

Sistem Temu Kembali Informasi Pasal-Pasal KUHP (Kitab Undang-Undang Hukum Pidana) Berbasis Android Menggunakan Metode *Synonym Recognition* dan *Cosine Similarity*

Safier Yusuf¹, M. Ali Fauzi², Komang Candra Brata³

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ¹safier.yusuf@gmail.com, ²moch.ali.fauzi@ub.ac.id, ³k.candra.brata@ub.ac.id

Abstrak

Kitab Undang-undang Hukum Pidana atau KUHP merupakan hal yang wajib dipelajari bahkan dihapalkan untuk kalangan orang yang bergerak dibidangnya, seperti kepolisian, aparat penegak hukum, hakim, pengacara atau orang-orang yang terkait dengan persidangan. KUHP merupakan kitab yang di dalamnya terdiri dari puluhan bab dan ratusan pasal dengan total mencapai 569 pasal, sehingga dengan kitab yang tebal tersebut akan menjadi sangat tidak efisien dan praktis apabila harus dibawa dan juga apabila ingin mencari pasal terkait yang harus membuka halaman satu-persatu secara manual. Berdasarkan kondisi tersebut dalam penelitian ini dikembangkan aplikasi menggunakan metode *synonym recognition* dan *cosine similarity*. *Synonym recognition* adalah teknik yang digunakan untuk mengenali kata dengan penulisan berbeda namun memiliki makna yang sama. Metode *cosine similarity* digunakan untuk menghitung kesamaan atau kedekatan dokumen pasal dengan *query*. Kinerja dari system ditunjukkan dengan hasil pengujian pada masing-masing variasi *threshold* yaitu 5, 10 dan 15, dengan kinerja optimal terdapat pada *threshold* 15 yang memiliki nilai *f.measure* sebesar 0,404.

Kata Kunci : KUHP, *Synonim Recognition* , *Cosine Similarity*, *Query*

Abstract

Kitab Undang-undang Hukum Pidana (KUHP) is a matter that must be studied and even memorized for people engaged in their field, such as police, law enforcement officials, judges, lawyers or persons associated with the trial. KUHP is a book in which it consists of dozens of chapters and hundreds of chapters with a total of 569 articles, so that with a thick book it will become very inefficient and practical if it must be brought and also if you want to find related chapters that must open the page one by one manual. Based on these conditions in this study developed applications using synonym recognition and cosine similarity methods. Synonym recognition is a technique used to recognize words with different writing but has the same meaning. The cosine similarity method is used to calculate the similarity or closeness of the chapter documents with the query. The performance of the system is indicated by the results of the tests on each threshold variation of 5, 10 and 15, with optimal performance is at threshold 15 which has f.measure value of 0.404.

Keywords : KUHP, *Synonim Recognition* , *Cosine Similarity*, *Query*

1. PENDAHULUAN

Berkembangnya teknologi perangkat mobile membuat sistem operasi Android menjadi salah satu sistem operasi yang mudah digunakan. Pemanfaatan teknologi tersebut dapat membantu penggunaanya dalam mendapatkan informasi. Salah satunya informasi mengenai Kitab Undang-Undang Hukum Pidana (KUHP). KUHP adalah semua dasar-dasar dan aturan-aturan yang dianut oleh suatu negara dalam menyelenggarakan

ketertiban hukum yaitu dengan melarang apa yang bertentangan dengan hukum dan mengenakan suatu sanksi kepada yang melanggar peraturan tersebut. KUHP dibentuk sebagai suatu aturan yang digunakan oleh negara untuk menyelenggarakan ketertiban umum. Induk peraturan hukum pidana di Indonesia mengacu kepada kitab tersebut.

Kitab Undang-undang Hukum Pidana atau KUHP merupakan hal yang wajib dipelajari bahkan dihapalkan untuk kalangan orang yang bergerak dibidangnya, seperti kepolisian, aparat

penegak hukum, hakim, pengacara atau orang-orang yang terkait dengan persidangan. KUHP merupakan kitab yang di dalamnya terdiri dari puluhan bab dan ratusan pasal dengan total mencapai 569 pasal, sehingga dengan kitab yang tebal tersebut akan menjadi sangat tidak efisien dan praktis apabila harus dibawa dan juga apabila ingin mencari pasal terkait yang harus membuka halaman satu-persatu secara manual.

