**BAB I**

# PENDAHULUAN

## 1.1 Latar Belakang

Kemajuan teknologi pada saat ini mendorong semua kalangan untuk mencari dan memberikan informasi dengan cepat dan mudah. Berbagai kegiatan yang dulu dilakukan secara manual pun sekarang dapat dilakukan secara online dan sistematis. Hal tersebut mendorong orang untuk mendapatkan dan menyalurkan berbagai informasi dan inovasi. Tidak salah jika saat ini banyak ditemui pemikiran-pemikiran baru yang mempunyai nilai guna tinggi. Saat ini banyak ditemui hal-hal baru, yang awalnya dianggap hanya hal yang tidak mungkin, namun dengan berkembangnya ilmu teknologi hal-hal tersebut dapat menjadi mungkin.Saat ini inovasi yang sangat berguna untuk membantu kehidupan manusia banyak dicari. Dengan berbagai cara, para ahli mencari temuan-temuan dan ciptaan-ciptaan baru yang diluar pemikiran manusia pada umumnya. Dari latar belakang tersebut, terciptalah suatu kompetisi online yang bergerak dibidang inovasi yaitu Indonesia ICT Award (INAICTA).

Indonesia ICT Award (INAICTA) sudah diadakan sejak tahun 2007

yang merupakan ajang lomba karya cipta kreativitas dan inovasi di bidang TIK terbesar di Indonesia. Tujuan diselenggarakannya INAICTA adalah untuk mendorong terus berkembangnya produk-produk TIK lokal dengan peningkatan kualitas maupun inovasi produk.Tidak hanya bagi pengembang individu, tapijuga bagi komunitas untuk pemberdayaan masyarakat [(www.inaicta.web.id/inaicta/)](http://www.inaicta.web.id/inaicta/). Ajang lomba ini dapat diikuti oleh kalangan pelajar (SD, SMP, SMA/SMK),

1

mahasiswa perguruan tinggi, maupun kalangan profesional

(www.inaicta.web.id/lomba). Untuk dapat berpartisipasi dalam INAICTA, para kontestan harus memiliki Hak Atas Kekayaan Intelektual dari tiap karya yang diikutsertakan (www.inaicta.web.id/lomba/syarat-dan-ketentuan).

Semakin tahun, jumlah kontestan yang mengikuti INAICTA semakin

bertambah. Hal tersebut berpengaruh terhadap tingkat kesulitan bagi para juri atau tim penilai untuk mengetahui kemiripan dari inovasi-inovasi dari para kontestan. Dibutuhkan suatu aplikasi yang dapat membantu dalam pendeteksian kemiripan tiap hasil karya yang diikutsertakan oleh para kontestan. Algoritma *Term Frequency – Inversed Document Frequency* (TF-IDF) serta *Vector Space Model* digunakan dalam pengembangan sistem pendeteksi kemiripan karya pada INAICTA dengan menggunakan kata kunci dari tiap dokumen penjelasan ringkas karya.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, maka rumusan masalah yang didapatkan adalah :

1. Bagaimana membangun sistem yang dapat mendeteksi kemiripan karya berdasarkan penjelasan ringkas karya pada INAICTA 2013.
2. Bagaimana menampilkan urutan tingkat kemiripan karya berdasarkan penjelasan ringkas karya pada INAICTA 2013 menggunakan algoritma *Term Frequency – Inversed Document Frequency* (TF-IDF) dan*Vector*

*Space Model*.

3

## 1.3 Batasan Masalah

Batasan masalah dalam pembuatan sistem ini, diantaranya sebagai

berikut :

1. Studi kasus yang diteliti adalah karya pada INAICTA tahun 2013.
2. Hanya diperuntukkan kata dalam bahasa Indonesia.
3. Data yang digunakan diperoleh berdasarkan data pada Indonesia ICT

Award (INAICTA) 2013

.

d.

Implementasi aplikasi menggunakan bahasa pemrograman

*PHP*

dan

*MySQL*

.

e.

*Input*

berupa berbagai

penjelasan ringkas karya

kontestan

INAICTA

2013

.

f.

*Output*

berupa hasil

peringkat kemiripan

.

**1.4**

**Tujuan Penelitian**

Berdasarkan rumusan masalah yang didapat, maka tujuan dari pembuatan

sistem ini adalah :

a.

Membangun sebuah sistem yang dapat mendeteksi kemiripan karya

berdasarkan penjelasan ringkas karya pada INAICTA

2013.

b.

Pengguna dapat mengetahui urutan tingkat kemiripan karya berdasarkan

penjelasan ringkas karya pada INAICTA 2013 menggunakan algoritma



*Term Frequency – Inversed Document Frequency* (TF-IDF) dan *Vector Space Model*.

## 1.5 Manfaat Penelitian

Manfaat yang diperoleh dari pembuatan sistem ini yaitu :

1. Memberikan informasi kemiripan karya berdasarkan penjelasan ringkas karya pada INAICTA 2013.
2. Memberikan informasi urutan tingkat kemiripan karya berdasarkan penjelasan ringkas karya pada INAICTA 2013menggunakan algoritma

*Term Frequency*

*–*

*Inversed Document Frequency*

(

TF

-

IDF) dan

*Vector*

*Space Model*

.



**BAB II**

# LANDASAN TEORI

## 2.1 Studi Penelitian Terdahulu

Berdasarkan hasil penelitian dari review terdahulu, ditemukan penelitian sejenis tentang *TF IDF*dan *VSM* yang dilakukan oleh Adhit Herwansyah, mahasiswa Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas

Gunadarma(2009), dengan judul “Aplikasi Pengkategorian Dokumen dan

PengukuranTingkat Similaritas Dokumen Menggunakan Kata KunciPada

Dokumen Penulisan IlmiahUniversitas Gunadarma”. Pada penelitian tersebut, aplikasi yang dibangun dapat memberikan informasi tentang kemmiripan dokumen dengan menentukan kata kunci sebagai pembandingnya.

Selain penelitian yang dilakuakn oleh Adhit Hermawansyah, penulis menemukan penelitian sejenis yang dilakukan oleh Nova Khairunnisa, Dadang Syarif SS, dan Ardianto Wibowo, mahasiswa Program Studi Teknik Informatika dan Multimedia, Politeknik Caltex Riau (2012), dengan judul “Aplikasi Pendeteksi Plagiat dengan Menggunakan Metode *Latent Semantic Analysis* (Studi

Kasus : Laporan TA PCR)”. Pada penelitian tersebut,aplikasi pendeteksian kemiripan dua atau lebih dokumen dapat digunakan untuk mendeteksi apakah suatu dokumen yang dibuat dapat dikatakan plagiat atau tidak.

5

## 2.2 Studi Pustaka

### 2.2.1 *Text Mining*

*Text mining* adalah suatu proses menggali informasi dimana seorang *user* berinteraksi dengan sekumpulan dokumen yang menggunakan *tools* analisis yang merupakan komponen-komponen dalam *data mining* yang salah satunya adalah kategorisasi. Tujuan dari *text mining* adalah untuk mendapatkan informasi yang bergunadari sekumpulan dokumen sehingga sumber data yang digunakan pada *text mining* adalah kumpulan teks yang memiliki format yang tidak terstruktur atau minimal semi terstruktur (Triawati,2009:1).

*Text mining* bisa dianggap subjek riset yang tergolong baru. *Text mining* dapat memberikan solusi dari permasalahan seperti pemrosesan, pengorganisasian / pengelompokkan dan menganalisa *unstructured text* dalam jumlah besar. Dalam memberikan solusi, *text mining* mengadopsi dan mengembangkan banyak teknik dari bidang lain, seperti *Data mining*, *Information Retrieval, Statistik dan Matematik, Machine Learning, Linguistic, Natural Languange Processing, dan Visualization*. Kegiatan riset untuk *text mining* antara lain ekstraksi dan penyimpanan *text, preprocessing* akan konten *text*, pengumpulan data statistik dan *indexing* dan analisa konten(Triawati,2009:1).

Didalam proses *text mining* dilakukan beberapa tahapan umum diantaranya adalah tokenizing, filtering, stemming, tagging, dan analyzing.

Tahap *tokenizing* adalah tahap pemotongan string input berdasarkan tiap

kata yang menyusunnya. Setelah teks input dilakukan proses *tokenizing*, maka tahap selanjutnya dilakukan tahap *filtering* yaitu tahap mengambil kata-kata penting dari hasil token. Tahap selanjutnya adalah tahap *stemming* adalah tahap mencari dasar kata dari tiap kata hasil *filtering*. Setiap kata yang memiliki imbuhan seperti imbuhan awalan dan akhiran maka akan diambil kata dasarnya. Tahap berikutnya adalah Tahap *tagging* yang merupakan tahap mencari bentuk awal/*root* dari tiap kata lampau atau kata hasil *stemming*. Tahap ini tidak dipakai untuk teks bahasa Indonesia dikarenakan bahasa Indonesia tidak memiliki bentuk lampau. Tahap yang terakhir dalam *text mining* adalah tahap *analyzing* yaitu tahap penentuan seberapa jauh keterhubungan antar kata-kata antar dokumen yang ada (Herwansyah,2009).

#### 2.2.1.1 Tokenizing

Tokenisasi merupakan proses pemisahan suatu rangkaian karakter berdasarkan karakter spasi, dan mungkin pada waktu yang bersamaan dilakukan juga proses penghapusan karakter tertentu, seperti tanda baca.

Sebagai contoh, kata-kata “*computer*”, “*computing*”, dan

“*compute*” semua berasal dari term yang sama yaitu “*comput*” , tanpa pengetahuan sebelumnya dari morfologi bahasa Inggris. Token seringkali disebut sebagai istilah (*term*) atau kata, sebagai contoh sebuah token merupakan suatu urutan karakter dari dokumen tertentu yang dikelompokkan sebagai unit semantik yang berguna untuk diproses (Amin,2012).

#### 2.2.1.2 Filtering

Eliminasi *stopwords* memiliki banyak

keuntungan, yaitu akan mengurangi *space* pada tabel *term index* hingga 40% atau lebih (Baeza, 1999). Proses

*stopword removal*

merupakan proses penghapusan

*term*

yang tidak memiliki arti atau tidak relevan. Proses ini

dilakukan pada saat proses tokenisasi. Proses

*Filtering*

menggunakan daftar

*stopword*

yang digunakan oleh Tala

(2003)

, yang merupakan

*stopword*

bahasa Indonesia yang

berisi kata

-

kata seperti; ada, yang,

ke, kepada, dan lain

sebagainya

(

Amin,

2012).

**2.2.1.3**

***Stemming***

Proses

*Stemming*

digunakan untuk mengubah

*term*

yang masih melekat dalam term tersebut awalan,

sisipan, dan akhiran. Proses stemming dilakukan dengan

cara menghilangkan semua imbuhan (

*affixes*

)

baik yang

t

erdiri dari awalan

*(*

*prefixes*

)

, sisipan

*(*

*infixes*

*)*

, akhiran



*(suffixes)* dan *confixes* (kombinasi dari awalan dan akhiran) pada kata turunan. *Stemming* digunakan untuk mengganti bentuk dari suatu kata menjadi kata dasar dari kata tersebut yang sesuai dengan struktur morfologi bahasa Indonesia yang benar (Amin,2012).

