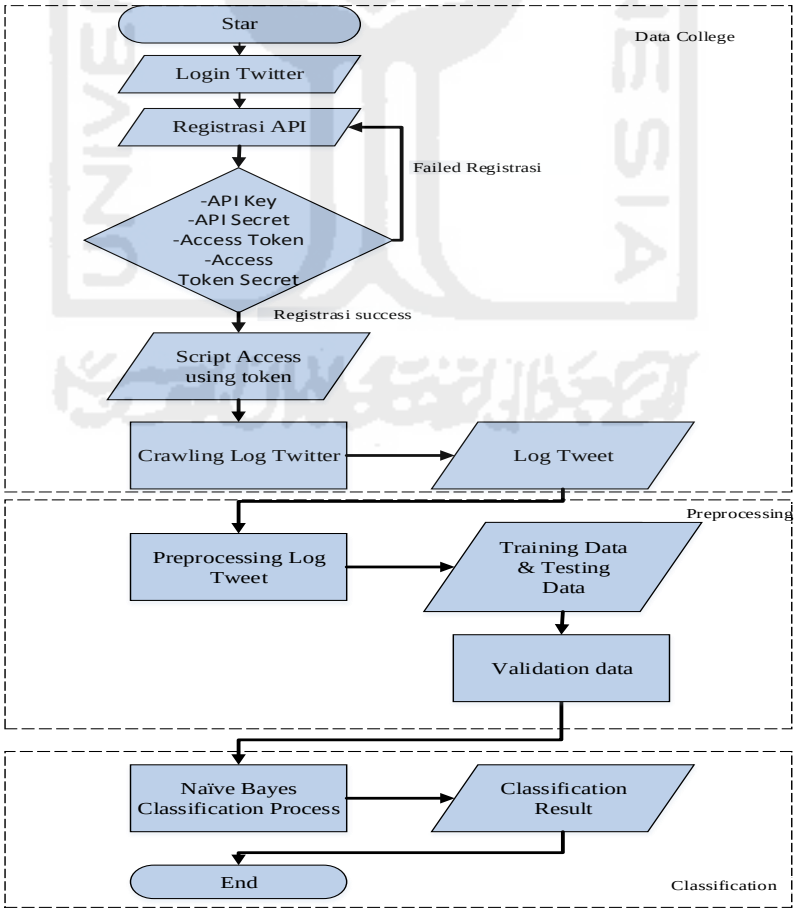
Bab 3 Metodologi Penelitian

3.1 Metodologi Penelitian

Penelitian ini terdiri dari tiga tahap yang pertama adalah teknik pengumpulan *Log* Data. *Log* data *Tweet* data dari jejaring sosial *Twitter* di *crawling* menggunakan API Twitter yang telah disediakan oleh *developer Twitter*. kedua adalah *preprocessing* data. *Log* Data yang telah di *crawling* dari twitter menghasilkan data mentah yang tidak terstruktur, *preprocessing* atau pembersihan data dilakukan agar data menjadi terstruktur dan memudahkan pada saat analisis, Ketiga adalah Klasifikasi, dari *log twitter* yang telah dibersihkan selanjutnya akan dirubah dalam bentuk vector untuk kemudian diklasifikasikan dengan metode *Naïve Bayes Classification* (NBC) menggunakan *Machine Learning* WEKA. *Flowchart* penelitian dapat dilihat pada **Gambar 3.1:**



Gambar 3. 1 Alur Penelitian

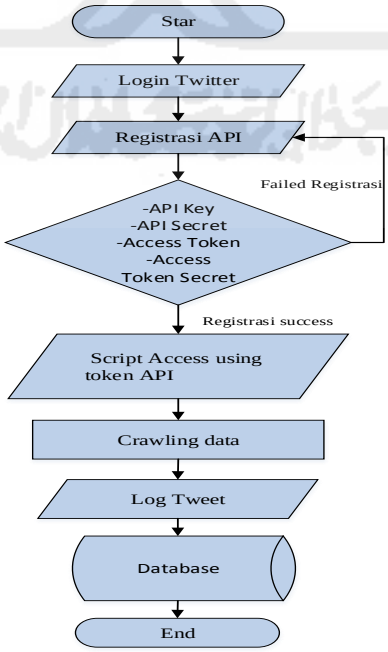
3.2 Perangkat Pendukung Penelitian

Untuk mendukung penelitian ini dibutuhkan beberapa perangkat keras maupun perangkat lunak diantaranya adalah:

1. Perangkat keras:
   a. Prosesor Intel(R) Core(TM) i5-3210M CPU @ 2.40GHz (4 CPUs)
   b. Hard Disk Drive 500 GB
   c. RAM 4 GB
2. Perangkat Lunak:
   a. Sistem operasi Windows 8.1 Pro 64-bit
   b. Notepad ++ v6.6.9 sebagai editor
   c. Anaconda 4.0.0
   d. WEKA 3.8.0
   e. Akun Twitter
   f. API Twitter
   g. Sastrawi master untuk stemming data
   h. Stopword Tala
   i. JsonLint untuk validasi file *Json*
   j. Convert csv untuk mengubah file *Json* ke CSV atau ke Excel

3.3 Pengumpulan Data *Log Tweet*

Proses pengambilan *Log Tweet* dapat dilihat pada alur dibawah ini:

