

IMPLEMENTASI TEKS MINING PADA APLIKASI PENGAWASAN PENGGUNAAN INTERNET ANAK "DODO KIDS BROWSER"

Firdaus Akhmad Muttaqin¹, Adam Mukaharil Bachtiar²

^{1,2}Teknik Informatika - Universitas Komputer Indonesia
Jl. Dipati Ukur No. 112-116, Bandung 40132

E-mail: firdaus.akhmad66@gmail.com¹, adammb@outlook.com²

ABSTRAK

Dodo Kids Browser merupakan perangkat lunak parental control untuk kegiatan pencarian atau surfing di internet oleh anak. Pengawasan dilakukan dengan cara memblokir setiap kata yang memiliki konteks negatif yang kemudian muncul pesan pada aplikasi mobile milik orang tua untuk memberikan aksi, namun kurangnya informasi mengenai sentimen dari *keyword* yang dimasukan menyulitkan orang tua untuk mengetahui apakah *keyword* tersebut termasuk pada sentimen negatif atau tidak. Hal tersebut berdampak pada pemilihan aksi yang akan diberikan oleh orang tua. Penerapan text mining dapat dijadikan sebagai solusi.

Implementasi teks mining digunakan untuk melakukan proses klasifikasi terhadap pencarian anak dalam mendapatkan informasi mengenai sentimennya. Tahapan yang dilakukan untuk proses klasifikasi yang pertama yaitu preprocessing data. Selanjutnya hasil data hasil preprocessing tersebut diterapkan pada algoritma *Naïve Bayes Classifier* untuk proses klasifikasi. Hasil klasifikasi yang ditampilkan yaitu berupa informasi mengenai saran dalam menentukan aksi oleh orang tua.

Hasil implementasi text mining terhadap sistem yang telah dilakukan pengujian fungsionalitas sistem, menguji algoritma *naïve bayes classifier*, dan pengujian terhadap beberapa sampel data uji. Hasil dari pengujian tersebut menyimpulkan bahwa sistem yang dibangun mampu memberikan informasi berupa saran yang mampu membantu orang tua dalam memutuskan pemberia aksi terhadap aktifitas internet anaknya.

Kata Kunci: Text Mining, Analisis Sentimen, *Naïve Bayes Classifier*, Klasifikasi.

1. PENDAHULUAN

Layanan internet sebagai media informasi yang semakin meningkat sudah mulai merambah ke setiap kalangan, tidak hanya remaja atau orang dewasa akan tetapi kalangan anak-anak pun sudah menggunakan layanan internet sebagai media pencarian informasi baik untuk kepentingan pribadi maupun untuk pendidikan. Hal ini memiliki dampak positif maupun

negatif, sehingga terdapat beberapa vendor yang menyediakan aplikasi atau layanan untuk memonitoring serta dapat membatasi kegiatan internet anak. Dodo Kids Browser merupakan salah satu perangkat lunak yang berfungsi sebagai *parental controlling* untuk kegiatan internet anak. Aplikasi ini dapat memberikan pemberitahuan kepada orang tua ketika anak melakukan pencarian.

Berdasarkan hasil observasi yang dilakukan dengan cara mencoba layanan yang disediakan pada aplikasi Dodo Kids Browser di antaranya yaitu penyaringan konten pada kata kunci yang dimasukan anak ketika melakukan pencarian, aplikasi ini akan melakukan pemblokiran pada setiap kata kunci yang bermakna negatif sehingga setiap kata yang memiliki makna negatif akan selalu dilakukan pemblokiran walaupun kata kunci yang dimasukan memiliki makna yang positif ketika menjadi sebuah frase atau kalimat. Hal ini menyebabkan masalah pada ketersediaan informasi yang seharusnya dapat diakses oleh anak akan tetapi menjadi tidak bisa dilakukan karena pada kata kunci yang dimasukan tersebut terdapat kata yang bermakna negatif. Sebagai contoh ketika anak melakukan pencarian dalam bahasa inggris dengan kata kunci "*how to avoid violence*", dalam kata kunci yang dimasukan tersebut terdapat kata "*violence*" yang memiliki makna negatif bagi anak akan tetapi jika dalam sebuah kalimat, kata kunci yang dimasukan tersebut memiliki makna yang positif. Hal tersebut disebabkan karena keterbatasan kemampuan untuk menghasilkan kesimpulan dari kata kunci pencarian yang dimasukan oleh anak. Hal ini dapat menyulitkan orang tua dalam mendapatkan referensi untuk menentukan aksi yang tepat kepada anak.

