Implementasi dan Analisis Mesin Focused Crawler untuk Web Musik dengan Menggunakan Learning Anchor Algorithm

Madina Ulfa[1], Shaufiah, ST., MT[2], Hetti Hidayati, S.Kom., MT [3]

Fakultas Informatika Telkom University, Bandung

Abstrak

ABSTRAKSI: Perkembangan world-wide web yang semakin pesat diikuti oleh kebutuhan informasi yang semakin meningkat, menjadi tantangan yang belum pernah terjadi sebelumnya untuk *general-purpose crawler* dan *search engines*. *Search engine* seperti Google, Yahoo!, Altavista dan sebagainya telah diperkenalkan dan digunakan untuk mempermudah pencarian informasi di Internet. *Web Crawler (crawler)* adalah sebuah *program/script* otomatis yang memproses halaman web untuk sebuah mesin pencari, yang banyak digunakan saat ini. Namun mengingat banyaknya halaman web yang ada, maka seringkali *search engine* dengan *crawler* biasa tidak dapat memberikan hasil yang maksimal. Untuk itu dikembangkanlah *focused crawler*. *Focused crawler* akan men-*download* halaman web yang sesuai topik dan berhati-hati memutuskan URL mana yang akan di-*scan* dan dalam urutan apa dilanjutkan berdasarkan informasi halaman download sebelumnya. Untuk tugas akhir ini, yang akan diproses adalah web musik.
Focused crawler membutuhkan classifier untuk membedakan halaman web yang relevan dan tidak. Yang pada tugas akhir ini digunakan Naïve Bayes Classifier. Halaman web yang relevan akan diekstrak outgoing-linknya dan disimpan kedalam *frontier*. Link dapat dicrawl dengan menggunakan algoritma penelusuran. Pemilihan algoritma penelusuran yang tepat akan berpengaruh pada efisiensi web crawler. Pada tugas akhir ini yang akan digunakan adalah *Learning Anchor Algorithm*.
Berdasarkan implementasi, dihasilkan akurasi terbaik 100% pada link pengujian http://gigsplay.com/ dengan dataset 100 musik dan 100 nonmusik. Sedangkan akurasi terendah 86.7% saat dataset 300 musik dan 200 nonmusik pada link pengujian http://musik.kapanlagi.com/.

Kata Kunci : *focused crawler*, web olahraga, *naïve bayes*, *Learning Anchor Algorithm*.

Abstract
The rapid growth of world-wide web followed by the increase on demand of information becomes a challenge which never occurs in the past for general-purpose crawler and search engines. Search engines, namely google, yahoo! Alavista and etc., have been introduced and utilized to ease the user browsing information on the internet. Web crawler (crawler) is an automatic program that processes a web page into a search engine on which commonly used nowadays. Nevertheless, due to the abundant presence of web page, search engine and crawler frequently cannot lead to a maximum result. As a result, Focused Crawler is developed. Focused crawler will automatically download a web page that are matched with desired topic and carefully decides which URL to be scanned on the basis either in order or previous download history. In this final year project, the subject matter to be highlighted is music web.
Focused crawler requires classifier to distinguish between relevant and irrelevant webs, in which in this final year project, the classier used is Naïve Bayes. All relevant web pages will be extracted in terms of outgoing-link and it will be saved into frontier. Link will be crawled by employing algorithm method/ formula. Algorithm method that is suitable will affect web crawler efficiency. And the author decided to use *Learning Anchor Algorithm.*
Based on the implementation, the resulting of best accuracy is 100% when the testing link is http://gigsplay.com/ *with dataset 100 music and 100 nonmusic. While the lowest accuracy is 86.7% when the dataset is 300 music and 200 nonmusic at testing link* http://musik.kapanlagi.com/**.**

Keywords: focused crawler, sport web, naïve bayes, *Learning Anchor Algorithm*.

1. Pendahuluan

1.1 Latar Belakang

Dengan perkembangan teknologi informasi yang semakin pesat, saat ini dunia semakin menuju kepada ledakan informasi. Berbagai informasi di dunia dapat dengan mudah diperoleh hanya dengan Internet melalui halaman *World Wide Web* (www). Halaman web yang sangat banyak dan semakin hari semakin berkembang cepat ini menjadi salah satu sarana penyebaran informasi yang sangat baik. Melihat tingginya kebutuhan masyarakat akan informasi dan sangat banyaknya jumlah web yang akan terus

bertambah, maka diperlukan adanya *search engine*. *Search engine* seperti Google, Yahoo!, Altavista dan sebagainya telah diperkenalkan dan digunakan untuk mempermudah pencarian informasi di Internet. Dengan pertumbuhan informasi di www, ada permintaan yang besar untuk mengembangkan metode yang efisien dan efektif untuk mengatur dan mengambil informasi yang tersedia. Ini menjadi salah satu alasan para peneliti mengembangkan *crawler* agar dapat memberikan hasil yang maksimal saat pengguna Internet melakukan pencarian. *Web Crawler (crawler)* adalah sebuah *program/script* otomatis yang memproses halaman web untuk sebuah mesin pencari, yang banyak digunakan saat ini.