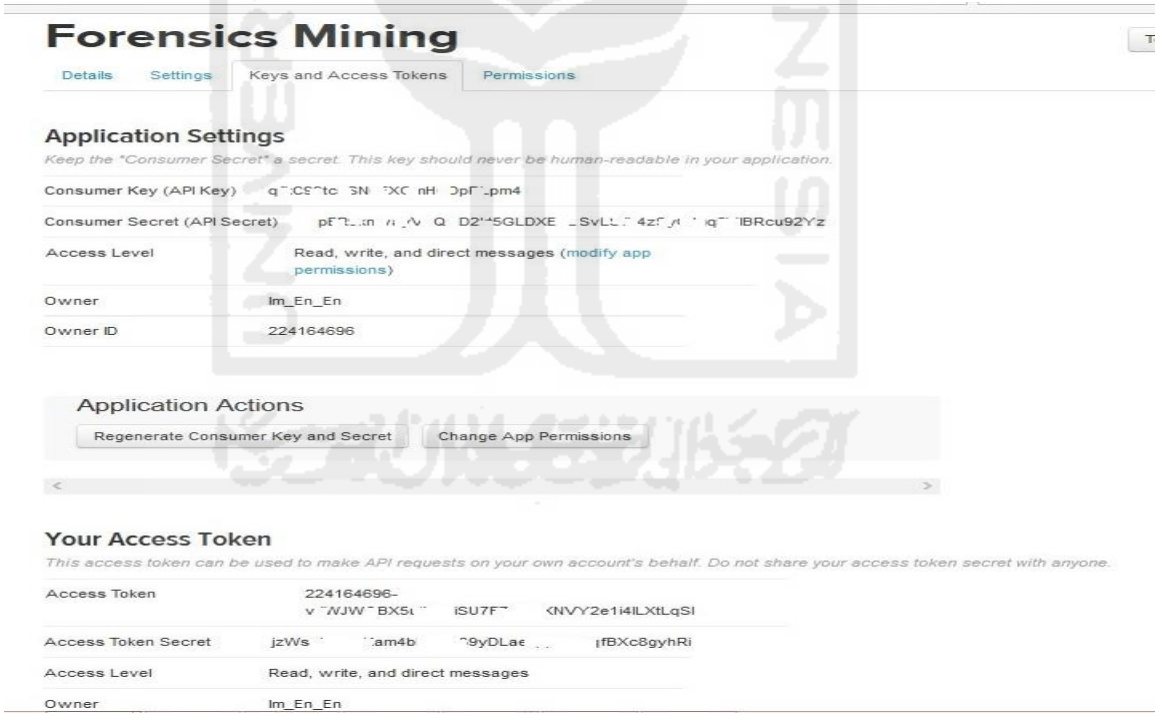


Gambar 3. 2 Teknik Pengumpulan Data

Proses pengumpulan data diawali dengan melakukan *Login* pada akun Twitter. Setelah *Login* lakukan daftar aplikasi untuk mendapatkan *access tokens* berupa *consumer_key*, *consumer_secret*, *access_token*, dan *access_secret*, hal ini dibutukan agar bisa mengakses *Twitter Search API*. Untuk bisa melakukan komunikasi antara *token Access* dan *Twitter Search API* maka dibuat sebuah *script* sebagai media untuk *crawling*. Meskipun demikian terdapat beberapa batasan dengan *Twitter Search API* diantaranya adalah:

a. Hanya dapat melakukan indeks pada 1500 *tweets* terakhir.
b. Data yang umurnya lebih dari seminggu tidak dapat di *crawling*
c. Pencarian data yang umurnya lebih dari seminggu maksimal 100 tweet
d. Pencarian dengan Query yang kompleks kemungkinan tidak berhasil