Perangkat bergerak dengan *Operating System* (OS) Android mampu menyimpan data dalam jumlah besar dalam penyimpanan basis datanya yang memungkinkan pengguna menggunakan aplikasi untuk mengambil data dengan cepat. Untuk menangani teks berukuran besar yang banyak, diperlukan suatu aplikasi yang dapat mengotomatisasi proses pendeteksian kemiripan teks. Sedangkan untuk mengatasi struktur teks yang tidak konsisten dan kompleks, diperlukan konsep *text mining*.

Berfokuskan pada teks berbahasa Indonesia dan berkonsepkan *text mining*, penelitian ini bertujuan untuk mengembangkan sebuah aplikasi yang mengimplementasikan cosine similarity yang berguna untuk mengukur kesamaan teks berdasarkan kemunculan kata-kata dalam teks tersebut. Dikombinasikan dengan *synonym recognition* yang merupakan pendekatan semantik terhadap dokumen teks. Pendekatan ini memanfaatkan kesamaan makna dalam kata yang kemungkinan banyak terjadi. Dengan mendeteksi kata-kata yang sama (sinonim) antara teks dokumen satu dengan yang lain, pendekatan ini menambah tingkat keakuratan proses deteksi keterkaitan pasal.

Telah terdapat penelitian sebelumnya tentang *Information Retrieval* yang memuat tentang pencarian pasal yang ditulis oleh Baskoro (2015) dan Ali (2015). Diketahui dari penelitian tersebut terdapat sebuah kelemahan yaitu hanya dapat mendeteksi pencarian sesuai kata yang diinputkan oleh user, sehingga jika ada kata lain yang mirip atau yang bersinonim dengan kata yang diinputkan tersebut yang berpotensi berhubungan atau berkaitan dengan pasal lainnya menjadi tidak terdeteksi oleh sistem tersebut. Pada penelitian lainnya tentang plagiarisme yang ditulis oleh Yoga (2012), pada

penelitian tersebut menganut metode *synonym recognition* sehingga sistem tersebut dapat mendeteksi kata-kata yang bersinonim yang apabila plagiat mengubah struktur kata dari suatu kalimat, oleh sistem tersebut akan tetap terbaca karena kalimat tersebut memiliki makna yang sama.

Berdasarkan uraian di atas, peneliti mengusulkan pengembangan sebuah sistem pencarian pasal yang dapat melengkapi kekurangan aplikasi sebelumnya yaitu dapat mendeteksi kemiripan pada kata yang bersinonim, sehingga pasal lainnya yang berkaitan dengan kata-kata tersebut dapat ditampilkan yang diharapkan dapat memberikan hasil yang lebih akurat dari pencarian yang dilakukan. Oleh karena itu, diusulkan penelitian yang berjudul Sistem Temu Kembali Informasi Pasal-Pasal KUHP (Kitab Undang-Undang Hukum Pidana) Berbasis Android Menggunakan Metode *Synonym Recognition* Dan *Cosine Similarity*.

2. LANDASAN KEPUSTAKAAN

2.1 Kitab Undang-Undang Hukum Pidana

Kitab Undang-Undang Hukum Pidana (KUHP) terdiri dari 569 pasal, secara sistematis dibagi dalam (Prasetyo, 2014):

Buku I : Memuat tentang Aturan Umum. Pasal 1-103.

Buku II : Mengatur tentang Kejahatan. Pasal 104-488.

Buku III : Mengatur tentang Pelanggaran. Pasal 489-569.

2.2 Text Mining

Text mining dapat diartikan sebagai menambang data yang berupa teks, dimana sumber data biasanya didapatkan dari suatu dokumen dan tujuannya adalah mencari kata – kata yang dapat mewakili isi dari dokumen sehingga dapat dilakukan analisa keterhubungan antar dokumen.

Tahapan proses – proses *text mining* dibagi menjadi 4 tahap utama, yaitu proses awal terhadap teks (*text preprocessing*), transformasi teks (*text transformation*), pemilihan fitur – fitur yang sesuai (*feature selection*) dan penemuan pola (*pattern discovery*). Masukan

awal dari proses ini adalah suatu data teks dan menghasilkan keluaran berupa pola sebagai hasil interpretasi.

Manfaat dari *text mining* adalah untuk mempermudah pencarian dan membuat inovasi yang dapat membantu manusia untuk mengerti dan menggunakan informasi dari sebuah repository dokumen.