Manusia membutuhkan berbagai jenis informasi. Namun mengingat banyaknya halaman web yang ada, maka seringkali hasil yang diberikan oleh *search engine* dengan *crawler* biasa tidak dapat memberikan hasil yang maksimal. Karena sumber daya komputasi yang terbatas dan waktu yang terbatas, *focused crawler* telah dikembangkan. *Focused crawler* akan men-*download* halaman web yang sesuai topic dan berhati-hati memutuskan URL mana yang akan di-*scan* dan dalam urutan apa dilanjutkan berdasarkan informasi halaman download sebelumnya. Untuk tugas akhir ini, yang akan diproses adalah web musik.

Untuk membedakan halaman web yang relevan dan tidak dibutuhkan classifier, yang merupakan proses untuk menentukan kelas (label) dari suatu objek yang tidak memiliki label. Di tugas akhir ini yang akan digunakan adalah *Naïve Bayes Classifier* karena memiliki kompleksitas waktu yang optimal dan akurasinya cukup tinggi. Naïve Bayes adalah salah satu algoritma pembelajaran yang paling efektif dan efisien untuk machine learning dan data mining. Keuntungan dari penggunaan Naïve Bayes adalah membutuhkan jumlah data training yang kecil dan memiliki kompleksitas waktu yang optimal serta akurasinya cukup tinggi.

Focused crawler dirancang untuk selektif mengambil konten yang relevan dengan topic tertentu yang menarik menggunakan struktur hyperlink web. Halaman web yang relevan akan diekstrak outgoing-linknya dan disimpan kedalam *frontier*. Link dapat dicrawl dengan menggunakan algoritma penelusuran. Pemilihan algoritma penelusuran yang tepat akan berpengaruh pada efisiensi web crawler dalam mencari informasi. Ada beberapa macam algoritma penelusuran, diantaranya *Best First Search, Breadth-First Search, Dept First Search, Fish Search* dan lain-lain. Namun pada tugas akhir ini yang akan digunakan adalah algoritma *Learning Anchor*.

Performansi web crawler dilihat dari seberapa banyak link yang relevan yang terjaring. Saat pengujian akan dilakukan crawling langsung ke internet dengan diberikan batasan halaman maksimum. System akan mengembalikan sekumpulan dokumen sebagai jawaban dari perintah user. Terdapat dua kategori dokumen yang dihasilkan oleh sistem, yaitu *relevant documents* (dokumen yang relevan dengan query) dan *retrieved documents* (dokumen yang diterima pengguna). Banyak metode yang dapat digunakan untuk mengevaluasi kualitas dari focused crawler diantaranya akurasi, precision, recall dan F-Measure. Yang akan digunakan pada tugas akhir ini adalah akurasi untuk mengevaluasi ketepatan sistem dalam melakukan klasifikasi.

1.2 **Tujuan**

Tujuan dari tugas akhir ini adalah:
1. Merancang dan menganalisis suatu *focused crawling* menggunakan algoritma *Learning Anchor* yang mampu melakukan klasifikasi terhadap halaman web musik maupun yang bukan.
2. Mengukur performansi yang dihasilkan oleh crawler.

2. Landasan Teori

2.1 Internet dan World Wide Web (www)

Internet merupakan kumpulan jaringan komputer yang saling terhubung dari seluruh dunia. Internet merupakan media yang menguhubungkan semua informasi dari email, web dan sebagainya. World Wide Web, atau yang biasa disebut web, merupakan cara untuk mengakses informasi melalui media internet. Web adalah model sharing informasi yang dibangun diatas Internet. Dengan semakin berkembangnya web membuat lalu lintas internet sangat padat.

Web menggunakan konsep hypertext yang dapat menghubungkan sebuah dokumen dengan dokumen lainnya. Dalam konsep hypertext, informasi disimpan di dalam dokumen atau file. Setiap dokumen dapat saling terhubung menggunakan hyperlink. Link tersebut yang akan menghubungkan sebuah informasi dengan bagian lain di dokumen yang sama atau dengan dokumen lainya di computer yang sama atau bahkan pada computer lain di Internet.