Berdasarkan permasalahan yang telah dipaparkan tersebut dibutuhkan suatu solusi yang dapat mengklasifikasikan kata kunci yang dimasukan oleh anak ketika melakukan pencarian untuk menghasilkan kesimpulan positif atau negatif dari kata kunci yang dimasukan tersebut. Hal ini dimungkinkan dengan penggunaan text mining yaitu proses yang semi otomatis melakukan klasifikasi dari pola yang berasal dari database tidak terstruktur. Hasil dari klasifikasi dapat dijadikan sebagai media untuk memberikan saran untuk orang tua dalam

menentukan aksi terhadap anak ketika melakukan suatu pencarian di internet.

Dalam melakukan pengklasifikasian terdapat banyak algoritma yang dapat digunakan dalam mengklasifikasikan kata kunci pencarian ke dalam kelas negatif atau positif salah satunya yaitu naïve bayes. Berdasarkan beberapa penelitian mengenai perbandingan kinerja algoritma *naïve bayes* dengan algoritma lain disimpulkan naïve bayes memiliki tingkat akurasi 87.88% untuk data kategori yang lebih baik dari tingkat akurasi algoritma decision tree yang memiliki 84.85% [1]. Selain itu pula terdapat penelitian mengenai penerapan naïve bayes pada pengklasifikasian spam dari data latih 80 sms memiliki tingkat akurasi 85,11% [2]. Berdasarkan hal tersebut memungkinkan algoritma *naïve bayes* untuk diterapkan dalam mengklasifikasikan kata kunci pencarian. Selain itu *naïve bayes* merupakan salah satu algoritma yang konvensional dan sederhana oleh karena itu naïve bayes cocok untuk diimplementasikan dalam pada aplikasi pengawasan penggunaan internet anak "Dodo Kids Browser".

1.1 Text Mining

Text Mining merupakan suatu langkah dari analisis teks yang dilakukan secara otomatis oleh sistem komputer untuk menghasilkan informasi baru yang belum diketahui sebelumnya yang diambil dari suatu rangkaian teks yang terangkum dalam sebuah dokumen [3]. Text Mining adalah bidang multi disiplin yang melibatkan information retrieval, text analysis, information extraction, clustering, categorization, visualization, machine learning dan teknik lainnya [4]. Text mining menggunakan penerapan data mining untuk mengubah data tidak terstruktur menjadi data terstruktur melalui tahap-tahap yaitu [4]:

1. *Text Preprocess* yaitu pemecahan sekumpulan karakter ke dalam kata-kata.
2. *Feature Generation / Text Transformation* yaitu mengubah kata-kata ke dalam bentuk dasar sekaligus mengurangi jumlah kata-kata tersebut.
3. *Feature Selection* yaitu seleksi feature untuk mengurangi dimensi dari suatu kumpulan teks.
4. *Text Mining / Pattern Discovery* yaitu dapat berupa unsupervised learning (clustering) atau supervised learning (classification).
5. *Interpretation / Evaluation* yaitu pengukuran efektifitas untuk mengevaluasi metode yang diterapkan menggunakan parameter precision.

1.2 Analisis Sentimen

Analisis sentimen atau dapat disebut juga *opinion mining* merupakan proses memahami, mengekstrak dan mengolah data teks tekstual secara otomatis untuk mendapatkan informasi sentiment yang terkandung dalam suatu kalimat opini [5]. Analisa sentimen bertujuan menentukan suatu isi dari dataset yang

berbentuk tesktual atau kalimat apakah bernilai sentimen positif atau negatif [6]. Opinion mining dapat dianggap pula sebagai kombinasi antara text mining dan natural language processing. Metode klasifikasi merupakan metode yang dapat digunakan untuk menyelesaikan masalah pada text mining. Salah satunya yaitu dengan menggunakan algoritma *Naïve Bayes Classifier* (NBC). Sedangkan Natural language processing berfungsi untuk memberikan kelas kata (tag) ke setiap kata dalam suatu kalimat.

