Implementasi Mesin Pencarian Berbasis Ontologi pada Twitter untuk Membantu Pengukuran Happiness Index Kota Bandung

Muhammad Arifino Setyawan[1)], Anisa Herdiani[2)], Nungki Selviandro[3)]
[1),2),3)]Prodi S1 Teknik Informatika, Fakultas Informatika, Universitas Telkom
[1)]ariffinsetya@gmail.com, [2)]anisaherdiani@telkomuniversity.ac.id,
[3)]nselviandro@telkomuniversity.ac.id

Abstrak

Media sosial merupakan tren teknologi yang terus meningkat dalam beberapa tahun terakhir. Seiring dengan peningkatan penggunanya, semakin banyak pengguna yang menyalurkan opininya melalui media sosial. Opini masyarakat merupakan salah satu hal yang bisa digunakan untuk menjadi input dari pengukuran happiness index. Happiness index merupakan suatu nilai yang dapat menggambarkan tingkat kesejahteraan masyarakat suatu wilayah. Namun untuk mengukur happiness index dibutuhkan data yang banyak, sehingga diperlukan suatu sistem yang dapat mengumpulkan data dari media sosial. Search engine merupakan salah satu cara yang bisa digunakan untuk mengumpulkan data, namun karena domain pencarian yang spesifik, penggunaan search engine biasa yang berbasis keyword dianggap kurang efisien sehingga diperlukan search engine yang dapat mencari data berdasarkan suatu domain yang spesifik. Penggunaan ontologi dalam pencarian dapat menangani masalah tersebut. Ontologi merupakan salah satu cara merepresentasikan pengetahuan, dengan bermula dari domain happiness index dapat dibangun sebuah ontologi yang dapat merepresentasikan happiness index. Paradigma top-down dan metodologi Noy McGuinness digunakan dalam konstruksi ontologi. Dalam melakukan penilaian dengan ontologi, metode dari Ehrig akan dipilih karena karakteristik data yang memiliki kemiripan dengan data yang akan diperoleh dari media sosial. Kemudian data tersebut akan diklasifikasi menggunakan metode klasifikasi berbasis kedalaman ontologi. Pada penelitian ini akan diuji performa search engine yang menggunakan ontologi menggunakan komputasi Ehrig dan klasifikasi menggunakan ontologi. Berdasarkan hasil pengujian yang dilakukan, diperoleh hasil yang memuaskan dengan nilai F1-Measire 86% untuk penilaian data berdasarkan ontologi dan hasil yang mencapai 84% dan 100%, untuk perbandingan relevansi hasil pencarian dibandingkan dengan search engine tanpa ontologi. Untuk klasifikasi diperoleh nilai accuracy mencapai 81% yang cukup baik untuk metode klasifikasi yang diusulkan.
Kata kunci: ***happiness index, media sosial, ontologi, search engine, klasifikasi***

Abstract

Social media is a trend of technology that keep increasing in the last few years. With the increase of users, more users share their opinions via social media. People opinion is one of thing that can be used as an input for happiness index measurement. Happiness index is a value that is able to show level of people happiness in a particular area. But to measure happiness index, it needs a lot of data, so it needs a system of collect data from social media. Search engine is one of the way to collect data but because the searching domain is specific, using keyword-based search engine is considered not efficient enough so domain-specific search engine is needed. Use of ontology in searching can handle that problem. Ontology is one of the way to represent knowledge, starting with happiness index domain it can be built to an ontology that can represent happiness index. According to that idea, top-down paradigm and Noy McGuinness methodology will be used to construct ontology. In searching process, a method from Ehrig and Maedche is chosen because the data characteristic is similar with data collected from social media. In this research, testing will be performed to measure performance of ontological-search engine that use Ehrig computation and ontology- using classification. Based on the result, it shows that the result is satisfying. Filtering functionality achieves 86% of F1-Measure score to determine the relevance of data to ontology and result of 84% and 100% of comparing ontological search engine and standard search engine. For classification, it achieves 81% accuracy score, quite good score for a proposed method.
Keywords: happiness index, social media, ontology, search engine, classification

1 Pendahuluan

Media sosial merupakan suatu layanan yang berjalan di atas internet yang memungkinkan penggunanya saling berkomunikasi baik melalui teks, suara, gambar maupun video kepada sesama penggunanya. Penggunaan media sosial menjadi tren yang diminati oleh masyarakat Indonesia. Hal tersebut terlihat dari meningkatnya pengguna media sosial yang mencapai 23% atau sebanyak 64 juta pengguna pada tahun 2015 [1].

Penggunaan media sosial yang semakin meningkat membuat banyak opini masyarakat yang dapat dilihat dari media sosial tersebut. Opini-opini masyarakat tersebut dapat diolah untuk mengukur happiness index. Happiness index sendiri merupakan suatu indeks atau nilai yang dapat menggambarkan tingkat kepuasan masyarakat terhadap suatu wilayah berdasarkan sepuluh aspek esensial [2]. Pengukuran happiness index dapat berfungsi melihat bagaimana kinerja pemerintah kepada rakyatnya. Bandung sebagai salah satu kota besar di Indonesia merupakan salah satu contoh yang tepat untuk dilakukan pengukuran happiness index. Wali Kota Bandung Ridwan Kamil sendiri mengatakan bahwa happiness index merupakan salah satu proyek yang beliau ingin kerjakan selama menjabat sebagai wali kota [3], [4]. Semakin banyaknya pengguna media sosial maka pengukuran happiness index bisa dilakukan berdasarkan opini yang ada di media sosial.

