

PENGLASIFIKASIAN LAMAN WEB BERDASARKAN GENRE MENGGUNAKAN URL FEATURE

Hendri Noviyanto¹, Teguh Bharata Adji², Indriana Hidayah³

Jurusan Teknik Elektro dan Teknologi Informasi

Universitas Gadjah Mada

Jl. Grafika No.2 Yogyakarta - 55281

E-mail: nhendri0@gmail.com, adji@mti.ugm.ac.id, indriana.h@ugm.ac.id

ABSTRACT

The Internet as a source of all information required to be able to present the relevant information as the user desires. In a search using search engine sometimes generate a lot of information that is too abroad and the topic is not always in accordance with the genre. The genre of the web is a group of web pages or posts that contain the contents or meaning almost the same. Classification of web page is very important in order to generate a more specific search. Genre classification is done by using the url of a web page where all the tags have been broken down into text that only contains a collection of words and numbers. The text will be grouped based on the words which are represent each field or have the same meaning, for example system information include in the category of informatics, and so on. The classification of URLs can be done with the SVM-KNN method. SVM-KNN work by providing feedback that will do the pruning KNN and SVM will fix his mistake. Weighting is done using TF-IDF then similarity be calculated using the longest resemblance common subsequence (LCS) and evaluated by looking at the value of precission and best recall. The classification based on genre is expected to produce the desired information more quick and accurate.

Keyword : Genre, Classification, URL, Web Page, Search Engine

ABSTRAKS

Internet sebagai sumber dari segala jenis informasi dituntut untuk dapat menyajikan informasi yang relevan sesuai keinginan pengguna. Pencarian menggunakan mesin pencari banyak menghasilkan informasi yang terlalu luas dan topiknya tidak selalu sesuai dengan genre yang dicari. Genre dari sisi web adalah sebuah kelompok dari laman web atau postingan yang mengandung isi atau makna yang hampir sama. Klasifikasi laman web sangat diperlukan agar dapat menghasilkan pencarian yang lebih spesifik. Klasifikasi genre dilakukan dengan memanfaatkan URL sebuah laman web dimana semua tag sudah dipecah sehingga hanya berisi teks berupa kumpulan kata dan angka. Teks tersebut akan dikelompokkan berdasarkan kata-kata yang mewakili setiap bidang atau memiliki makna yang sama, misalnya Sistem Informasi masuk ke dalam kategori informatika, dan sebagainya. Pengklasifikasian URL bisa dilakukan dengan metode SVM-KNN. SVM-KNN bekerja dengan cara memberikan umpan balik yaitu KNN akan melakukan pruning dan SVM akan memperbaiki kesalahannya. Cara pembobotan dilakukan menggunakan TF-IDF kemudian kemiripan akan dihitung menggunakan longest common subsequence (LCS) dan di evaluasi dengan melihat nilai precission dan recall terbaik. Dengan adanya pengklasifikasian berdasarkan genre diharapkan dapat menghasilkan informasi yang diinginkan secara lebih cepat dan akurat.

Kata Kunci: genre, klasifikasi, url, laman web, mesin pencari

1. PENDAHULUAN

1.1 Latar Belakang

Perkembangan Internet yang semakin pesat membawa tingkat popularitas *World Wide Web* (WWW) menjadi lebih terkenal. WWW menyajikan beragam konten yang direpresentasikan melalui sebuah laman web yang menyediakan berbagai bentuk informasi, seperti penyedia layanan berita (*news site*), toko online (*online shop*), dan beragam penyedia layanan-layanan yang lain.

Melimpahnya sumber informasi di Internet dapat dilihat dari meningkatnya persentase penggunaan Internet dengan melihat data dari ("Total Number of Websites", 2014). Proses yang sering dilakukan seperti transaksi jual beli, memperoleh informasi

terhadap suatu produk *online* maupun pengetahuan yang lain. Jumlah informasi yang sangat besar tidak selalu berdampak positif, salah satunya yaitu pengguna kesulitan menemukan laman web dengan informasi yang relevan secara cepat sesuai kebutuhan menggunakan *search engine*.

Seperti yang kita ketahui, sebuah laman web atau informasi yang berada di Internet bersifat tidak terstruktur (Krutil, 2012), artinya informasi yang tersedia sangat banyak dan beragam jenisnya, serta sumber informasi berasal dari berbagai tempat yang berbeda. Untuk mengatasi tingkat kesulitan pencarian informasi yang relevan, dibutuhkan sebuah mekanisme proses pengklasifikasian informasi dari laman web dengan tujuan agar

informasi lebih terstruktur. Klasifikasi termasuk dalam cabang ilmu *data mining* yang dikenal dengan teknik-teknik pengolahan data.