2.3 Information Retrieval

Information Retrieval merupakan bagian dari computer science yang berhubungan dengan pengambilan informasi dari dokumen-dokumen yang didasarkan pada isi dan konteks dari dokumen-dokumen itu sendiri. *Information Retrieval* merupakan suatu pencarian informasi (biasanya berupa dokumen) yang didasarkan pada suatu *query* (inputan user) yang diharapkan dapat memenuhi keinginan user dari kumpulan dokumen yang ada. Sedangkan, definisi *query* dalam *Information Retrieval* menurut referensi merupakan sebuah formula yang digunakan untuk mencari informasi yang dibutuhkan oleh *user*, dalam bentuk yang paling sederhana, sebuah *query* merupakan suatu keywords (kata kunci) dan dokumen yang mengandung keywords merupakan dokumen yang dicari dalam IRS (*Information Retrieval System*).

Secara prinsip, penyimpanan informasi dan penemuan kembali informasi adalah hal yang sederhana. Misalkan terdapat tempat penyimpanan dokumen-dokumen dan seseorang (user) merumuskan suatu pertanyaan (*request* atau *query*) yang jawabannya adalah himpunan dokumen yang mengandung informasi yang diperlukan yang diekspresikan melalui pertanyaan user. User bisa saja memperoleh dokumen-dokumen yang diperlukannya dengan membaca semua dokumen dalam tempat penyimpanan, menyimpan dokumen-dokumen yang relevan dan membuang dokumen lainnya. Hal ini merupakan *perfect retrieval*, tetapi solusi ini tidak praktis. Karena user tidak memiliki waktu atau tidak ingin menghabiskan waktunya untuk membaca seluruh koleksi dokumen, terlepas dari kenyataan bahwa secara fisik user tidak mungkin dapat melakukannya.

2.4 Text Preprocessing

Tahap ini melakukan analisis semantik (kebenaran arti) dan sintaktik (kebenaran susunan) terhadap teks. Tujuan dari pemrosesan awal adalah untuk mempersiapkan teks menjadi data yang akan mengalami pengolahan lebih lanjut. Pada tahap ini pada umumnya terdapat beberapa proses, antara lain tokenizing, filtering, stemming, dan term weighting.

Tahap proses awal terhadap teks bertujuan untuk mempersiapkan teks menjadi data yang akan mengalami pengolahan lebih lanjut. Tahap ini diawali dengan melakukan pemecahan sekumpulan karakter ke dalam kata – kata (*token*). Setiap *token* adalah objek dari suatu tipe sehingga jumlah token lebih banyak daripada tipenya.

2.4.1 Tokenizing

Proses *tokenizing* adalah proses pemotongan *string* masukan berdasarkan tiap kata yang terdapat pada kalimat. Setiap huruf *input* akan diubah menjadi huruf kecil. Semua tanda baca dan tanda hubung akan dihapuskan, termasuk semua karakter selain huruf alfabet.

2.4.2 Filtering

Pada tahap *filtering* adalah tahap mengambil kata – kata penting dari hasil *token*. Pada proses tersebut akan ditentukan istilah yang mewakili isi dari dokumen tersebut sehingga dapat digunakan untuk menggambarkan isi dari dokumen tersebut dan membedakan dokumen dari dokumen lain dalam koleksi. Dalam proses ini dilakukan pembuangan kata yang tidak penting (*stoplist*) atau menyimpan kata yang penting (*wordlist*). *Stoplist* adalah daftar kata yang sering digunakan dan tidak menjelaskan isi dari dokumen, atau dapat disebut dengan istilah *stopword*. Contoh *stopwords* adalah "yang", "dan", "di", "dari" dan seterusnya.

2.4.3 Stemming

Stemming merupakan tahap mencari *root* kata dari tiap kata hasil *filtering*. Pada tahap ini dilakukan proses pengembalian berbagai bentuk kata ke dalam suatu representasi yang sama. Pada *stemming* dengan menggunakan bahasa Inggris yang digunakan adalah proses

penghilangan *suffix* sedangkan untuk bahasa Indonesia proses *stemming* yang digunakan meliputi penghilangan *suffix*, *prefix*, *infix* dan *konfix*.

2.5 Term Weighting

Dalam *text mining*, *term weighting* merupakan salah satu tahapan yang sangat penting. Tahapan ini dilakukan dengan tujuan untuk memberikan suatu nilai/bobot pada term yang terdapat pada suatu dokumen. Bobot yang diberikan terhadap sebuah term bergantung kepada metode yang digunakan untuk membobotinya.

Metode yang digunakan dalam penelitian ini adalah TF-IDF (*Term Frequency - Inverse Document Frequency*).