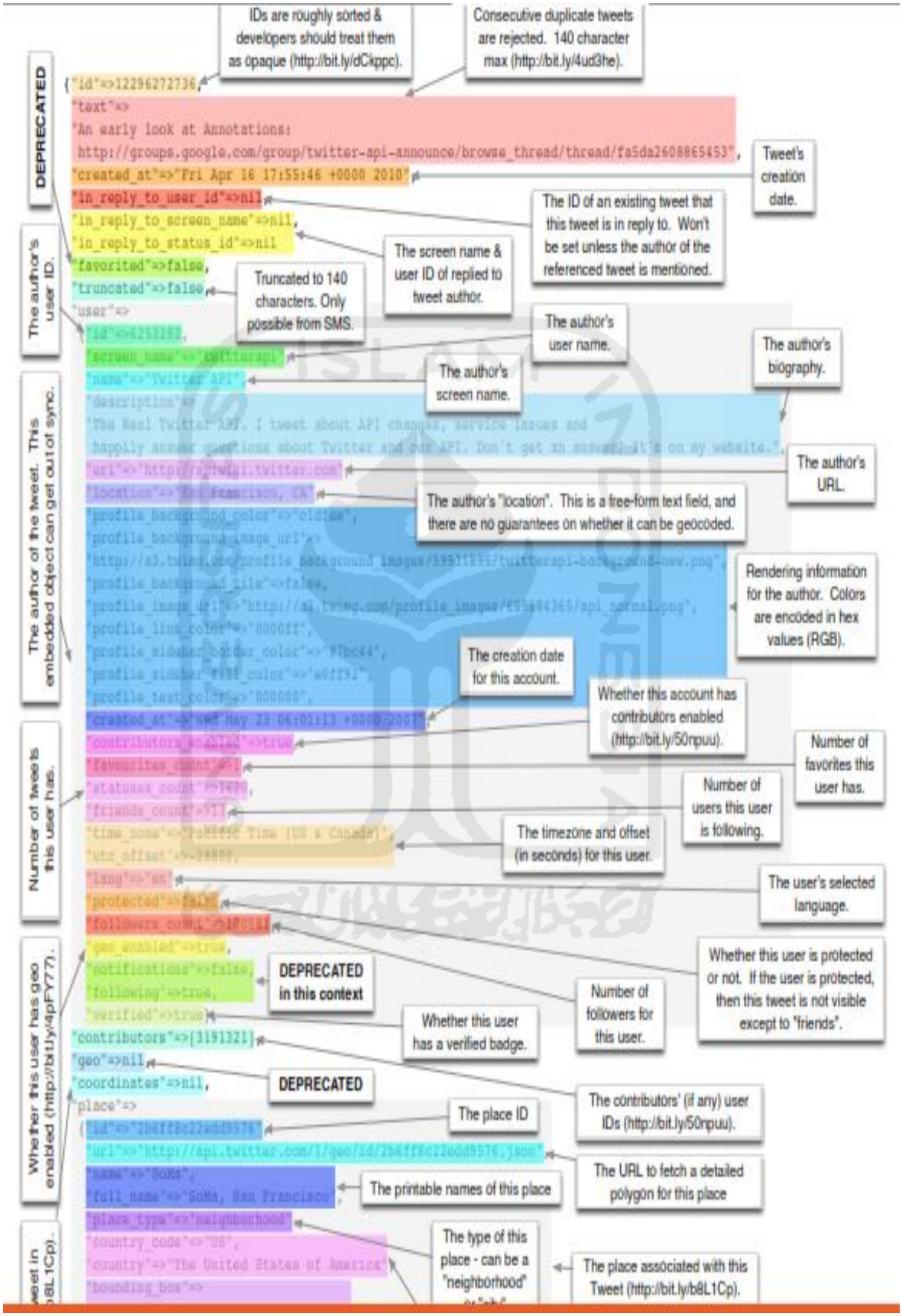
Dengan mengetahui batasan tersebut diatas maka dapat dilakukan pencarian data sesuai akses yang diberikan. Selanjutnya, setelah *token acssess* didapatkan maka dapat dilakukan pencarian data menggunakan *script* yang telah dibuat sebelumnya sesuai Query yang di inginkan. Pengumpulan data dilakukan secara random pada periode November-Desember 2016 sebanyak 1000 data, namun setelah melakukan filter data menjadi 583 data. **Gambar 3.3** dibawah adalah contoh daftar aplikasi untuk mendapatkan akses token.



Gambar 3. 3 Access Token

Untuk satu akun *Twitter* banyak objek yang bisa menjadi sumber Informasi, seperti ID unik, *text*, waktu pembuatan akun twitter, ID user yang me-*reply tweet author*, *biography user*, lokasi, Negara, bahkan sampai type lokasi user dapat diketahui, maka untuk memudahkan penelitian ini, dilakukan parsing data pada *script* sehingga saat *crawling* hanya

sebagian objek yang di ekstrak. Sebelum dilakukan parsing struktur data pada twitter terlihat seperti pada **Gambar 3.4** (Raffi Krikorian 2010).



Gambar 3. 4 Map Twitter

Namun setelah dilakukan parsing objek yang diekstrak menjadi beberapa objek diantaranya adalah *created_at*, *text*, *Location*, *followers*, *friends* dan *id_str*, seperti terlihat pada **Gambar 3.5** dibawah:


```
[{
    "id": 2634687793,
    "text": "Gua tau kok gua ini goblok,tolol,bego,Yakalo lu mau ngatain silahkan,Lebih baik gua munduur!\n\
    "friends": 161,
    "created_at": "2016-12-06 06:13:08",
    "location": "",
    "followers": 14
}, {
    "id": 724728729197375488,
    "text": "RT @teguh_har: puas puaaaas \ndasar otak dodol loe bukannya minta maaf malah ngenye,gara2 loe k
    "friends": 136,
    "created_at": "2016-12-06 05:59:20",
    "location": "",
    "followers": 21
}, {
    "id": 1624589670,
    "text": "Lu gak bisa nyepam lagi kan?? GOBLOK BANGET lu Njingg!! bruakakakakak :v @MekelSungg",
    "friends": 3,
    "created_at": "2016-12-06 05:49:55",
    "location": "Jakarta",
    "followers": 8
}, {
    "id": 2600084004,
    "text": "RT @jaewonfk: @exkjl ah goblok lu",
    "friends": 7737,
    "created_at": "2016-12-06 05:44:18",
    "location": "oky babu",
    "followers": 8089
```


Normal text file | length : 72,046 lines : 2,101 | Ln : 1 Col : 1 Sel : 0 | 0 | Windows (CR LF) | UTF-8 | INS

Gambar 3. 5 Hasil Ekstrak Data Setelah di Parsing

Proses pengumpulan data ini menggunakan beberapa *library* dalam pengambilan data *tweet* diantaranya adalah *library OAuth*, *Twitter REST API v1.1*, *Jsonpickle* dan *Tweepy*. *Library OAuth* digunakan untuk proses otentikasi sedangkan *Twitter REST API* digunakan untuk mengirimkan pesan kepada Twitter dan menerima *status update*, *Jsonpickle* digunakan untuk hasil pencarian data agar data tersebut dalam bentuk format *Json* sementara *Tweepy* digunakan untuk menghubungkan pemrograman yang digunakan ke Twitter. Untuk teknik pencarian data menggunakan operator pencarian. Metode ini menggabungkan kata-kata pencarian mencakup sinonim dan dikenal juga sebagai *Boolean Searching*. Pada metode ini memungkinkan untuk memasukkan banyak kata ataupun konsep dalam pencarian. Metode ini mengindikasikan hasil yang didapat berdasarkan operator “AND”, “OR”, dan ”NOT”.

