PRE-PROCESSING TEXT MINING PADA DATA TWITTER

Siti Mujilahwati[1]
[1]Program Studi Teknik Informatika, Fakultas Teknik, Universitas Islam Lamongan
Jl, Veteran No.53 A Lamongan
Telp (0322)324706
E-mail: moedjee@gmail.com

ABSTRAK
Pertumbuhan sosial media yang sangat pesat tidak membuat twitter ditinggal oleh penggunanya. Twitter merupakan sebuah sosial media yang dimanfaatkan oleh penggunanya untuk berbagi informasi. Tidak banyak karakter yang dapat dimasukkan pada komentar di twitter. Keterbatasan karakter tersebut membuat para peneliti memakai data tersebut untuk penelitiannya. Komentar ditwitter mengandung banyak ragam type data dan beragam gaya bahasa. Oleh sebab itu diperlukan penanganan khusus pada data komentar dari twitter. Penelitian kali ini akan membahas teknik penanganan data preprocessing data komentar dari twitter. Untuk mengetahui hasil teknik preprocessing yang dihasilkan maka pada penelitian ini akan di ujikan untuk proses klasifikasi layanan sebuah perusahaan telekomunikasi dan didapatkan hasil akurasi mencapai 93,11%.

Kata Kunci: Text Mining, Data Mining, Pre-processing, Twitter

ABSTRACT
Growth social media is very rapid does not make twitter left by users. Twitter is a social media used by users to share information. Not a lot of characters that can be inserted in the comments on twitter. The character makes the researchers used these data for research. Comment on twitter contains a wide variety of data types and diverse style. It therefore requires special handling of the data comments from twitter. The present study will discuss data handling techniques of data preprocessing comments from twitter. To find out the results generated preprocessing techniques, this research will test to the classification of services a telecommunications company until 93,11 % accuracy rate is achieved.

Keyword : Text Mining, Data Mining, Pre-processing, Twitter

1. PENDAHULUAN

1.1. Latar Belakang

Melihat pola hidup manusia saat ini lebih cenderung dengan kehidupan dunia maya, aktifitas sehari-hari yang tidak lepas dari internet. Baik untuk bekerja, usaha, belajar dan juga untuk bersosialisasi sesama teman. Hal tersebut mengakibatkan banyaknya bermunculan sebuah situs yang dinamakan sosial media, salah satunya adalah twitter. Twitter mengalami pertumbuhan yang pesat dan dengan cepat meraih popularitas di seluruh dunia. Hingga bulan Januari 2013, terdapat lebih dari 500 juta pengguna terdaftar di twitter, 200 juta diantaranya adalah pengguna aktif. Pertambahan penggunaan twitter umumnya berlangsung saat terjadinya peristiwa-peristiwa populer. Pada awal 2013, pengguna twitter mengirimkan lebih dari 340 juta komentar (*tweet*) per hari, dan twitter menangani lebih dari 1,6 miliar permintaan pencarian per hari. Twitter memiliki tingkat pertumbuhan pengguna bulanan sebesar 40 persen. Data tersebut membuat minat para peneliti untuk memanfaatkan data komentar (*tweet*) dan melakukan teknik *mining* terhadap data tersebut (Alexander,2013). Baik untuk analisis, klasifikasi ataupun juga asosiasi. Pada disiplin ilmu hal tersebut termasuk kategori *text mining*. Karena komentar pada twitter mengandung beragam jenis data seperti text, angka, *emoticon*, *hastag*, *mention* dan lain-lain menjadikan komentar tersebut memiliki tipe yang komplek (Apoorv, dkk. 2011) Dari uraian tersebut maka diperlukan adanya penanganan yang ekstra pada saat tahap *pre-processing* atau tahap persiapan data. Pada penelitian kali ini akan membahas beberapa teknik penanganan data komentar dari twitter untuk proses data mining.