Pengumpulan opini dari media sosial dapat dilakukan untuk melakukan pengukuran happiness index. Media sosial yang menjadi pilihan pengambilan data adalah Twitter. Twitter merupakan salah satu media sosial yang paling disukai oleh masyarakat Indonesia dengan pengguna mencapai 11% dari pengguna media sosial di Indonesia [1]. Keunggulan Twitter adalah kemudahan pengguna menuliskan tweet baru dengan cepat sehingga pengguna lebih sering menuliskan hal-hal baru yang baru saja terjadi. Hal tersebut membuat lebih banyak hal yang dibagikan oleh pengguna ke jejaring sosial. Namun, dalam mengumpulkan opini-opini dari media sosial dibutuhkan suatu sistem yang praktis. Penggunaan search engine merupakan salah satu cara yang tepat karena dapat menemukan data dalam jumlah yang besar tanpa perlu menghabiskan banyak tenaga jika dilakukan secara manual. Namun, search engine biasa dianggap belum cukup [5]. Search engine biasa mengandalkan keyword dalam pencariannya, sehingga search engine bisa memasukkan data selama keyword yang dicari muncul tanpa melihat konteks dari data yang ditemukan, hal ini bisa membuat data-data yang tidak relevan muncul ke dalam hasil pencarian. Misalkan pencarian dengan dilakukan dengan meggunakan kata "Kemiskinan Bandung", maka hasil pencarian yang mengandung satu atau kedua kata tersebut akan muncul, namun dengan penggunaan ontologi dalam pencarian dapat meningkatkan akurasi dengan melihat makna/konteks dari data yang ditemukan sehingga data yang dihasilkan lebih relevan [5].

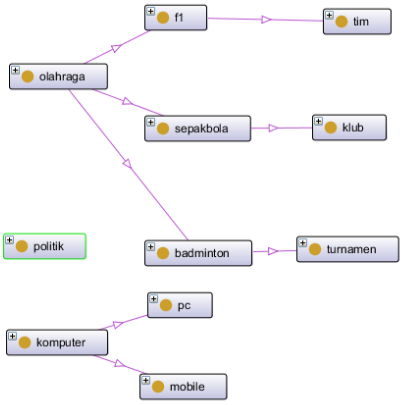
Pembuatan search engine berbasis ontologi ini dapat membantu pengumpulan data dalam pengukuran happiness index. Ontologi yang digunakan dikonstruksi menggunakan paradigma top-down. Hal ini dilakukan karena konstruksi ontologi berangkat dari suatu domain yang sudah ditetapkan yaitu happiness index. Kemudian metodologi yang digunakan untuk melakukan konstruksi ontologinya adalah metodologi Noy McGuinness. Metodologi yang diusulkan oleh Noy McGuinness merupakan metodologi yang bersifat umum dan dapat diterapkan pada semua domain pada pembuatan ontologi sehingga dapat diaplikasikan pada pembuatan ontologi happiness index. Search engine yang akan mampu melakukan filtering data berdasarkan kemiripan dengan ontologi. Metode yang digunakan adalah metode komputasi relevansi Ehrig. Metode ini digunakan karena karakteristik data yang mirip sehingga cocok digunakan pada search engine yang akan dibuat. Setelah itu data yang telah difilter tadi akan dilakukan klasifikasi menggunakan metode klasifikasi berbasis kedalaman dan bobot yang merupakan metode yang diusulkan dalam penelitian ini.

2 Dasar Teori

2.1 Ontologi

Ontologi merupakan representasi formal dari pengetahuan melalui kumpulan konsep pada sebuah domain dan hubungan antar konsep tersebut [1]. Ontologi dapat mendeskripsikan sebuah domain dengan membaginya ke beberapa konsep dan mendeskripsikan relasinya. Konsep atau kelas adalah komponen terbesar dari sebuah ontologi. Kelas dapat dijelaskan oleh beberapa atribut, yaitu atribut yang konkrit seperti tipe data. Kelas juga dapat menyimpan informasi secara langsung dengan *instance* [2]. Gambar

2-1 menunjukkan contoh gambaran ontologi sederhana.



Gambar 2-1 Gambaran Sistem

OWL (Web Ontology Language) merupakan bahasa semantik yang didesain untuk merepresentasikan pengetahuan. OWL merupakan bahasa yang dapat digunakan untuk merepresentasikan ontologi. OWL dapat merepresentasikan pengetahuan yang kompleks mengenai benda, kumpulan benda dan relasi antar benda tersebut [3].