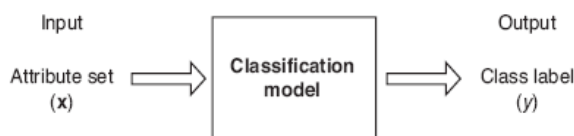
Data Mining (DM) berkembang dengan cepat dalam beberapa area penelitian dan disiplin ilmu seperti *parallel computing*, *databases*, *statistic*, *visualization* (Mikanshu dkk, 2013). DM adalah sebuah proses untuk menggali pengetahuan (*Knowledge Discovery*) dari sekumpulan data yang memiliki volume sangat besar. Proses yang dilakukan memiliki kemungkinan untuk menggali sebuah data yang tidak terstruktur atau belum diketahui menjadi sebuah data yang berguna dan bisa dimanfaatkan untuk melakukan manajemen atau mengelola sumber daya menjadi lebih baik. *Knowledge Discovery in Databases* (KDD) adalah satu kesatuan dengan DM yang memiliki beberapa tahapan sebelum bisa diproses dalam *Machine Learning* (ML) seperti pembersihan data (*data cleaning*), integrasi data (*data integration*), pemilihan data (*data selection*), transformasi data (*data transformation*), evaluasi pola (*pattern evaluation*), menyajikan pengetahuan (*knowledge presentation*) (Ayub, 2007).

DM memiliki beberapa aplikasi yang dapat digunakan untuk melakukan pemrosesan tugas pengolahan data, salah satunya adalah WEKA. Aplikasi WEKA dapat digunakan untuk menyelesaikan berbagai tugas yang berbeda, seperti *association* (membentuk sebuah pola dimana terjadi hubungan antara satu data dengan data yang lain), *classification* (mengidentifikasi pola baru dengan target data yang sudah dikenal), dan *clustering* (mengelompokkan identitas atau kesamaan sebuah objek) (Rani, 2013).

WEKA adalah sebuah aplikasi DM yang bersifat *open sources software*, dikembangkan oleh Universitas Waikato di New Zealand menggunakan bahasa java (WEKA, 2014) dan termasuk salah satu aplikasi yang memiliki kumpulan algoritme di dalam direktorinya, sehingga memiliki kemampuan untuk menangani beberapa tugas seperti *regression*, *classification*, *clustering*, *association rule mining* dan *attribute selection*. WEKA menggunakan format ARFF sebagai *source file* untuk melakukan pemrosesan data, maka mengubah format file menjadi ARFF bersifat wajib agar dapat diolah oleh WEKA. ARFF adalah sebuah format file yang digunakan untuk mengindikasikan perbedaan *attribute names*, *attribute type*, dan *attribute value* (Rani, 2013). Selain penyajian dengan angka WEKA juga mampu memberikan hasil dalam bentuk visual seperti tabel dan kurva.

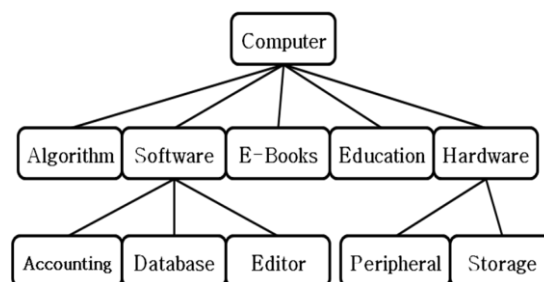
Klasifikasi adalah sebuah teknik yang cukup baik untuk mengolah data yang bervariasi. Pada penelitian (Baykan, 2009; Zhaohui, 2011; Rajalakshmi, 2013; Chaker, 2014) klasifikasi digunakan untuk menentukan nilai *precision* dan *recall* dari sebuah dataset. Klasifikasi termasuk dalam *supervised learning* (Rani, 2013) yang

melakukan pembelajaran secara terbimbing karena memiliki data *training* untuk menentukan *class*-nya. Langkah melakukan klasifikasi, yaitu: 1) Proses pengumpulan data yang masih bersifat mentah (*raw data*); 2) Melakukan proses *preprocessing* untuk membersihkan *noise* dalam data. Data yang dihasilkan proses *preprocessing* diolah dalam *machine learning* seperti proses *filtrasi*, *agregasi*, *classification* atau proses lain sesuai kebutuhan. Klasifikasi dapat digunakan untuk mengatasi masalah pengelompokan suatu objek sesuai dengan definisi dan kesamaan menjadi sekumpulan *genre* (Chaker, 2014).