1.2. Metode Penelitian

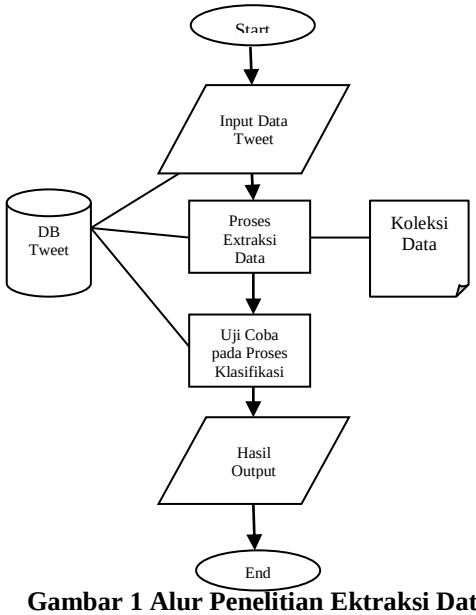
Tahap *pre-processing* atau praproses data merupakan proses untuk mempersiapkan data mentah sebelum dilakukan proses lain. Pada umumnya, praproses data dilakukan dengan cara mengeliminasi data yang tidak sesuai atau mengubah data menjadi bentuk yang lebih mudah diproses oleh sistem. Praproses sangat penting dalam melakukan analisis sentimen, terutama untuk media sosial yang sebagian besar berisi kata-kata atau kalimat yang tidak formal dan tidak terstruktur serta memiliki *noise* yang besar. Ada tiga model praproses untuk kalimat atau teks dengan *noise* yang besar (A Clark, 2003). Tiga model tersebut adalah :

1. *Orthographic Model*. Model ini dipergunakan untuk memperbaiki kata atau kalimat yang memiliki kesalahan dari segi bentuk kata atau kalimat. Contoh kesalahan yang diperbaiki

dengan *Orthographic model* adalah huruf kapital di tengah kata.

2. *Error Model.* Model ini dipergunakan untuk memperbaiki kesalahan dari segi kesalahan eja atau kesalahan penulisan. Ada dua jenis kesalahan yang dikoreksi dengan model ini yaitu kesalahan penulisan dan kesalahan eja. Kesalahan penulisan mengacu pada kesalahan pengetikan sedangkan kesalahan eja muncul ketika penulis tidak tahu ejaannya benar atau salah.
3. *White Space Model*. Model ke tiga ini mengacu pada pengoreksian tanda baca. Contoh kesalahan untuk model ini adalah tidak menggunakan tanda titik '.' di akhir kalimat. Namun, model ini tidak terlalu signifikan, terutama ketika berhadapan dengan media sosial yang jarang mengindahkan tanda baca.

Rangkaian dari penelitian ini adalah melakukan ekstraksi data menjadi data yang siap untuk digunakan teknik mining. Tahap praproses data ini dapat kita sebut sebagai ektraksi data. Alur dari penelitian ini dapat ditunjukkan pada Gambar 1. Pertama yang dilakukan adalah pengambilan data dari twitter secara otomatis dan disimpan dalam database. Sesuai dengan tujuan dari teknik mining yang akan dilakukan, misalkan pada kasus ini yang nanti hasilnya akan dipakai untuk teknik klasifikasi maka data mentah sebelum dilakukan tahap praproses terlebih dahulu harus dilabeli secara manual untuk menentukan kelas setiap masing-masing komentar. Paling penting data yang diambil adalah data komentar (*tweet*) berdasarkan topik yang diinginkan. Selanjutnya data yang sudah tersimpan pada database akan dilakukan ektraksi data, hasil ektraksi atau praproses akan dilakukan pengujian untuk kasus klasifikasi.



Gambar 1 Alur Penelitian Ektraksi Data

Merujuk pada penelitian sebelumnya yang dilakukan oleh Himalatha (Himalatha, dkk, 2012) maka pada penelitian ini akan dibahas beberapa proses ektraksi data antara lain *case folding, remove punctuation, remove username, remove hashtag, clean number, clean one char, remove url, remove RT, convert number*dan *remove number*.