2.5.1 TF-IDF

TF-IDF adalah metode pembobotan yang paling umum digunakan untuk menggambarkan dokumen dalam *vector space model*. Dalam klasifikasi teks, fungsi pembobotan ini terkait dengan dua metode pembelajaran yang penting yaitu *k*-NN dan SVM. TF-IDF umumnya digunakan untuk membandingkan vektor *query* dengan vektor dokumen menggunakan kesamaan (*similarity*).

Term Frequency (TF) adalah faktor yang menentukan bobot *term* pada suatu dokumen berdasarkan jumlah kemunculannya dalam dokumen tersebut. Nilai kumulatif kemunculan suatu kata (*term frequency*) diperhitungkan dalam pemberian bobot terhadap suatu kata (*term frequency*) diperhitungkan dalam pemberian bobot terhadap suatu kata. Semakin besar jumlah kemunculan suatu *term* dalam dokumen, semakin besar pula bobotnya dalam dokumen atau akan memberikan nilai kesesuaian yang semakin besar.

Inverse Document Frequency (IDF) adalah pengurangan dominasi *term* yang sering muncul di berbagai dokumen. Hal ini diperlukan karena *term* yang banyak muncul di berbagai dokumen, dapat dianggap sebagai *term* umum (*common term*) sehingga tidak penting nilainya. Sebaliknya faktor kejarangmunculan kata (*term scarcity*) dalam koleksi dokumen harus diperhatikan dalam pemberian bobot. Kata yang muncul pada sedikit dokumen harus dipandang

sebagai kata yang lebih penting (*uncommon term*) daripada kata yang muncul pada banyak dokumen. Pembobotan akan memperhitungkan faktor kebalikan frekuensi dokumen yang mengandung suatu kata (*inverse document frequency*). Hal ini merupakan usulan dari George Zipf. Zipf mengamati bahwa frekuensi dari sesuatu cenderung kebalikan secara proposional dengan urutannya.

TF-IDF dihitung menggunakan Persamaan (1) :

$$IDF(t_i) = \log\left(\frac{|D|}{DF(t_i)}\right) \quad (1)$$

Keterangan :

$IDF(t_i)$: *inverse document frequency* dari kata (*term*) t_i

D : jumlah dokumen latih keseluruhan

$DF(t_i)$: jumlah dokumen latih yang memiliki kata (*term*) t_i

Kemudian persamaan untuk menghitung bobot kata (w_i) dalam dokumen dihitung menggunakan Persamaan 2.2 :

$$w_i = TF(t_i, d) \times IDF(t_i) \quad (2)$$

Keterangan :

w_i : bobot kata (*term*) dalam dokumen d

$TF(t_i, d)$: banyaknya kata (*term*) t_i yang muncul dalam dokumen d

$IDF(t_i)$: *inverse document frequency* dari kata (*term*) t_i

2.6 Vector Space Model

Vector Space Model adalah suatu model yang digunakan setelah proses pembobotan, dimana kumpulan dokumen direpresentasikan ke dalam sebuah vektor matrik. Dari matrik tersebut akan didapatkan titik koordinat tertentu.

Kasus klasifikasi teks untuk suatu dokumen biasanya tersusun dari kumpulan kata – kata (*string*), oleh karena itu harus ditransformasi ke dalam suatu representasi yang sesuai untuk proses pembelajaran (*learning*)

dan klasifikasi sistem. Untuk kasus ini digunakan fitur representasi vektor dokumen, dimana dokumen akan diset sebagai suatu rangkaian kata.

Dari hasil pembobotan dengan *TF-IDF* selanjutnya akan dilakukan pembentukan *Vector Space Model*. Gambar 2.1 merupakan contoh hasil pembentukan *Vector Space Model*.

$$\begin{matrix} & T_1 & T_2 & T_3 & T_4 & T_5 \\ \begin{matrix} D_1 \\ D_2 \\ D_3 \end{matrix} & \begin{bmatrix} 0.352 & 0.176 & 0 & 0 & 0 \\ 0.176 & 0 & 0.352 & 0 & 0.528 \\ 0 & 0.528 & 0.176 & 0.954 & 0 \end{bmatrix} \end{matrix}$$

Gambar 1. Contoh Hasil *Vector Space Model*

2.7 Cosine Similarity

Metode *cosine similarity* adalah metode yang digunakan untuk kesamaan atau kedekatan antar dokumen. Semakin besar nilai kesamaan vector query dengan vektor dokumen maka query tersebut dipandang semakin relevan dengan dokumen. Dengan demikian dua vektor dengan orientasi yang sama memiliki kesamaan cosinus dari 1, dua vektor pada 90° memiliki kesamaan 0. *Cosine similarity* terutama digunakan dalam ruang positif, dimana hasilnya dibatasi dengan (0,1). *Cosine similarity* kemudian memberikan tolok ukur seberapa mirip dua dokumen.