Gambar 1. *Classification* input output (Rani, 2013)

Pengertian *genre* dalam buku *Genre Analysis* (Swales, 1990) adalah sebuah pengelompokan suatu objek yang memiliki kesamaan. Klasifikasi laman web berdasarkan *genre* bertujuan untuk memudahkan dalam pencarian informasi yang relevan dikarenakan pertumbuhan informasi yang berada di Internet terus meningkat dengan cepat. Pengklasifikasian terhadap *genre* dipilih karena pada saat ini *search engine*, misalnya Google masih menggunakan kombinasi metode *Keyword* dan *PageRank* ("Mesin Pencari Web" 2015). Pendekatan dengan *keyword* memiliki cara kerja yaitu dengan memecahkan kata masukan dalam pencarian dokumen atau informasi. Misalnya contoh pencarian "*Machine Learning*" maka *search engine* akan melakukan pemecahan kata menjadi "*Machine*" + "*Learning*" kemudian melakukan pencarian pada *directory* yang dimilikinya. *PageRank* bertujuan untuk membantu proses pencarian dokumen atau informasi dengan menampilkan laman web yang sering dikunjungi tanpa melihat isi dari *content* sebuah laman web yang ditampilkan.



Gambar 2. Skema pengklasifikasian laman web dari *Open Directory Project* (ODP) ("DMOZ" 2014)

1.2 Literatur Review

Penelitian dengan topik klasifikasi laman web sudah banyak dilakukan dengan berbagai macam

feature, salah satunya URL. Beberapa peneliti telah melakukan penelitian terhadap klasifikasi dengan *feature* tersebut, berbagai macam metode telah diterapkan untuk mendapatkan hasil *precision* dan *recall* yang terbaik.

Pada tahun 2007, M. Indra Devi dkk melakukan penelitian terkait dengan klasifikasi URL dengan judul “*Machine Learning Techniques for Automated Web Pages Classification using URL Feature*”. Penelitian tersebut hanya menggunakan *feature* dari URL untuk melakukan pengklasifikasian dengan alasan bahwa URL bersifat unik, memiliki arti, serta dapat digunakan untuk identifikasi. Dengan membandingkan 3 buah algoritme yaitu Naïve Bayes, SVM dan RBF Network, diperoleh hasil bahwa SVM lebih unggul dari kedua algoritme yang lain. Kekurangan dari RBF adalah tidak mampu mengeksekusi *instances* yang bersifat negative, namun dalam *instances* yang bersifat positive RBF lebih unggul dibanding dengan SVM dan Naïve Bayes (Devi, 2007).

Pada tahun 2009, Penelitian Eda Baykan dkk dengan judul “*Purely URL-based Topic Classification*”, subject penelitian hanya menggunakan URL sebagai *feature* utama tanpa ada *feature* pendukung yang lainnya. Pertimbangan Eda Baykan adalah URL mengandung informasi yang sudah cukup mewakili dan menggambarkan isi dari sebuah laman web, selain itu kecepatan dan *resource storage* sangat di pertimbangkan. Penelitian ini melibatkan metode *n*-gram sebagai pemecah kata, sedangkan *classifier* menggunakan metode SVM, NaiveBayes, dan ME, dengan cara melakukan perbandingan diantara ketiga metode tersebut. Namun hasil penelitian yang dilakukan masih memiliki kelemahan yaitu tidak semua *single* URL bisa di klasifikasikan (Baykan, 2009).

Pada tahun 2011, Zhaohui Xu dkk dalam penelitiannya yang berjudul “*A Web Page Classification Algorithm Based On Link Information*”, menjelaskan bahwa *traditional classification* biasanya melakukan pengeksekusian terhadap *content* web namun metode tersebut memiliki beberapa kelemahan, antara lain besarnya tingkat kesalahan informasi yang dihasilkan, ukuran teks yang terlalu besar sehingga sering mengalami *error*, dan tidak bisa digunakan untuk mengklasifikasikan video, musik, dan gambar. Karena dalam prosesnya membutuhkan *preprocessing* terhadap semua text yang termuat dalam isi laman web, maka dikembangkan klasifikasi dengan *Link Information Categorization* (LIC) yang dikembangkan dari KNN. KNN termasuk *lazy learning algorithm* yang membutuhkan *storage* dan *computing cost* yang cukup besar. Dengan memperbaiki metode KNN agar lebih cepat dan tingkat akurasi yang tinggi maka LIC hadir untuk mengatasi masalah tersebut. Penelitian yang dikerjakan memperoleh hasil dengan membandingkan 3 algoritme yaitu LIC, KNN, dan

SVM. Sehingga diperoleh bahwa algoritme LIC mampu menunjukkan kemampuannya dengan melebihi kemampuan KNN dan SVM (Zhaohui, 2011) dengan menunjukan hasil *precision* dan *recall* yang lebih baik.