1. *Case Folding*, bertujuan membuat semua text menjadi huruf kecil.
2. *Remove Punctuation*. Bertujuan menghapus semua karakter *non alphabet* misalnya simbol, spasi dan lain-lain.
3. *Remove Username*. Bertujuan menghapus nama *user* biasanya diawali dengan simbol "@" karena dalam suatu kasus dapat dianggap tidak penting maka perlu dihilangkan, apabila dibutuhkan maka proses ini tidak perlu dilakukan.
4. *Remove Hashtag*. *Hashtag* hanyalah suatu penunjuk sebuah kata yang dibicarakan oleh sesama pengguna twitter yang memiliki simbol "#". Biasanya akan digunakan sebagai judul topik pembicaraan dan juga berfungsi sebagai pengelompokan terhadap percakapan yang berhubungan dengan kata yang diberi simbol *hashtag*. Proses ini juga dapat dikategorikan antara penting dan tidak penting, dapat dilakukan ataupun tidak dilakukan proses *Remove Hashtag*.
5. *Clean Number*. Berfungsi untuk menghapus angka yang selalu ada di depan dan di belakang kata. Meskipun dalam penulisan komentar selalu menyertakan sebuah angka di setiap awal atau akhir kalimat untuk menunjukkan bahwa kalimat tersebut diulang-ulang maka dalam bahasa Indonesia yang baik itu merupakan hal yang salah. Begitu juga pada sebuah penelitian, apabila menemukan sebuah kata yang menggunakan tambahan angka maka perlu dihapus. Contohnya hujan2 maksudnya hujan-hujan, i2 maksudnya itu.
6. *Clean One Character.* Berfungsi menghapus jika terdapat hanya satu huruf saja, karena tidak mengandung arti. Seringnya muncul sebuah huruf pada komentar twitter membuat sebuah hasil data ektraksi yang banyak dan tidak baik. Satu huruf yang dimaksud adalah sebagai contoh y, g, k dan lain sebagainya. Walaupun maksud dari penulis komentar bahwa y adalah ya, g adalah tidak, k adalah kok. Maka untuk proses ekstraksi data itu merupakan sebuah kata yang tidak mudah dideklarasikan karena tidak memiliki arti yang jelas.
7. *Removal URL*. Seringnya muncul sebuah url dari data twitter membuat data tidak efektif dan tidak memiliki arti. Untuk itu perlu adanya penghapusan url tersebut. Kemunculan alamat

web atau url ini disebabkan karena banyaknya *user* mempromosikan sebuah produk pada situs mereka supaya *user* yang lain langsung bisa masuk pada halaman web yang dimaksud.

8. *Remove RT*. Pada twitter untuk menunjuk atau mengajak teman berkomunikasi langsung adalah dengan menambahkan simbol "@" sebelum *user name* yang dituju. Pada suatu penelitian tidak memperhatikan sebuah nama *user* dan banyaknya *user* yang komentar. Peneliti hanya memanfaatkan data atau komentar *user* tersebut, untuk itu perlu dihapus.
9. *Convert Number*. Seringnya pemakaian bahasa *gaul* pada twitter melibatkan angka menjadi variasi dalam menulis seperti "s4y4n9" dan lainnya. Dalam Bahasa Indonesia yang baik kata "s4y4n9" tidak memiliki makna, padahal maksud dari kata tersebut adalah sayang. Untuk itu perlu adanya proses *convert number* untuk mengkonversi angka menjadi huruf.
   Sebelum melakukan konversi nomor maka perlu dideskripsikan perubahan yang diinginkan. Pemodelan ini sebenarnya ada keuntungan dan kerugian. Apabila penelitian pada kasus layanan sinyal atau berhubungan dengan produk operator maka bisa saja proses ektraksi ini tidak dilakukan. Karena dapat merubah arti sebuah kata pada komentar. Seperti sinyal 3G, apabila dilakukan c*onvert number* maka angka 3 akan dihapus dan untuk huruf G bisa saja dilakukan proses selanjutnya yaitu proses *convert word.* Dalam penelitian ini perubahan c*onvert number* yang dipakai datanya dapat direpresentasikan seperti pada Tabel 1.