$$\cos(d_j, q) = \frac{d_j \cdot q}{|d_j| |q|} = \frac{\sum_{i=1}^t (w_{ij} \cdot w_{iq})}{\sqrt{\sum_{i=1}^t w_{ij}^2} \sqrt{\sum_{i=1}^t w_{iq}^2}}$$

Keterangan :

t = kata di *database*

d = dokumen

q = kata kunci/query

w_{ij} = bobot kata ke i pada dokumen j

w_{iq} = bobot kata ke i pada dokumen q

2.8 Synonym Recognition

Synonym Recognition atau pengenalan kata bersinonim adalah teknik yang digunakan untuk mengenali kata dengan penulisan berbeda namun memiliki makna yang sama.

Pada proses pencocokan sinonim, kaitannya antara isi dokumen dengan kamus sinonim yang terdapat pada database. Jika kata tidak tersedia pada kamus sinonim, maka tidak akan melakukan proses *synonym recognition*. Seluruh isi dokumen akan discan dan dicocokkan dengan kata yang ada dalam kamus, kemudian akan diubah berdasarkan isi kamus sinonim tersebut.

Adapun contoh kinerja sistem untuk *synonym recognition* dengan kalimat “komputer dipakai untuk mengolah data”.

Dan hasil pendeteksian sinonimnya adalah sebagai berikut:

Dipakai=Digunakan

Kata “dipakai” akan diganti dengan kata “digunakan”. Sehingga kalimat yang semula “komputer dipakai untuk mengolah data” diganti dengan kata “komputer digunakan untuk mengolah data”.

2.9 Evaluasi

Tujuan evaluasi percobaan pada klasifikasi (*classifier*) yaitu untuk mengukur keefektifan apakah sistem mengklasifikasi secara benar. Evaluasi biasanya membutuhkan sebuah matriks yang disebut berupa *matriks confusion*. *Matriks confusion* ditunjukkan pada tabel 1.

Tabel 1. *Matriks Confusion*

Ck	<i>Classifier positive label</i>	<i>Classifier negative label</i>
<i>True positive label</i>	A	B
<i>True negative label</i>	C	D

Dari Tabel 1 menunjukkan bahwa jika diberikan kategori Ck, parameter A adalah jumlah dokumen yang berhasil dikategorikan oleh sistem ke dalam kategori Ck, parameter B adalah jumlah dokumen yang mempunyai kategori Ck namun sistem tidak mengklasifikasikannya ke dalam kategori Ck, parameter C adalah jumlah dokumen yang bukan kategori Ck namun sistem mengklasifikasikannya ke dalam kategori Ck,

dan parameter D adalah jumlah dokumen yang tidak termasuk kategori Ck dan sistem juga tidak mengklasifikasikannya ke dalam kategori Ck.

Recall adalah ukuran keberhasilan sistem dalam mengenali dokumen pada setiap kategori tanpa melihat ketepatan klasifikasi yang dilakukan. *Recall* dihitung menggunakan persamaan 7.

$$Recall = \frac{A}{A+B} \quad (7)$$

Precision adalah ukuran keberhasilan sistem dalam melakukan ketepatan klasifikasi tanpa melihat seberapa banyak dokumen yang berhasil dikenali. *Precision* dihitung menggunakan persamaan 8.

$$Precision = \frac{A}{A+C} \quad (8)$$

F-measure mewakili pengaruh relative antara *precision* dan *recall*, yang dihitung dengan persamaan berikut. *F-measure* dihitung menggunakan persamaan 9.

$$F.measure = \frac{2.recall.precision}{(recall+precision)} \quad (9)$$

2.10 SQLite Database

SQLite adalah salah satu software yang embedded yang sangat populer, kombinasi SQL interface dan penggunaan memory yang sangat sedikit dengan kecepatan yang sangat cepat (Safaat, 2011). Dengan adanya SQLite database ini, banyak membantu dalam pembuatan database dari setiap versi android karena SQLite database ini termasuk dalam Android runtime. Untuk pengembangannya, dalam membuat dan membuka database dipergunakan sebuah libraries yang harus kita import, yaitu android.database.sqlite.SQLiteOpenHelper yang menyediakan tiga metode, adapun tiga metode tersebut adalah sebagai berikut (Safaat, 2011):

1) Constructor

Menyediakan representasi versi dari database dan skema yang kita gunakan.