Pada operator “AND”, misalkan untuk kata kunci pencarian data dengan kata bullying “Bangsat and Bajingan” akan menghasilkan data yang terdapat kata bangsat saja maupun bajingan saja. Untuk operator “OR”, misalkan untuk kata kunci pencarian data kata bullying “Bangsat or Bajingan” akan menghasilkan data yang berisikan bangsat saja, atau bajingan saja maupun data yang berisikan bajingan dan bangsat. Untuk operator “NOT”, misalkan

untuk kata kunci pencarian data kata bullying “Bangsat not Bajingan” akan menghasilkan data yang berisikan kata bangsat saja.

Keywords dalam pencarian data menggunakan kata yang sering digunakan untuk melakukan bullying misalnya kata, “bangsat”, “dasar monyet” dan lain-lain. Metode ini mengikuti peneliti terdahulu yang pada penelitiannya menganalisis *Gender Bullying* sehingga kata kunci hanya berdasar pada LGBT seperti kata “gay” dan “bitch” (Sanchez 2011). Sementara untuk *keywords* pencarian juga mengacu pada penelitian terdahulu yang mana penelitian tersebut terdapat kata-kata bullying yang banyak digunakan di Indonesia untuk melakukan bullying pada jejaring sosial (Margono, 2011). Meskipun *keywords* pencarian adalah kata-kata bullying namun tidak semua maksud dari kata tersebut adalah untuk membullying, misalnya pada kalimat berikut “siapa yang kamu sebut babi” dan “babi kamu”. Walaupun kedua kalimat ini menggunakan kata “babi” namun tidak semuanya bermakna untuk membullying seseorang.

Implementasi pengambilan data dilakukan dengan membuat *file script parsing* yang bertugas melakukan *cron job*, untuk penelitian ini menggunakan *python*. **Gambar 3.6** merupakan kode program proses *query* pengaksesan data ke Twitter yang telah di parsing.

```
import tweepy,sys,jsonpickle

consumer_key = 'qfkC99tcnSN0FXQmHCDpRLpm4'
consumer_secret = 'pER9knwayVPQQD2kt5GLDXEHDSvLSfb4zSyGJgq5YIBRcu92Yz'

#inisialisasi
qry='bangsat’
maxTweets = 300
tweetsPerQry = 100
fName='Parse_Data_tweet.json'

#Proses Parsing
parseTweets=[]

#getData
auth = tweepy.AppAuthHandler(consumer_key,consumer_secret)
api = tweepy.API(auth, wait_on_rate_limit=True,wait_on_rate_limit_notify=True)
if (not api):
    sys.exit('Autentikasi gagal, cek "Consumer Key" & "Consumer Secret" Twitter anda')
parseTweets = []
#inisialisation
sinceId=None;max_id=-1;tweetCount=0

print("Mulai mengunduh maksimum {0} tweets".format(maxTweets))
with open(fName,'w') as f:
    while tweetCount < maxTweets:
```

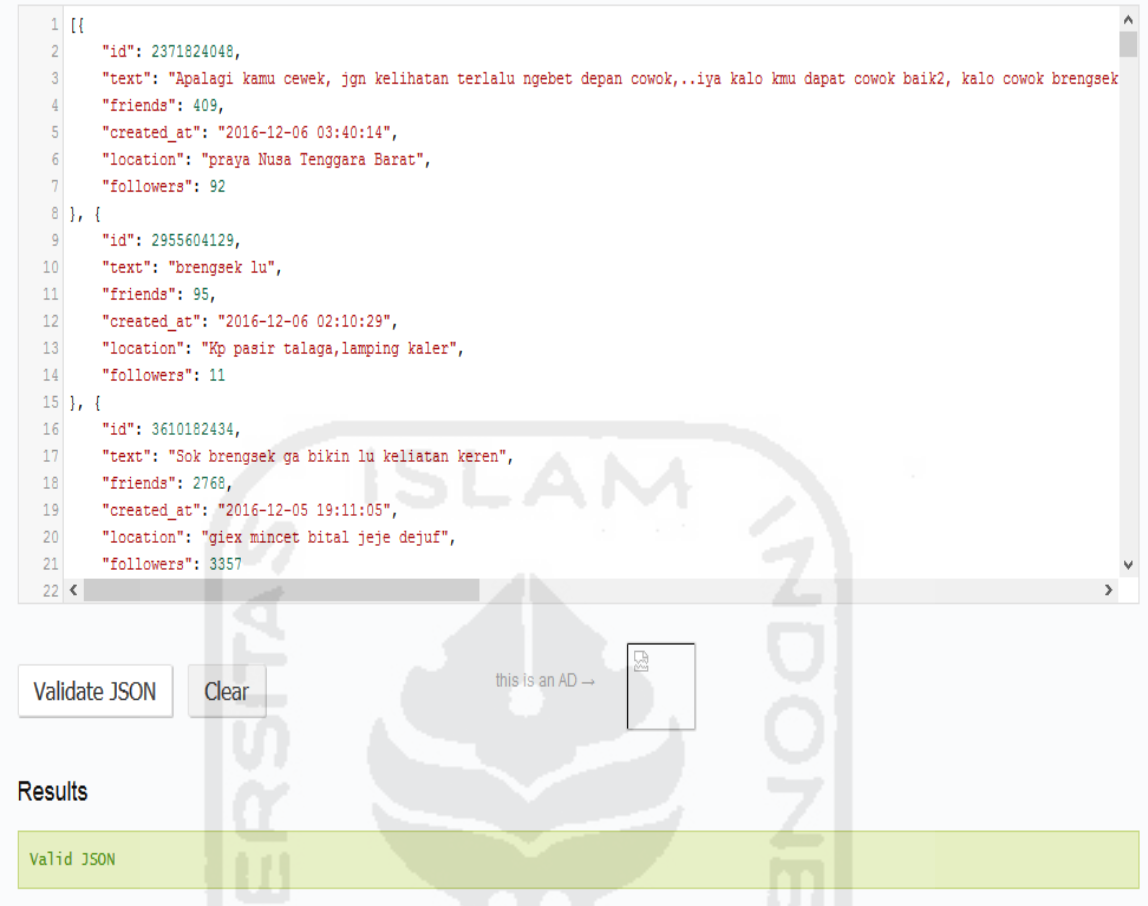
Gambar 3. 6 Script Program Pengumpulan Data

```
        try:
        if (max_id <= 0):
        if (not sinceId):
            new_tweets=api.search(q=qry,count=tweetsPerQry)
          else:
            new_tweets=api.search(q=qry,count=tweetsPerQry,since_id=sinceId)
        else:
          if (not sinceId):
            new_tweets=api.search(q=qry,count=tweetsPerQry,max_id=str(max_id - 1))
          else:
            new_tweets=api.search(q=qry,count=tweetsPerQry,max_id=str(max_id -
1),since_id=sinceId)
        if not new_tweets:
          print('Tidak ada lagi Tweet ditemukan dengan Query="{0}'".format(qry));break
        for tweet in new_tweets:
         # f.write(jsonpickle.encode(tweet._json,unpicklable=False)+'\n')
          userTweet = {}
          userTweet = tweet.user
          parseTweets.append({
                    "id" : userTweet.id,
                    "location" : userTweet.location,
                    "text" : tweet.text,
                    "created_at" : tweet.created_at,
                    "followers": userTweet.followers_count,
                    "lang": userTweet.lang,
                    "friends": userTweet.friends_count
                    })

        f.write(jsonpickle.encode(parseTweets,unpicklable=False))

        tweetCount+=len(new_tweets)
        sys.stdout.write("\r");sys.stdout.write("Jumlah Tweets yang tersimpan: %.0f"
%tweetCount);sys.stdout.flush()
        max_id=new_tweets[-1].id
      except tweepy.TweepError as e:
        print("some error : " + str(e));break
print ('\nSelesai! {0} tweets tersimpan di "{1}'".format(tweetCount,fName))
```

