasa digunakan dalam melakukan pengujian untuk menghitung akurasi pada sebuah sistem untuk mengevaluasi kemampuan sistem dalam melakukan pencarian. Pengujian dilakukan sebanyak 50 kali menggunakan query, dimana setiap query yang dimasukkan mewakili 1 dokumen. 1 query seharusnya memiliki minimal 1 jawaban yang relevan dengan koleksi dokumen yang tersedia. Pada saat melakukan pengujian terhadap 50 query, terdapat 44 jawaban yang relevan dengan dokumen yang ditampilkan dan terdapat 6 jawaban yang tidak relevan. Berdasarkan pengujian tersebut, didapatkan tingkat akurasi dari sistem dengan rumus *confusing matrix* sebagai berikut :

$$Precision = \frac{44}{44+0} \times 100\% = 100\%$$

$$Recall = \frac{44}{44+6} \times 100\% = 88\%$$

$$Accuracy = \frac{44}{44+0+0+6} \times 100\% = 88\%$$

$$Error\ Rate = \frac{6+0}{44+0+0+6} \times 100\% = 12\%$$

4. KESIMPULAN

4.1 Kesimpulan

Penelitian telah mencapai hasil akhir, dimana didapatkan kesimpulan bahwa metode *Term Frequency – Inverse Document Frequency* (TF-IDF) dan *Cosine Similarity* berhasil diterapkan pada aplikasi "*Lost & Found*". Aplikasi dapat menghasilkan dokumen yang relevan sampai dengan tingkat akurasi dan *recall* 88%. *Error Rate* hanya 12%. Berdasarkan hasil pengujian tersebut dapat dikatan bahwa aplikasi dapat berjalan dengan baik.

4.2 Saran

Aplikasi "*Lost & Found*" yang telah dibuat tentu masih ada kekurangan. Diharapkan kedepan ada pengembangannya agar aplikasi menjadi lebih baik lagi. Berikut beberapa saran pengembangan untuk kedepannya :

1. Aplikasi ini bisa diperluas ruang lingkupnya, bukan hanya dapat digunakan pada 1 kota aja, tetapi bisa untuk seluruh kota di Indonesia.
2. Aplikasi ini kedepannya dapat dikembangkan agar lebih *user friendly.*
3. Aplikasi bisa dikembangkan dengan metode yang dapat memberikan *output* yang hasilnya lebih akurat.

DAFTAR PUSTAKA

[1] D. Rizki and E. Supraptono, "Penerapan Algoritma Cosine Similarity dan Pembobotan TF-IDF pada Sistem Klasifikasi Dokumen," *Jurusan Teknik Elektro,* 2017.

[2] I. Sommerville, Software Engineering 9th Edition, Addison-Wesley, 2011.

[3] Y. Latif, "Sistem Pencarian dan Pengumuman Barang Hilang Berbasis Android," *Jurnal Jurusan Teknik Informatika. Fakultas Sains dan Teknologi.,* 2016.

[4] A. A. Maarif, "Penerapan Algoritma TF-IDF untuk Pencarian Karya Ilmiah," *Jurnal Teknik Informatika,* 2016.

[5] V. H. d. T. Ria, "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: Syarah Umdatil Ahkam).," *Jurnal Teknik Informatika*, 2018.

[6] A. Rachman, "Pengembangan Aplikasi Lost And Found Berbasis Website Dengan Fitur Pencarian Menggunakan Cosine Similarity.," Jurusan Teknik Informatika. Fakultas Teknik. Universitas Muhammadiyah Malang., Malang, 2017.

[7] A. d. M. Setyoko, "Pencarian Pasal Pada Kitab Undang-Undang Hukum Pidana (Kuhp) Berdasarkan Kasus Menggunakan Metode Cosine Similarity Dan Latent Semantic Indexing (Lsi).," *Journal of Environmental Engineering & Sustainable Technology.,* 2015.

[8] D. d. Azhari, "Peringkasan Sentimen Esktraktif di Twitter Menggunakan Hybrid TF-IDF dan Cosine Similarity," *IJCCS,* vol. Volume 10 Nomor 2.

[9] R. N. Safitri, "Temu Kembali Informasi Pada Pencarian Jurnal Skripsi dengan Menggunakan Metode Single Pass Clustering," Universitas Muhammadiyah Gresik, 2013.

[10] R. d. S. J. Feldman, The Text Mining Handbook Advanced Approaches in Analyzing Unstructured Data., New York: Cambridge University Press, 2007.

[11] D. d. E. S. Rizki, "Penerapan Algoritma Cosine Similarity dan Pembobotan TF-IDF pada Sistem Klasifikasi Dokumen.," Universitas Negeri Semarang, 2017.

[12] I. Arwanda, "Penerapan Metode Text Mining pada Aplikasi Chatbot.," Universitas Ilmu Komputer. Bandung., 2013.

[13] A. A. A. Putra, "Implementasi Text Summarization Model pada Artikel Berita Bahasa Indonesia," Universitas Komputer Indonesia, 2016.

[14] A. d. A. Dewa, "Pengukuran Kemiripan Dokumen Teks Bahasa Indonesia Menggunakan Metode Cosine Similarity.," *Jurnal Teknik Informatika*, 2016.

[15] I. R. d. S. R. Rozas, "Sistem Pemilihan Kontrol Keamanan Informasi Berbasis ISO 27001," in *Seminar Nasional Pascasarjana XI. IT.*, Surabaya, 2011.

[16] S. Ferdinandus, "Sistem Information Retrieval Layanan Kesehatan Untuk Berobat dengan Metode Vector Space Model berbasis WebGis.," *Jurnal Teknik Informatika.*

[17] R. Pressman, Rekayasa Perangkat Lunak: PendekatanPraktisi(Buku Dua)., Yogyakarta: Andi, 2002.

LAMPIRAN

		
Tampilan Login	**Tampilan Register**	**Tampilan Add Kasus**
		
Tampilan Barang Ketemu	**Tampilan Barang Hilang**	**Tampilan Pengiriman Pesan**