2.2 Information Retrieval

Information retrieval (IR) merupakan cara untuk mendapatkan informasi (biasanya berupa dokumen) yang bersifat tidak terstruktur dan berasal dari dokumen yang sangat banyak untuk memenuhi kebutuhan informasi. Tujuan dari sistem IR adalah memenuhi kebutuhan informasi pengguna dengan me-retrieve semua dokumen yang mungkin relevan, pada waktu yang sama me-retrieve sesedikit mungkin dokumen yang tak-relevan. Sistem ini menggunakan fungsi heuristik untuk mendapatkan dokumen-dokumen yang relevan dengan query pengguna. Sistem IR yang baik memungkinkan pengguna menentukan secara cepat dan akurat apakah isi dari dokumen yang diterima memenuhi kebutuhannya. Agar representasi dokumen lebih baik, dokumen-dokumen dengan topik atau isi yang mirip dikelompokkan bersama-sama.
Proses pada IR terbagi menjadi dua bagian pokok, yaitu:

1. Indexing Subsistem
   Indexing merupakan proses yang paling awal penciptaan file indeks dari suatu dataset, berupa proses persiapan yang

dilakukan terhadap dokumen sehingga dokumen siap untuk diproses. Proses index terdiri dari 2 proses, yaitu *document indexing* dan *term indexing*. Dimana untuk memperoleh kumpulan term yang akan dimasukkan ke dalam indeks, diperlukan tahap preprocessing yang dilakukan secara umum dalam text preprocessing pada dokumen. Dari term indexing ini akan dihasilkan koleksi kata yang akan digunakan untuk meningkatkan performansi pencarian pada tahap selanjutnya.

1.1 Tokenizing

Tahap tokenizing atau parsing adalah tahap pemotongan string input berdasarkan kata yang disusunnya dan mengubah kumpulan *term* menjadi *lowercase*.

1.2 Stoplist

Tahap stoplist atau filtering adalah tahap mengambil kata-kata penting dari hasil token dengan menghapus kata-kata yang sering muncul seperti di, dan, dan sebagainya.

1.3 Stemming

Tahap stemming adalah tahap mencari kata dasar dari tiap kata hasil stoplist.

1.4 *Word Weighting*

Tahap word weighting adalah tahap pembobotan setiap *term* di dalam dokumen.

2. Searching Subsistem (Matching Sistem)

Searching merupakan proses menemukan kembali informasi dokumen relevan terhadap query yang diberikan oleh user. Proses yang terjadi adalah *text preprocessing, Boolean operation,* dan perankingan hasil retrieve.

2.3 Pembobotan

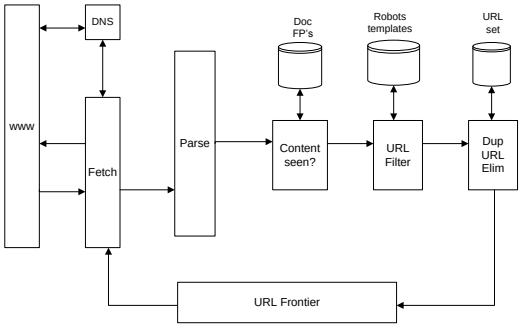
Ada beberapa metode pembobotan term baik untuk cakupan lokal, global atau kombinasi keduanya.

1. *Term Frequency (tf)*
2. *Inverse Document Frequency (idf)*
3. *tf-idf*

2.4 Web Crawler

Web crawler adalah salah satu komponen penting dalam sebuah mesin pencari modern. Fungsi utama *Web crawler* adalah untuk melakukan penjelajahan dan pengambilan halaman-halaman Web yang ada di Internet. Hasil pengumpulan situs Web selanjutnya akan diindeks oleh mesin pencari sehingga mempermudah pencarian informasi di Internet.Berikut ini proses yang dilakukan *Web crawler* pada saat bekerja :

1. Mengunduh halaman Web.
2. Memparsing halaman yang didownload dan mengambil semua link.
3. Untuk setiap link yang diambil, ulangi proses.



Mendesain sebuah *crawler* yang baik saat ini menemui banyak tantangan. Secara eksternal, *crawler* harus mengatasi besarnya situs Web dan link jaringan. Secara internal, *crawler* harus mengatasi besarnya volume data. Sehubungan dengan terbatasnya sumber daya komputasi dan keterbatasan waktu, maka harus hati-hati memutuskan link apa yang harus di scan dan bagaimana urutannya. Penting bagi *crawler* untuk memilih halaman dan mengunjungi halaman yang penting dulu dengan memprioritaskan link yang penting tersebut dalam antrian.

Ada 2 metode utama crawling: *exhaustive crawling* dan *focused crawling*. *Exhaustive crawling* bertujuan mengumpulkan seluruh halaman web dengan melakukan strategi pencarian secara transversal, seperti *Breadth-first Search.* Karena *exhaustive crawling* melewati semua halaman terlepas dari apapun isinya, dia sering dipakai oleh *general search engines* seperti Google dan Yahoo!. *Exhaustive crawling* berguna untuk crawling secara umum tapi membutuhkan ruang penyimpanan yang sangat besar, sehingga tidak efisien jika hanya halaman web dengan subjek khusus yang dibutuhkan.