Gambar 3.6 Script Program Pengumpulan Data (Lanjutan)

Proses pengambilan data *tweet* dilakukan dengan memanggil fungsi *search* dari *library twitter*. Namun sebelum proses pencarian dilakukan, terlebih dahulu dideklarasikan variabel *$consumerKey*, *$consumerSecret*, *$accessToken*, *$accessTokenSecret*. Variabel *$consumerKey* dan *$consumerSecret* berisi *OAuth setting* aplikasi yang didaftarkan ke Twitter. Variabel *$accessToken* dan *$accessTokenSecret* merupakan akses token untuk mengakses Twitter. Hasil yang diperoleh dalam proses ini yaitu berupa file dalam bentuk file *Json*. Untuk memastikan file tersebut file *Json* maka peneliti melakukan verifikasi secara

online menggunakan *JsonLint*, **Gambar 3.7** dibawah adalah contoh verifikasi *Json* menggunakan *Jsonlint:*



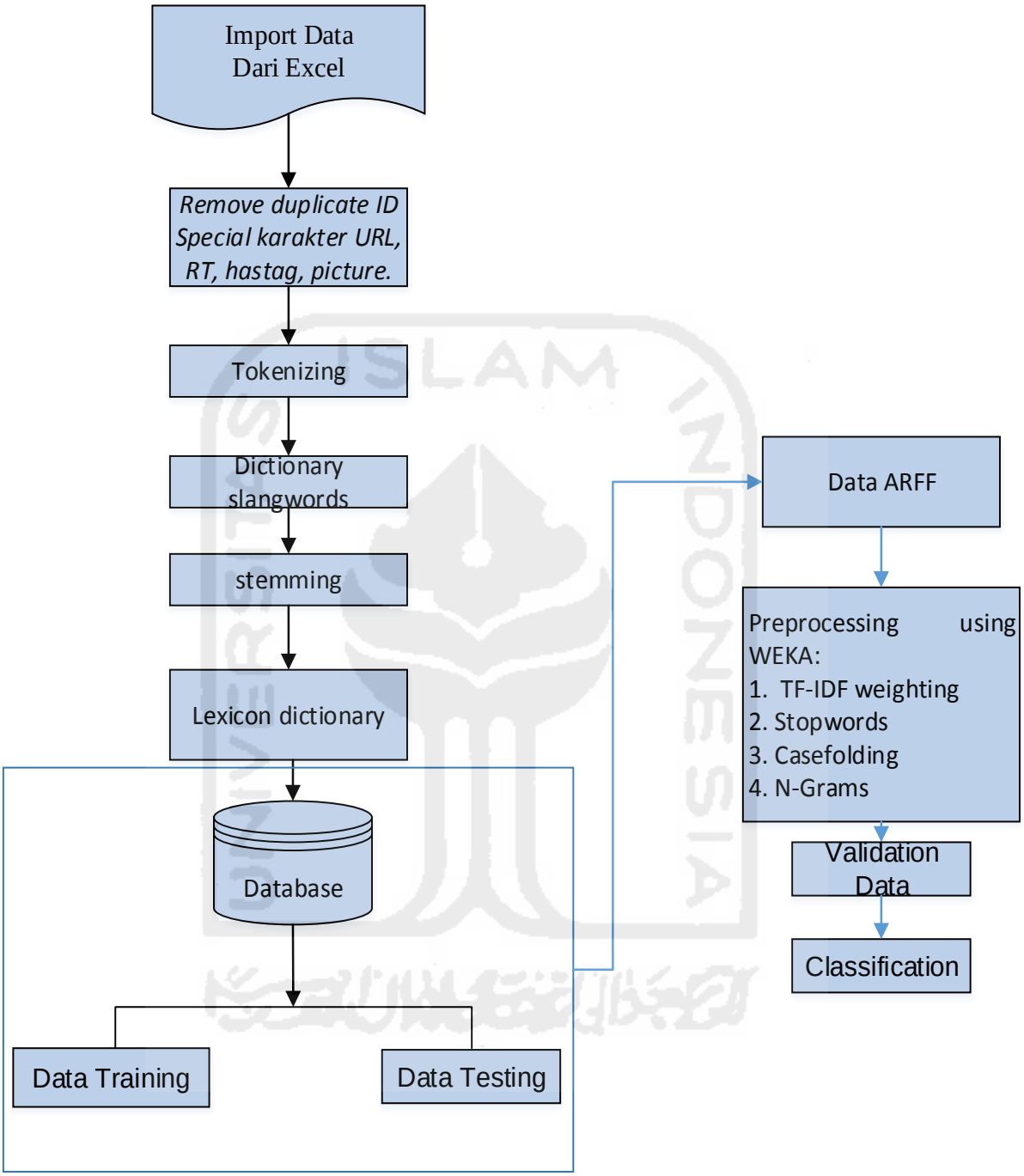
Gambar 3. 7 Hasil Validasi file Json Menggunakan JsonLint