Gambar

2.1

*Flowchart Stemming*

Bahasa Indonesia

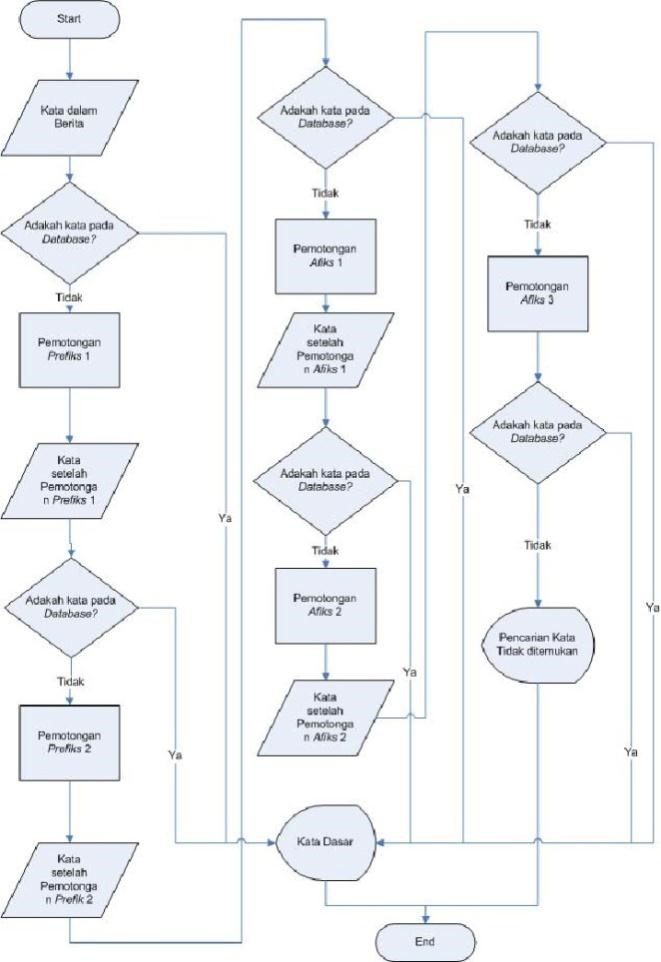
Menurut Agusta (2009), a

lgoritma yang dibuat

oleh Bobby Nazief dan Mirna Adriani ini memiliki tahap

-

tahap sebagai berikut:



1. Cari kata yang akan distem dalam kamus. Jika ditemukan maka diasumsikan bahwa kata tesebut adalah *root word*. Maka algoritma berhenti.
2. Inflection Suffixes(“-lah”, “-kah”, “-ku”, “-mu”, atau “nya”) dibuang. Jika berupa particles (“-lah”, “-kah”, “tah” atau “-pun”)maka langkah ini diulangi lagi untuk menghapus *Possesive Pronouns*(“-ku”, “-mu”, atau “-

nya”), jika ada.

1. Hapus *Derivation Suffixes*(“-i”, “-an” atau “-kan”). Jika

kata ditemukan di kamus, maka algoritma berhenti. Jika

tidak maka ke langkah 3a

a.

Jika “

-

an” telah dihapus dan huruf terakhir dari kata

tersebut adalah “

-

k”, maka “

-

k” juga ikut dihapus.

Jika kata tersebut ditemukan dalam kamus maka

algoritma berhenti. Jika tidak ditemukan maka

lakukan langkah 3b.

b.Akhiran yang dihapus (“

-

i”, “

-

an” at

au “

-

kan”)

dikembalikan, lanjut ke langkah 4.

4.

Hapus

*Derivation Prefix*

. Jika pada langkah 3 ada

sufiks yang dihapus maka pergi ke langkah 4a, jika

tidak pergi ke

langkah 4b.

a.

Periksa tabel kombinasi awalan

-

akhiran yang tidak



diijinkan. Jika ditemukan maka algoritma berhenti, jika tidak pergi ke langkah 4b.

b. For i = 1 to 3, tentukan tipe awalan kemudian hapus awalan. Jika *root word* belum juga ditemukan lakukan langkah 5, jika sudah maka algoritma berhenti. Catatan: jika awalan kedua sama dengan awalan pertama algoritma berhenti.

* 1. Melakukan *Recoding*.
  2. Jika semua langkah telah selesai tetapi tidak juga berhasil maka kata awal diasumsikan sebagai *root*

*word*

. Proses selesai.

Tipe awalan ditentukan melalui langkah

-

langkah

berikut:

1.

Jika awalannya adalah: “di

-

”, “ke

-

”, atau “se

-

” maka

tipe awalannya secara berturut

-

turut adalah “di

-

”, “ke

-

”,

atau “se

-

”.

2.

Jika awalannya adalah “te

-

”, “me

-

”, “be

-

”, atau “pe

-

”

maka dibutuhkan sebuah proses tambahan untuk

menentukan tipe awalannya.

3.

Jika

dua karakter pertama bukan “di

-

”, “ke

-

”, “se

-

”,

“te

-

”, “be

-

”, “me

-

”, atau “pe

-

” maka berhenti.

4.

Jika tipe awalan adalah “none” maka berhenti. Jika tipe

awalan adalah bukan “none” maka awalan dapat dilihat



pada tabel 2. Hapus awalan jika ditemukan.

Tabel 2.1Kombinasi Awalan Akhiran Yang Tidak

|  |  |
| --- | --- |
| Awalan | Akhiran yang tidak diijinkan |
| be- | -i |
| di- | -an |
| ke- | -i, -kan |
| me- | -an |
| se- | -i, -kan |

Diijinkan

“

-

r

-

“

not

(

vowel

or “

-

r

-

”)

not “

-

er

-

“

-

Ter

not

(

vowel

or “

-

r

-

”)

“

-

er

-

“

vowel

-

none

not

(

vowel

or “

-

r

-

”)

“

-

er

-

“

not

vowel

-

Te



Tabel 2.3Jenis Awalan Berdasarkan Tipe Awalannya

Tipe Awalan

Awalan yang harus dihapus

di

-

di

-

ke

-

ke

-

se

-

se

-

te

-

te

-

ter

-

ter

-

ter

-

luluh

Ter

Untuk mengatasi keterbatasan pada algoritma di

atas, maka ditambahkan aturan

-

aturan dibawah ini:

1.

Aturan untuk reduplikasi.

a.

Jika kedua

kata yang dihubungkan oleh kata

penghubung adalah kata yang sama maka root

wordadalah bentuk tunggalnya, contoh : “buku

-



buku” root wordnya adalah “buku”.

b. Kata lain, misalnya “bolak-balik”, “berbalasbalasan, dan ”seolah-olah”. Untuk mendapatkan

root wordnya, kedua kata diartikan secara terpisah.

Jika keduanya memiliki root wordyang sama maka diubah menjadi bentuk tunggal, contoh: kata

“berbalas-balasan”, “berbalas” dan “balasan” memiliki root wordyang sama yaitu “balas”, maka root word“berbalas-balasan” adalah “balas”. Sebaliknya, pada kata “bolak-balik”, “bolak” dan

metode

“balik” memiliki root wordyang berbeda, maka root

wordnya

adalah “bolak

-

balik”

2.

Tambahan bentuk awalan dan akhiran serta aturannya.

a.

Untuk tipe awalan “mem

-

“, kata yang diawali

dengan awalan “memp

-

” memiliki tipe awalan

“mem

-

”.

b.

Tipe awalan “meng

-

“, kata yang diawali dengan

awalan “mengk

-

” memiliki tipe awalan “meng

-

”.

**2.2.2**

***Term Frequency***

***–***

***Inversed Document Frequency***

**(**

**TF**

**-**

**IDF)**

Metode

TF

-

IDF

(

Robertson,

2004)

merupakan

untukmenghitung bobot setiap kata yang paling umum digunakan

pada

*information retrieval*

. Metode ini juga terkenal efisien, simpel

danmemiliki hasil yang akurat (Ramos, 2010). Metode ini akan



menghitungnilai *Term Frequency* (TF) dan *Inverse Document Frequency* (IDF) padasetiap token (kata) di setiap dokumen dalam korpus (Suliantoro,

2012).

Dalam tahapan perhitungan *term frequency (tf)* menggunakan persamaan(1)

## *tf = tfij (1)*

Dengan *tf* adalah *term frequency,* dan *tfij*adalah banyaknya kemunculan *term* ti dalam dokumen dj, *Termfrequency (tf)* dihitung dengan

menghitung banyaknya kemunculan term t

i

dalam dokumen d

j

.

Perhitungan

*Inverse Document Frequency (idf),*

menggunakan

persamaan (2)

***(2)***

Dengan

*idf*

*i*

adalah

*inverse document frequency, N*

adalah jumlah

dokumen yang terambil oleh sistem, dan

*df*

*i*

adalah banyaknya dokumen

dalam koleksi dimana

*termt*

*i*

muncul di dalamnya, maka perhitungan

*idf*

*i*

digunakan untuk mengetahui banyaknya

*term*

yang dicari (

*df*

*i*

)

yang muncul

dalam dokumen lain yang ada pada

*database*

.

Perhitungan

*termfrequency Inverse Document Frequency (tfidf)*

,

menggunakan persamaan (3)

***idf***

***i***

***=***

***log***

𝑵

𝒅𝒇𝒊



## *Wij = tfi . idfi (3)*

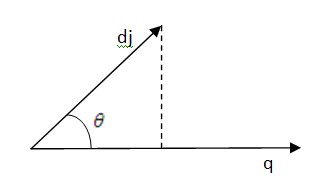
Dengan *Wij*adalahbobot dokumen. Bobot dokumen (*Wij*) dihitung untuk didapatkannya suatu bobot hasil perkalian atau kombinasi antara *termfrequency* (*tfi*) dan *inverse document frequency* (*idfi*) (Amin, 2012).

### 2.2.3 *Vector Space Model*

*Vector Space Model (VSM)* adalah metode untuk melihat tingkat kedekatan atau kesamaan *(similarity)* term dengan cara pembobotan *term*. Dokumen dipandang sebagi sebuah vektor yang memiliki *magnitude* (jarak) dan *direction* (arah). Pada *Vector Space Model,* sebuah istilah direpresentasikan dengan sebuah dimensi dari ruang vektor. Relevansi sebuah dokumen ke sebuah *query* didasarkan pada similaritas diantara vektor dokumen dan vektor *query* (Baeza, 1999)(Amin, 2012).

VSM memberikan sebuah kerangka pencocokan parsial adalah mungkin. Hal ini dicapai dengan menetapkan bobot non-biner untuk istilah indeks dalam query dan dokumen. Bobot istilah yang akhirnya digunakan untuk menghitung tingkat kesamaan antara setiap dokumen yang tersimpan dalam sistem dan permintaan user. Dokumen yang terambil disortir dalam urutan yang memiliki kemiripan, model vektor memperhitungkan pertimbangan dokumen yang relevan dengan permintaan user. Hasilnya adalah himpunan dokumen yang terambil jauh lebih akurat (dalam arti sesuai dengan informasi yang dibutuhkan oleh *user*) (Amin, 2012).

Sebuah dokumen dj dan sebuah *query* q direpresentasikan sebagai vektor t-dimensi seperti pada Gambar 2.



Gambar 2

.

2

The Cosines of is adopted as sim dj, q

(

Amin,

2012)

Dalam VSM koleksi dokumen direpresentasikan sebagai sebuah

matrik

*term*

document (atau matrik

*term frequency*

. Setiap sel dalam matrik

)

bersesuaian dengan bobot yang diberikan dari suatu

*term*

dalam dokumen

yang ditentukan. Nilai nol berarti bahwa term tersebut t

idak ada dalam

dokumen. Gambar

3

menunjukkan matrik

*term document*

d

engan n

dokumen dan t

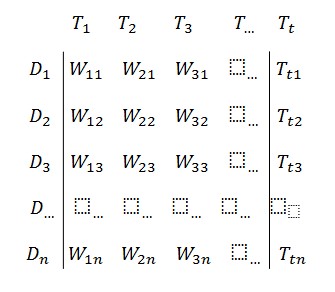
*term*

(

Amin,

2012)

.



Gambar 2.3 Matrik *term-document* (Amin, 2012) Perhitungan jarak *query* menggunakan persamaan (4) dan untuk dokumen menggunakan persamaan (5)

# |*q*| =

Dengan |q| adalah jarak *query* dan *Wiq*adalah bobot *query* dokumen ke-i, maka jarak *query* |q| dihitung untuk didapatkan jarak *query* dari bobot *query* dokumen (*Wiq*) yang terambil oleh sistem. Jarak *query* bisa dihitung dengan persamaan akar jumlah kuadrat dari *query*(Amin, 2012).