Focused crawling dirancang untuk selektif mengambil konten yang relevan dengan topic tertentu yang menarik menggunakan struktur hyperlink web. Focused crawler dimulai dari seed list dari topic link. Seed adalah web-page yang mengandung banyak link page dengan topic terkait dan merupakan awal yang jenius untuk crawling. Ini memperkirakan kemungkinan bahwa setiap kandidat link berikutnya akan mengarahkan ke konten yang relevan berikutnya, dan dapat memprioritaskan urutan crawling pada basis itu dan/atau menolak kemungkinan link yang kecil. *Focused crawling* biasanya diaplikasikan untuk suatu tujuan tertentu misalnya aplikasi musik yang akan hanya men-download link-link yang berhubungan dengan musik.

Link dapat dicrawl dengan menggunakan algoritma penelusuran. Ada beberapa macam algoritma penelusuran, diantaranya *Best First Search, Breadth-First Search, Dept First Search, Fish Search* dan lain-lain. Namun pada tugas akhir ini yang akan digunakan adalah algoritma *Learning Anchor.*

2.4.1 Learning Anchor Algorithm

Learning Anchor Algorithm merupakan algoritma yang diperkenalkan oleh Ignacio Garc´ıa Dorado di tesisnya yang berjudul "*Focused Crawling: algorithm survey and new approaches with a manual analysis*". Algoritma

ini diimplementasikan menggunakan ide dari dua paper. Di paper pertama menunjukkan bahwa klasifikasi berdasarkan pada anchor text lebih akurat daripada klasifikasi berdasarkan seluruh halaman. Selain itu, mereka menggabungkan anchor dan seluruh parent dokumen (menggunakan representasi Bayesian). Kesimpulannya adalah bahwa campuran mencapai hasil yang lebih baik. Hasil itu diperoleh dari relasi antara web universitas dan web researcher, contohnya dari halaman departemen Computer Science mencoba menemukan informasi tentang, perkuliahan, departemen, homepages, projek dan lain-lain. Karena sebagian besar web universitas memiliki struktur web yang mirip, hasilnya tidak dapat diekstrapolasi ke topic lain.

Dalam paper yang berikutnya dijelaskan bahwa tidak perlu menemukan manual training set (seperti yang dilakukan kebanyakan algoritma). Mereka menunjukkan pendekatan yang berbeda: dilakukan sebuah TFIDF focused crawling, informasi yang diambil sekarang digunakan untuk melatih Naïve Bayes yang menjadi focused crawler yang sebenarnya. Awalnya sebuah proses crawl yang normal dilakukan untuk mengumpulkan informasi yang akan digunakan ditraining akhir. Mereka menggunakan seluruh struktur DOM sebagai training set di langkah kedua. Karena tidak ada penjelasan mengapa DOM merupakan solusi yang lebih baik dari pada normal anchor dan karena di paper pertama yang telah dijelaskan sebelumnya anchor memberikan hasil yang lebih menjanjikan, struktur DOM dihilangkan dan hanya anchor text yang akan digunakan. Karena tidak ada alasan untuk berpikir dikasus umum bahwa struktur web mirip satu sama lain.

Dengan kedua ide tersebut sebuah learning anchor algorithm diimplementasikan dengan beberapa modifikasi. Berikut adalah pseudocode dari *Learning Anchor Algorithm:*

insert in ready queue(seeds)
while *true* **do**
  if *more links in ready queue* **then**
    link := dequeue best
    doc := fetch(link)
    svm score := classify(doc)
    out link[] := extract links(doc)
    out anchor[] := extract links(doc)
    anchor score[] := class\ify(out anchor[])
    *final score := svm score * factor - anchor score * (1−factor)*
    save score(out links, final score)
    for *i = 0 to num out links* **do**
      if *abs(anchor score[i]) > threshold* **then**
        add to training(anchor score[i])
      end if
    end for
  else
    threshold := retrain anchor
    sorted links := sort decreasing(non processed queue)
    insert in ready queue(sorted links)
  end if
end while

Pada algoritma diatas digunakan skor SVM untuk menghitung *final scor*, dimana SVM digunakan sebagai metode klasifikasi. Karena di tugas akhir ini metode klasifikasi yang digunakan adalah *Naïve Bayes* maka yang akan digunakan skor *Naïve Bayes* untuk menghitung *final score.*

2.4.2 Classifier

Klasifikasi merupakan proses untuk menentukan kelas (label) dari suatu objek yang tidak memiliki label. Penentuan kelas dari suatu dokumen dilakukan dengan cara membandingkan nilai probabilitas suatu sampel berada di kelas yang satu dengan nilai probabilitas suatu sampel berada di kelas yang lain. Dalam klasifikasi sebuah pengklasifikasi dibuat dari sekumpulan data latih dengan kelas yang telah ditentukan sebelumnya.