2) onCreate()

Menyediakan SQLiteDatabase object yang kita gunakan dalam definisi tabel dan inisialisasi data.

3) onUpgrade()

Menyediakan fasilitas konversi database dari database versi yang lama ke database versi yang baru atau sebaliknya.

3. PERANCANGAN

3.1 Daftar Kebutuhan

Daftar kebutuhan terdiri dari kebutuhan fungsional dan non-fungsional. Kebutuhan fungsional berisikan proses-proses apa saja yang diberikan oleh perangkat lunak yang akan dibangun, sedangkan kebutuhan non-fungsional tidak secara langsung terkait pada suatu fitur yang ada diperangkat lunak. Pada tabel 2 kebutuhan fungsional ditunjukkan dengan kode penomoran KFX, sedangkan kebutuhan non-fungsional ditunjukkan dengan kode penomoran KNFX pada table 2.

Tabel 2. Kebutuhan Fungsional

Kode	Kebutuhan	Use case
KF01	Aplikasi harus bisa menampilkan daftar buku.	Melihat daftar buku
KF02	Aplikasi harus bisa menampilkan daftar bab-bab.	Melihat daftar bab-bab
KF03	Aplikasi harus bisa menampilkan pasal-pasal KUHP.	Melihat data pasal-pasal
KF04	Aplikasi harus bisa mencari pasal-pasal menggunakan kata yang diinputkan disertai pasal yang terkait dengan pasal tersebut	Mencari data pasal-pasal

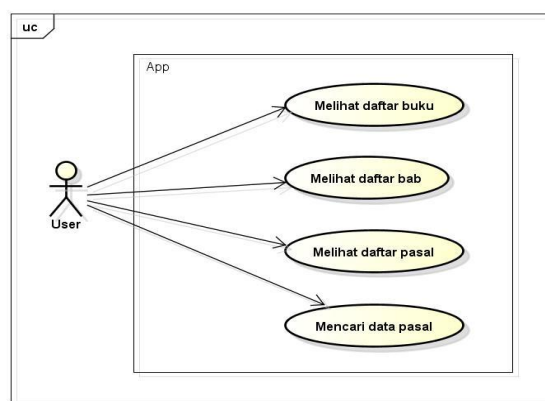
Tabel 3 Kebutuhan Non-Fungsional

Parameter	Kode	Deskripsi Kebutuhan
<i>Compatibility</i>	KNF01	Aplikasi harus bisa diakses pada perangkat bergerak dengan sistem operasi Android pada beberapa versi

Usability	KNF02	Aplikasi harus dapat digunakan dengan mudah oleh pengguna
-----------	-------	---

3.2 Diagram Use Case

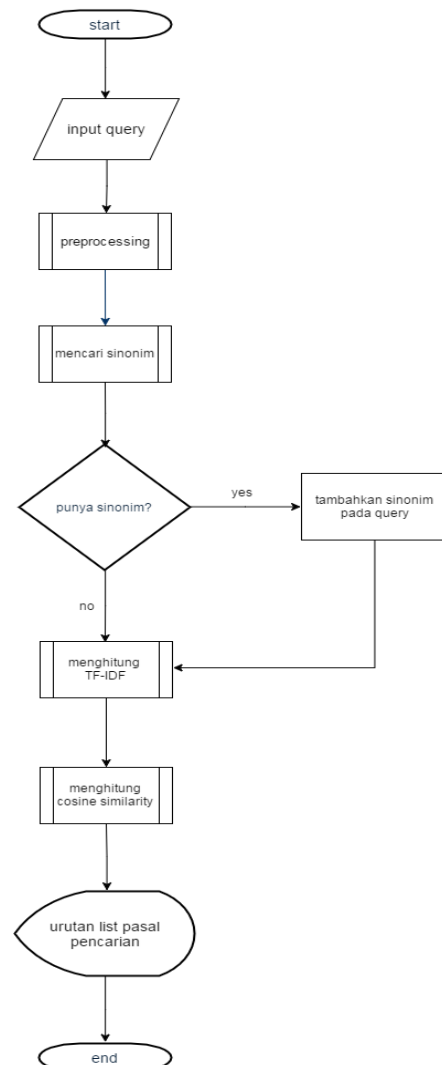
Pemodelan *use case* sistem diperoleh dari kebutuhan fungsional yang digunakan untuk menggambarkan interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat.