# |*dj*| = *(5)*

Dengan |dj| adalah jarak dokumen dan *Wij* adalah bobot dokumen ke-i, maka jarak dokumen (|dj|) dihitung untuk didapatkan jarak dokumen dari bobot dokumen dokumen (*Wij*) yang terambil oleh sistem. Jarak dokumen bisa dihitung dengan persamaan akar jumlah kuadrat dari dokumen(Amin, 2012).

√

∑

(

𝑾

𝒊𝒋

)

𝟐

𝒕

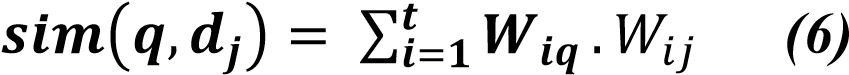
𝒊

=

𝟏



Perhitungan pengukuran *similarity query document* (*inner product*), menggunakan persamaan (6)



Dengan *Wij* adalah bobot *term* dalam dokumen, *Wiq* adalah bobot *query*, dam *sim*(*q,dj*) adalah similaritas antara *query* dan dokumen. Similaritas antara *query* dan dokumen atau *inner product / sim*(*q,dj*) digunakan untuk mendapatkan bobot dengan didasarkan pada bobot *term* dalam dokumen (*Wij*) dan bobot *query* (*Wiq*) atau dengan cara menjumlahkan bobot *q* dikalikan dengan bobot dokumen(Amin, 2012).

***(7)***

Pengukuran

*cosine similarity*

(

menghitung nilai kosinus sudut antara

dua vektor) menggunakan persamaan (7)

𝒔𝒊𝒎

(

𝒒

,

𝒅

𝒋

)

=

𝒒

.

𝒅

𝒋

|

𝒒

|

∗

|

𝒅

𝒋

|

**=**

∑

𝑾

𝒊𝒒

.

𝑾

𝒊𝒋

𝒕

𝒊

=

𝟏

√

∑

(

𝑾

𝒋

)

𝟐

𝒕

𝒋

=

𝟏

∗

√

∑

(

𝑾

𝒊

)

𝟐

𝒕

𝒊

=

𝟏

Similaritas antara

*query*

dan dokumen

*sim*

(

*q,d*

*j*

)

berbanding lurus

terhadap jumlah bobot

*query*

(

*q*

dikali bobot dokumen

)

(

*d*

*j*

)

dan berbanding

terbalik terhadap akar jumlah kuadrat

*q*

|

(

*q*

|) dikali dengan akar

jumlah

kuadrat dokumen (|

*d*

*j*

|). Perhitungan similaritas menghasilkan bobot

dokumen yang mendekati nilai 1 atau menghasilkan bobot dokumen yang

lebih besar dibandingkan dengan nilai yang dihasilkan dari perhitungan

*inner product*

Amin

(

, 2012).

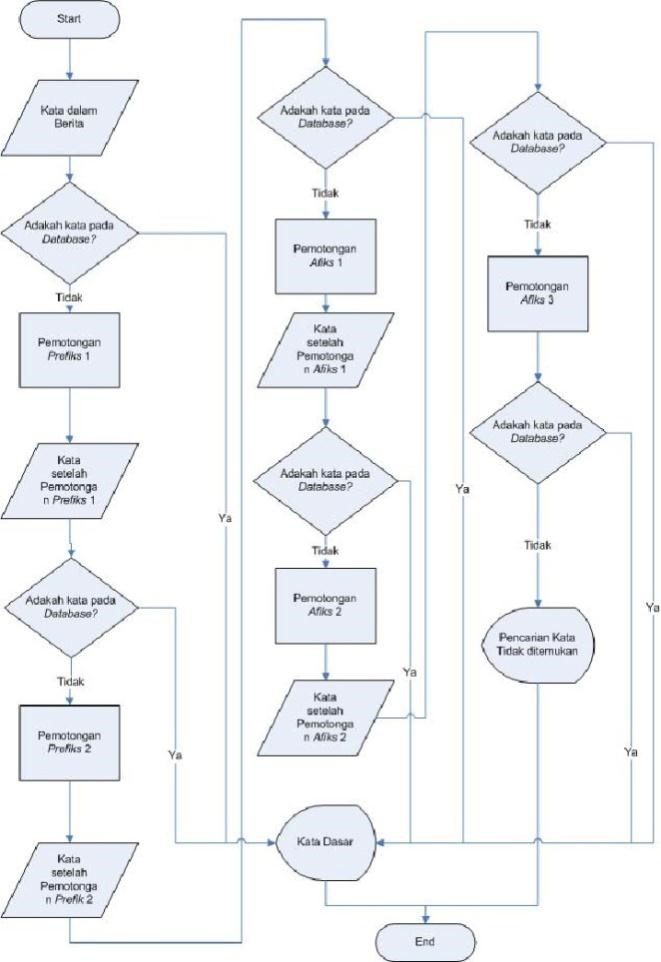


## 2.2.4 *PHP*

*PHP* singkatan dari *Personal Home Page Tools*, adalah sebuah bahasa scripting yang dibundel dengan *HTML*, yang di jalankan di sisi server.

Sebagian besar perintahnya berasal dari *C*, *Java* dan *Perl* dengan beberapa tambahan fungsi khusus *PHP*. Bahasa ini memungkinkan para pembuat aplikasi *web* menyajikan halaman *HTML* dinamis, interaktif dengan cepat dan mudah, yang dihasilkan *server*. *PHP* juga dimaksudkan untuk mengganti teknologi lama seperti *CGI (Common Gateway Interface)*.

*PHP* bisa berinteraksi dengan hampir semua teknologi *web* yang sudah ada. *Developer* bisa menulis sebuah program *PHP* yang

mengeksekusi suatu program *CGI* di *server* lain. Fleksibilitas ini sangat bermanfaat bagi pemilik situs-situs *web* yang besar dan sibuk, karena pemilik masih bisa mempergunakan aplikasi-aplikasi yang sudah terlanjur dibuat sebelumnya dengan *CGI, ISAP*, atau dengan *script* seperti *Perl, Awk,* atau *Python* selama proses migrasi ke aplikasi yang baru dibuat oleh *PHP*. Ini mempermudah dan memperhalus peralihan antara teknologi lama dan teknologi baru (Subekti, Susanto, 2013).

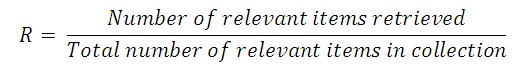
## 2.2.5 *MySQL*

Dalam bahasa SQL pada umumnya informasi tersimpan dalam tabeltabel yang secara logik merupakan struktur dua dimensi terdiri dari baris (row atau record) dan kolom (column atau field). Sedangkan dalam sebuah database dapat terdiri dari beberapa table. MySQL adalah database jenis RDBMS (Relational Database Management System). Jadi dalam MySQL tetap menggunakan Table, Baris dan Kolom. Sebuah Database dalam MySQL mengandung beberapa table dan satu table dalam database terdiri dari sejumlah baris dan kolom(Anymous, 2010).

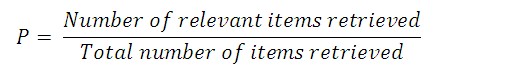
## 2.2.6 Uji *Recall* dan *Precision*

Tujuan uji *Recall* dan *Precision* adalah untuk mendapatkan informasi hasil pencarian yang didapatkan oleh STKI*.* Hasil pencarian STKI bisa dinilai tingkat *recall* dan *precision* nya. *Precision* dapat dianggap sebagai ukuran ketepatan atau ketelitian, sedangkan *recall* adalah kesempurnaan. Nilai *precision* adalah proporsi dokumen yang terambil oleh sistem adalah relevan. Nilai *recall* adalah proporsi dokumen relevan yang terambil oleh sistem (Amin, 2012).

Nilai *recall* dan *precision* bernilai antara 0 sd 1. Sistem temu kembali informasi diharapkan untuk dapat memberikan nilai *recall* dan *precision* mendekati 1. Pengguna rata-rata ingin mencapai nilai *recall* tinggi dan *precision* tinggi, pada kenyataannya hal itu harus dikompromikan karena sulit dicapai

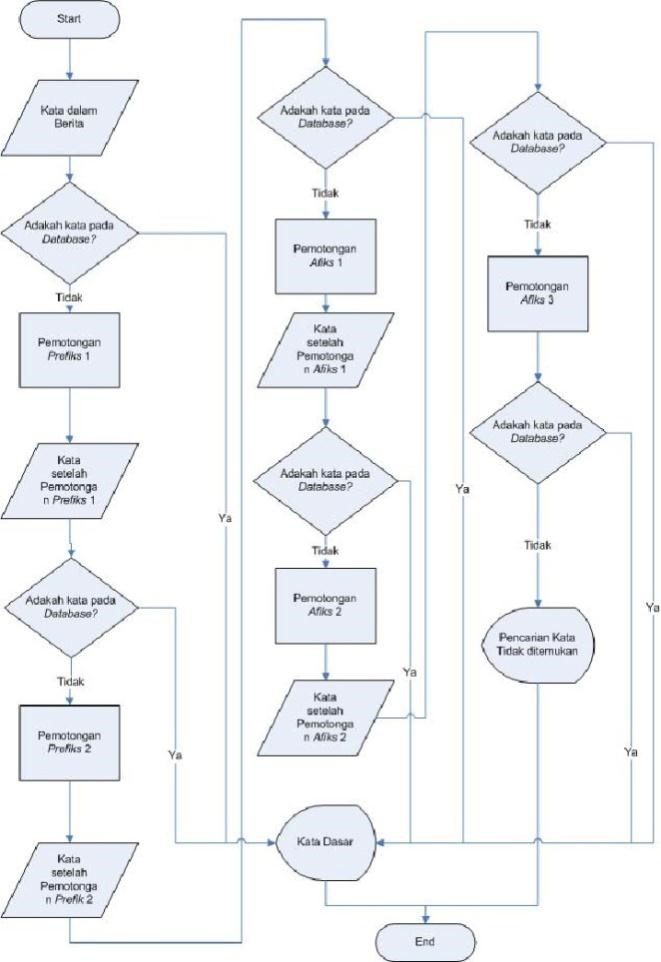


Dengan *R* adalah *recall,*maka nilai *R* didapatkan dengan membandingkan *Number of relevant items retrieved* dengan *Total number of relevant items in collection*. *Recall* adalah dokumen yang terpanggil dari STKI sesuai dengan permintaan *user* yang mengikuti pola dari STKI. Nilai *recall* makin besar belum bisa dikatakan suatu STKI baik atau tidak.



Dengan *P* adalah *Precision.* maka nilai *P* didapatkan dengan membandingkan *Number of relevant items retrieved* dengan *Total number of items retrieved. Precision* adalah jumlah dokumen yang terpanggil dari *database* relevan setelah dinilai *user* dengan informasi yang dibutuhkan. Semakin besar nilai *precision* suatu STKI, maka STKI bisa dikatakan baik

(Amin, 2012).