Tabel 1 Konversi Angka ke Huruf

No	Angka	Huruf
1	1	i
2	3	e
3	4	a
4	5	s
6	6 dan 9	g
7	7	t
8	8	b

   Konversi angka ke huruf pada penelitian ini hanya menggunakan data seperti pada Tabel 1, angka 1 diganti dengan huruf i, angka 3 diganti dengan huruf e dan angka 4 diganti dengan huruf a. Angka lima diganti dengan huruf s angka 6 dan 9 diganti dengan huruf g. untuk angka 7 diganti dengan huruf t dan angka 8 diganti huruf b.
10. *Remove Stop Word*. *Stop word* diproses pada sebuah kalimat jika mengandung kata-kata yang sering keluar dan di anggap tidak penting seperti waktu, penghubung, dan lain sebagainya (Vijayarani). Untuk itu perlu dilakukan penghapusan. Untuk melakukan proses penghapusan kata ini diperlukan sebuah data atau daftar kata yang diinginkan untuk dihapus.

Tabel 2 Data untuk Stop Word

#Kata hubung	# Waktu	# Kata tanya
dengan	senin	apa
di	selasa	bagaimana
karena	rabu	dimana
ke	kamis	kapan
is	jumat	mengapa
yang	sabtu	siapa
jika	minggu	
bagi	januari	
akan	februari	
sebagai	maret	
seperti	april	
kalau	mei	

11. *Remove Negation Word*. Untuk *negation word* sebenarnya prosesnya tidaklah menghapus kata melainkan diambil untuk menilai bahwa kalimat yang diproses mengandung kalimat negatif. Selanjutnya akan ditambahkan ke sebuah variabel yang sudah ditentukan untuk dihitung. Misalnya kasus sentimen analisis yang membutuhkan penilaian pada kalimat positif dan negatif.
    Sama dengan penggunaan fungsi penghapusan kata *stop word*, pada fungsi penghapusan *negation word* ini juga menggunakan sebuah *file path* berupa *file text* sebagai penyimpan data yang dikoleksi seperti pada Tabel 3.

Tabel 3 Daftar Kata Negation Word

No	Kata
1	Gak
2	ga
3	bkn
4	bukan
5	enggak
6	g
7	jangan
8	nggak
9	tak
10	tdk
11	tidak

12. *Convert Word*. Pentingnya *convert word* adalah untuk mengkonversi kalimat yang tidak baku, saat ini penggunaan kalimat *alay* atau bahasa *gaul* mengakibatkan penggunaan Bahasa Indonesia tidak baku.

Tabel 4 Contoh Daftar Kata untuk Convert Word

Sebelum	Sesudah
Akyu	Aku
akuwh	Aku
akku	Aku
aq	Aku
aquwh	Aku
awak	Aku
amaca	Ahmasak
alluw	Hallo
atw	Atau
bb	Blackberry
bwt	Buat
bs	Bisa
bsa	Bisa
bli	Beli
binun	Bingung
btw	Ngomong-ngomong
bnerin	Benerin
bapuk	Jelek
bnr	Benar
cemungud	Semangat
ciyus	Serius
cuxin	Cuekin
coz	Sebab
cz	Karena
cay	Saying
cayank	Saying
dmn	Dimana
ett	Add
enelan	Beneran
engga	Enggak
eank	Yang
fren	Teman
gantii	Ganti
gantiii	Ganti
gnt	Ganti
gmn	Gimana
gni	Gini
grtis	Gratis
gituu	Begitu
Hhumz	Rumah

13. *Convert Emoticon*. Seringnya ekpresi diungkapkan dengan sebuah gambar atau simbol *emoticon* dalam twitter menyebabkan perlu adanya pengkonversian ke dalam bentuk *string* yang dapat diartikan maknanya.
Fungsi untuk mengkonversi simbol *emoticon* ini hampir sama dengan fungsi melakukan *convert negation* dan *convert word*. Hanya saja isi atau koleksi data yang dipakai yang berbeda. Dalam penelitian ini digunakan tiga ekspresi pada symbol emoticon (Read, 2005; Go et al., 2009). Seperti pada Tabel 5 Daftar emoticon yang dipakai.