Gambar 1. Diagram *use case* aplikasi KUHP

3.3 Diagram Alir Sistem

Diagram alir atau Flowchart merupakan visualisasi dari algoritma yang diterapkan untuk memecahkan persoalan dalam sistem



Gambar 2. Flowchart pencarian pasal menggunakan sistem pencarian

Pada sistem pencarian pasal ini pertama pengguna memasukkan kata yang ingin dicari pada kotak pencarian. Kemudian sistem akan melakukan proses *preprocessing* pada data pasal-pasal dan kata yang inputkan oleh pengguna. Setelah itu sistem akan mencari sinonim dari kata yang diinputkan pengguna, apabila kata tersebut memiliki sinonim dengan kata lain maka kata tersebut akan ditambahkan ke dalam daftar pencarian bersama dengan kata yang diinputkan sebelumnya. Setelah melalui proses pengecekan sinonim (baik yang memiliki sinonim atau tidak) maka proses selanjutnya adalah melakukan perhitungan TF-IDF pada data pasal-pasal dan kata yang diinputkan tersebut, kemudian melakukan

proses perhitungan *cosine similarity* untuk memperoleh hasil pasal mana saja yang terkait dengan kata yang diinputkan oleh pengguna sebelumnya.

4. PENGUJIAN DAN ANALISIS

4.1 Query Pengujian

Query yang digunakan dalam pengujian ini adalah sebagai berikut ;

- 1) Makar
- 2) Pencurian
- 3) Pemerkosaan
- 4) Pembunuhan berencana
- 5) Hukuman mati
- 6) Menjual minuman keras
- 7) Hukuman seumur hidup
- 8) Pelanggaran lalu lintas
- 9) Kejahatan terhadap negara
- 10) Mengedarkan mata uang palsu

4.2 Hasil Pengujian

- 1) Makar

Pasal	Hasil
108	benar
363	salah
48	salah
89	salah
107	benar
286	salah
106	benar
139 b	benar
87	benar
140	benar
139 a	benar
104	benar
90	salah
378	salah
379	salah

Threshold = 5

	Retrieved	Non-Retrieved
Actual	2	6
Non-Actual	3	322

$$\text{Precision} = 0,4$$

$$\text{Recall} = 0,25$$

$$\text{F.measure} = 0,31$$

Threshold = 10

	Retrieved	Non-Retrieved
Actual	6	2
Non-Actual	4	322

$$\text{Precision} = 0,6$$

$$\text{Recall} = 0,75$$

$$\text{F.measure} = 0,66$$

Threshold = 15

	Retrieved	Non-Retrieved
Actual	6	2
Non-Actual	4	322

$$\text{Precision} = 0,53$$

$$\text{Recall} = 1$$

$$\text{F.measure} = 0,7$$

Kemudian Query 2 sampai Query 10 dilakukan perhitungan seperti pada Query 1 di atas untuk mencari nilai *precision*, *recall* dan *f.measure* pada threshold 5, 10 dan 15.

Hasil dari pencarian *precision*, *recall* dan *f.measure* pada semua query dapat dilihat pada table berikut:

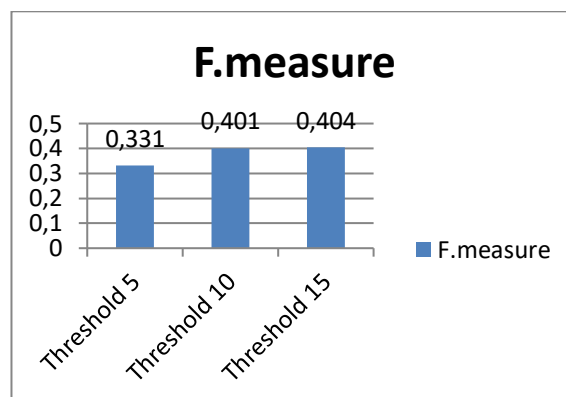
Threshold 5

	Precision	Recall	F.measure
Query 1	0.4	0.25	0.31
Query 2	1	0.23	0.37
Query 3	0.2	1	0.33
Query 4	1	0.18	0.3
Query 5	0	0	0
Query 6	0.6	1	0.75
Query 7	0.2	0.06	0.1
Query 8	0.6	0.43	0.5
Query 9	0.2	0.08	0.12
Query 10	1	0.36	0.53
Rata-rata	0.52	0.359	0.331