1.3 Preprocessing

Preprocessing merupakan tahapan sebelum proses pengklasifikasian yang diperlukan untuk membersihkan, menghilangkan, mengubah sumber data, baik itu berupa karakter non alfabet maupun kata-kata yang tidak diperlukan. Hal ini bertujuan agar data yang digunakan lebih optimal ketika digunakan pada proses pengklasifikasiannya. Tahapan *preprocessing* setiap kasus dapat berbeda-beda. Berikut ini merupakan tahapan preprocessing dan penjelasannya yang digunakan dalam penelitian ini.

1. *Cleansing*

Cleansing merupakan proses membersihkan data yang akan digunakan dari karakter-karakter bahkan kata-kata yang tidak diperlukan. Hal ini bertujuan untuk mengurangi noise yang dapat menimbulkan proses perhitungan dalam pengklasifikasian tidak optimal.

2. *Case Folding*

Case Folding merupakan proses pengubahan data menjadi format yang sesuai. Hal ini bertujuan mengurangi redundansi data yang akan digunakan dalam proses pengklasifikasian sehingga proses perhitungan pun menjadi optimal. Contohnya mengubah format data menjadi lowercase atau uppercase sesuai dengan kebutuhan yang dibutuhkan dalam proses pengklasifikasiannya.

3. *Tokenizing*

Tokenizing merupakan proses pemisahan atau memotong data baik berupa frasa, klausa, atau kalimat menjadi kata perkata berdasarkan delimiter yang digunakan yaitu space.

1.4 Algoritma Naïve Bayes

Naïve Naïve Bayes Classifier merupakan suatu metode *classifier* yang mengacu pada teorema *bayes* yaitu teorema yang mengacu pada konsep probabilitas bersyarat. Pada metode ini diperlukan kombinasi pengetahuan sebelumnya dengan pengetahuan yang baru [7]. Dalam melakukan pengklasifikasiannya diperlukan *training* set sebagai data latih. Pada setiap *sample* dari data latih tersebut memiliki kelas label tersendiri. Berikut merupakan model matematis *naïve bayes classifier* yaitu:

$$p(H|X) = \frac{p(H)p(X|H)}{p(X)} \dots [2]$$

Di mana:

X = Data dengan kelas yang belum diketahui

H = Hipotesa data X merupakan suatu kelas spesifik

$p(H|X)$ = Probabilitas hipotesis H berdasarkan kondisi X (*posterior probability*)

$p(H)$ = Probabilitas hipotesis H (*prior probability*)

2. ISI PENELITIAN

Isi penelitian ini bertujuan untuk memaparkan penelitian yang dilakukan dari proses analisis hingga implementasi ke dalam sistem. Berikut bahasan dari penelitian ini.

2.1 Analisis Masalah

Permasalahan yang terjadi pada penelitian ini yaitu orang tua sebagai pengguna perlu menentukan aksi yang tepat terhadap pencarian yang dilakukan oleh anak apakah bersifat positif atau negative, sehingga diperlukan informasi hasil klasifikasi pencarian tersebut berupa saran untuk menentukan aksi yang akan diberikan.

2.2 Sumber Data

Sumber data yang digunakan berupa kata kunci pencarian berupa URL suatu *search engine*. Dalam melakukan *request* pencarian, suatu *search engine* akan melakukan request data dengan menggunakan metode GET dengan contoh yaitu dengan mengirimkan suatu parameter yang berisi kata kunci yang dimasukan. Berikut merupakan contoh dari sumber data tersebut yang disajikan pada Tabel 1.