Aplikasi ini tersedia di *http://jsonlint.com/*, setelah melakukan verifikasi selanjutnya data di *convert* dalam bentuk .csv atau excel agar lebih mudah dalam *cleansing* data, pada penelitian ini menggunakan *convert* .csv secara online dan tersedia di *http://www.convertcsv.com/json-to-csv.htm*, selanjutnya data disimpan ke dalam Database untuk diolah lebih lanjut.

3.4 Preprocessing Data

Data yang telah diubah dalam bentuk .csv selanjutnya dilakukan pembersihan atau *preprocessing*. Hal ini dilakukan agar mendapatkan data terstruktur yang mudah diolah baik secara manual maupun menggunakan *machine learning*. *preprocessing* ini terdiri dari penghapusan ID duplicate, penghapusan URL, penghapusan karakter khusus *hastag*, *RT* dan gambar, Normalisasi dari kata tidak baku menjadi kata baku, *tokenizing* yaitu memecah kalimat menjadi kata, *casefolding* yang merupakan pengubahan semua kalimat dalam bentuk huruf kecil, *stopword* yaitu membuang akhiran kata *possessive* seperti –kah, -lah, -pun,

Stemming atau pencarian akar kata, dan *N-grams*. Alur dari *preprocessing* ini dapat dilihat pada **Gambar 3.8** berikut:



Gambar 3. 8 Teknik Preprocessing Data

3.4.1 Menghapus Special Karakter

Pada tahap ini dilakukan proses penghapusan karakter-karakter yang dapat mengganggu proses analisis, baik data training maupun data testing. Karakter yang dihapus adalah Duplicate ID, URL, RT, Gambar, *hastag* dan *special character* lainnya seperti tanda baca koma, kurung dll. Proses penghapusan dilakukan secara manual menggunakan Excel dengan

metode *fine* dan *replace* (Saputra 2015). **Tabel 3.1** menunjukkan penghapusan dengan *Fine* dan *Replace*:

Tabel 3. 1 Tabel Find dan Replace

Yang dihapus	Find	Replace	Keterangan
URL	http*[spasi]	[spasi]	Link di depan
URL	[spasi]http*[spasi]	[spasi]	Link ditengah
URL	[spasi]http*	[spasi]	Link dibelakang
Gambar	Pic.Twitter*[spasi]	[spasi]	Gambar didepan
Gambar	[spasi]pic.Twitter*	[spasi]	Gambar dibelakang
Gambar	[spasi]pic.twitter*[spasi]	[spasi]	Gambar ditengah
@	@*[spasi]	[spasi]	Akun didepan
@	[spasi]@*	[spasi]	Akun dibelakang
@	[spasi]@* [spasi]	[spasi]	Akun ditengah
#	#*[spasi]	[spasi]	Hastag didepan
#	[spasi]#*	[spasi]	Hastag ditengah
#	[spasi]#* [spasi]	[spasi]	Hastag dibelakang

3.4.2 Normalisasi kalimat

Normalisasi kalimat di perlukan untuk menyetarakan kata pada kalimat. Adapun langkah-langkah dalam Normalisasi adalah sebagai berikut:

a. Tokenizing

Pada tahap ini dilakukan secara manual pada Excel dengan cara mengganti spasi menjadi koma. *Tokenizing* merupakan pemotongan string input berdasarkan tiap kata menyusunnya. Sebagai contoh pada *tweet*:

"Kamu fikir nemu dompet orang di jalan itu berkah Tolol" akan di pecah menjadi kata kamu, kata fikir, kata nemu, kata dompet, kata orang dan seterusnya. Setelah dilakukan tokenizing selanjutnya akan mudah dilakukan normalisasi kalimat dari kata tidak baku menjadi baku atau kata *slang* menjadi baku dengan merujuk pada Kamus Besar Bahasa Alay (KBBA).