2.4.2.1Teorema Bayes

Teorema Bayes adalah teorema yang digunakan dalam statistika untuk menghitung peluang untuk suatu hipotesis. Ide dasar dari teorema bayes adalah ketika kita dihadapkan pada masalah yang sifatnya hipotesis, dimana mendesain fungsi klasifikasi untuk memisahkan 2 jenis objek, dalam tugas akhir ini adalah web musik dan bukan web musik. Bayesian classifier punya tingkat akurasi dan kecepatan tinggi ketika diaplikasikan dalam database yang besar [han kamber]

$$P(c_j|d) = \frac{P(d|c_j)P(c_j)}{P(d)}$$

$P(c_j|d)$: probabilitas kemunculan kelas c_j dengan kondisi d
c_j : kategori teks yang akan diklasifikasi
d : dokumen teks yang dapat direpresentasikan sebagai himpunan kata ($w_1, w_2, \ldots, w_n$)
$P(c_j)$: probabilitas dari kategori teks c_j
P(d) : probabilitas dari kategori d, nilainya konstan untuk semua c_j
$P(d|c_j)$: probabilitas kemunculan kejadian d dengan kondisi c_j
Pada saat proses pengklasifikasian dokumen teks, maka pendekatan Bayes akan menyeleksi kategori teks yang memiliki probabilitas paling tinggi (c_{MAP}) yaitu [10]

$$c_{MAP} = \underset{c_j}{\operatorname{argmax}}\, P(c_j)\, P(d|c_j)$$

Pengklasifikasian menggunakan Teorema Bayes ini membutuhkan biaya komputasi yang mahal (waktu prosessor dan ukuran memory yang besar) karena kebutuhan untuk menghitung nilai probabilitas untuk tiap nilai dari perkalian kartesius untuk tiap nilai atribut dan tiap nilai kelas. [11]

2.4.2.2 Naïve Bayes Classifier

Naïve Bayes adalah salah satu algoritma pembelajaran yang paling efektif dan efisien untuk machine learning dan data mining. Classifier ini adalah Bayesian Classifier sederhana yang dapat dibandingkan dalam kinerjanya dengan *decision tree* dan *selected neural network classifier* [5]. **Naïve Bayes** merupakan pengklasifikasian dengan metode probabilitas dan statistik yang dikemukakan oleh ilmuwan Inggris Thomas Bayes, yaitu memprediksi peluang di masa depan berdasarkan pengalaman di masa sebelumnya sehingga dikenal sebagai **teorema Bayes**. Teorema tersebut dikombinasikan dengan "**naive**" dimana diasumsikan kondisi antar atribut saling bebas. Karena asumsi atribut tidak saling terkait, maka:

$$P(d|c_j) = \prod_{i=1}^{n} P(W_i|c_j)$$

Dengan menggunakan persamaan di atas, maka persamaan (2._) menjadi:

$$c_{MAP} = \underset{c_j}{\operatorname{argmax}}\, P(c_j) \prod_{i=1}^{n} P(W_i|c_j)$$

$P(c_j)$: probabilitas setiap dokumen terhadap sekumpulan dokumen

$P(W_i|c_j)$: probabilitas kemunculan kata W_i pada suatu dokumen dengan kategori kelas c_j

Nilai $P(c_j)$ dan $P(W_i|c_j)$ akan dihitung pada saat proses training dijalankan.

$$P(c_j) = \frac{n(W_j)}{n(sampel)}$$

$$P(W_i|c_j) = \frac{1 + n_i}{|C| + n(kosakata)}$$

$n(W_j)$: jumlah dokumen pada kategori j

n(sampel) : jumlah dokumen sampel yang digunakan dalam proses training

n_i : jumlah kemunculan kata W_i pada kategori c_j

$|C|$: jumlah sua kata pada kategori c_j

n(kosakata) : jumlah kata yang unik pada semua data training

Penentuan kelas dilakukan dengan cara membandingkan nilai probabilitas suatu sampel berada di kelas yang satu dengan nilai probabilitas suatu sampel berada di kelas yang lainnya.

Keuntungan penggunan Naïve bayes adalah bahwa metoda ini hanya membutuhkan jumlah data pelatihan (training data) yang kecil untuk menentukan estimasi parameter yang diperlukan dalam proses pengklasifikasian. Naïve bayes juga memiliki kompleksitas waktu yang optimal, itu sebabnya mengapa Naïve Bayes merupakan metode klasifikasi teks yang paling popular disamping akurasinya yang cukup tinggi [2]. Karena diasumsikan sebagai variable independent, maka hanya varians dari suatu variable dalam sebuah kelas yang dibutuhkan untuk menentukan klasifikasi, bukan keseluruhan dari matriks kovarians.