**BAB III**

# KERANGKA KONSEP PENELITIAN

## 3.1. Kerangka Konsep

Penelitian ini mengacu pada penerapan algoritma *Term Frequency –*

*Inversed Document Frequency*

(

TF

-

IDF) dan

*Vector Space*

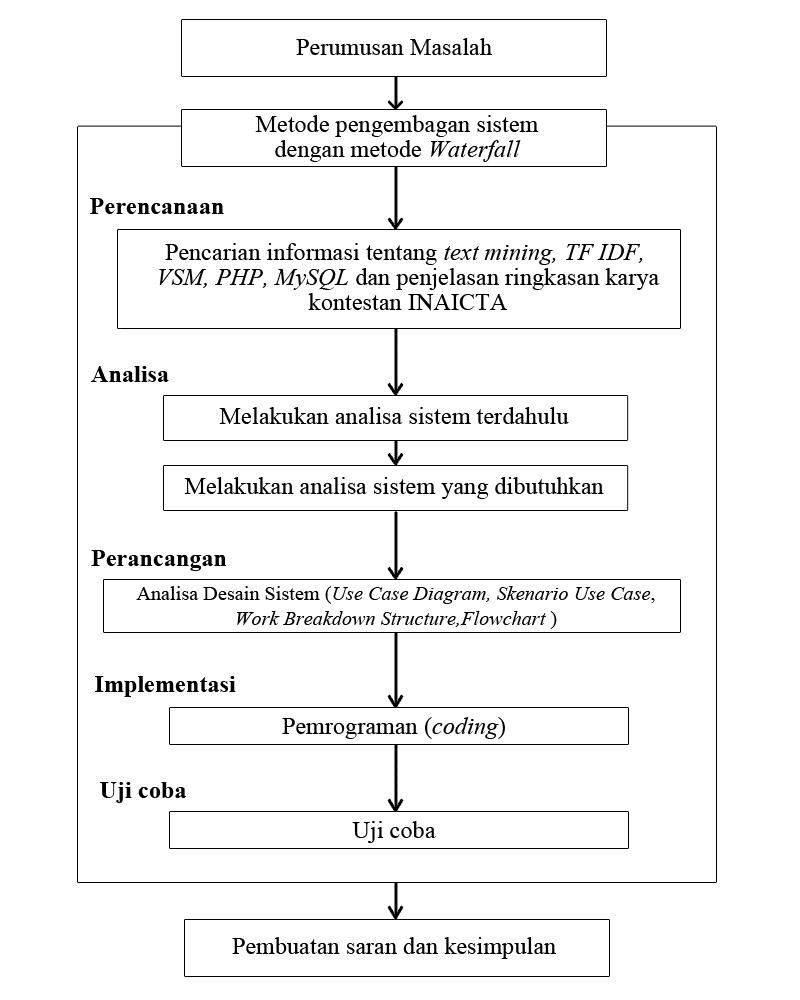
*Model*

, Kerangka

konseptual dibangun agar penelitian yang dilakukan dapat berjalan terarah sesuai

dengan rumusan masalah dan tujuan yang akan dicapai. Berikut ini adalah

kerangka konsep yang akan dilakukan:



Gambar 3.1 Kerangka Konsep

24

### 3.1.1 Perumusan Masalah

Pada tahap ini, penulis merumuskan segala permasalahan yang menyangkut sistem yang akan dibangun, dan diperoleh rumusan masalah

sebagai berikut:

c. Bagaimana membangun sistem yang dapat mendeteksi kemiripan karya berdasarkan penjelasan ringkas karya pada

Sistem

INAICTA 2013.

d.

Bagaimana menampilkan urutan tingkat kemiripan karya

berdasarkan penjelasan ringkas karya pada INAICTA 2013

menggunakan algoritma

*Term Frequency*

*–*

*Inversed Document*

*Frequency*

(

TF

-

IDF) dan

*Vector Space Model*

.

**3.1.2**

**Metode Pengembangan Sistem**

Metode

yang

digunakan

dalam

Pengembangan

Pen

deteksiKemiripan Karya pada INAICTA

ini adalah SDLC (

*Software*

*Development Life Cycle*

)

dengan model

*waterfall*

, dengan rincian sebagai

berikut:

a.

Perencanaan.

Pada proses perencanaan, penulis mengumpulkan



segala informasi terkait sistem yang akan dibangun, seperti *text mining, TF IDF, VSM, PHP, MySQL* dan penjelasan ringkasan karya kontestan INAICTA.

26

b. Analisa

Analisa yang dilakukan penulis yaitu analisa sistem terdahulu yang pernah dibangun dan analisis kebutuhan sistem. Sistem terdahulu kemudian diteliti oleh penulis dan diambil kesimpulannya, kemudian pada analisis kebutuhan sistem terdapat kebutuhan fungsional, kebutuhan perangkat keras dan

kebutuhan perangkat lu

nak.

c.

Perancangan

Pada tahap ini, kegiatan yang dil

akukan meliputi

analisa desain sistem dan perancangan database. Analisa desain

sistem mencakup pembuatan

*Use C*

*ase Diagram, Skenario Use*

*Case,*

*Work Breakdown Structure*

*, Flowchart*

*.*

Hal ini bertujuan

untuk

memenuhi kebutuhan para pemakai siste

m serta memberi

g

ambaran yang jelas dan rancang bangun yang lengkap kepada

penulis.

d.

Implementasi

Proses implementasi bertujuan

untuk menerapkan

perancangan yang telah

dilakukan terhadap sistem, pada tahap

ini terdapat proses pemrograman yaitu pembuatan coding



sistem.

e. Uji Coba

Setelah proses implementasi, maka sistem harus

melalui uji coba sistem.

## 3.2. Hipotesis

Berdasarkan latar belakang, tinjauan pustaka, kerangka konsep serta dukungan dari rumusan masalah dan tujuan dalam penelitian maka dapat diambil hipotesis sebagai berikut:

a. Sistem Pendeteksi Kemiripan Karya pada INAICTA 2013 dapat menghasilkan informasi urutan tingkat kemiripan karya berdasarkan

INAICTA.

penjelasan ringkas karya

.

b.

Sistem Pen

deteksi Kemiripan Karya pada INAICTA 2013

, diduga dapat

membe

rikan

bantuan

bagi

tim

penilai



**BAB IV**

# ANALISA DESIGN SISTEM

## 4.1 Metode Perancangan

Metode yang digunakan dalam Pengembangan Sistem Pendeteksi

Kemiripan Karya pada INAICTA 2013

ini adalah SDLC (

*Software Development*

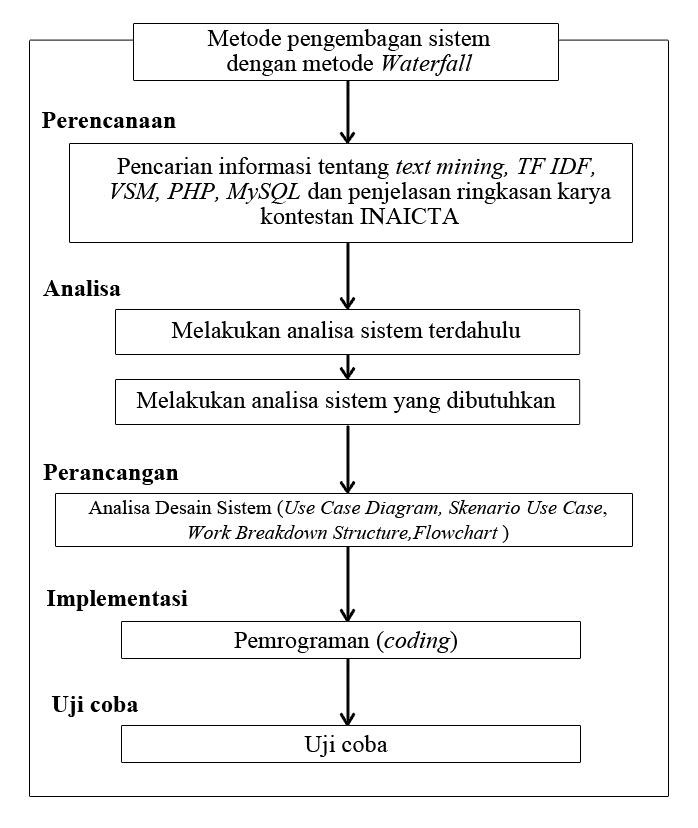
*Life Cycle*

)

dengan model

*waterfall*

,dengan gambaran sebagai berikut:



Gambar 4.1 Blok Diagram Perancangan

28

## 4.2 Analisis Kebutuhan Sistem

Pada analisis kebutuhan sistem terdiri atas analisis kebutuhan fungsional sistem, kebutuhan perangkat keras (*hardware*) dan kebutuhan perangkat lunak

(*software*), *Use Case Diagram, Skenario Use Case,Work Breakdown*

*Structure,Flowchart*.

### 4.2.1 Analisis Kebutuhan Fungsional

K

ebutuhan fungsional yang harus dipenuhi oleh sistem, dalam hal

ini adalah pengembangan sistem

pendeteksi kemiripan karya pada

INAICTAadalah m

enampilkan hasil

peringkat urutan kemiripan

.

**4.2.2**

**Kebutuhan Perangkat Keras**

Perangkat keras (

*hardware*

)

yang dibutuhkan adalah Personal

Computer (PC) atau notebook yang terhubung dengan jaringan internet

dengan spesifikasi minimal sebagai berikut :

a.

Processor Cor

e 2 Duo RAM 2 GB

b.

*Harddisk*

berkapasitas 40 GB

c.

Monitor

d.

Mouse

e.

Keyboard



### 4.2.3 Kebutuhan Perangkat Lunak

Untuk membangun sistem ini diperlukan perangkat lunak

(*software*) sebagai berikut :

1. Sistem operasi Microsoft Windows XP Professional
2. *Sublime Text 2* sebagai tools untuk menulis kode program
3. *MySQL* sebagai penyimpanan database

## 4.3 Deskripsi Sistem

SistemPendeteksi Kemiripan Karya pada INAICTA 2013 ini bertujuan menampilkan informasi peringkat urutan kemiripan karya berdasarkan penjelasan ringkas karya.

**4.4**

**Desain Sistem**

Arsitektur

Sistem

Pen

deteksi Kemiripan Karya pada INAICTA 2013

yang

akan dibangun merupakan sebuah

sistem

yang

berbasis

*localhost*

.

S

ecara

sederhana deskripsi umum sistem dapat dilihat melalui gambar :

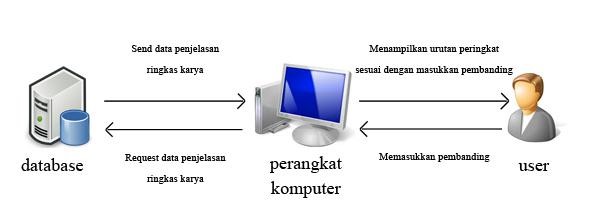
Gambar

4.2

Cara Kerja Sistem

**4.5**

**Perancangan Alur Sistem**



### 4.5.1 Work Breakdown Structure (WBS)

WBS adalah suatu metode pengorganisaian proyek menjadi struktur pelaporan hierarakis. WBS digunakan untuk melakukan

Breakdown atau memecahkan tiap proses pekerjaan menjadi lebih detail.hal ini dimaksudkan agar proses perencanaan proyek memiliki tingkat yang lebih baik.

Gambar 4.3

*Work Breakdown Structure*

**4.5.2**

***Use Case***

**Diagram**

*Use case*

adalah rangkaian / uraian sekelompok yang saling terkait

dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh

sebuah aktor. Berikut

*Use Case*

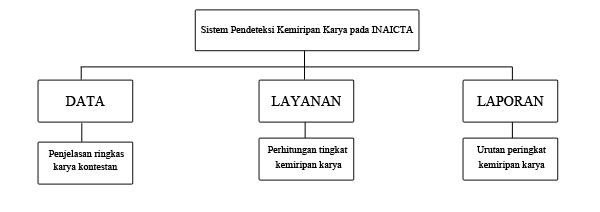
Diagram sistem yang dibangun :

Gambar

4.4

*Use Case*

Diagram



Memasukkan pembanding

Melihat urutan peringkat

kemiripan

USER



Berikut adalah definisi aktor pada Sistem Pencarian Lokasi Kuliner di Kota Malang dengan Location Based Service.

|  |  |  |
| --- | --- | --- |
| **No.** | **Aktor** | **Deskripsi** |
| 1 | User | Merupakan tim penilai atau juri dari INAICTA 2013. |

a. Definisi Aktor

b.