b. Kamus KBBA

Komentar yang diberikan seseorang tidak semuanya bahasa baku, banyak sekali yang menggunakan bahasa gaul, misalnya: "gue", "loe" dan lain-lain, serta tidak jarang pula yang menggunakan potongan kata, misalnya: "yg", "brp","bgm" dan lain-lain. Kata yang tidak dinormalisasi lebih dahulu akan dikenali oleh *machine learning* sebagai kata yg berbeda, misalnya: 'semoga' dan 'smoga' yang seharusnya memiliki makna yang sama akan menjadi

beda makna dikarenakan penulisannya yang berbeda. Untuk itu dilakukan normalisasi kata dari yang tidak baku menjadi baku. Untuk normalisasi ini menggunakan bantuan kamus KBBA. Dibawah ini adalah Contoh tabel dari kata tidak baku menjadi kata baku:

Tabel 3. 2 Contoh kata tidak baku menjadi baku

Kata Tidak Baku	Kata Baku
Brp	Berapa
Sm	Sama
Njir	Anjing
Syg	Sayang
Klw, Low, Klo	Kalau
Kamuh, Kamyu, ello, elu	Kamu
Aj	Saja
Nyet	Monyet

c. Penggunaan Rumus

Rumus yang digunakan untuk mengganti kata tidak baku menjadi kata baku adalah =IF(ISNA(VLOOKUP('data training'!$1:$1048576,KBBA!$1:$1048576,2,FALSE))=TRUE,'data training'!$1:$1048576,VLOOKUP('data training'!$1:$1048576,KBBA!$1:$1048576,2,FALSE)).

Keterangan:

Data Training = tabel data training berisi kalimat yang dipecah menjadi kata

KBBA = nama table yang berisi kata tidak baku dan kata baku

ISNA = rumus yang digunakan untuk mengatasi output berupa #N/A yang artinya Not Available, sehingga tidak perlu dilakukan penghapusan satu persatu.

IF = adalah fungsi (kondisi jika benar, jika salah).

VLOOKUP = rumus yang berfungsi untuk mencari kolom pertama dalam satu rentang sel, kemudian mengembalikan nilai apapun yang ada pada baris yang sama.

Penjelasan rumus:

Apabila kata pada workseet pada table Data training $1 sampai $1048576 tidak terdapat satupun pada workseet KBBA dari rentang $1 sampai $1048576, maka kata tidak diubah

menjadi kolom ke-2 atau kolom B pada workseet KBBA, melainkan akan dikeluarkan output berupa #N/A, karena dituliskan rumus ISNA, maka kata akan dikembalikan seperti semula atau kata tidak terjadi perubahan. Dan apabila kata tersebut terdapat pada workseet KBBA pada kolom $1 sampai $1048576, maka kata tersebut akan diubah menjasi kata yang terdapat pada kolom berikutnya atau ke-2 pada workseet KBBA.

3.4.3 Stemming

Proses ini adalah tahap mencari akar kata dari tiap kata hasil *filtering*. Proses ini mengubah kata menjadi kata dasarnya dengan menghilangkan imbuhan-imbuhan pada kata dalam dokumen atau mengubah kata kerja menjadi kata benda. *Stem* (akar kata) adalah kata inti setelah imbuhan dihilangkan (awalan dan akhiran). Misalnya kata "perancangan" dan "merancang" akan diubah menjadi sebuah kata yang sama, yaitu "rancang". Proses stemming sangat tergantung kepada bahasa dari kata yang akan di-stem.

Penelitian ini menggunakan Sastrawi Master. Sastrawi master adalah *library* php sederhana yang menyediakan *stemming* kata bahasa Indonesia. kamus kata dasar yang digunakan Sastrawi berasal dari kateglo.com dengan sedikit perubahan dan masing-masing mempunyai lisensi Sastrawi dan lisensi kateglo. Sastrawi dapat diunduh secara gratis di alamat https://github.com/sastrawi/sastrawi.

Untuk melakukan proses ini peneliti menggunakan bahasa pemrograman python. Namun terlebih dahulu install Library master sastrawi pada python, kemudian buat *script* pada *console* python seperti berikut:

```
# import StemmerFactory class
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
# create stemmer
factory = StemmerFactory()
stemmer = factory.create_stemmer()
# stemming process
sentence = 'kamu memang politisi kampungan yang sedang membela tuannya'
output   = stemmer.stem(sentence)
print(output)
# hasil stemming
# “kamu memang politis kampungan yang sedang bela tuan”
```

Gambar 3. 9 Script Stemming

3.4.4 Penggunaan Lexicon

Penggunaan lexicon pada prosesnya sama dengan tahap normalisasi cleansing, yaitu *preprocessing* data menggunakan excel. Tetapi perbedaanya adalah pada workseet yang berisi kamus, pada normalisasi kolom A berisi kata tidak baku dan pada kolom B berisi kata baku, sedangkan pada proses pemanfaatan lexicon ini, kolom A berisi kamus bullying, pronoun atau kata ganti orang kedua dan ketiga misanya, “kamu”, “kau”, “anda” dan kamus negasi seperti “bukan” dan “tidak”, sementara kolom B berisi kata bullying yang saya ubah menjadi “badword” untuk kamus bullying dan kata pronoun untuk kamus pronoun dan kata negasi untuk kamus negasi. **Gambar 3.10** dibawah adalah Contoh kamus lexicon.