2.5 Parameter Pengukuran Performansi

Sistem IR akan mengembalikan sekumpulan dokumen sebagai jawaban dari query pengguna. Terdapat dua kategori dokumen yang dihasilkan oleh sistem IR terkait pemrosesan query, yaitu *relevant documents* (dokumen yang relevan dengan query) dan *retrieved documents* (dokumen yang diterima pengguna). [4]. Untuk mengevaluasi kualitas dari system dapat dilihat dengan menghitung tingkat akurasi, precision, recall dan F-Measure. Yang akan digunakan pada tugas akhir ini adalah F-Measure.

2.5.1 Akurasi

Akurasi dihitung untuk mengevaluasi ketepatan sistem untuk melakukan klasifikasi.

$$Accuracy = \frac{tp + tn}{tp + tn + fp + fn}$$

2.5.2 Recall

Recall mengevaluasi kemampuan sistem IR untuk menemukan semua item yang relevan dari dalam koleksi dokumen dan didefinisikan sebagai persentase dokumen yang relevan terhadap query pengguna dan yang diterima.

$$Recall = \frac{tp}{tp + fn}$$

2.5.3 Precision

Precision didefinisikan sebagai persentase dokumen yang di*retrieve* yang benar-benar relevan terhadap *query* pengguna. Precision mengevaluasi kemampuan sistem IR untuk menemukan kembali dokumen *top-ranked* yang paling relevan.

$$Precision = \frac{tp}{tp + fp}$$

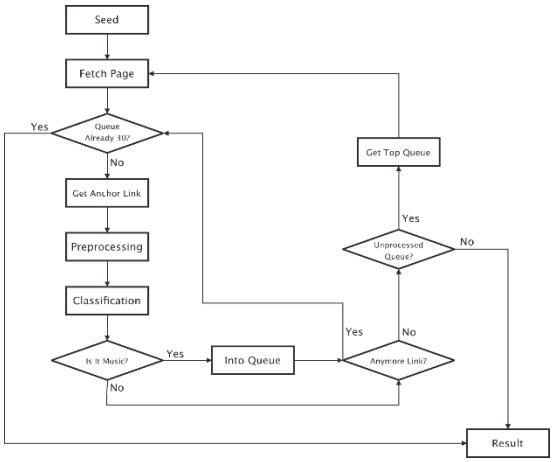
2.5.4 F-Measure

F-Measure dihitung untuk mengevaluasi kualitas dari information retrieval dengan mengkombinasikan recall dan precision. Yang merupakan rata-rata bobot harmonic precision dan recall.

$$F = 2.\frac{precision \times recall}{precision + recall}$$

3. Perancangan Sistem

Secara umum system yang akan dibangun di Tugas Akhir ini adalah system focused crawler yang menggunakan *Learning Anchor Algorithm* saat penelusuran dan Naïve bayes sebagai classifier. Adapun inputan dari user adalah seed sebagai link awal penelusuran.



Gambar 3-1 : Proses kerja sistem focused crawler

4. Analisis Hasil Pengujian

Pada saat pengujian dibutuhkan seed atau link awal untuk memulai penelusuran. Link tersebut merupakan link yang telah dikategorisasikan sebagai web musik. Sebab jika tidak termasuk kedalam kategori web music, maka proses crawling tidak akan berjalan. Link dipilih manual oleh pengguna. Berikut adalah link yang dipilih:

- http://musik.kapanlagi.com/
- http://musik.kapanlagi.com/

Adapun data yang digunakan untuk training berasal dari halaman web yang berbahasa Indonesia dengan kategori music dan non music. Data training yang digunakan tidak redundan sehingga pengetahuan ketika membangun model semakin banyak. Jumlah data training yang digunakan dalam proses crawling juga akan mempengaruhi hasil focused crawler.

4.1 Pengaruh komposisi dan jumlah dataset training terhadap akurasi

Pengujian dilakukan untuk mengetahui hasil maksimal pada proses crawling. Dataset yang digunakan antara 0,100, 200 dan 300 untuk setiap kategori. Untuk kategori music tidak terdapat jumlah dataset 0 karena yang ingin dihasilkan adalah page yang berkaitan dengan music. Berikut adalah akurasi dari pengujian dengan setiap seed dengan maksimum halaman 30:

non / musik	0	100	200	300
100	96.7%	100%	96.7%	96.7%
200	96.7%	96.7%	93.3%	93.3%
300	93.3%	93.3%	93.3%	93.3%