Tahun 2013, R. Rajalakshmi dkk dalam penelitian berjudul “*Web Page Classification using n-gram based URL Features*”, menitikberatkan penggunaan *feature* URL untuk klasifikasi dan didukung dengan penggunaan metode *n*-gram sebagai pemecah kata, seperti yang dilakukan (Devi, 2007) pada penelitian sebelumnya. Dalam penelitian ini dibandingkan metode SVM dan ME untuk mengetahui metode mana yang lebih baik untuk klasifikasi. Penggunaan dataset yaitu dengan WebKB dan *Open Directory Project* (ODP) (“DMOZ” 2014) yang berasal dari *directory* yang dimiliki oleh mozilla. Hasil yang diperoleh menunjukan bahwa ME lebih unggul saat mengeksekusi data yang lebih kecil, namun kemampuan kedua algoritme tersebut seimbang saat melakukan eksekusi pada data yang relatif berjumlah besar (Rajalakshmi, 2013).

Tahun 2014, Chaker Jebari dalam penelitiannya tentang klasifikasi laman web ke dalam *genre* berdasarkan URL *feature*, penelitian tersebut menggunakan pendekatan dengan memberikan pembobotan pada URL, Seperti *Domain name* (DOMN), *Document path* (DOCP), dan *Document name and query string* (DOCN). Pendekatan yang lain adalah dengan menggunakan metode *n*-gram untuk membantu proses pemecahan kata. Chaker membandingkan metode pada penelitiannya terdahulu tentang klasifikasi *genre* (C. Jebari and Wani 2012) yaitu RakEL, BR-SVM, MLKNN, dan BPMLL. Hasil yang diperoleh adalah metode RakEL lebih baik dalam pengklasifikasian laman web menggunakan URL (Chaker, 2014).

Dari penjelasan diatas dapat disimpulkan bahwa penggunaan *feature* URL sudah bisa mewakili isi dari sebuah laman web, sehingga sudah dapat digunakan untuk melakukan proses klasifikasi. Pengklasifikasian berdasarkan *genre* akan memudahkan pencarian spesifik sesuai keinginan pengguna. Beberapa metode umum yang digunakan meliputi Naive Bayes, SVM, ME, RakEL, BR-SVM, MLKNN, dan BPMLL. Setiap metode memiliki kelemahan dan kelebihan masing-masing dalam melakukan klasifikasi, sehingga perlu adanya eksplorasi terhadap metode lain untuk tujuan klasifikasi yang sama yaitu menggunakan SVM-KNN (Yun Lin, 2014).

2. PEMBAHASAN

2.1 URL Feature

URL adalah singkatan dari *Uniform Resource Locator*, URL adalah serangkaian karakter yang sesuai dengan format berstandar yang digunakan untuk menunjukan alamat suatu sumber atau *resource* seperti dokumen, gambar dan aplikasi di

Internet (“URL” 2014). URL memiliki fungsi antara lain sebagai berikut (“Situs Web” 2014):

- Pengidentifikasi sebuah dokumen di situs web.
- Memudahkan dalam pengaksesan suatu dokumen melalui situs web.
- Memberikan penamaan terhadap suatu berkas atau dokumen pada situs web.
- Memudahkan kita dalam mengingat sebuah alamat situs web.

2.1.1 N-gram

URL adalah sebuah kumpulan kata yang memudahkan kita dalam mengingat sebuah alamat situs atau laman web, namun dalam proses pengklasifikasian tidak mungkin memproses data mentah, tingkat kesulitan akan semakin membesar atau bahkan tidak dapat dilakukan. Maka diperlukan pemrosesan terlebih dahulu supaya menjadi token-token yang dimengerti oleh algoritme dan mempermudah dalam pemrosesan. Contoh sebuah url <https://www.machinelearning.com/tag/ngram>, *machine learning* belum mengetahui maksud dari serangkaian kata tersebut bahkan sulit untuk memprosesnya. Oleh sebab itu kualitas data masukan pada *machine learning* berperan penting terhadap kesuksesan pengklasifikasian. *N-gram* adalah sebuah metode yang digunakan untuk melakukan pemotongan *n* karakter dalam suatu *string* tertentu atau potongan kata dalam suatu kalimat tertentu. Semisal “*machine learning*”, jika dilakukan pemrosesan dengan *n-gram* maka akan didapatkan hasil sebagai berikut.

Tabel 1. Pemrosesan *n-gram*

<i>N-gram by each adjacent character</i>	
<i>Bi-gram</i>	“ma”, “ac”, “ch”, “hi”, “in”, “ne”, “el”, “le”, “ea”, “ar”, “rn”, “ni”, “in”, “ng”
<i>Tri-gram</i>	“mac”, “ach”, “chi”, “hin”, “ine”, “nel”, “ele”, “lea”, “ear”, “arn”, “rni”, “nin”, “ing”
<i>4-gram</i>	“mach”, “achi”, “chin”, “hine”, “inel”, “nele”, “elea”, “lear”, “earn”, “arni”, “rnin”, “ning” Dst.....