Tabel 5 Daftar *Emoticon*

Emoticon	**Konversi**	**Masuk kelas**
>:] :-) :) :o) :] :3 :c) :> =] 8) =) :} :^)	Senang	Positif
>:D :-D :D 8-D 8D x-D xD XD XD =-D =D =-3 =3	Tertawa	Positif
>:[:-(:(:-c :c :-< :< :-[:[:{ > .><. <> .< :'(	Sedih	Negative
D :< D : D 8 D ; D = D X v.v D-':	Horror	Netral
>:P :-P :P X-P x-p xp XP :- p :p =p :-Þ :Þ :-b :b	Tongue	Netral
>:o>:O :-O :O °o° °O° :O o_O o.O 8-0	Shock	Positif
>:\ >:/ :-/ :-. :/ :\ =/ =\ :S	Kesal	Negative
:| :-|	Ekspesi Datar	Negative

2. PEMBAHASAN

Melakukan tahap praproses dinilai sangat penting dalam teknik data mining, terutama pada data yang bersumber dari sosial media yang berupa sebuah text. Untuk menilai dari hasil penelitian ini peneliti akan membahas beberapa point berikut diantaranya adalah pre-processing dan uji coba untuk kasus klasifikasi.

2.1 Pre Processing (Extraksi Data)

Aplikasi *Pre-processing* ini dibangun dengan menggunakan bahasa pemprograman java. Pada point ini akan dibahas beberapa formula (code) yang digunakan dalam penelitian ini.

1. *Case Folding*

```
1. public String foldCase(String
   myString) {
2. return myString.toLowerCase();}
```

2. *Remove Punctuation*

```
1. public String
   removePunctuation(String myString) {
2. String myPattern = "[^A-Za-z0-
   9\\s]+";
3. String newString =
   myString.replaceAll(myPattern, "");
4. return newString}
```

3. *Remove User Name*

```
1. public String removeUsers(String
   myString) {
2. Extractor myExtractor = new
   Extractor();
3. List<String> myUsers =
   myExtractor.extractMentionedScreenna
   mes(myString);
4. String myResult =
   this.removeWords(myUsers,myString,"@
   ");
5. return myResult;}
```

4. *Remove Hashtag*

```
1. public String removeHashtags(String
   myString) {
2. Extractor myExtractor = new
   Extractor();
3. List<String> myHashtags =
4. myExtractor.extractHashtags(myString
   );
5. String myResult =
   this.removeWords(myHashtags,
   myString,      "#");
6. return myResult;}
```

5. *Clean Number*

```
1. public String cleanNumber(String
   myString) {
2. String myPatternEnd = "([0-
   9]+)(\\s|$)";
3. String myPatternBegin = "(^|\\s)([0-
   9]+)";
4. myString =
   myString.replaceAll(myPatternBegin,
   "$1");
5. myString =
   myString.replaceAll(myPatternEnd,
   "$2");
6. return myString;}
```

6. *Clean One Character*

```
1. public StringremoveSingleChar(String
    myString) {
2. String newString = "";
3. String[] listWords = yString.split("
    ");
4. for (String myWord : listWords) {
5. if (myWord.length() > 1) {
6. if (newString.length() != 0) {
7. newString += " " + myWord;
8. } else {
9. newString = myWord;}}}
10.return newString;}
```

7. *Removal URL*

```
1. public    String    removeURLs(String
   myString) {
2. Extractor    myExtractor    =    new
   Extractor();
3. List<String>         myURLs         =
   myExtractor.extractURLs(myString);
4. String           myResult           =
   this.removeWords(myURLs, myString);
5. return myResult; }
```

8. *Remove RT*

```
1. public String removeRT(String
   myString) {
2. String myPattern = "(\\s)(RT)(\\s)";
3. String newString =
   myString.replaceAll(myPattern,
   "$1");
4. return newString; }
```

9. *Convert Number*

```
1. public String convertNumber(String
   myString) {
2. myString = myString.replace("00",
   "u");
3. Iterator myIterator =
   numberMap.entrySet().iterator();
4. while (myIterator.hasNext()) {
5. Map.Entry myPair = (Map.Entry)
   myIterator.next();
6. String myKey = (String)
   myPair.getKey();
7. String myValue = (String)
   myPair.getValue();
8. myString =
   myString.replaceAll(myKey, myValue);
   }
9. return myString; }
```