Definisi Use Case

No

Usecase

Deskripsi

1

Memasukkan pembanding

Pengguna

memberikan

*input*

atau masukkan satu

data penjelasan ringkas karya sebagai pembanding.

2

peringkat

Melihat

urutan

kemiripan

Pengguna

melihat

dapat

peringkat

urutan

kemiripan

berdasarkan pembanding yang sudah

dimasukkan

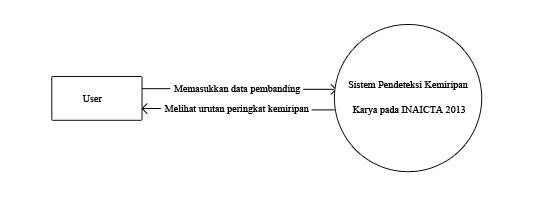
.



### 4.5.3 Diagram Konteks

Diagram konteks merupakan arus data yang berfungsi untuk menggambarkan keterkaitan aliran-aliran data antara sistem dengan bagian-bagian luar (kesatuan luar). Berikut diagram konteks sistem :

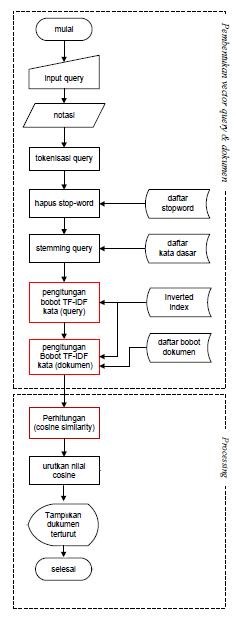
Gambar 4.5 Diagram Konteks



### 4.5.4 *Flowchart*

Berikut merupakan *flowchart*pada Sistem Pendeteksi Kemiripan

Karya pada INAICTA 2013:



Gambar 4.6 *Flowchart*Sistem Pendeteksi

Berikut merupakan urutan atau alur perhitungan bobot menggunakan TFIDF yang terdapat pada gambar 4.7*flowchart* TF-IDF.

Gambar 4.7

*Flowchart*

TF

-

IDF

Masukkan

*term*

***tf = tf***

***ij***

***idf***

***i***

***=***

***log***

𝑵

𝒅𝒇𝒊

***W***

***ij***

***tf***

***=***

***i***

***. idf***

***i***

***W***

(

Nilai bobot

)

Mulai

Selesai



Pada gambar 4.8 akan dijelaskan alur atau urutan dalam perhitungan nilai kesamaan menggunakan *Vector Space Model* yang menggunakan nilai bobot yang sudah dihitung dalam TF IDF sebelumnya.

Masukkan

bobot

query

**|**

***q***

**| =**

√

∑

(

𝑾

𝒊𝒒

)

𝟐

𝒕

𝒋

=

𝟏

Nilai Kemiripan

Mulai

Selesai

Masukkan

bobot

dokumen

**|**

***d***

***j***

**| =**

√

∑

(

𝑾

𝒊𝒋

)

𝟐

𝒕

𝒊

=

𝟏

𝒔𝒊𝒎

(

𝒒

,

𝒅

𝒋

)

=

∑

𝑾

𝒊𝒒

𝒕

𝒊

=

𝟏

.

𝑊

𝑖𝑗

𝒔𝒊𝒎

(

𝒒

,

𝒅

𝒋

)

=

𝒒

.

𝒅

𝒋

|

𝒒

|

∗

|

𝒅

𝒋

|

**=**

∑

𝑾

𝒊𝒒

.

𝑾

𝒊𝒋

𝒕

𝒊

=

𝟏

√

∑

(

𝑾

𝒋

)

𝟐

𝒕

𝒋

=

𝟏

∗

√

∑

(

𝑾

𝒊

)

𝟐

𝒕

𝒊

=

𝟏



Gambar 4.8 *FlowchartVector Space Model*

**BAB V**

**IMPLEMENTASI**

Setelah melakukan tahapan analisa desain sistem, maka pada pembahasan bab ini akan dibahas tentang proses implementasi perancangan yang telah disusun. Bagian pada bab ini meliputi implementasi basis data dan implementasi sistem. Implementasi basis data menggunakan MySQL dan implementasi sistem dibangun menggunakan bahasa pemrograman PHP sebagai bahasa pemrograman berbasis *web*.Tujuan implementasi adalah untuk menerapkan perancangan yang telah dilakukan pada bab sebelumnya.

## 5.1 Implementasi Basis Data

Implementasi basis data merupakan hasil implementasi dari perancangan basis data dan merupakan lanjutan dari perancangan sistem yang telah dilakukan pada bab sebelumnya. Wujud dari hasil implementasi ini nantinya adalah sebuah sistem yang siap untuk diuji dan digunakan. Basis data yang dibuat diberi nama jaringide. Gambar 5.1 merupakan hasil implementasi basis data yang dibangun menggunakan MySQL.

37

Gambar 5.1

*Database*

inaicta

Basis data

smpn1

memiliki

enam belas

tabel yaitu

ide dan katadasar

.

Implementasi tabel

-

tabel tersebut adalah sebagai berikut :

1.

Tabel

ide

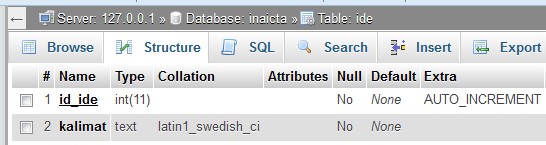
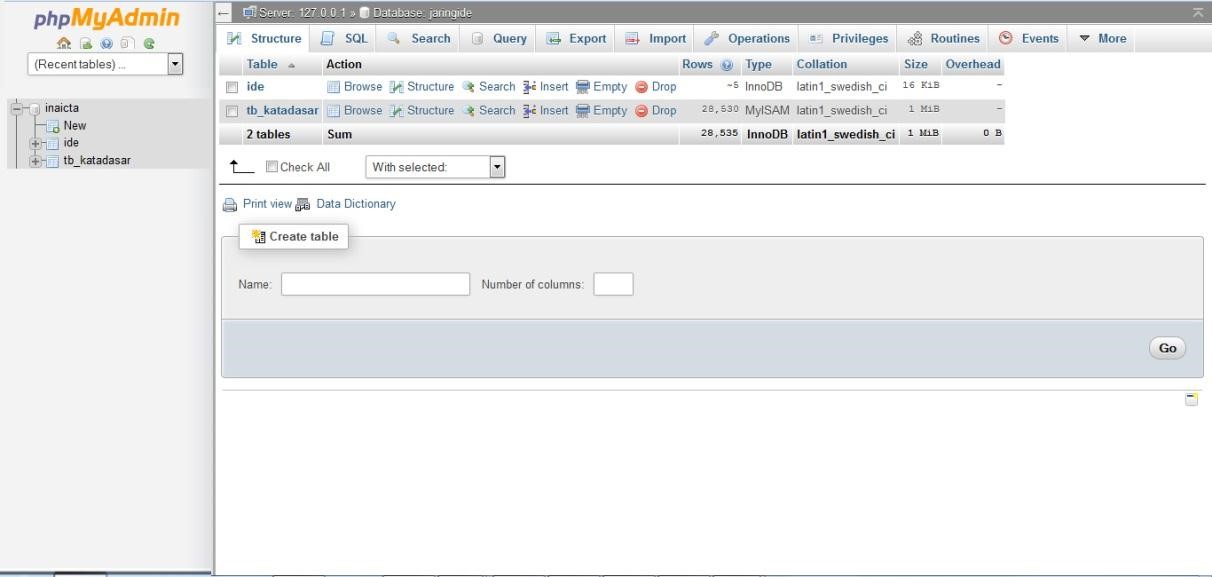
Gambar 5.2 Implementasi Tabel

ide

Gambar 5.2 adalah implementasi dari rancangan tabel

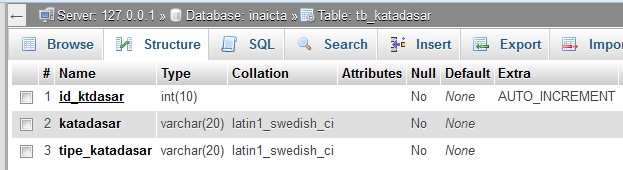
ide

yang



berfungsi menampung data penjelasan ringkas karya. Tabel ini memiliki *field* antara lain : id\_ide dan kalimat.

2. Tabel katadasar



Gambar 5.3 Implementasi Tabel katadasar

Gambar 5.

3

adalah implementasi dari rancangan tabel

katadasar

yang berfungsi menampung data

kata dasar

. Tabel ini memiliki

*field*

antara

lain :

id\_ktdasar

(

*primary key*

)

,

katadasar

,

tipe\_katadasar

.

Pada gambar 5.4

menjelaskan isi dari tabel katadasar.



Gambar 5.4 Tabel katadasar

## 5.2 Implementasi Sistem

Tahapan implementasi sistem merupakan implementasi darihasil tampilang antarmuka.Pengimplementasian aplikasi menggunakan PHP sebagai bahasa pemrograman berbasis *web*. Antar muka sistem pada aplikasi antara lain:

### 5.2.1 Implementasi Halaman Utama

Implementasi

rancangan

*interface*

pada

halaman

utama

digambarkan sebagai berikut:

Gambar 5.5 Halaman Utama

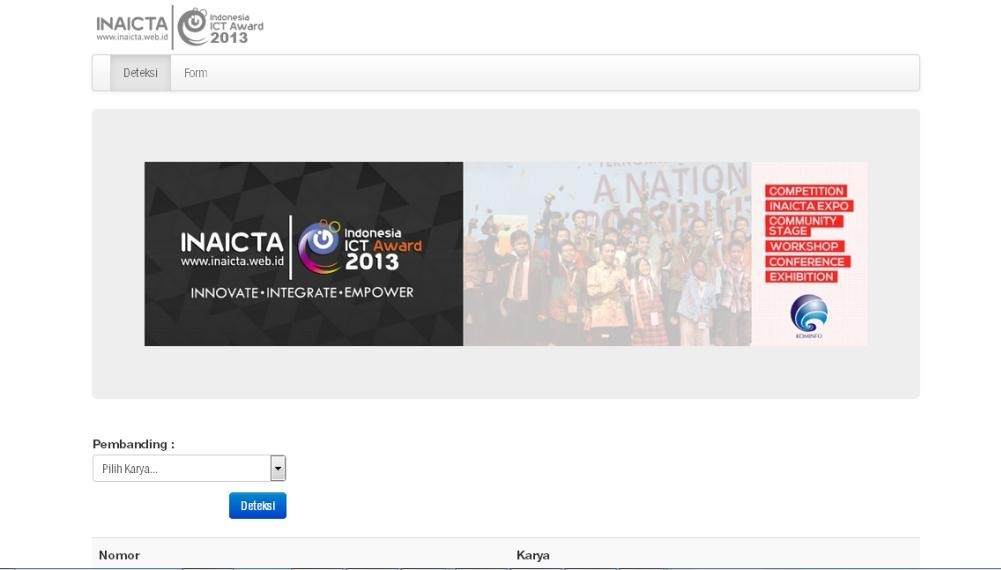
Saat aplikasi dibuka atau dijalankan, maka tampilan pada gambar

5.5

yang akan muncul sebagai halaman utama. Terdapat dua menu pada

halamann tersebut, yaitu deteksi dan form. Menu deteksi merupakan

*link*



untuk menuju ke halaman utama dan *linkform* digunakan untuk menuju ke halaman *form* dimana di dalam halaman tersebut pengguna dapat mengisikan penjelasan ringkas karya untuk menambahkan data yang akan dibandingkan.

Selain menu, pada halaman utama juga terdapat *dropdown* menu yang digunakan untuk memilih data sebagai pembanding pada proses pendeteksi kemiripan karya INAICTA. Di bawah *dropdown* menu terdapat tampilan data penjelasam ringkas karya yang sudah dimasukkan ke dalam *database*. Tampilan data tersebut berfungsi untuk memudahkan pengguna dalam pencarian data yang akan digunakan sebagai pembanding.