Untuk pemrosesan terhadap dokumen menggunakan metode *n-gram* dapat diformulasikan sebagai berikut (Zhaohui, 2011):

$$tNgram = tWord - depth + 1 \quad (1)$$

dimana:

tNgram adalah total dari jumlah *n-gram*

tWord adalah total dari pemberian kata

depth adalah type dari *n-gram*

Contoh pada kata “SAYASAJA”, terdiri dari (*tWord*) sebanyak 8 buah. Dengan menggunakan *bigram* (*depth* = 2) dapat membentuk *n-gram* (*tNgram*) sebanyak: $8 - 2 + 1 = 7$ kemungkinan yaitu {SA, AY, YA, AS, SA, AJ, JA}.

Dengan kata lain *n-gram* melakukan proses sesuai masukan pada *depth* dimana nilai masukan akan mempengaruhi pemrosesan terhadapnya, seperti *bigram*, *trigram*, *fourgram*, dan seterusnya diperoleh dari seberapa besar nilai *gram* masukannya.

2.1.2 Teknik Pembobotan

Pengukuran tingkat kemiripan sangat penting dalam mekanisme pengolahan dokumen berbasis teks. Dalam pengolahan dokumen langkah yang sering digunakan adalah menghitung kesamaan *query* masukan dengan dokumen lain. Teks atau dokumen akan di representasikan sebagai vektor untuk mempermudah dalam perhitungan.

a. Term Frequency-Inverse Document Frequency (TF-IDF)

TF adalah metode dasar untuk menghitung frekuensi kemunculan kata atau istilah dalam sebuah dokumen (Riboni, 2002). Frekuensi kemunculan tersebut dijadikan sebuah bobot dokumen yang akan direpresentasikan dalam bentuk lingkungan vektor sehingga terbentuk vektor berdimensi *n* yang mana nilainya dapat digunakan untuk proses selanjutnya. TF dihitung dengan persamaan (2). Sedangkan IDF adalah algoritme dari rasio jumlah seluruh dokumen yang dimiliki oleh korpus dengan dokumen term yang ditulis secara matematis pada persamaan (3). Kemudian nilai total akan didapatkan dengan melakukan perkalian antara TF dan IDF dengan formula (4) (Saadah, 2013). Formula fungsi dapat dilihat sebagai berikut.

$$t = \frac{freq_i(d)}{\sum_{i=1}^k freq_i} \quad (2)$$

$$idf_i = \log \frac{|D|}{|\{d : t_i \in d\}|} \quad (3)$$

$$(tf - idf)_{ij} = tf_i(d)_j * idf_i \quad (4)$$

Dimana (*t_i*) menunjukkan jumlah frekuensi kemunculan istilah atau kata dalam sebuah dokumen (*d_j*).

b. Longest Common Subsequence (LCS)

LCS adalah sebuah pendekatan yang digunakan untuk menghitung relasi berurutan yang paling panjang antara *query* masukan dengan dokumen. LCS digunakan sebagai pendukung fitur pembobotan tf-idf sebelumnya. Dokumen yang memiliki kesamaan urutan kata dengan *query* masukan akan memiliki bobot yang tinggi. Nilai *query* q dengan dokumen j yang telah didapatkan kemudian dinormalisasi dengan persamaan (5) seperti pada (Saadah, 2013). Yaitu m adalah jumlah *term* dalam *query* dan n adalah jumlah *term* dalam dokumen.

$$LSC_{(q,j)normalisasi} = \frac{LSC_{q,j}}{m+n} \quad (5)$$

2.1.3 Preprocessing

Preprocessing adalah proses mengubah data mentah menjadi format yang sesuai untuk tahap analisis berikutnya. Selain itu *preprocessing* juga digunakan untuk membantu dalam pengenalan atribut dan data segmen yang relevan dengan tugas *data mining*.

Seperti yang telah dijelaskan sebelumnya bahwa setiap masukan data harus memiliki kualitas dan penyampaian yang jelas, agar *machine learning* dapat bekerja secara maksimal. Dalam proses *Preprocessing* dan *Extraction URL* sebelum dilakukan training dan testing maka beberapa langkah yang dilakukan antara lain:

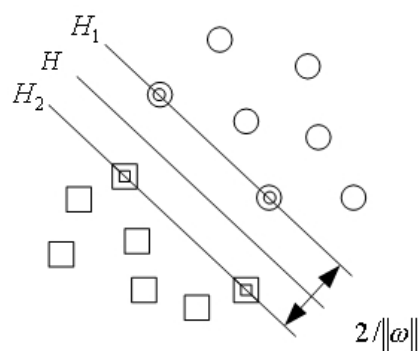
- Stoplist, yaitu menghilangkan karakter yang tidak berguna atau *noise* seperti “http”, “www”, “.”, “/”.
- Pemrosesan menggunakan *n-gram*, yaitu melakukan pemecahan kata seperti yang ditunjukkan di tabel 1.
- Pembobotan TF-IDF, yaitu memberikan bobot kepada setiap kata hasil pemrosesan *n-gram*. Dengan melakukan pembobotan maka akan diketahui tingkat *similarity* dari suatu kata dan kemudian diijadikan sebagai acuan untuk pengklasifikasian laman web ke dalam *genre* masing-masing.

Setelah melakukan tahap *preprocessing* maka dapat dilakukan klasifikasi URL berdasarkan genre dengan *machine learning classifier* untuk mendapatkan hasil akhir. Data yang diperoleh kemudian akan dilakukan analisis untuk mengetahui hasil yang paling baik.

2.2 Metode Klasifikasi

a. SVM-KNN

Support Vector Machine (SVM) diusulkan oleh Vapnik (Vapnik V, 1995), SVM dikenal dengan sistem pembelajaran *machine learning* (ML) yang cukup baik. Pada dasarnya SVM dikembangkan agar dapat menyelesaikan masalah klasifikasi linier kemudian dikembangkan agar mampu bekerja pada masalah non-linier dengan konsep kernel trick pada ruang berdimensi tinggi. SVM secara sederhana berusaha mencari *hyperplane* terbaik yang berfungsi sebagai pemisah dua kelas pada *input space*, seperti pada gambar 3. Pola positif ditunjukkan dengan tanda kotak dan negatif dengan tanda bulat.

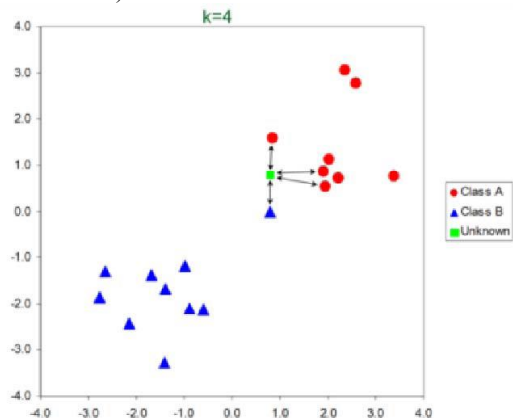


Gambar 3. Sketsa pemetaan *hyperplane* (Weimin, 2006)

Hyperplane pemisah terbaik antara dua kelas dapat digunakan untuk mengukur batas tepi *hyperplane* tersebut. SVM yang mampu bekerja pada ruang berdimensi tinggi memiliki beberapa kelebihan, salah satunya proses generalisasi. Generalisasi dikategorikan sebagai kemampuan metode SVM untuk mengklasifikasikan suatu pola yang tidak termasuk dalam data yang dipakai dalam fase pembelajaran. Dalam fase metode tersebut Vapnik (Vapnik V, 1995) menjelaskan generalisasi error disebabkan oleh dua faktor yaitu error terhadap training set dan dipengaruhi oleh Vapnik-Chervokinesis (VC). Strategi yang digunakan SVM untuk mengatasi kedua masalah tersebut adalah dengan cara *Empirical Risk Minimization* (ERM) dengan meminimalkan error pada training set dan *Structural Risk Minimization* (SRM) pada VC untuk memilih *hyperplane* dengan margin terbesar (Chaker, 2014), (Yun Lin, 2014).

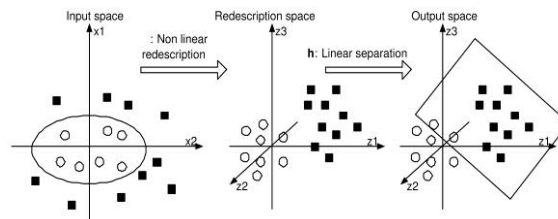
K-Nearest Neighbour (KNN) adalah sebuah metode untuk melakukan klasifikasi terhadap object berdasarkan data pembelajaran yang paling dekat (Chaker, 2014). Data pembelajaran diproyeksikan ke ruang berdimensi banyak, dimana masing-masing merepresentasikan fitur dari data dan ruangan dibagi berdasarkan klasifikasi data pembelajaran. Pada fase pembelajaran, algoritme ini hanya melakukan penyimpanan vektor-vektor fitur dan klasifikasi dari

data pembelajaran. Perhitungan jarak dari vektor biasanya dihitung berdasarkan jarak *Euclidean*. Untuk mengetahui nilai terbaik dengan metode ini adalah dengan mengambil nilai k terbaik pada data. Nilai k bisa dipilih secara acak, misalnya dengan menggunakan *Cross Validation*. Kasus khusus di mana klasifikasi pembelajaran paling dekat adalah $k=1$. Namun nilai k yang tergantung kepada data membuat k mengurangi *noise*, tetapi membuat batasan antara setiap klasifikasi menjadi kabur (“KNN” 2015).