OWL merupakan salah satu dari rekomendasi oleh W3C yang berhubungan dengan Semantic Web yaitu XML, XML Schema, RDF, RDF Schema, dan OWL [4]. OWL sendiri merupakan dokumen RDF, namun jika RDF hanya merupakan model data untuk objek serta relasinya, OWL menambahkan kata benda untuk menjelaskan properti dan kelas, relasi antar kelas, kardinalitas, kesamaan, jenis-jenis properti serta karakteristik properti [4].

2.2 *Noy McGuinness Methodology*

Metodologi Metodologi ini diusulkan oleh Natalya F. Noy dan Deborah L. McGuinness ini menjadi salah satu alternatif dalam membangun ontologi. Metodologi ini dibuat karena tidak ada cara terbaik dalam mengembangkan ontologi dan dibuatlah proses pembangunan ontologi dengan pendekatan iteratif yang bersifat umum [5].

Konstruksi ontologi dengan metodologi Noy McGuinness menggunakan tahap-tahap sebagai berikut [5]:

1. Menentukan domain dan lingkup dari ontologi yang akan dibangun.
2. Mempertimbangkan ontologi-ontologi yang sudah ada untuk digunakan kembali.
3. Menuliskan istilah-istilah penting yang berhubungan dengan domain pada ontologi.
4. Mendefinisikan kelas dan hierarki yang akan digunakan.
5. Mendefinisikan properti dari kelas yang telah didefinsikan.
6. Mendefinisikan *facet* dari properti.
7. Membuat *instance.*

2.3 Komputasi Relevansi Ehrig

Ehrig menerapkan suatu metode berbasis *in stance based matching* pada salah satu karya il miah berjudul "Ontology-Focused Crawling of Web Documents". Ehrig membuat *search en gine* berbasis ontologi untuk dokumen web. M etode komputasi tersebut diterapkan pada *cr awler* untuk membandingkan data yang se dang pada proses *crawling* dengan ontologi yang digunakan. Beberapa langkah pada metode tersebut adalah [6]

1. Membangun *entity reference.*
   Melihat kemunculan dari *term* dari data, *term* yang dihitung adalah *term* yang ada di *lexicon*.
2. *Background Knowledge Compilation*
   Mencari relevansi tiap entitas yang ada di referensi entitas berdasarkan set tertentu. Terdapat empat set yang bisa digunakan yaitu:
   a. *Single sets*: tiap entitas berdiri sendiri tidak terhubung ke entitas lain.
   b. *Taxonomic sets*: entitas memiliki hubungan dengan entitas yang berada dalam satu taksonomi.
   c. *Relational sets*: entitas memiliki hubungan lebih dari hubungan taksonomi dan mempertimbangkan konsep relasional.
   d. *Total sets:* melihat keseluruhan ontologi dengan pengurangan bobot tiap jarak.
3. Melakukan *summarization.*
   Berdasarkan hasil kompilasi maka akan dilakukan *summarization. Summarization* bisa dilakukan dengan tiga cara yaitu:
   a. Fungsi total: menjumlahkan semua nilai entitas yang ada di hasil kompilasi.
   b. Fungsi *max:* mencari nilai maksimal dari semua nilai entitas yang ada di hasil kompilasi.
   c. Fungsi *min:* mencari nilai minimal dari semua nilai entitas yang ada di hasil kompilasi.

   Gambar 2.3 menunjukkan gambaran bagaimana proses komputasi Ehrig berjalan menggunakan *relational sets* dan *max summarization*.

2.4 *Preprocessing*

*Preprocessin*g adalah proses pengubahan bentuk data yang terstruktur sembarang menjadi data yang terstruktur sesuai kebutuhan untuk proses dalam *text mining* [7], berikut adalah langkah-langkah *preprocessing* yang digunakan dalam penelitian ini :

1. *Case Folding*
   Case folding adalah tahap mengubah semua huruf dalam dokumen menjadi huruf kecil [7].
2. *URL Removal*
   URL removal adalah tahap menghapus URL dari dalam dokumen.
3. *Symbol Removing*
   Symbol removing adalah tahap menghapus symbol dari dalam dokumen.
4. *Number Replacement*
   Number replacement adalah pendekatan yang mengubah semua angka menjadi simbol lain [8].
5. *Tokenizing*
   Tokenizing adalah tahap pemecahan kalimat berdasarkan tiap kata yang menyusunnya [7].
6. *Stopword Removal*
   Stopword removal adalah tahap menghapus kata-kata yang berupa *stopword* seperti ini, itu, di, dsb.
7. *Phrases Lookup*
   Phrases lookup adalah tahap melihat apakah ada frase pada hasil token. Jika ada maka token-token tersebut akan digabung menjadi satu.
8. *Synonymy Recognition and Word Generalization*
   Synonymy recognition adalah tahap melihat sinonim dari sebuah kata dan menggantinya berdasarkan kamus [8]. *Word generalization* adalah tahapan mengganti sebuah kata menjadi kata yang lebih umum [8].

2.5 *Training-less Ontology-based Using Depth and Weight Classification* (TRODEW Classification)

Klasifikasi data merupakan salah satu fungsi pada *data mining* yang mengelompokkan suatu data ke sebuah koleksi atau kategori [9]. Tujuan dari klasifikasi adalah memprediksi target kelas secara akurat untuk tiap kasus pada data [9].