Tabel 1 Sumber Data

Kata kunci Pencarian	Format URL	Nama Parameter
<i>Good people example</i>	http://www.bing.com/search?q=good+people+example&go=Submit&qsn=&form=QBRLH&pq=good+people+example&sc=0-0&sp=-1&sk=&cvid=f8323014b9c64795b68123461eb2a982	q
<i>Kind of case</i>	https://www.google.com/search?q=Search+something&gws_rd=ssl	q
<i>How to avoid violence</i>	https://www.google.com/search?q=Search+something&gws_rd=ssl#q=what+do+i+search	q
<i>How to bully people</i>	https://www.google.com/search?q=How+to	q

Kata kunci Pencarian	Format URL	Nama Parameter
	o+bully+people&gws_rd=ssl	
<i>Example of violence</i>	https://www.google.com/search?q=How+to+bully+a+people&gws_rd=ssl	q
<i>Good violence</i>	http://www.bing.com/search?q=Good+violence&go=Submit&qsn=&form=QBRE&pq=Good+violence&sc=8-10&sp=-1&sk=&cvid=166e131d89424cefa4e2aec4be4891fd	q

2.3 Implementasi Preprocessing

Proses preprocessing dilakukan guna mengubah sumber data menjadi format yang sesuai dan mudah untuk dilakukan proses pengklasifikasian sehingga proses klasifikasi tersebut dapat lebih optimal. Tahapan *preprocessing* yang dilakukan pada penelitian ini yaitu mulai dari proses *cleansing*, *case folding*, dan yang terakhir adalah proses *tokenizing*. Tahapan-tahapan tersebut dilihat pada Gambar 1.



Gambar 1 Tahapan Preprocessing

Berikut ini merupakan implementasi penjelasan dari tahapan-tahapan tersebut.

1. Cleansing

Pada tahap ini dilakukan pembersihan simbol dan huruf yang tidak diperlukan. Selain itu dilakukan pula perubahan simbol tertentu yang berhubungan dengan kata kunci pencarian yaitu format *space*, yang dalam hal ini *space* akan berubah menjadi simbol + (plus), sehingga dilakukan pengubahan kembali menjadi *space*. Langkah-langkah proses *cleansing* disajikan dalam bentuk *flowchart* yang dapat dilihat pada Gambar 2.



Gambar 2 Flowchart Proses Cleansing

Data masukan pada proses *cleansing* ini berupa URL yang dihasilkan ketika melakukan pencarian pada web browser. Berikut merupakan contoh dari penerapan proses *cleansing* yang disajikan pada Tabel 2

Tabel 2 Proses Cleansing

Data Masukan	Hasil Cleansing
http://www.bing.com/search?q=good+people+example&go=Submit&qs=n&form=QBLH&pq=good+people+example&sc=0-0&sp=-1&sk=&cvid=f8323014b9c64795b68123461eb2a982	<i>Good people example</i>
https://www.google.com/search?q=Search+something&gws_rd=ssl	<i>Kind of cats</i>
https://www.google.com/search?q=Search+something&gws_rd=ssl#q=what+do+i+search	<i>How to avoid violence</i>
https://www.google.com/search?q=How+to+bully+people&gws_rd=ssl	<i>How to bully people</i>
https://www.google.com/search?q=How+to+bully+a+people&gws_rd=ssl	<i>Example of violence</i>
http://www.bing.com/search?q=Good+violence&go=Submit&qs	<i>Good violence</i>

Data Masukan	Hasil Cleansing
=n&form=QBRE&pq=Good+violence&sc=8-10&sp=-1&sk=&cvid=166e131d89424cefa4e2aec4be4891fd	

2. Case Folding

Pada tahap *case folding* dilakukan pengubahan data hasil proses *cleansing* menjadi ke dalam bentuk yang sama. Dalam kasus ini pengubahan dilakukan menjadi ke dalam format *lower case*. Berikut merupakan tahapan proses *case folding* yang disajikan dalam bentuk *flowchart* pada gambar.



Gambar 3 Flowchart Proses Cleansing

Berdasarkan Gambar 3, berikut merupakan contoh dari penerapan proses *case folding* yang disajikan pada Tabel 3.