Gambar 4. KNN Sketch Map ($k=4$) (Yun Lin, 2014)

SVM-KNN merupakan penggabungan dua metode yang berbeda, tujuan dari kombinasi ini adalah untuk bisa mendapatkan nilai *precision* dan *recall* yang baik. Studi klasifikasi menggunakan SVM ditemukan sebuah kesalahan sampel dekat dengan margin, hal ini menunjukkan bahwa informasi tersebut dapat digunakan untuk meningkatkan kinerja klasifikasi dengan menutup kelemahan tersebut. Dengan mengkombinasikan SVM dan KNN, sampel yang didistribusikan ke dalam ruang dapat digunakan untuk mencari klasifikasi dengan menggunakan *Nearest Neighbors*. Kombinasikan KNN di gunakan dengan menghitung fungsi jarak yang sederhana dan untuk menghasilkan keputusan yang lebih baik. Pemangkasan dilakukan menggunakan KNN, dan perbaikan menggunakan SVM. Fitur KNN bekerja dengan syarat bahwa semua sampel poin adalah perwakilan sebuah titik. Dengan demikian KNN akan melakukan pelatihan terhadap semua sampel titik. Oleh karena itu harus dilakukan perhitungan semua jarak uji sampel x untuk semua pelatihan sampel, dengan menggabungkan dua metode SVM dan KNN (Yun Lin, 2014).



Gambar 5. Ide penggunaan algoritme SVM (Alani, 2010)

b. RBF

RBF network adalah metode dari pemodelan matematik, dasar fungsi RBF adalah jaringan syaraf tiruan yang menggunakan dasar radial sebagai aktivasi. Output adalah kombinasi linier radial fungsi input dan neuron parameter. RBF biasanya memiliki 3 lapisan, 1) lapisan masukan, 2) lapisan tersembunyi dengan aktivasi RBF non-linier, 3) lapisan output linier. Masukan dapat dimodelkan sebagai vektor bilangan real, output dari jaringan ini fungsi saklar vektor masukan (Devi, 2007), (“Radial Basis Function Network” 2015).

2.3 Performa Measure

Pengukuran performa dari pengklasifikasian adalah dengan cara mengetahui *Error rate*, *Recall*, dan *Precision*. Pertama pendefinisian *Error rate* seperti dibawah ini.(Weimin Xue et al. 2006)

$$ErrorRate = \frac{JumlahPrediksiSalah}{JumlahTotalPrediksi} \quad (6)$$

Mendefinisikan *Recall*, *Precision*, *Acuracy*, dan *F-measure*.

$$Recall = \frac{tp}{tn + fp} \quad (7)$$

$$Precision = \frac{tp}{tp + tn} \quad (8)$$

$$Accuracy = \frac{tp + tn}{tp + tn + fp + fn} \quad (9)$$

$$F = 2 \times \frac{precision * recall}{precision + recall} \quad (10)$$

Dimana:

tp adalah True Positif

tn adalah true negatif

fp adalah false positif

fn adalah false negatif

F adalah F-measure

2.4 Dataset

Klasifikasi laman web termasuk dalam kategori *supervised learning* dengan kebutuhan data yang

relative simple, pada penelitian ini penulis menggunakan dataset dari ODP("DMOZ" 2014) dengan laman web versi Bahasa Indonesia. Dalam directory ODP terdapat berbagai variasi laman web yang dapat diklasifikasikan menurut *genre*, namun dalam penelitian ini kami melakukan proses pengklasifikasian hanya terhadap 12 *genre* laman web, seperti penyedia layanan berita (*news site*), *Business*, *Computer*, *Games*, *Health*, *Home*, *Arts*, *Shopping*, *Sciences*, *Sport*, *References*, *Society*. Dalam penelitian ini akan digunakan 2400 data URL, dengan masing-masing *genre* sebanyak 200 URL.

3. KESIMPULAN

Dari paper yang telah dikaji diatas dapat ditarik sebuah gambaran tentang pengklasifikasian terhadap laman web ke dalam *genre* masing-masing menggunakan *feature* URL. Beberapa metode seperti Naive Bayes, SVM, ME, KNN, NN, RakEL, BR-SVM, MLKNN, dan BPMLL yang telah diterapkan untuk meningkatkan *Precision* dan *Recall* memiliki hasil yang berbeda, semua kondisi ini dikarenakan terdapat perbedaan terhadap penggunaan dataset, penggunaan *feature*, serta kendala saat penelitian berlangsung.