Klasifikasi menggunakan ontologi merupakan salah satu topik riset yang intensif [10]. Pada penelitian ini penulis mengusulkan metode klasifikasi berbasis ontologi menggunakan kedalaman dan bobot yang bersifat *training-less.* Metode ini menggunakan ontologi sebagai salah satu input parameter yang dibutuhkan selain data yang ingin diklasifikasi. Metode ini menggunakan ontologinya sendiri sebagai *classifier*nya. Kedalaman dari suatu konsep dan *instance* berpengaruh terhadap bobot dari *term* yang sedang dinilai bobotnya. Dalam satu koleksi data, maka kelas yang memiliki bobot terberat akan menjadi hasil prediksi kelas dari metode klasifikasi ini.

Sudah ada metode klasifikasi yang menggunakan ontologi seperti *Janik's Training-less Ontology-based Text Categorization* [10] atau TODWEB [11]. Metode klasifikasi Janik mengubah data menjadi *graph* dan dalam pembobotannya menggunakan kesamaan *property* dan *literal* [10], sedangkan TODWEB menganggap data sebagai *bag of words* dan *bag of concepts* lalu mencari kemiripan dengan menghitung *semantic relatedness* [11]. Sedangkan pada metode klasifikasi TRODEW, data yang digunakan akan diubah menjadi sebuah *list* yang kemudian tiap *term* di *list* tersebut akan dicari kelas yang berhubungan dan dihitung bobotnya untuk menambahkan bobot kelas tersebut. Perhitungan bobot dilakukan tergantung dengan kesamaan *property, literal* serta kedalaman dari *term* tersebut di ontologi.

2.6 *Precision, Recall, Accuracy, F1-Measure*

Precision dan *recall* merupakan salah satu cara paling dasar dan sering digunakan untuk mengukur efektivitas dari *information retrieval* [12]. *Precision* adalah nilai yang menunjukkan nilai dari dokumen yang relevan. Dalam perhitungan precision dan recall ada empat nilai yang digunakan yaitu *TP (True Positive), TN (True Negative), FP (False Positive)* dan *FN (False Negative).*

TP (True Positive) adalah nilai dari data jika label dan hasil klasifikasi / *assignment* sama-sama bernilai *true*. *TN (True Negative)* adalah nilai dari data jika label dan hasil klasifikasi / *assignment* sama-sama bernilai *false*. *FP (False Positive)* adalah nilai dari data jika label bernilai *false* namun hasil klasifikasi / *assignment* bernilai *true*. *FN (False Negative)* adalah nilai dari data jika label bernilai *true* namun hasil klasifikasi / *assignment* bernilai *false*.

Accuracy merupakan salah satu alternatif dalam menilai suatu sistem

information retrieval, nilai *accuracy* didapat dari pembagian data yang benar diklasifiksi dibagi semua data [12].

Suatu penilaian yang menjembatani nilai *precision* dan *recall* adalah *F-Measure* [12]. *F-Measure* merupakan rata-rata *harmonic* antara *precision* dan *recall,* perhitungan *F-Measure* sendiri bisa dikondisikan untuk lebih menekankan pada *precision* atau *recall,* namun perhitungan yang biasa dilakukan adalah *precision* dan *recall* memiliki bobot yang seimbang , perhitungan ini disebut *F1-Measure* [12].

3 Pembahasan

3.1 Rancangan Sistem

Perancangan sistem yang akan dibangun dilakukan berdasarkan studi literatur serta kebutuhan-kebutuhan yang telah ditemukan sebelumnya. Pada tahap ini juga dilakukan perancangan konstruksi ontologi. Konstruksi ontologi akan dilakukan berdasarkan studi literatur berdasarkan domain *happiness index* untuk menjadi dasar dari sistem.

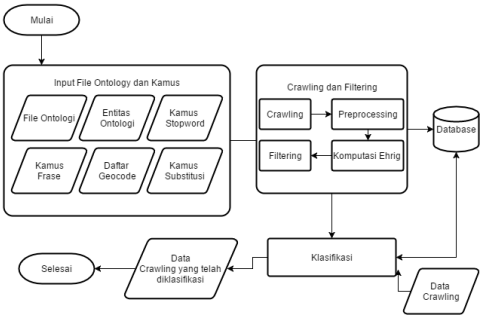
Dalam konstruksi ontologi akan digunakan paradigma *top-down.* Paradigma *top-down* dipilih karena ontologi yang akan dibuat bermula dari domain yang sudah ditentukan yaitu *happiness index.* Teknik yang akan digunakan dalam melakukan konstruksi ontologi adalah metodologi dari Noy McGuinness. Teknik tersebut dipilih karena dokumentasi yang cukup lengkap mengenai tahapan proses yang dilakukan. Selain itu, metodologi ini dipilih karena sifatnya yang umum tidak terikat pada suatu domain tertentu seperti metodologi dari Uschold dan King, dan Grunninger dan Fox.