10. *Remove Stopword*

```
1. public String removeStopWords(String
   myString) {
2. for (String myStopWord :
   this.stopWordsList) {
3. String myPattern = "(^|\\s)(" +
   myStopWord + ")(\\s|$)";
4. myString =
   myString.replaceAll(myPattern, " ");
   }
5. return myString;    }
```

11. *Remove Negation word*

```
1. public String
   removeNegationWords(String myString)
   {
2. for (String myStopWord :
   this.negationWordsList) {
3. String myPattern = "(^|\\s)(" +
   myStopWord + ")(\\s|$)";
4. myString =
   myString.replaceAll(myPattern, " ");
   }
5. return myString; }
```

12. *Convert Word*

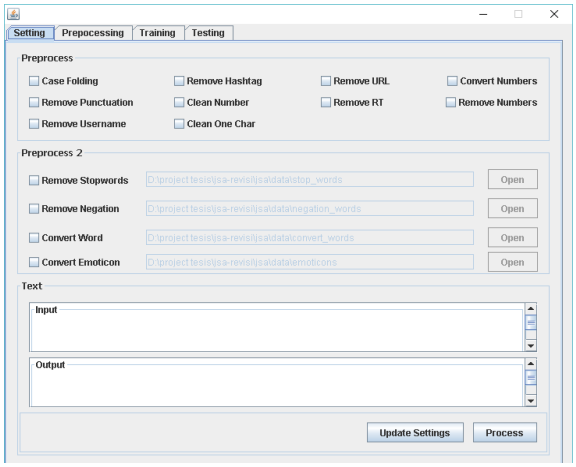
```
1. public String convertWords(String
   myString) {
2. terator myIterator =
   this.convertWordMap.entrySet().itera
   tor();
3. while (myIterator.hasNext()) {
4. Map.Entry myPair = (Map.Entry)
   myIterator.next();
5. String myKey = (String)
   myPair.getKey();
6. String myValue = (String)
   myPair.getValue();
7. myString =
   this.convertWord(myString, myKey,
   myValue);}
8. return myString;}
```

13. *Convert Emoticon*

```
1. public String
   convertEmoticons(String myString) {
```

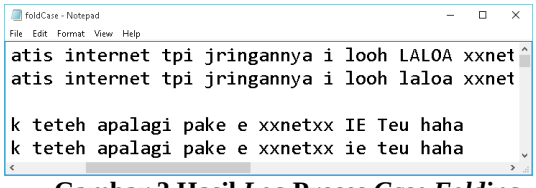
```
2.  Iterator myIterator =
    this.emoticonsMap.entrySet().iterato
    r();
3.  while (myIterator.hasNext()) {
4.  Map.Entry myPair = (Map.Entry)
    myIterator.next();
5.  String myKey = (String)
    myPair.getKey();
6.  String myValue = (String)
    myPair.getValue();
7.  myString =
    this.convertWord(myString,Pattern.qu
    otemyKey), myValue);}
8.  return myString;     }
```

Hasil implementasi pada *desain user interface* adalah seperti nampak pada Gambar 2.



Gambar 2 Implementasi Code Preprocessing Data Twitter dengan Java

Pada Gambar 2 dapat dijelaskan bahwa setiap tahap proses dapat dipilih sesuai dengan kebutuhan, untuk itu di desain dengan model pilihan ceklist, sehingga pengguna dapat menyesuaikan topik penelitian mana saja proses yang dapat digunakan. Sedangkan untuk daftar *stopward, Negation, Conver Word* dan *Convert Emoticon* digunakan model *load*. File yang disimpan berupa file text, sehingga apabila *user* menghendaki penambahan dan pengurangan data *list* nya dapat menambahkan langsung ke dalam *file text*. Hasil yang diperoleh dari sistem *pre-processing* yang dibuat adalah langsung berupa hasil data yang disebut dengan data latih. Akan tetapi pada sistem ini dibuat sebuah *log* untuk mendokumentasikan setiap proses yang dilakukan pada proses *pre-processing*. Pada Gambar 3 akan diberikan contoh hasil *log* proses *case folding*.