Tabel 3 Penerapan Proses Case Folding

Data Masukan	Hasil Case Folding
<i>Good people example</i>	<i>good people example</i>
<i>Kind of cats</i>	<i>kind of cats</i>
<i>How to avoid violence</i>	<i>how to avoid violence</i>
<i>How to bully people</i>	<i>how to bully people</i>
<i>Example of violence</i>	<i>example of violence</i>
<i>Good violence</i>	<i>good violence</i>

3. Tokenizing

Tokenizing merupakan tahapan melakukan pemecahan suatu gabungan dua kata atau lebih atau dapat disebut juga seperti suatu frasa atau kalimat sehingga menjadi persatu satu. Dalam kasus ini pemisahan yang dilakukan berdasarkan *space* sebagai *delimiter*. Berikut merupakan

tahapan proses *tokenizing* yang disajikan dalam bentuk *flowchart* pada Gambar 4.



Gambar 4 Flowchart Proses Tokenizing

Berdasarkan Gambar 4, berikut merupakan contoh dari penerapan proses *tokenizing* yang disajikan pada Tabel 4.

Tabel 4 Penerapan Proses Tokenizing

Data Masukan	Hasil <i>Tokenizing</i>
<i>good people example</i>	<i>good people example</i>
<i>kind of cat</i>	<i>kind of cat</i>
<i>how to avoid violence</i>	<i>how to avoid violence</i>
<i>how to bully people</i>	<i>Howe to bully people</i>
<i>example of violence</i>	<i>example of violence</i>
<i>good violence</i>	<i>Good violence</i>

Berdasarkan tabel, data masukan tersebut merupakan data hasil dari proses *case folding* yang kemudian dilakukan proses *tokenizing* sehingga dihasilkan pemisahan setiap kata.

2.4 Implementasi Algoritma Naïve Bayes

Tahap Pada tahap ini dilakukan proses analisis algoritma *Naïve Bayes* yang merupakan hal penting dalam melakukan pengklasifikasian terhadap sumber

data atas *sentiment* yang dimilikinya apakah positif atau negatif. Dalam tahap ini terdapat dua buah proses utama untuk melakukan pengklasifikasian tersebut yaitu proses pembelajaran dan proses pengklasifikasian. Berikut merupakan penjelasan masing-masing proses.

1. Proses Pembelajaran

Dalam proses ini *naïve bayes classifier* perlu diberi pengetahuan awal untuk dijadikan acuan agar dapat melakukan pengklasifikasian terhadap data tekstual berdasarkan sentimennya. Dalam proses pembelajaran atau *learning* ini terdapat tiga langkah utama. Berikut merupakan ketiga langkah utama tersebut beserta penjelasannya.

a. Penentuan Kelas Data Latih

Pada tahap ini dilakukan penentuan kelas suatu data. Penentuan kelas tersebut ditentukan dengan bantuan pengguna dengan cara memberikan opini terhadap suatu keyword pencarian apakah termasuk pada kelas positif atau kelas negatif. Berikut merupakan contoh dari penentuan kelas data latih yang disajikan pada Tabel 5.

Tabel 5 Penentuan Kelas Data

Data	Kata	Kelas Sentimen
D1	food people example	Positif
D2	kind of cat	Positif
D3	how to avoid violence	Positif
D4	how to bully people	Negatif
D5	example of violence	Negatif
D6	good violence	Negatif

b. Perhitungan Probabilitas

Pada tahap ini dilakukan perhitungan probabilitas terhadap data yang telah ditentukan kelasnya. Tabel 6 merupakan perhitungan probabilitas dari setiap kelas.

Tabel 6 Perhitungan Probabilitas

Kelas sentimen	Jumlah Kata						Probabilitas
	D 1	D 2	D 3	D 4	D 5	D 6	
Positif	3	3	4	0	0	0	10/19

Kelas sentimen	Jumlah Kata						Probabilitas
	D 1	D 2	D 3	D 4	D 5	D 6	
Negatif	0	0	0	4	3	2	9/19
Total	3	3	4	4	3	2	1

c. Menentukan Probabilitas Item

Setelah probabilitas setiap kelas dihitung, selanjutnya dilakukan perhitungan probabilitas setiap itemnya. Berikut merupakan rumus untuk menghitung probabilitas per-item tersebut.

$$p(i) = \frac{f(i)}{f(u)}$$

Dimana,

$p(i)$ = Probabilitas item

$f(i)$ = Frekuensi item

$f(u)$ = Frekuensi total item berdasarkan kelas sentimennya.