Setelah konstruksi ontologi, maka perancangan sistem *search engine* dilakukan. *Search engine* yang dibuat nantinya akan memiliki beberapa fungsi yang bisa dilakukan seperti:

1. Membaca file ontologi.
2. Membaca kamus-kamus untuk *preprocessing*.
3. Melakukan *crawling* di Twitter.
4. Menerapkan ontologi untuk menyeleksi data mana yang memiliki relevansi dengan ontologi dengan komputasi Ehrig
5. Mengklasifikasikan data sesuai ontologi dengan klasifikasi TRODEW.
6. Menyimpan hasil *crawling*.

Sistem harus dapat membaca file ontologi serta kamus-kamus yang dibutuhkan untuk *preprocessing.* Kemudian pada proses *crawling* yang dilakukan akan menggunakan Twitter Search API. Setelah *crawling* dilakukan maka data tweet tersebut akan dinilai mengunakan komputasi Ehrig yang memanfaatkan ontologi. Metode tersebut dipilih karena karakteristik data yang mirip, dimana Ehrig membuat *search engine* pada dokumen berupa teks. Sedangkan data yang digunakan pada Tugas Akhir ini adalah data dari media sosial Twitter berupa teks. Setelah data selesai dipilih maka data tersebut akan dilakukan klasifikasi berdasarkan ontologi yang digunakan menggunakan metode TRODEW yang diusulkan. Kemudian hasil dari semua proses tersebut akan disimpan ke dalam file.

Berdasarkan hasil dari proses yang sudah dilakukan, terbentuklah rancangan sistem yang akan diimplementasi pada tahap selanjutnya.



Gambar 3-1 Gambaran Sistem

Penjelasan mengenai gambaran sistem yang diusulkan yaitu:

a. Gambar di atas menjelaskan mengenai alur proses dan parameter dari sistem yang akan dibangun.
b. Konstruksi ontologi harus dilakukan karena file ontologi akan menjadi input sebelum *seach engine* bisa digunakan.
c. Kamus-kamus dan daftar *geocode* harus ada sebelum *search engine* bisa digunakan.
d. Ontologi *happiness index* merupakan hasil dari konstruksi ontologi.
e. *Crawling* akan dilakukan menggunakan *Twitter Search API*.
f. Hasil *crawling* akan dilakukan *preprocessing*.
g. Hasil *crawling* akan disimpan ke *database.*
h. *Filtering* dilakukan dengan komputasi Ehrig.
i. Klasifikasi menggunakan metode klasifikasi TRODEW menggunakan ontologi berbasis kedalaman dan bobot.

j. Klasifikasi bisa dilakukan menggunakan data dari *database* ataupun input data berupa .xlsx
k. Data hasil klasifikasi berupa .xlsx

3.2 Komponen Perangkat Keras dan Lunak

Berikut penjelasan mengenai spesifikasi perangkat keras dan lunak yang digunakan:

3.2.1 Komponen Perangkat Lunak

Perangkat lunak yang digunakan memiliki spesifikasi sebagai berikut :

a. Netbeans 8.0.2
b. MySQL
c. Windows 10 Professional 64bit
d. Protégé

3.2.2 Komponen Perangkat Lunak

Komponen perangkat keras yang digunakan adalah sebagai berikut :

a. Intel(R) Core(TM) i7 – 4700HQ @ 2.4 GHz
b. RAM 8GB
c. Storage 1 TB HDD
d. Koneksi internet via modem

3.3 Gambaran Sistem Detail

3.3.1 Konstruksi Ontologi

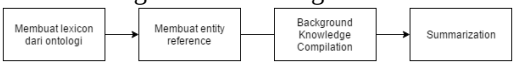
Dalam mengkonstruksi ontologi akan digunakan paradigma *top-down*. Paradigma ini dipilih karena pembuatan ontologi bermula dari domain yang sudah ditentukan yaitu *happiness index*. Sedangkan metodologi yang akan digunakan adalah metodologi dari Noy McGuinness. Konstruksi ontologi dengan metodologi Noy McGuinness menggunakan tahap-tahap sebagai berikut [5]:

1. Menentukan domain dan lingkup dari ontologi yang akan dibangun.
2. Mempertimbangkan ontologi-ontologi yang sudah ada untuk digunakan kembali.
3. Menuliskan istilah-istilah penting yang berhubungan dengan domain pada ontologi.
4. Mendefinisikan kelas dan hierarki yang akan digunakan.
5. Mendefinisikan properti dari kelas yang telah didefinsikan.
6. Mendefinisikan *facet* dari properti.
7. Membuat *instance*.

Konstruksi ontologi akan dilakukan menggunakan tools Protégé. Tools ini digunakan karena Protégé merupakan salah satu tools yang mendukung konstruksi ontologi serta memiliki banyak fitur yang mendukung dalam konstruksi ontologi.

3.3.2 *Komputasi Ehrig*

Komputasi Ehrig yang bekerja pada sistem ini digambarkan oleh gambar 3-2.