Table 4-2 : Hasil pengujian komposisi dataset (http://qiqsplay.com/)

non / musik	0	100	200	300
100	93.3%	93.3%	93.3%	90%
200	90%	90%	93.3%	93.3%
300	93.3%	90%	86.7%	96.7%

Table 4-3 : Hasil pengujian komposisi dataset (http://musik.kapanlagi.com/)

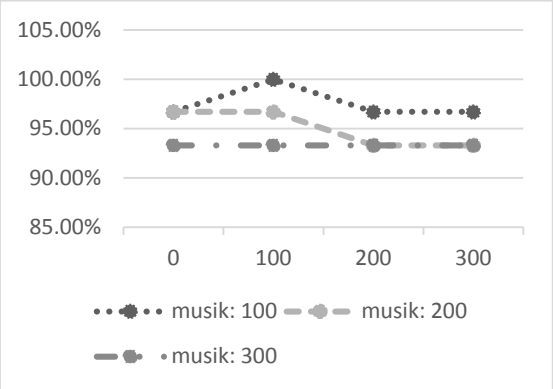
Dari hasil pada table 4-2 dapat dilihat bahwa tidak terdapat perubahan berarti pada akurasi yang dihasilkan. Akurasi tertinggi adalah pada dataset 100 musik dan 100 non music yaitu sebesar 100%. Sedangkan akurasi terendah 93.3% terdapat pada beberapa komposisi dataset. Di table 4-3 dapat dilihat juga tidak terdapat perubahan yang berarti pada akurasi yang dihasilkan. Akurasi tertinggi diperoleh saat dataset 300 musik dan 300 non music yaitu sebesar 96.7%. Dan akurasi terendah 86.7% terdapat pada komposisi dataset 300 musik dan 200 non musik. Perbedaan ini disebabkan oleh pemilihan dan jumlah data training yang mempengaruhi jumlah parameter yang ada. Banyaknya jumlah dataset pada data training yang digunakan tidak menjamin akurasi akan lebih baik. Hal ini dikarenakan naïve bayes memiliki sifat independensi kondisional dimana kemunculan suatu term tertentu tidak mempengaruhi kemunculan term lainnya, serta terdapat probabilitas bersyarat untuk setiap term sesuai dengan kelasnya. Dalam hal ini kelas music dan non music. Pemilihan term-term dalam data training juga sangat berpengaruh terhadap akurasi. Saat data training memiliki lebih banyak term-term penting (kata-kata yang dapat menjadi pembeda yang tepat), maka akurasinya akan meningkat.

Perbedaan hasil antar dataset pun terlihat pada urutan link yang dihasilkan. Perbedaan ini dikarenakan perbedaan jumlah parameter yang mempengaruhi bobot setiap link.

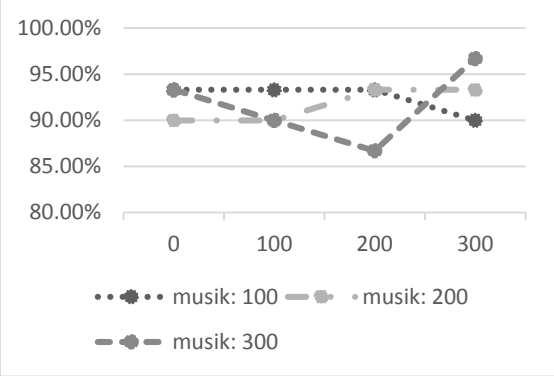
Hasil akurasi pada table diatas dapat naik ataupun turun jika dilakukan pada link pengujian yang berbeda. Hal ini dikarenakan perbedaan isi konten web. Walaupun secara actual dikatakan sebagai web musik, namun halaman tersebut juga berisi konten-

konten topic lain yang tidak berhubungan dengan music.

Berikut ini ditampilkan hasil pengujian dalam bentuk grafik:



Gambar 4-1 : Grafik hasil pengujian komposisi dataset (http://gigsplay.com/)



Gambar 4-2 : Grafik hasil pengujian komposisi dataset (http://musik.kapanlagi.com/)

4.2 Pengaruh kombinasi skor Naïve Bayes dan skor anchor terhadap akurasi

Kombinasi skor ini diperlukan karena seringkali anchor tidak merupakan informasi tentang dirinya sendiri (seperti ‘disini’ atau ‘press here’), jadi perlu menggunakan bagian dari skor Naïve Bayes untuk menyempurnakan skor anchor yang baru. Terdapat kombinasi yang berbeda dari skor Naïve Bayes dan skor anchor dari 0% anchor dan 100% Naïve Bayes hingga 100% anchor dan 0% Naïve Bayes.

Dari percobaan yang telah dilakukan disimpulkan penting untuk menggunakan persentase skor Naïve Bayes yang tinggi untuk mendapatkan hasil yang bagus sehingga pada program ini digunakan 100% skor Naïve Bayes.