Berikut merupakan perhitungan probabilitas setiap item yang disajikan pada Tabel 7.

Tabel 7 Perhitungan Probabilitas Item

Data	Kelas Sentimen	
	Positif	Negatif
good	$\frac{1}{10}$	$\frac{1}{9}$
people	$\frac{1}{10}$	$\frac{1}{9}$
example	$\frac{1}{10}$	-
kind	$\frac{1}{10}$	-
of	$\frac{1}{10}$	$\frac{1}{9}$
cats	$\frac{1}{10}$	-
how	$\frac{1}{10}$	$\frac{1}{9}$
do	$\frac{1}{10}$	$\frac{1}{9}$
avoid	$\frac{1}{10}$	-
violence	$\frac{1}{10}$	$\frac{2}{9}$
bully	-	$\frac{1}{9}$
people	-	$\frac{1}{9}$

2. Proses Klasifikasi

Dalam tahap ini akan dilakukan klasifikasi terhadap data baru yaitu sebagai data uji dengan menggunakan *naïve bayes classifier*. Berikut merupakan alur dari proses klasifikasinya yang dapat dilihat pada Gambar 5.



Gambar 5 Flowchart Proses Klasifikasi

Berdasarkan pada Gambar 5 tersebut tahap awal yang dilakukan yaitu melakukan input data uji. Data uji yang digunakan berasal dari sumber data yang telah dilakukan *preprocessing*. Berikut merupakan contoh data uji yang disajikan pada Tabel 8.

Tabel 8 Contoh Data Uji

Data Latih	
Sebelum <i>preprocessing</i>	Setelah <i>preprocessing</i>
https://www.google.com/search?q=Violence+in+classroom&gws_rd=sal	<i>violence in classroom</i>

Tahap selanjutnya merupakan proses yang sangat penting yaitu menghitung probabilitas tiap kelas, baik negatif maupun positif. Proses tersebut akan dijelaskan sebagai berikut.

a. Menghitung Probabilitas

Pada tahap ini dilakukan proses perhitungan probabilitas dengan menggunakan *naïve bayes classifier*. Data uji tiap kelas akan ditentukan nilai probabilitasnya berdasarkan dari proses learning. Berikut merupakan proses perhitungannya.

1). Perhitungan untuk probabilitas positif

$$P = P_{c+} * P(\text{violence}|\text{positif}) * P(\text{in}|\text{positif}) * P(\text{classroom}|\text{positif})$$

$$P = 0.5263 * \frac{1}{10} * \frac{1}{10} * \frac{1}{10}$$

$$P = 0,000005263$$

2). Perhitungan untuk probabilitas negative

$$P = P_{c-} * P(\text{violence}|\text{negatif}) * P(\text{in}|\text{negatif}) * P(\text{classroom}|\text{negatif})$$

$$P = 0.4737 * \frac{2}{9} * \frac{1}{9} * \frac{1}{9}$$

$$P = 0,001299588$$

b. Menentukan Probabilitas Sentimen Maksimum.

Dari hasil perhitungan sebelumnya dibandingkan antara nilai Ppositif dan Pnegatif, didapatkan nilai tertinggi yaitu Pnegatif sehingga dapat disimpulkan bahwa pencarian yang dilakukan diklasifikasikan ke dalam sentiment negatif.

2.5 Implementasi Sistem

Tahap implementasi merupakan tahap penerapan elemen-elemen yang telah dilakukan pada tahap analisis dan perancangan sistem untuk diimplementasikan ke dalam sebuah sistem. Tahapan ini meliputi lingkungan implementasi, implementasi data, dan implementasi antarmuka.

1. Lingkungan Implementasi

Lingkungan implementasi merupakan penjelasan dari penerapan sistem yang terdiri atas dua lingkungan yaitu pada lingkungan perangkat lunak dan lingkungan perangkat keras. Spesifikasi perangkat lunak dalam implementasi sistem yaitu sebagai berikut.

- Sistem Operasi Windows 8.1 Pro
- WeBuilder 2014
- MySQL DBMS
- Visual Studio 2013
- MySQL Workbench 6.3
- StarUML 5.0.2.1570

Spesifikasi perangkat keras pada sistem yang dibangun adalah sebagai berikut.