Gambar 3-2 Gambaran proses komputasi Ehrig

Berikut adalah penjelasan masing-masing proses dari komputasi Ehrig:

1. Membuat *lexicon* dari ontologi
   Pada tahap ini *lexicon* untuk *classes* dan *instances* diisi berdasarkan ontologi yang digunakan.
2. Membuat *entity reference*
   Pada tahap ini *entity reference* dibuat. Data berupa *list* hasil *preprocessing* akan dibandingkan dengan *lexicon* yang telah dibuat. Algoritma yang digunakan ditunjukkan oleh gambar 3.10.
3. *Background knowledge compilation*
   Berdasarkan hasil *entity reference*, akan dilihat apakah ada relevansi dari masing-masing entitas. Ada beberapa *relevance sets* yang bisa digunakan dalam penelitian kali ini yaitu *single* dan *taxonomic*. *Taxonomic sets* merupakan opsi teerbesar dari set yang bisa digunakan karena ontologi yang dikonstruksi hanya memiliki hubungan taksonomi. Kemudian dihitung *score* nya dengan mengalikan kemunculan dengan *term*.
4. *Summarization*
   Berdasarkan hasil *background knowledge compilation* maka akan dilakukan *summarization*. Terdapat beberapa cara yang bisa digunakan seperti penjumlah, mencari nilai *max/min*.

3.3.3 Klasifikasi dengan Trodew

Klasifikasi TRODEW dilakukan untuk mengklasifikasi tweet ke sepuluh kelas utama dari ontologi yang digunakan. Metode ini meminjam satu proses dari komputasi Ehrig yaitu pembuatan *lexicon* dengan ontologi. Gambar 3-3 menggambarkan algoritma dari klasifikasi dengan TRODEW.

```
Algorithm TRODEW-Classification (input : inputData, PVW, PNW; output : list of result)
For each word in inputData
    check if the word is instance or class in ontology
        if the word is instance do
            score := 0
            find the root class of the instance
                depth := depth of class of instance to root class + 1
            put the root class name to map result if it is not available
            scoreProperties := 0
            for each properties of the instance
                if match then
                find the value of the properties in the rest of inputData
                    if available then
                        scoreProperties := scoreProperties + PVW + PNW
                    else
                        scoreProperties := scoreProperties + PNW
                score := score + scoreProperties
            score := score + (DIWR*depth)
            key := root class name
            add score to map according to the key
        else
            score := 0
            find the root class of the instance
                depth := depth of class of instance to root class + 1
            put the root class name to map result if it is not available
            score := score + (DIWR*depth)
            key := root class name
            add score to map according to the key
for each key in map
    find the max value
    if there is more than one key with same max value then
        output all the keys
    else
        output the key with max value
```

Gambar 3-3 Algoritma klasifikasi TRODEW

3.4 Pengujian dan Analisis Sistem

Dataset yang digunakan dalam pengujian merupakan hasil *crawling* dari sistem yang kemudian dilakukan *labelling* manual untuk menentukan apakah tweet tersebut berhubungan dengan ontologi dan termasuk kelas apakah tweet tersebut. Terdapat 3 dataset yang digunakan yaitu:

1. Dataset berjumlah 2385 tweet dengan pencarian tanpa *keyword*.
2. Dataset berjumlah 72 tweet dengan *keyword* pencarian "sehat".
3. Dataset berjumlah 28 tweet dengan *keyword* pencarian "menteri, plastik, laptop".

Ontologi yang digunakan ada 2 yaitu:

1. Ontologi HappinessV1
2. Ontologi HappinessV2, perbaikan dari V1 dengan penambahan *instances* serta penambahan pada kamus.

Dataset untuk klasifikasi sudah divalidasi oleh expert sebelum digunakan, hasil kesalahan labelling < 1%.

3.4.1 Pengujian Performansi Komputasi Ehrig untuk Menghasilkan Nilai yang Menunjukkan Relevansi Tweet dengan Ontologi

Pengujian ini dilakukan untuk mengetahui apakah komputasi Ehrig bisa digunakan untuk membedakan tweet mana yang relevan dengan ontologi atau tidak. Skenario pengujian yang digunakan yaitu:

1. Tanpa preprocessing menggunakan ontologi HappinessV1.
2. Menggunakan preprocessing menggunakan ontologi HappinessV1.
3. Tanpa preprocessing menggunakan ontologi HappinessV2.
4. Menggunakan preprocessing menggunakan ontologi HappinessV2.

Hasil pengujian dari skenario ini ditunjukkan di tabel 3-1.

Tabel 3-1 Performansi Komputasi Ehrig dalam Menilai Relevansi Tweet ke Ontologi

Skenario	precision	recall	accuracy	F1-measure
1	94.44%	38.51%	67.90%	54.07%
2	91.95%	77.12%	88.82%	83.28%
3	96.22%	38.94%	68.20%	54.79%
4	94.06%	90.65%	92.94%	91.76%

Dari semua skenario yang telah dilakukan di atas, terbukti bahwa komputasi Ehrig mampu memberi nilai kepada tweet yang menunjukkan relevansinya dengan ontologi. Namun, dalam pengujian terlihat, beberapa syarat untuk komputasi Ehrig dapat memiliki performa yang baik, yaitu

1. Proses *preprocessing* dibutuhkan dalam proses komputasi Ehrig. Dari semua skenario terlihat bahwa penggunaan *preprocessing* meningkatkan performa secara signifikan, hal ini terjadi karena proses komputasi Ehrig masih berbasis kepada *string matching* sehingga *preprocessing* yang baik diperlukan.
2. Ontologi dan kamus yang lengkap diperlukan dalam proses komputasi Ehrig untuk mencapai performa yang baik. Terlihat dari pengujian dengan dataset 1 bahwa penggunaan ontologi dan kamus yang lebih lengkap dapat meningkatkan performa dari komputasi Ehrig secara signifikan.