**5.2.2**

**Implementasi Halaman Form**

Pada

*interface*

halaman

*form*

digambarkan sebagai berikut:

Gambar 5.6 Halaman

*Form*

Pada halaman

*Form*

terdapat dua menu yang fungsinya sama

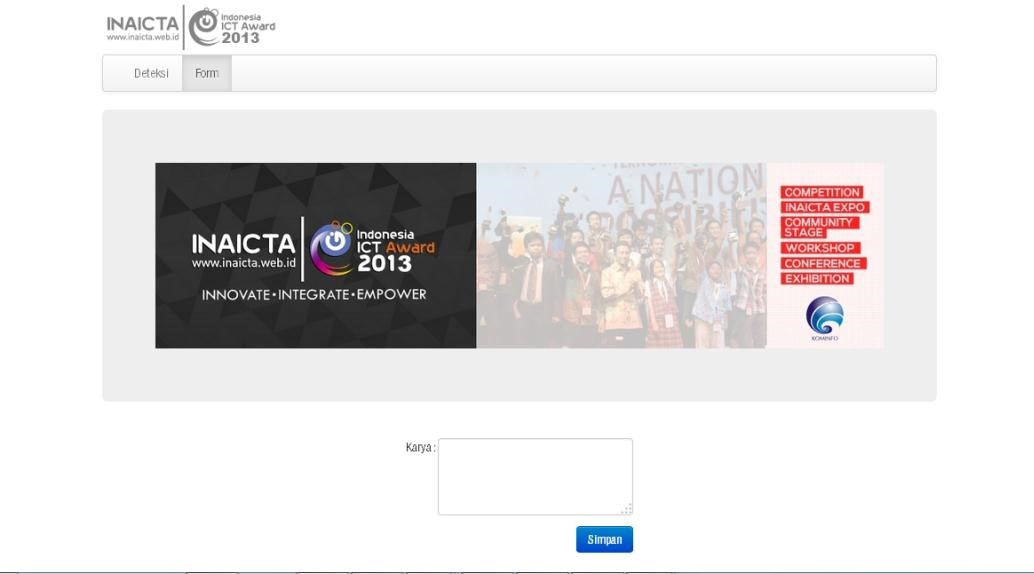
seperti p

ada halaman utama. Terdapat

*text area*

yang digunakan untuk

memasukkan isian berupa penjelasan ringkas karya INAICTA.



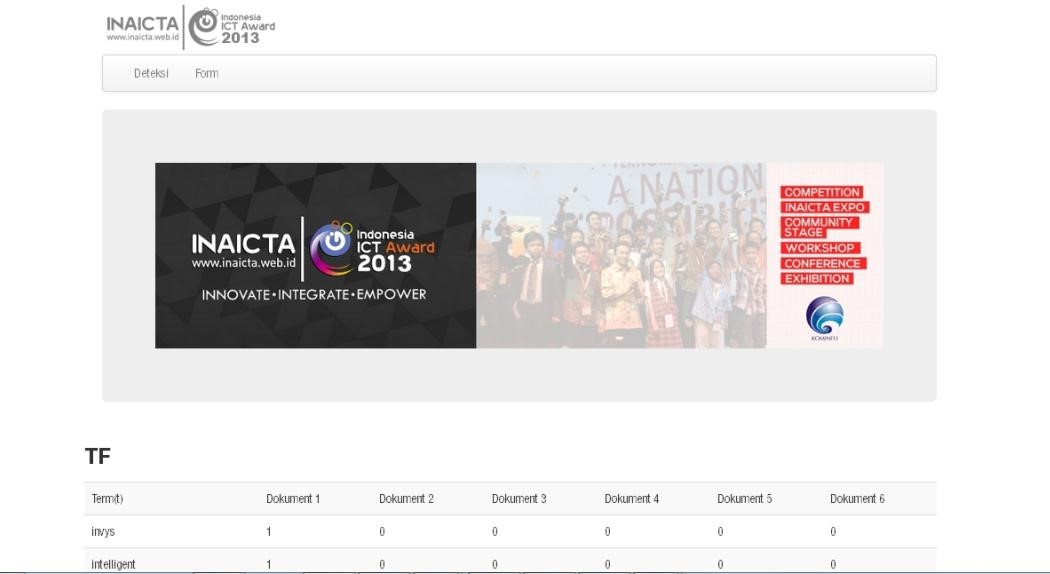
### 5.2.3 Implementasi Halaman Deteksi

Pada *interface* halaman deteksi digmabrakan sebagai berikut:

Gambar 5.7 Halaman Deteksi

Pada halaman deteksi terdapat perhitungan kemiripan berdasarkan

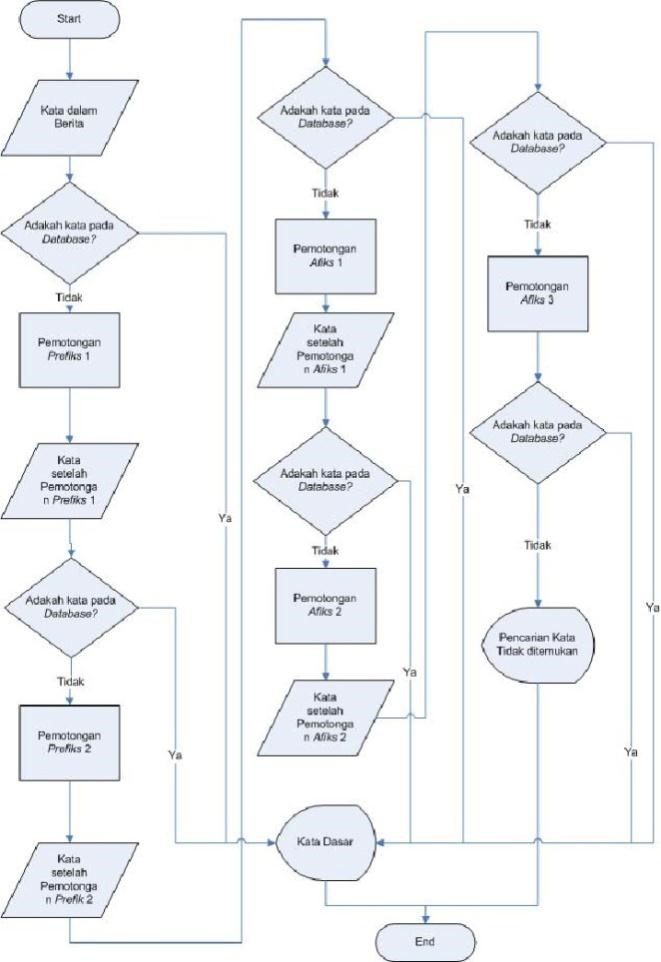
pembanding yang dipilih oleh pengguna.



**BAB VI**

**PENGUJIAN DAN PEMBAHASAN**

## 6.1 Pengujian

Setelah melakukan implementasi terhadap aplikasi, pada bab ini dilakukan pengujian. Pengujian digunakan untuk memastikan algoritma *Term Frequency – Inversed Document Frequency* (TF-IDF) dan *Vector Space Model* menghasilkan perhitungan yang sesuai. Perhitungan kemiripan menggunakan algoritma *Term Frequency – Inversed Document Frequency* (TF-IDF) dan *Vector Space Model* dalam sistem dan perhitungan kemiripan menggunakan *Term Frequency – Inversed Document Frequency* (TF-IDF) dan *Vector Space Model*secara manual akan menghasilkan nilai yang sama.

### 6.1.1 Pengujian Hasil

Pada tahap pengujian dilakukan dengan menggunakan 5 sampel data penjelasan ringkas karya dari INAICTA 2013. Berikut data yang akan diuji adalah

Tabel 6.1 Tabel Data Penjelasan Ringkas Karya

|  |  |
| --- | --- |
| **NOMOR KARYA** | **PENJELASAN RINGKAS KARYA** |
| **1.** | Invys (Intelligent Vision System) adalah sebuah sistem cerdas yang dapat digunakan sebagai assistive technology dalam hal penglihatan (vision), baik untuk digunakan pada (1) manusia, terutama mereka yang mengalami permasalahan dalam penglihatan, seperti tuna |

43

|  |  |
| --- | --- |
|  | netra, ataupun pada (2) robot, misalnya untuk membantu navigasi robot pengantar makanan di restoran.  Invys dapat mendeteksi adanya halangan yang ada di depan subyek (manusia/robot), memberikan peringatan dan rekomendasi untuk menghindari tabrakan dengan halangan tersebut, serta menunjukkan interest point yang telah ditandai dengan suatu marker, misalnya nama ruangan dalam sebuah gedung atau nomor meja pengunjung restoran. |
| **2.** | jalur  Sistem pencari dan penanda korban bencana alam di peta Google  dengan GPS (Globa  l Positioning System) dan Quadcopter ini  merupakan sistem yang berfungsi untuk membantu Tim SAR dalam  melakukan proses penyelamatan dan evakuasi korban pasca  terjadinya bencana alam di suatu daerah yang sulit dijangkau oleh  manusia dengan menunjukkan posis  i koordinat korban yang  sebenarnya.    Sistem ini memanfaatkan Quadcopter sebagai robot terbang (Aerial  Robot) yang dikendalikan jarak jauh oleh seorang Pilot dengan  remote control (RC) untuk mencari korban pada suatu area bencana  alam  dengan  dipantau  pergera  kkannya,  yaitu  berupa  perpindahan yang ditampilkan pada Peta Google (Google Maps) yang diamati oleh Operator di Ground Station. Pencarian korban pada lokasi bencana alam dilakukan dengan mengamati kamera wireless yang telah dipasang pada Quadcopter untuk melihat |

|  |  |
| --- | --- |
|  | kondisi sekitar lokasi bencana alam dan mencari korban di lokasi tersebut.  Apabila pada proses penelusuran lokasi bencana alam oleh Quadcopter, kamera wireless menangkap objek yang mirip dengan tubuh manusia maka operator yang mengamati program Ground Station dapat segera memberikan tanda (marker) di Google Maps dengan penanda yang berbeda dari marker pergerakan Quadcopter. Penanda atau marker yang menunjukkan titik koordinat korban kemudian diteruskan ke Tim SAR agar proses evakuasi dapat segera dilakukan. |
| **3.** | diamati dengan menggunakan sistem vision camera yang dimuat dalam AUV ini. Sistem omunikasi yang digunakan dalam AUV ini adalah cable-wireless.  Autonomous underwater vehicle (AUV) merupakan sebuah robot  yang dioperasikan dibawah air dengan menggunakan sistem  propulsi, dikontrol dan diarahkan oleh sebuah komputer. AUV  dirancang untuk misi surveillance di bawah air. Sistem navigasi    AUV  ini dilengkapi dengan kompas    dan GPS, sehingga posisi AUV dapat dilacak dan dapat diprogram  untuk mengikuti trajektori navigasi yang kita harapkan.    AUV ini dilengkapi dengan sensor pergerakan IMU sehingga  kestabilan dari gerakan AUV dapat dijaga. Kond  isi bawah air dapat  Sistem kabel digunakan untuk mentransmisikan data di dalam air, |
|  | sedangkan sistem wireless digunakan untuk mentransmisikan data di atas permukaan air. |
| **4.** | DRU13SAR adalah robot yang berfungsi sebagai alat bantu  memeriksa kondisi wilayah yang sulit atau berbahaya seperti kondisi pasca bencana. Robot ini didesain dengan dimensi 45x11x10 cm, sehingga memudahkan pergerakan robot.  Robot ini dilengkapi dengan kamera serta sensor gas untuk mendeteksi adanya kebocoran gas yang mudah terbakar atau gas yang membahayakan nyawa tim SAR. Robot ini juga dilengkapi dengan sensor suhu yang berguna untuk mengukur suhu  lingkungan, termasuk suhu korban “jika ditemukan- sehingga dapat ditentukan jenis penanganan yang harus diberikan nantinya. |
| **5.** | dan sebuah kompartemen pemberat. Sumber energi VOREBOT  VOREBOT (Navodneniye reshatel’ Robot/ Flood Solver Robot)  merupakan sebuah Underwater ROV(Remote Operated Vehicle)  karya tim    dari Robotic SMA Negeri 28 Jakarta yang memiliki    fungsi observasi untuk tujuan search and rescue.    VOREBOT memiliki rangka utama dari besi (dengan panjang 30 cm,  lebar 50 cm, tinggi 31 cm), dan dua buah kompartemen kedap air  yang terdiri dari sebuah  kompartemen berisi komponen elektronika,  adalah sebuah Accu 12 Volt, 18 Ah. |

Dari data pada tabel 6.1 maka dapat dihitung untuk mendapatkan nilai *TF IDF* dengan memproses mulai dari *tokenizing, filtering* dan *stemming*. Pada tabel 6.2 berikut ini ditampilkan hasil perhitungan *TF IDF* untuk 5 data penjelasan ringkas karya INAICTA 2013.