Gambar 3 Hasil *Log* Proses *Case Folding*

Pada Gambar 3 dapat dijelaskan bahwa desain untuk hasil prosesnya adalah terdiri dari dua baris, baris pertama adalah teks asli dan baris ke dua adalah hasil *case folding*. Pada hasil *log* tersebut sebenarnya disertakan tanggal dilakukannya proses tersebut. Masing-masing proses yang ada pada *pre-processing* memiliki satu *log* berupa *text file*.

Hasil uji coba untuk semua proses pada tahap *pre-processing* yang dilakukan dapat digambarkan pada Tabel 6.

Tabel 6 Hasil Tahap Pre-Processing

Proses	Kata asli	Hasil Ekstraksi
Casefolding	SurabayaPOS	Surabayapos
Remove punctuation	Ka:bar*ba!k	Kabarbak
Remove Username	@sandiwahono	(hilang)
Remove Hashtag	#makanan	(hilang)
Clean Number	Teknik12	Teknik
Clean OneChar	G makan	Makan
RemoveUrl	http://stts.edu	(hilang)
RemoveRT	iya mel RT @moedjee	Iya mel@moedjee
Convert Numbers	M4m4	Mama
Remove Number	Hanya 5 jam	Hanya jam
Remove stopword	Mau ke kampus	Mau kampus
Remove Negationword	Tidak tahu	Tahu
Convertword	Aq lagi tlp	Aku lagi telepon
Convert Emoticon	☺	Pos

2.2 Pengujian Hasil Pre-Processing untuk Kasus Klasifikasi

Hasil dataset yang diperoleh dari tahap *pre-processing* selanjutnya akan diuji coba untuk proses klasifikasi layanan produk suatu perusahaan telekomunikasi. Dengan menggunakan data dari twitter sebanyak 680 *record* untuk data latih dan 450 *record* untuk data uji. Contoh data yang digunakan yang berasal dari komentar di twitter dapat dilihat pada Tabel 7. (Mujilahwati, 2015)

Tabel 7 Contoh Data Komentar dari Twitter

No	Komentar
1	serasa mati tanpa internet dalam beberapa hari... :'(gara gara indosat @**indosat** fuck*

2	Hadeuhhh...bebenya udh benerr .skrg providernya yg error **@indosat#mentari**
3	SinyaL indosat gila' naik turun trus **@indosat**
4	**@XLCare** sinyal bb cm GPRS doang nih,,, ada apa ya ?
5	2tahun pake simpati bb gue sinyal 3G terus tapi belakangan kenapa sekarang EDGE mulu gan @Telkomsel -_-
6	Sapu-sapu dada saja sama jaringannya telkomsel ini ☹
7	Mending INDOSAT ayeuna mah beli paket 25ribu dapat 2GB cobaTelkomsel 100ribu lebih pelit

Dengan menggunakan teknik *pre-processing* yang telah dibuat data komentar tersebut akan berubah menjadi dataset seperti pada Tabel 8.

Tabel 8 Hasil Ekstraksi (Pre-Processing)

No	Komentar
1	Serasa mati tanpa internet dalam beberapa hari….sedih gara gara indosat indosat fuck
2	Hadeuhhh…bebenya udh benerr..skrg providernya yang error indosat mentari
3	Sinyal indosat gila naik turun trus indosat
4	Xlcare sinyal bb cm gprs doing nih….ada apa ya?
5	Tahun pake simpati sinyal terus belakangan kenapa sekarang edge mulu gan telkomsel
6	Sapu sapu dada saja sama jaringannya telkomsel ini (ekspresi datar)
7	Mending indosat ayeuna mah beli paket ribu dapat gb coba telkomsel ribu lebih pelit

Dari yang digunakan pada uji coba ini dapat di gambarkan dengan sebuah tabel matrix sperti ditunjukkan pada Tabel 9.