3.4.2 Pengujian Perbandingan Performansi Antara Hasil Pencarian Search Engine yang Menggunakan Ontologi dan Seach Engine Biasa Hanya dengan Twitter Search API

Pengujian ini dilakukan untuk menguji apakah sistem yang dibangun dapat memiliki peningkatan dari segi hasil pencarian yang lebih relevan, maka akan dilakukan pengujian dengan *precision*, *recall*, *accuracy* dan *F1-measure* untuk membandingkan performa antara *search engine* yang menerapkan ontologi dan yang tidak. Hasil pengujian skenario ini ditunjukkan di tabel 3-2.

Tabel 3-2 Perbandingan Performa Search Engine Dengan dan Tanpa Ontologi

Dataset	Nilai	Dengan Ontologi	Tanpa Ontologi
1	Precision	86.41 %	26.25%
	Recall	100 %	100 %
	Accuracy	86.41 %	26.25 %

	Fmeasure	92.71 %	41.58 %
	True Positive	604	626
	False Positive	95	1759
2	Precision	95.77 %	95.83 %
	Recall	100 %	100 %
	Accuracy	95.77 %	95.83 %
	Fmeasure	97.84 %	97.87 %
	True Positive	68	69
	False Positive	3	3
3	Precision	100 %	46.43 %
	Recall	100 %	100 %
	Accuracy	100 %	46.43 %
	Fmeasure	100 %	63.41 %
	True Positive	10	13
	False Positive	0	15

Dari ketiga pengujian terlihat performa *search engine* dengan ontologi sangat baik untuk menghasilkan hasil pencarian yang relevan dengan ontologi yang digunakan terbukti dengan nilai *F1-Measure* maupun *accuracy* yang tinggi, walaupun *search engine* tanpa ontologi menghasilkan jumlah tweet yang relevan lebih banyak dari pencarian *search engine* dengan ontologi untuk ketiga pengujian, namun *search engine* tanpa ontologi juga menghasilkan banyak tweet yang tidak relevan di hasil pencariannya yang terlihat dari nilai *F1-Measure* dan *accuracy.*

3.4.3 Pengujian Akurasi dari Metode Klasifikasi yang Diusulkan

Pengujian ini dilakukan untuk menguji apakah metode klasifikasi yang diusulkan dapat digunakan untuk melakukan klasifikasi data berdasarkan ontologi. Pengujian dilakukan untuk melihat *accuracy* dari metode TRODEW. Skenario yang digunakan adalah sebagai berikut:

1. Melakukan klasifikasi dari hasil komputasi Ehrig pada data tweet yang tidak dilakukan *preprocessing* menggunakan ontologi HappinessV1.
2. Melakukan klasifikasi dari hasil komputasi Ehrig pada data tweet yang dilakukan *preprocessing* menggunakan ontologi HappinessV1.
3. Melakukan klasifikasi dari hasil komputasi Ehrig pada data tweet yang tidak dilakukan *preprocessing* menggunakan ontologi HappinessV2.
4. Melakukan klasifikasi dari hasil komputasi Ehrig pada data tweet yang dilakukan *preprocessing* menggunakan ontologi HappinessV2.

Hasil pengujian dari skenario ini ditunjukkan pada tabel 3-3.

Tabel 3-3 Akurasi Klasifikasi TRODEW

Skenario	Accuracy
1	43.16406 %
2	67.1875 %
3	48.24219 %
4	82.03125 %

Dari hasil pengujian yang ada dicek kembali hasil dari klasifikasi yang dihasilkan. Hasil kesalahan klasifikasi ditunjukkan di tabel 3-4.

Tabel 3-4 Kesalahan Klasifikasi Skenario 4

Kesalahan	Kemunculan
Kesalahan pembobotan	10
Ambiguitas kata	24
Kalimat eksplisit	2
Multi-class result	9
Kata tidak ada dalam ontologi	9
Kata merupakan properties	3
Belum ada sinonim di kamus	2

Dari tabel 3-4 terlihat bahwa kesalahan terbanyak dihasilkan oleh ambiguitas kata. Walaupun dalam pembuatan ontologi HappinessV2, ambiguitas kata juga menjadi salah satu pertimbangan, namun karena pada pembuatannya hanya memperhatikan aspek relevansi dengan ontologi, tidak melihat secara detail kelas tertentunya maka masih banyak ditemukan kesalahan akibat ambiguitas kata. Selain itu, kesalahan karena kesalahan pembobotan dan *multi-class result* juga masih menjadi penyebab kurangnya nilai *accuracy*. Untuk kedua masalah ini, murni karena cara kerja dari metode klasifikasi TRODEW sendiri, yang menggunakan bobot dari kedalaman suatu *term.*

Dengan hasil yang ditunjukkan di tabel 3-3, performa dari klasifikasi TRODEW cukup baik dengan *accuracy* mencapai 80%. Namun dari hasil tabel 3-4, masih terlihat bahwa metode TRODEW yang diusulkan dapat dikembangkan lebih jauh lagi untuk meningkatkan performanya dalam melakukan klasifikasi khususnya pemilihan bobot.