Tabel 6.2 Perhitungan TFIDF

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Term(t) | Karya 1 | Karya 2 | Karya 3 | Karya 4 | Karya 5 |
| invys | 1.6094379124341 | 0 | 0 | 0 | 0 |
| intelligent | 1.6094379124341 | 0 | 0 | 0 | 0 |
| vision | 1.8325814637483 | 0 | 0.91629073187416 | 0 | 0 |
| cerdas | 1.6094379124341 | 0 | 0 | 0 | 0 |
| assistive | 1.6094379124341 | 0 | 0 | 0 | 0 |
| technology | 1.6094379124341 | 0 | 0 | 0 | 0 |
| lihat | 3.2188758248682 | 0 | 0 | 0 | 0 |
| 1 | 1.6094379124341 | 0 | 0 | 0 | 0 |
| masalah | 1.6094379124341 | 0 | 0 | 0 | 0 |
| tuna | 1.6094379124341 | 0 | 0 | 0 | 0 |
| netra | 1.6094379124341 | 0 | 0 | 0 | 0 |
| 2 | 1.6094379124341 | 0 | 0 | 0 | 0 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Term(t) | Karya 1 | Karya 2 | Karya 3 | Karya 4 |  | Karya 5 |
| robot | 0 | 0 | 0 | 0 | 0 |  |
| navigasi | 0.91629073187416 | 0 | 1.8325814637483 | 0 | 0 |  |
| antar | 1.6094379124341 | 0 | 0 | 0 | 0 |  |
| restoraninvys | 1.6094379124341 | 0 | 0 | 0 | 0 |  |
| deteksi | 0.91629073187416 | 0 | 0 | 0.91629073187416 | 0 |  |
| halang | 3.2188758248682 | 0 | 0 | 0 | 0 |  |
| subyek | 1.6094379124341 | 0 | 0 | 0 | 0 |  |
| manusiarobot | 1.6094379124341 | 0 | 0 | 0 | 0 |  |
| ingat | 1.6094379124341 | 0 | 0 | 0 | 0 |  |
| rekomendasi | 1.6094379124341 | 0 | 0 | 0 | 0 |  |
| hindar | 1.6094379124341 | 0 | 0 | 0 | 0 |  |
| tabrak | 1.6094379124341 | 0 | 0 | 0 | 0 |  |
| point | 1.6094379124341 | 0 | 0 | 0 | 0 |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Term(t) | Karya 1 | Karya 2 | Karya 3 | Karya 4 |  | Karya 5 |
| tanda | 0.91629073187416 | 3.6651629274966 | 0 | 0 | 0 |  |
| marker | 0.91629073187416 | 2.7488721956225 | 0 | 0 | 0 |  |
| ruang | 1.6094379124341 | 0 | 0 | 0 | 0 |  |
| meja | 1.6094379124341 | 0 | 0 | 0 | 0 |  |
| unjung | 1.6094379124341 | 0 | 0 | 0 | 0 |  |
| restoran | 1.6094379124341 | 0 | 0 | 0 | 0 |  |
| pencar | 0 | 1.6094379124341 | 0 | 0 | 0 |  |
| bencana | 0 | 5.4977443912449 | 0 | 0.91629073187416 | 0 |  |
| peta | 0 | 3.2188758248682 | 0 | 0 | 0 |  |
| google | 0 | 6.4377516497364 | 0 | 0 | 0 |  |
| gps | 0 | 0.91629073187416 | 0.91629073187416 | 0 | 0 |  |
| positioning | 0 | 1.6094379124341 | 0 | 0 | 0 |  |
| quadcopter | 0 | 8.0471895621705 | 0 | 0 | 0 |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Term(t) |  | Karya 1 | Karya 2 | Karya 3 | Karya 4 |  | Karya 5 |
| fungsi | 0 |  | 0.91629073187416 | 0 | 0.91629073187416 | 0 |  |
| sar | 0 |  | 1.8325814637483 | 0 | 0.91629073187416 | 0 |  |
| selamat | 0 |  | 1.6094379124341 | 0 | 0 | 0 |  |
| evakuasi | 0 |  | 3.2188758248682 | 0 | 0 | 0 |  |
| pasca | 0 |  | 0.91629073187416 | 0 | 0.91629073187416 | 0 |  |
| sulit | 0 |  | 0.91629073187416 | 0 | 0.91629073187416 | 0 |  |
| jangkau | 0 |  | 1.6094379124341 | 0 | 0 | 0 |  |
| koordinat | 0 |  | 3.2188758248682 | 0 | 0 | 0 |  |
| sebenarnyasistem | 0 |  | 1.6094379124341 | 0 | 0 | 0 |  |
| manfaat | 0 |  | 1.6094379124341 | 0 | 0 | 0 |  |
| aerial | 0 |  | 1.6094379124341 | 0 | 0 | 0 |  |
| kendali | 0 |  | 1.6094379124341 | 0 | 0 | 0 |  |
| jarak | 0 |  | 1.6094379124341 | 0 | 0 | 0 |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Term(t) |  | Karya 1 | Karya 2 | Karya 3 | Karya 4 |  | Karya 5 |
| pilot | 0 |  | 1.6094379124341 | 0 | 0 | 0 |  |
| remote | 0 |  | 1.6094379124341 | 0 | 0 | 0 |  |
| control | 0 |  | 1.6094379124341 | 0 | 0 | 0 |  |
| rc | 0 |  | 1.6094379124341 | 0 | 0 | 0 |  |
| area | 0 |  | 1.6094379124341 | 0 | 0 | 0 |  |
| pantau | 0 |  | 1.6094379124341 | 0 | 0 | 0 |  |
| gerak | 0 |  | 1.6094379124341 | 0 | 0 | 0 |  |
| pindah | 0 |  | 1.6094379124341 | 0 | 0 | 0 |  |
| tampil | 0 |  | 1.6094379124341 | 0 | 0 | 0 |  |
| maps | 0 |  | 3.2188758248682 | 0 | 0 | 0 |  |
| amat | 0 |  | 2.7488721956225 | 0.91629073187416 | 0 | 0 |  |
| operator | 0 |  | 3.2188758248682 | 0 | 0 | 0 |  |
| ground | 0 |  | 3.2188758248682 | 0 | 0 | 0 |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Term(t) |  | Karya 1 | Karya 2 | Karya 3 | Karya 4 |  | Karya 5 |
| station | 0 |  | 3.2188758248682 | 0 | 0 | 0 |  |
| cari | 0 |  | 1.6094379124341 | 0 | 0 | 0 |  |
| kamera | 0 |  | 1.8325814637483 | 0 | 0.91629073187416 | 0 |  |
| wireless | 0 |  | 3.2188758248682 | 0 | 0 | 0 |  |
| pasang | 0 |  | 1.6094379124341 | 0 | 0 | 0 |  |
| tersebutapabila | 0 |  | 1.6094379124341 | 0 | 0 | 0 |  |
| telusur | 0 |  | 1.6094379124341 | 0 | 0 | 0 |  |
| tangkap | 0 |  | 1.6094379124341 | 0 | 0 | 0 |  |
| objek | 0 |  | 1.6094379124341 | 0 | 0 | 0 |  |
| tubuh | 0 |  | 1.6094379124341 | 0 | 0 | 0 |  |
| gera | 0 |  | 0.51082562376599 | 1.021651247532 | 0.51082562376599 | 0 |  |
| titik | 0 |  | 1.6094379124341 | 0 | 0 | 0 |  |
| terus | 0 |  | 1.6094379124341 | 0 | 0 | 0 |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Term(t) |  | Karya 1 | Karya 2 | Karya 3 | Karya 4 | Karya 5 |
| autonomous | 0 |  | 0 | 1.6094379124341 | 0 | 0 |
| underwater | 0 |  | 0 | 0.91629073187416 | 0 | 0.91629073187416 |
| vehicle | 0 |  | 0 | 0.91629073187416 | 0 | 0.91629073187416 |
| auv | 0 |  | 0 | 11.266065387039 | 0 | 0 |
| operasi | 0 |  | 0 | 1.6094379124341 | 0 | 0 |
| bawah | 0 |  | 0 | 1.6094379124341 | 0 | 0 |
| propulsi | 0 |  | 0 | 1.6094379124341 | 0 | 0 |
| kontrol | 0 |  | 0 | 1.6094379124341 | 0 | 0 |
| arah | 0 |  | 0 | 1.6094379124341 | 0 | 0 |
| komputer | 0 |  | 0 | 1.6094379124341 | 0 | 0 |
| rancang | 0 |  | 0 | 1.6094379124341 | 0 | 0 |
| misi | 0 |  | 0 | 1.6094379124341 | 0 | 0 |
| surveillance | 0 |  | 0 | 1.6094379124341 | 0 | 0 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Term(t) |  | Karya 1 | Karya 2 | Karya 3 | Karya 4 |  | Karya 5 |
| lengkap | 0 |  | 0 | 1.8325814637483 | 1.8325814637483 | 0 |  |
| kompasdan | 0 |  | 0 | 1.6094379124341 | 0 | 0 |  |
| lacak | 0 |  | 0 | 1.6094379124341 | 0 | 0 |  |
| program | 0 |  | 0 | 1.6094379124341 | 0 | 0 |  |
| ikut | 0 |  | 0 | 1.6094379124341 | 0 | 0 |  |
| trajektori | 0 |  | 0 | 1.6094379124341 | 0 | 0 |  |
| harapkanauv | 0 |  | 0 | 1.6094379124341 | 0 | 0 |  |
| sensor | 0 |  | 0 | 0.91629073187416 | 1.8325814637483 | 0 |  |
| i | 0 |  | 0 | 1.6094379124341 | 0 | 0 |  |
| stabil | 0 |  | 0 | 1.6094379124341 | 0 | 0 |  |
| jaga | 0 |  | 0 | 1.6094379124341 | 0 | 0 |  |
| camera | 0 |  | 0 | 1.6094379124341 | 0 | 0 |  |
| muat | 0 |  | 0 | 1.6094379124341 | 0 | 0 |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Term(t) |  | Karya 1 | Karya 2 | Karya 3 | Karya 4 |  | Karya 5 |
| omunikasi | 0 |  | 0 | 1.6094379124341 | 0 | 0 |  |
| cablewirelesssistem | 0 |  | 0 | 1.6094379124341 | 0 | 0 |  |
| kabel | 0 |  | 0 | 1.6094379124341 | 0 | 0 |  |
| transmisi | 0 |  | 0 | 3.2188758248682 | 0 | 0 |  |
| wirelessdigunakan | 0 |  | 0 | 1.6094379124341 | 0 | 0 |  |
| muka | 0 |  | 0 | 1.6094379124341 | 0 | 0 |  |
| dru13sar | 0 |  | 0 | 0 | 1.6094379124341 | 0 |  |
| alat | 0 |  | 0 | 0 | 1.6094379124341 | 0 |  |
| bantu | 0 |  | 0 | 0 | 1.6094379124341 | 0 |  |
| periksa | 0 |  | 0 | 0 | 1.6094379124341 | 0 |  |
| bahaya | 0 |  | 0 | 0 | 3.2188758248682 | 0 |  |
| desain | 0 |  | 0 | 0 | 1.6094379124341 | 0 |  |
| dimensi | 0 |  | 0 | 0 | 1.6094379124341 | 0 |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Term(t) |  | Karya 1 | Karya 2 | Karya 3 | Karya 4 | Karya 5 |
| 45x11x10 | 0 |  | 0 | 0 | 1.6094379124341 | 0 |
| cm | 0 |  | 0 | 0 | 0.91629073187416 | 2.7488721956225 |
| mudah | 0 |  | 0 | 0 | 1.6094379124341 | 0 |
| robotrobot | 0 |  | 0 | 0 | 1.6094379124341 | 0 |
| gas | 0 |  | 0 | 0 | 4.8283137373023 | 0 |
| bocor | 0 |  | 0 | 0 | 1.6094379124341 | 0 |
| bakar | 0 |  | 0 | 0 | 1.6094379124341 | 0 |
| nyawa | 0 |  | 0 | 0 | 1.6094379124341 | 0 |
| suhu | 0 |  | 0 | 0 | 4.8283137373023 | 0 |
| guna | 0 |  | 0 | 0 | 1.6094379124341 | 0 |
| ukur | 0 |  | 0 | 0 | 1.6094379124341 | 0 |
| tentu | 0 |  | 0 | 0 | 1.6094379124341 | 0 |
| tangan | 0 |  | 0 | 0 | 1.6094379124341 | 0 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Term(t) |  | Karya 1 | Karya 2 | Karya 3 | Karya 4 | Karya 5 |
| vorebot | 0 |  | 0 | 0 | 0 | 3.2188758248682 |
| navodneniye | 0 |  | 0 | 0 | 0 | 1.6094379124341 |
| reshatel | 0 |  | 0 | 0 | 0 | 1.6094379124341 |
| flood | 0 |  | 0 | 0 | 0 | 1.6094379124341 |
| solver | 0 |  | 0 | 0 | 0 | 1.6094379124341 |
| rovremote | 0 |  | 0 | 0 | 0 | 1.6094379124341 |
| operated | 0 |  | 0 | 0 | 0 | 1.6094379124341 |
| robotic | 0 |  | 0 | 0 | 0 | 1.6094379124341 |
| sma | 0 |  | 0 | 0 | 0 | 1.6094379124341 |
| 28 | 0 |  | 0 | 0 | 0 | 1.6094379124341 |
| memilikifungsi | 0 |  | 0 | 0 | 0 | 1.6094379124341 |
| observasi | 0 |  | 0 | 0 | 0 | 1.6094379124341 |
| tuju | 0 |  | 0 | 0 | 0 | 1.6094379124341 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Term(t) |  | Karya 1 | Karya 2 | Karya 3 | Karya 4 | Karya 5 |
| search | 0 |  | 0 | 0 | 0 | 1.6094379124341 |
| rescuevorebot | 0 |  | 0 | 0 | 0 | 1.6094379124341 |
| rangka | 0 |  | 0 | 0 | 0 | 1.6094379124341 |
| besi | 0 |  | 0 | 0 | 0 | 1.6094379124341 |
| 30 | 0 |  | 0 | 0 | 0 | 1.6094379124341 |
| lebar | 0 |  | 0 | 0 | 0 | 1.6094379124341 |
| 50 | 0 |  | 0 | 0 | 0 | 1.6094379124341 |
| 31 | 0 |  | 0 | 0 | 0 | 1.6094379124341 |
| buah | 0 |  | 0 | 0 | 0 | 1.6094379124341 |
| kompartemen | 0 |  | 0 | 0 | 0 | 4.8283137373023 |
| kedap | 0 |  | 0 | 0 | 0 | 1.6094379124341 |
| isi | 0 |  | 0 | 0 | 0 | 1.6094379124341 |
| komponen | 0 |  | 0 | 0 | 0 | 1.6094379124341 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Term(t) | Karya 1 | Karya 2 | Karya 3 | Karya 4 | Karya 5 |
| elektronika | 0 | 0 | 0 | 0 | 1.6094379124341 |
| berat | 0 | 0 | 0 | 0 | 1.6094379124341 |
| energi | 0 | 0 | 0 | 0 | 1.6094379124341 |
| accu | 0 | 0 | 0 | 0 | 1.6094379124341 |
| 12 | 0 | 0 | 0 | 0 | 1.6094379124341 |
| volt | 0 | 0 | 0 | 0 | 1.6094379124341 |
| 18 | 0 | 0 | 0 | 0 | 1.6094379124341 |
| ah | 0 | 0 | 0 | 0 | 1.6094379124341 |
| invys | 1.6094379124341 | 0 | 0 | 0 | 0 |
| intelligent | 1.6094379124341 | 0 | 0 | 0 | 0 |
| vision | 1.8325814637483 | 0 | 0.91629073187416 | 0 | 0 |
| cerdas | 1.6094379124341 | 0 | 0 | 0 | 0 |
| assistive | 1.6094379124341 | 0 | 0 | 0 | 0 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Term(t) | Karya 1 | Karya 2 | Karya 3 | Karya 4 |  | Karya 5 |
| technology | 1.6094379124341 | 0 | 0 | 0 | 0 |  |