Tabel 9 Kebutuhan pada Data Uji

No	Kelas	Jumlah Data Training	Jumlah Data Uji
1	Sinyal	188	158
2	Tarif	112	45
3	Internet	196	130
4	Android	22	10
5	Blackberry	97	47
6	Other	65	60

Algoritma yang dipakai untuk uji coba ini adalah dengan menggunakan algoritma Naïve Bayes. Didapatkan hasil klasifikasi dari data tersebut seperti digambarkan pada matrik Tabel 10.

Tabel 10 Hasil Klasifikasi Uji Coba

		Prediksi					
	Kelas	Sinyal	Tarif	Internet	Android	Blackberry	Other
Fakta	Sinyal	150	1	2	1	0	4
	Tarif	0	43	2	0	0	0
	Internet	5	1	124	0	0	0
	Android	0	0	0	10	0	0
	Black berry	0	2	0	0	45	0
	Other	3	4	1	2	3	47

Tingkat akurasi yang diperoleh dari hasil klasifikasi per kategori pada kelas layanan adalah sebagai berikut.

1. Kelas sinyal = 94.93%.
2. Kelas tarif = 95.55%.
3. Kelas internet = 95.38%.
4. Kelas android = 100%.
5. Kelas blackberry = 100%.
6. Kelas other = 78.33%.

Dari 450 data uji yang dipakai masing-masing kategori layanan mendapatkan nilai presentasi keakuratan yang sangat baik. Apabila dihitung dari nilai keseluruhan pada kelas layanan dari 450 *record* data uji dan dari data latih yang sudah melalui tahap *pre-processing*, algoritma Naïve Bayes dapat mengklasifikasikan sebanyak 419 *record*. Sehingga hasil nilai presentase akurasi untuk klasifiasi kategori layanan adalah 93.11%.

3. KESIMPULAN

Setiap proses pada tahap *pre-processing* data text dari twitter yang telah dibuat, memiliki hasil yang sangat baik, sehingga dapat digunakan untuk penelitian lebih lanjut tentang data mining atau text mining.

Penelitian ini tidak membahas proses stemming dan terbukti hasil *pre-processing* yang lakukan untuk uji coba klasifikasi mendapatkan hasil yang baik hingga mencapai tingkat akurasi 93.11%. Hasil uji dari kasus klasifikasi ini dianggap tidak maksimal dikarenakan adanya data *mention* terhadap *customer support* dan dilakukannya *remove usename* dan *remove hashtag*.

Walaupun tanpa menggunakan proses stemming hasil yang didapatkan sudah sangat baik, proses stemming dapat ditambahkan untuk mendapatkan hasil dataset yang lebih efisien, lebih ringkas dan

akan berpengaruh pada proses data mining yang tentunya prosesnya akan lebih cepat.

4. PUSTAKA

Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, Rebecca Passonneau, (2011), Sentiment Analysis Of Twitter Data. Department of Computer Science, Columbia University, New York, USA

Clark, A. (2003). Pre-processing Very Noisy Text. *Proceedings of Workshop on Shallow Processing of Large Corpora* (pp. 12-22). Lancaster: Lancaster University.

Hemalatha, P Saradhi Varma, G. Govardhan, A. Preprocessing the Informal Text for efficient Sentiment Analysis. International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), vol 1 Issue 2,July–August 2012

Jonathon Read, (2005). Using emoticons to reduce dependency in machine learning techniques for sentiment classification. In *ACL*. The Association for Computer Linguistics.

Mujilahwati. Siti, (2015), Klasifikasi dan sentiment analysis dari twitter untuk komentar pada penyedia jasa seluler di Indonesia. Makalah disajikan dalam Seminar Nasional pengembangan actual teknologi informasi. Proceding. UPN "Veteran", Surabaya, Jawa Timur, 02 Desember.

Pak. Alexander dan Paroubek. Patrick, (2013). Twitter as a Corpus for Sentiment Analysis and Opinion Mining. Laboratoire LIMSI-CNRS, Bˆatiment 508, Universit´e de Paris-Sud, France

Vijayarani, S., Ilamathi, J.,Nithya. Preprocessing Techniques for Text Mining - An Overview. International Journal of Computer Science & Communication Networks,Vol 5(1),7-16