5. Kesimpulan dan Saran

5.1 Kesimpulan

Berdasarkan pengujian yang telah dilakukan diatas, diperoleh kesimpulan sebagai berikut:

1. Klasifikasi Naïve Bayes dapat diimplementasikan pada focused crawler karena kemampuannya mengklasifikasikan data dengan baik, sehingga diperoleh hasil crawler yang bagus.
2. Karena naïve bayes classification bersifat independen kondisional, tidak hanya banyaknya jumlah data training yang dibutuhkan tetapi juga data training dengan kata-kata yang mewakili topic dengan baik.
3. Pemilihan data training yang sesuai dengan topic sangat berpengaruh pada hasil testing karena hasil training tersebut yang menjadi parameter pengujian.

5.2 Saran

Berdasarkan hasil yang telah dicapai pada sistem ini, adapun saran yang diharapkan dapat dilakukan untuk penelitian selanjutnya agar mencapai hasil yang lebih baik adalah:

1. Dapat digunakan algoritma penelusuran (crawling strategy) lain selain LAA, misalnya best first search, fish search, shark search atau yang lainnya.
2. Dapat melakukan penambahan variasi kategori dalam 1 mesin focused crawler sehingga pengguna dapat memilih topic yang diinginkan dengan lebih dinamis.
3. Dapat melakukan teknik klasifikasi lain yang telah dikenal, misalnya support vector machine.

DAFTAR PUSTAKA

[1] Cami, Aurel and Narsingh Deo, *“Evaluation of a Graph-based Topical Crawler.* Available at: http://www.eecs.ucf.edu/~deo/deo/icomp06-topical.pdf . Diakses tanggal 29 November 2012

[2] Christopher D. Manning, Prabhakar Raghavan & Hinrich Schütze. 2008. “*An Introduction to Information Retrieval”.* Cambridge University

[3] Dorado, Ignacio Garc´ıa. 2008. Focused Crawling: algorithm survey and new approaches with a manual analysis. *Thesis.* Sweden: Lund University. Available at: http://combine.it.lth.se/documentation/publ/Ignacio_Garcia_Dorado_MastersThesis.pdf . 9 Januari 2013

[4] Goker, Ayse, John Davies. 2009. *“Information Retrieval: searching in 21st century”.*United Kingdom: John Wiley & Sons, Ltd.

[5] Han, Jiawei dan Micheline Kamber. 2006. ”*Data Mining Concepts and Techniques*”. San Francisco: Morgan Kaufmann Publishers.

[6] Husni, *“IR dan Klasifikasi”,* diktat kuliah, Teknik Informatika Universitas Trunojoyo. Available at: http://husni.trunojoyo.ac.id/wp-content/uploads/2010/03/Husni-IR-dan-Klasifikasi.pdf. 30 November 2012

[7] Jamali, Mohsen., Hassan Sayyadi, Babak Bagheri Hariri and Hassan Abolhassani. *" A Method for Focused Crawling Using Combination of Link Structure and Content Similarity".* Available at: http://www.cs.umd.edu/~sayyadi/files/papers/4A_Method_for_Focused_Crawling_Using_Combination_of_Link_Structure_and_Content_Similarity.pdf . Diakses tanggal 29 November 2012

[8] Massandy, Danang Tri, Masayu Leylia Khodra. 2014. Klasifikasi Kategori Berita Dengan Metode Pembelajaran Semi Supervised.

[9] Pal, Anshika., Deepak Singh Tomar, S.C. Shrivastava. *"Effective Focused Crawling Based on Content and Link Structure Analysis".* Available at: **http://arxiv.org/ftp/arxiv/papers/0906/0906.5034.pdf** . diakses tanggal 29 November 2012

[10] Samodra, Joko, Surya Sumpeno dan Mochamad Hariadi. 2009. Klasifikasi Dokumen Teks Berbahasa Indonesia dengan Menggunakan Naïve Bayes. *Seminar Nasional Electrical, Informatics, and It's Educations*.

[11] Tang, Thanh Tin., David Hawking , Nick Craswell , Kathy Griffiths. 2005. Focused Crawling for both Topical Relevance and Quality of Medical Information. Available at: http://research.microsoft.com/pubs/65240/tang_cikm05.pdf . Diakses tanggal 29 November 2012

[12] Tsoi, Ah Chung., Daniele Forsali, Marco Gori, Markus Hagenbuchner, Franco Scarselli.*"A Simple Focused Crawler.* Available at: http://www2003.org/cdrom/papers/poster/p181/p181-tsoi/p181-tsoi.html . Diakses tanggal 29 November 2012

[13] Xuan, Wang, 2006," *Augmenting Focused Crawling using Search Engine Queries".* School of Computing, National University of Singapor