Tabel 6.3 Perhitungan

*Vector Space Model*

NILAI KEMIRIPAN

KARYA NOMOR 2

KARYA NOMOR 3

KARYA NOMOR 4

KARYA

NOMOR 5

0.067540916598598

0.019874702241438

0.0061038951217969

0



Dari perhitunga pada tabel 6.3 maka dapat disimpulkan bahwa karya yang memiliki kemiripan terhadap karya nomor 1 adalah karya nomor 2 dengan nilai *VSM* sebesar0.067540916598598.

Berikut tabel nilai hasil pendeteksi kemiripan pada setiap karya yang diambil dari nilai kemiripan tertinggi terhadap seluruh dokumen pada INAICTA.

Tabel 6.4 Nilai Kemiripan Karya

|  |  |  |
| --- | --- | --- |
| Pembanding | Urutan Karya Memiliki Kemiripan | Nilai Kemiripan |
| Karya 1 | Karya 2 | 0.067540916598598 |
| Karya 3 | 0.019874702241438 |
| Karya 4 | 0.0061038951217969 |
| Karya 5 | 0 |
| Karya 2 | Karya 4 | 0.041839154328761 |
| Karya 3 | 0.022963214970868 |
| Karya 1 | 0.01791061130501 |
| Karya 5 | 0 |
| Karya 3 | Karya 2 | 0.028209742040656 |
| Karya 4 | 0.020813327755827 |
| Karya 1 | 0.015445615057625 |
| Karya 5 | 0.010754343407157 |
| Karya 4 | Karya 2 | 0.11314095946043 |
| Karya 3 | 0.028983458838872 |

62

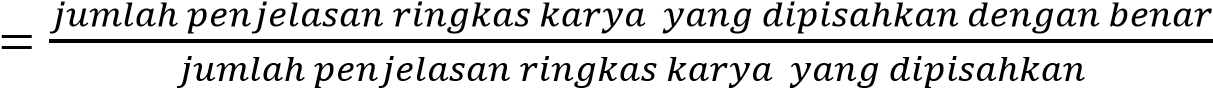
63

|  |  |  |
| --- | --- | --- |
|  | Karya 5 | 0.016131515110735 |
| Karya 1 | 0.0074879440985397 |
| Karya 5 | Karya 4 | 0.01522962991263 |
| Karya 3 | 0.0082647870395617 |
| Karya 1 | 0 |
| Karya 2 | 0 |

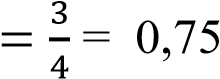
Hasil *similarity* yang didapat dari pengujian pada tabel 6.4 dinyatakan 10% karya yang memiliki kesamaan (*similarity*) diatas 0,05 dimana pada dokumen tersebut menggunakan kata-kata yang sama pada penulisan penjelasan ringkasan karya. Dan pengujian menghasilkan nilai diatas 0,1 sebanyak 5% dari karya yang telah diuji.

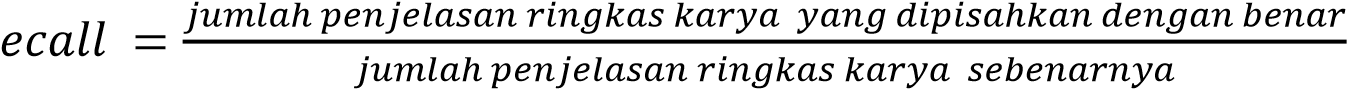
### 6.1.2 Uji *Recall* dan *Precision*

Pengujian kerja sistem menggunakan sistem uji *recall* dan *precision* yang bertujuan untuk mendapatkan informasi hasil pencarian yang didapatkan oleh STKI*.* Hasil pencarian STKI bisa dinilai tingkat *recall* dan *precision* nya. *Precision* adalah tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem. Sedangkan *recall* adalah tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi. Pada statistika terdapat juga penghitungan *accuracy*. *Accuracy* didefinisikan sebagai tingkat kedekatan antara nilai prediksi dengan nilai actual.

*precision*

64

*precision* x100% = 75%

*r*

*r*

𝑒𝑐𝑎𝑙𝑙

=

3

4

=

7

0,

5

x100% = 7

5

%

*accuracy*

=

𝑗𝑢𝑚𝑙𝑎

ℎ

𝑝𝑒

𝑛𝑗𝑒𝑙𝑎𝑠𝑎𝑛

𝑟𝑖𝑛𝑔𝑘𝑎𝑠

𝑘𝑎𝑟𝑦𝑎

𝑦𝑎𝑛𝑔

𝑑𝑖𝑝𝑖𝑠𝑎

ℎ

𝑘𝑎𝑛

𝑑𝑒𝑛𝑔𝑎𝑛

𝑏𝑒𝑛𝑎𝑟

𝑗𝑢𝑚𝑙𝑎

ℎ

𝑝𝑒𝑛𝑗𝑒𝑙𝑎𝑠𝑎𝑛

𝑟𝑖𝑛𝑔𝑘𝑎𝑠

𝑘𝑎𝑟𝑦𝑎

𝑡𝑜𝑡𝑎𝑙

*accuracy*

=

3

5

=

0,

8

x100% =

8

%

0



**BAB VII**

**PENUTUP**

## 7.1 Keimpulan

Dari pembahasan dan analisa di atas, maka penulis menyimpulkan bahwa:

1. Dalam pengembangan sistem pendeteksi kemiripan karya pada INAICTA menggunakan algoritma *Term Frequency – Inversed*

*Document Frequency* (TF-IDF) sebagai penentu bobot tiap karya dan *Vector Space Model*sebagai penentu peringkat urutan kemiripan karya.

1. Dari perhitungan meggunakan algoritma *Term Frequency – Inversed Document Frequency* (TF-IDF) dan *Vector Space Model*, maka pengguna atau tim penilai dapat mengetahui tingkat kemiripan karya pada INAICTA 2013 berdasarkan jumlah kemunculan kata yang

mirip.

## 7.2 Saran

Pada penelitian pengembangan sistem pendeteksi kemiripan karya ini terdapat beberapa saran sebagai berikut:

1. Penggunaan algoritma *Term Frequency – Inversed Document Frequency* (TF-IDF) dan *Vector Space Model* perlu dikembangkan lagi agar digunakan untuk mendeteksi kemiripan dokumen lain.

65

66

1. Proses *stemming* perlu dikembangkan untuk kata-kata yang masih belum termasuk kata dasar seperti, mempertanggungjawabkan, menganakemaskan dan sejenisnya.