Threshold 10

	Precision	Recall	F.measure
Query 1	0.6	0.75	0.66

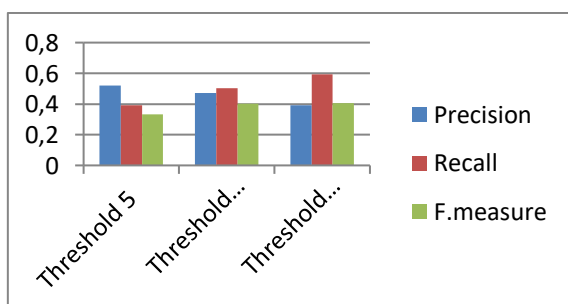
Query 2	0.8	0.3	0.48
Query 3	0.1	1	0.18
Query 4	1	0.36	0.53
Query 5	0	0	0
Query 6	0.3	1	0.46
Query 7	0.6	0.4	0.48
Query 8	0.4	0.57	0.47
Query 9	0.1	0.08	0.09
Query 10	0.8	0.57	0.66
Rata-rata	0.47	0.503	0.401



Gambar 4. Grafik nilai f.measure dari threshold 5, 10 dan 15

Threshold 15

	Precision	Recall	F.measure
Query 1	0.53	1	0.7
Query 2	0.6	0.39	0.47
Query 3	0.07	1	0.13
Query 4	0.87	0.46	0.6
Query 5	0.07	0.05	0.06
Query 6	0.2	1	0.33
Query 7	0.53	0.53	0.53
Query 8	0.33	0.71	0.45
Query 9	0.13	0.16	0.15
Query 10	0.6	0.64	0.62
Rata-rata	0.393	0.594	0.404



Gambar 3. Grafik average dari precision, recall dan f.measure pada tiap threshold

4.3 Hasil Analisa

Dari hasil pengujian di atas dapat diketahui bahwa hasil pencarian yang optimal diperoleh dari threshold 15 dengan nilai f.measure 0,404.

Threshold 15 dapat memiliki nilai f.measure tertinggi dikarenakan pada skenario pengujian pada 'query 5' ketika diambil 5 dan 10 pasal hasilnya adalah semua salah sehingga menghasilkan nilai *precision*, *recall* dan *f.measure* yaitu 0.

Sedangkan ketika diambil 15 pasal ada pasal yang benar sehingga *precision*, *recall* dan *f.measure* mempunyai nilai dan itu mempengaruhi jumlah rata-rata nilai *f.measure* secara keseluruhan.

5. KESIMPULAN

Setelah dilakukan penelitian dan juga pengujian diatas didapatkan kesimpulan sebagai berikut :

- 1) Pencarian pasal-pasal KUHP dapat diterapkan pada metode synonym recognition dan cosine similarity dengan langkah-langkah yang berawal dari preprocessing kemudian mencari kata yang bersinonim, setelah itu menghitung nilai dari TF-IDF dan yang terakhir adalah menghitung nilai cosine similarity
- 2) Tingkat precision, recall dan f.measure pada metode Synonym Recognition dan Cosine Similarity mempunyai nilai yang optimal pada threshold 15 dengan nilai

precision 0,393, recall 0,594 dan f.measure sebesar 0,404.

6. DAFTAR PUSTAKA

- Atiq ,Dian Astitatul. 2015. *Indonesia Negara Hukum*. [online] Tersedia di: <http://academia.edu/6234004/Indonesia_Negara_Hukum>
- B.M. Harwani.(2013).” The Android Tablet Developer's Cookbook”. Addison-Wesley.425-430.
- Godfrey Nolan, David Truxall, Onur Cinar.(2013).”Android Best Practices”. Addison-Wesley.167-168.
- Hearst, M. (2003). *What Is Text Mining*. Retrieved Juli 3, 2016, from <http://people.ischool.berkeley.edu/~hears/t/text-mining.html>
- Prasetyo, Teguh. 2014. *Hukum Pidana*. Jakarta: Rajawali Pers
- Ronan Schwarz, Phil Dutson, James Steele, Nelson To.(2013).” The Android Developer's Cookbook: Building Applications with the Android SDK”. Addison-Wesley.
- Safaat, Nazruddin (2011), *Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*, Informatika, Bandung.
- Wati, Arrosyida Rizqa Buana. 2012. *Sistem Penilaian Otomatis Jawaban Essay Menggunakan Deteksi Similarity*. Program Studi Teknik Informatika. Fakultas Teknologi Industri. Universitas Pembangunan Nasional “Veteran”. Jawa Timur
- Zafikri, Atika. 2008. Implementasi Metode Term Frequency Invers Document Frequency (TF-IDF) Pada Sistem Temu Kembali Informasi. Tugas Akhir Program Studi Ilmu Komputer Fakultas Matematika Dan Ilmu Pengetahuan Alam Universitas Sumatera Utara : Medan