Dalam penelitian ini diusulkan penggunaan SVM dan KNN yang dikombinasikan sebagai metode pengklasifikasian URL, diimplementasikan untuk klasifikasi laman web berdasarkan *genre*. *Feature n-gram* akan digunakan sebagai pemecah kata dengan mencari *n* terbaik. Pembobotan teks menggunakan TF-IDF, hasilnya akan dilakukan lagi pembobotan dengan LCS untuk meneliti tingkat *similarity* pada *query* dan dokumen supaya hasil pencarian lebih akurat.

PUSTAKA

- Al-ani T, and Dalila T. 2010. *Signal Processing and Classification Approaches for Brain-Computer Interface, Intelligent and Biosensors*, Vernon S. Somers (Ed.). InTech.
<http://www.intechopen.com/books/intelligent-and-biosensors/signal-processing-and-classification-approaches-for-brain-computer-interface>.
- Ayub, M. 2007. "Proses Data Mining Dalam Sistem Pembelajaran Berbantuan Komputer." *Jurnal Sistem Informasi* Vol. 2 No. 1 (March): 21–30.
- Baykan, E, Monika H, Ludmila M, and Ingmar W. 2009. "Purely URL-based Topic Classification." In *Proceedings of the 18th International Conference on World Wide Web*, 1109–10. Madrid, Spain: ACM.
- Devi, M. Indra, R. Rajaram, and K. Selvakuberan. 2007. "Machine Learning Techniques for Automated Web Page Classification Using URL Features." In *Proceedings of the International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007) - Volume 02*, 116–20. IEEE Computer Society.
- "DMOZ." 2014. Accessed December 17. <http://www.dmoz.com>.
- Jebari, C., and M.A. Wani. 2012. "A Multi-label and Adaptive Genre Classification of Web Pages." In , 1:578–81. doi:10.1109/ICMLA.2012.106.
- Jebari, C. 2014. "A Pure URL-Based Genre Classification of Web Pages." In , 233–37. doi:10.1109/DEXA.2014.56.
- "KNN." 2015. Accessed January 31. <http://id.wikipedia.org/wiki/KNN>.
- Krutil, J., M. Kudelka, and V. Snasel. 2012. "Web Page Classification Based on Schema.org Collection." In , 356–60. doi:10.1109/CASoN.2012.6412428.
- "Mesin Pencari Web." 2015. Accessed January 31. http://id.wikipedia.org/wiki/Mesin_pencari_web.
- "Radial Basis Function Network." 2015. Accessed January 29. http://en.wikipedia.org/wiki/Radial_basis_function_network.
- Rajalakshmi, R., and Chandrabose A. 2013. "Web Page Classification Using N-gram Based URL Features." In , 15–21. doi:10.1109/ICoAC.2013.6921920.
- Rani, M, Singh V, and Bhushan B. 2013. "Performance Evaluation of Classification Techniques Based on Mean Absolute Error" Vol 4 (Issue 1 January 2013).
- Riboni, D. 2002. *Feature Selection for Web Page Classification*. na.
- Saadah, M.N., Rigga W.A, Dyah S.R, and Agus Z.A. 2013. "Sistem Temu Kembali Dokumen Teks Dengan Pembobotan Tf-Idf Dan LCS" Vol 11: 17–20.
- "Situs Web." 2014. Accessed December 16. http://id.wikipedia.org/wiki/Situs_web#URL_28uniform_resource_locator.29.
- Swales, J. 1990. *Genre Analysis*. New York: Cambridge University Press.
- "Total Number of Websites." 2014. Internet Live Stats. Accessed October 22. <http://www.internetlivestats.com/total-number-of-websites/>.
- "URL." 2014. Accessed December 16. <http://id.wikipedia.org/wiki/URL>.
- Vapnik V. 1995. "The Nature of Statistical Learning Theory." Springer.
- Weimin Xue, Hong Bao, Weitong Huang, and Yuchang Lu. 2006. "Web Page

- Classification Based on SVM.*” In , 2:6111–14. doi:10.1109/WCICA.2006.1714255.
- WEKA. 2014. “*Machine Learning.*” Accessed December 8. <http://www.cs.waikato.ac.nz/ml/weka/>.
- Yun Lin, and Jie Wang. 2014. “*Research on Text Classification Based on SVM-KNN.*” In , 842–44. doi:10.1109/ICSESS.2014.6933697.
- Zhaohui Xu, Fuliang Yan, Jie Qin, and Haifeng Zhu. 2011. “*A Web Page Classification Algorithm Based on Link Information.*” In , 82–86. doi:10.1109/DCABES.2011.19.