- Processor Core i3 M380 @2.53GHz
- RAM 6 GB
- HDD 256 GB
- Monitor LED
- Keyboard dan Mouse

2. Implementasi Data

Data yang terlibat pada sistem yang dibangun yaitu berasal dari rangkaian teks pencarian pada suatu web browser dengan menggunakan extension. Berikut merupakan struktur dari implementasi data

yang digunakan dalam sistem yang dibangun disajikan pada Tabel 9.

Tabel 9 Penggunaan Data

No	Koleksi	Data yang Dikirim
1	Data Testing	a. uid: integer b. post_id : string c. konten: string d. sugest: string e. id_notif: integer f. uri
2	Data Training	a. id_post: integer b. word: string c. status: string

3. Implementasi Antarmuka

Implementasi antarmuka berisi tampilan antarmuka dari sistem yang dibangun hasil implementasi dari perancangan sebelumnya. Antarmuka yang dibangun ini diimplementasikan kepada antarmuka pada aplikasi Dodo: Kids Browser versi mobile Windows Phone yang di dalamnya merupakan visualisasi dari hasil klasifikasi yang telah dilakukan berupa informasi saran untuk orang tua. Berikut merupakan tampilan yang implementasi antarmuka dari sistem yang dibangun dapat dilihat pada Gambar 6.



Gambar 6 Antarmuka Visualisasi Hasil Klasifikasi

3. PENUTUP

Berdasarkan dari hasil penelitian yang telah dilakukan bahwa penerapan text mining dalam aplikasi pengawasan penggunaan internet anak “dodo kids browser” dengan memberikan solusi berupa suatu pemberian informasi berupa saran hasil klasifikasi dari kata kunci yang digunakan oleh anak dalam suatu pencarian di web browser untuk membantu orang tua dalam memberikan aksi terhadap anak ketika anak tersebut terindikasi menggunakan kata kunci dengan konteks negatif telah diimplementasikan sesuai dengan analisis dan perancangan sebelumnya. Sehingga dapat

disimpulkan bahwa hasil klasifikasi ini mampu membantu orang tua dalam mendapatkan informasi berupa saran dalam menentukan aksi yang tepat untuk anak yang terindikasi melakukan pencarian ketika *surfing* dengan kata yang mengandung makna buruk.

DAFTAR PUSTAKA

- [1] D. Oktafia and D. C. Pardede, "Perbandingan Kinerja Algoritma Decision Tree dan Naive Bayes dalam Prediksi Kebangkrutan," UG Repository, Jakarta, 2014.
- [2] E. A.W, M. and T. , "Penerapan Naive Bayes Untuk Sistem Klasifikasi SMS Pada Smartphone Android," EPrints 3 , Palembang, 2013.
- [3] I. F. Rozi, S. H. Pramono and E. A. Dahlan, "Implementasi Opinion Mining (Analisis Sentimen) Untuk Ekstraksi Data Opini Publik pada Perguruan Tinggi," *Jurnal EECCIS*, vol. 6, pp. 37-43, 2012.
- [4] J. Ling, I. P. E. N. Kencana and T. B. Oka, "Analisis Sentimen Menggunakan Metode Naive Bayes Classifier Dengan Seleksi Fitur Chi Square," *E-Jurnal Matematika*, vol. 3, pp. 92-99, 2014.
- [5] S. Andini, "Klasifikasi Dokument Teks Menggunakan Algoritma Naive Bayes Dengan Bahasa Pemograman Java," *Jurnal Teknologi Informasi & Pendidikan*, vol. 6, pp. 140-147, 2013.
- [6] A. Nurani, B. Susanto and U. Proboyekti, "Implementasi Naive Bayes Classifier Pada Program Bantu Penentuan Buku Referensi Matakuliah," *Jurnal Informatika*, vol. 3, pp. 32-36, 2007.
- [7] S. F. Rodiyansyah and E. Winarko, "Klasifikasi Posting Twitter Kemacetan Lalu Lintas Kota Bandung Menggunakan Naive Bayesian Classification," *IJCCS*, vol. 6, pp. 91-100, 2012.