4 Kesimpulan

Berdasarkan pengujian dan analisi yang dilakukan dalam Tugas Akhir ini, dapat disimpulkan bahwa:

1. Salah satu penerapan ontologi dalam *search engine* dapat dilakukan dengan menambahkan komputasi Ehrig setelah pencarian dengan *search engine* biasa.
2. Penggunaan komputasi Ehrig dapat memberi nilai mengenai relevansi sebuah

tweet ke ontologi yang ditunjukkan dari nilai *F1-Measure* tertinggi sebesar 86.95%.

3. Pengunaan komputasi Ehrig membutuhkan proses *preprocessing* serta ontologi serta kamus yang lengkap untuk meningkatkan performanya.
4. Penerapan ontologi dalam *search engine* dapat meningkatkan relevansi hasil pencarian yang ditunjukkan dengan nilai yang tinggi khususnya pada pencarian tanpa *keyword* dan pencarian menggunakan *keyword* yang tidak berhubungan dengan ontologi yang diinginkan yang ditunjukkan dengan nilai *F1-Measure* sebesar 86.4% dan 100%.
5. *Search engine* yang tidak menggunakan ontologi dapat menghasilkan hasil pencarian dengan data yang relevan dengan ontologi lebih banyak namun juga menghasilkan banyak data yang tidak relevan dengan ontologi.
6. Metode klasifikasi TRODEW merupakan metode klasifikasi yang menerapkan ontologi.
7. Metode klasifikasi TRODEW yang diusulkan dapat digunakan untuk melakukan klasifikasi yang ditunjukkan dengan *accuracy* yang mencapai 82%.
8. Metode klasifikasi TRODEW membutuhkan proses *preprocessing* serta ontologi yang lengkap untuk meningkatkan performa.

5 Saran

Berdasarkan sistem yang sudah dikerjakan pada penelitian ini, dapat dilakukan pengembangan lebih lanjut dengan beberapa cara yaitu:

1. Melakukan *preprocessing* yang lebih baik dan lebih lengkap dalam menangani data yang ada.
2. Melakukan konstruksi ontologi yang lebih lengkap dan kaya, baik dari segi *class*, *instance*, *properties* dan relasinya.
3. Menambahkan proses *machine learning* sistem.
4. Meningkatkan performa metode klasifikasi TRODEW dengan melakukan pengujian serta modifikasi algoritma yang digunakan.

6 Daftar Pustaka

[1] R. Subhashini and J. Akilandeswari, "A survey on ontology construction methodologies," *Int. J. Enterp. Comput. Bus. Syst.*, vol. 1, no. 1, pp. 60–72, 2011.

[2] K. S. Zaiss, "Instance-based ontology matching and the evaluation of matching systems," Düsseldorf, Univ., Diss., 2010, 2010.

[3] W3C, "OWL - Semantic Web Standards." [Online]. Available: https://www.w3.org/2001/sw/wiki/OWL. [Accessed: 30-May-2016].

[4] W3C, "OWL Web Ontology Language Overview." [Online]. Available: https://www.w3.org/TR/owl-features/. [Accessed: 30-May-2016].

[5] N. F. Noy, D. L. McGuinness, and others, *Ontology development 101: A guide to creating your first ontology*. Stanford knowledge systems laboratory technical report KSL-01-05 and Stanford medical informatics technical report SMI-2001-0880, Stanford, CA, 2001.

[6] M. Ehrig and A. Maedche, "Ontology-focused Crawling of Web Documents," in *Proceedings of the 2003 ACM Symposium on Applied Computing*, New York, NY, USA, 2003, pp. 1174–1178.

[7] R. V. Imbar, A. Adelia, M. Ayub, and A. Rehatta, "Implementasi Cosine Similarity dan Algoritma Smith-Waterman untuk Mendeteksi Kemiripan Teks," *J. Inform.*, vol. 10, no. 1, 2015.

[8] Z. Ceska and C. Fox, "The Influence of Text Pre-processing on Plagiarism Detection," in *International Conference on Electronic Publishing*, Leuven, 2009.

[9] "Data Mining Concepts." [Online]. Available: https://docs.oracle.com/cd/B28359_01/datamine.111/b28129/classify.htm. [Accessed: 31-May-2016].

[10] M. Janik and K. Kochut, "Training-less Ontology-based Text Categorization," in *Exploiting Semantic Annotations in Information Retrieval*, 2008.

[11] U. Noor, Z. Rashid, and A. Rauf, "TODWEB: Training-Less ontology based deep web source classification," in *Proceedings of the 13th International Conference on Information Integration and Web-based Applications and Services*, 2011, pp. 190–197.

[12] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press, 2008.