	A	B
16	menteri	pronoun
17	presiden	pronoun
18	walikota	pronoun
19	gubernur	pronoun
20	mereka	pronoun
21	bapak	pronoun
22	ibu	pronoun
23	kalian	pronoun
24	perempuan	pronoun
25	tidak	negasi
26	bukan	negasi
27	sableng	badword
28	gila	badword
29	jelek	badword
30	edan	badword
31	tolol	badword
32	bego	badword
33	geblek	badword
34	goblok	badword
35	dongo	badword
36	bodoh	badword
37	buta	badword
38	tuli	badword
39	sableng	badword

Gambar 3. 10 Kamus Lexicon

Rumus yang digunakan sama dengan rumus pada proses Normalisasi kata tidak baku menjadi kata baku. **Gambar 3.11** dibawah merupakan contoh data sebelum penggunaan lexicon:

1 | 2383283436

E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
396	brengsek	kamu	hebat	juga	gitu	saja	pakai	tanya								
1565	brengsek	kamu	hebat	juga	gitu	saja	pakai	tanya								
941179	brengsek	kamu	hebat	juga	gitu	saja	pakai	tanya								
122	kamu	terlalu	brengsek	buat	sayang	sama	dia	ohseh								
8	brengsek	juga	diko	main	sendiri	saja	kamu	modus	sempat	dasar	brengsek					
138	dasar	brengsek														
216	dasar	brengsek														
2825	arti	kamu	juga	alien	gayung	geblek										
97	jakarta	bekasi	tiga	seperdua	jam	dasar	geblek									
278	geblek	otak	kamu	balik												
514	geblek	kamu														
1253	saya	baca	dan	kamu	geblek											
735	tai	kontok	kamu	omong	apa	geblek	kutil									
1671	geblek	memang	jidat	kamu	jong	yang	jendol									
146	apa	apa	pc	gaya	banget	kamu	pc	orang	geblek	malah	nanya	saya	memang	saya	pacar	galer
1239	maksimal	jek	tidak	mungkin	positif	geblek	kamu									
214	dasar	gila														
1915	dasar	cewek	gila	memang												
3474	gila	kamu	allahu													
8892	kamu	gila														
322	malique	respect	gila	kamu												
521	bodor	gila	pokok	kamu	asik	cinta	kamu									
8646	ngapain	kamu	gila													
685	gila	baik	banget	kamu												

Gambar 3. 11 Data Sebelum Penggunaan Lexicon

Setelah menggunakan lexicon data tersebut menjadi seperti pada **Gambar 3.12** berikut:

D95 | =IF(ISNA(VLOOKUP('data 583 lengkap'!$1:$1048576,Kamus_bullying!$1:$1048576,2,FALSE))=TRUE,'data 583 lengkap'!$1:$1048576,VLOOKUP('data 583

	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
158	396	badword	pronoun	hebat	juga	gitu	saja	pakai	tanya								
159	1565	badword	pronoun	hebat	juga	gitu	saja	pakai	tanya								
160	941179	badword	pronoun	hebat	juga	gitu	saja	pakai	tanya								
161	122	pronoun	terlalu	badword	buat	sayang	sama	pronoun	ohseh								
162	8	badword	juga	diko	main	sendiri	saja	pronoun	modus	sempat	dasar	badword					
163	138	dasar	badword														
164	216	dasar	badword														
165	2825	arti	pronoun	juga	badword	gayung	badword										
166	97	jakarta	bekasi	tiga	seperdua	jam	dasar	badword									
167	278	badword	otak	pronoun	balik												
168	514	badword	pronoun														
169	1253	saya	baca	dan	pronoun	badword											
170	735	badword	kontok	pronoun	omong	apa	badword	kutil									
171	1671	badword	memang	jidat	pronoun	jong	yang	jendol									
172	146	apa	apa	pc	gaya	banget	pronoun	pc	orang	badword	malah	nanya	saya	memang	saya	pacar	galer
173	1239	maksimal	jek	negasi	mungkin	positif	badword	pronoun									
174	214	dasar	badword														
175	1915	dasar	pronoun	badword	memang												
176	3474	badword	pronoun	allahu													
177	8892	pronoun	badword														
178	322	malique	respect	badword	pronoun												
179	521	bodor	badword	pokok	pronoun	asik	cinta	pronoun									
180	8646	ngapain	pronoun	badword													
181	685	badword	baik	banget	pronoun												

Gambar 3. 12 Data Setelah Penggunaan Lexicon

3.4.5 Data training dan Data testing

Setelah penggunaan lexicon pada data, maka data dapat dibagi menjadi dua bagian yaitu data training dan data testing. Metode pembagian data ini dibagi seimbang yaitu 50% data training dan 50% untuk data testing karena data yang tidak seimbang klasifikasi yang dibangun

memiliki kecenderungan untuk mengabaikan *minority class* (Buntoro 2016). Selanjutnya data di ubah ke format ARFF, Namun untuk data training sebelum di ubah ke format ARFF terlebih dahulu dilakukan pelabelan secara manual berdasarkan pattern atau pola yang mengindikasikan bahwa kalimat tersebut mengandung bullying. Tabel dibawah menunjukkan pola bahwa suatu kalimat mengandung bullying jika terdapat unsur sebagai berikut (Yin 2009):

Tabel 3. 3 Pola Cyberbullying

BadWord!	Pronoun	…	…
Kamu	BadWord	…	**...**
…	BadWord	Pronoun	…
Pronoun	BadWord	….	….
Pronoun	….	BadWord	…

Tetapi untuk kalimat yang menggunakan kata negasi diikuti kata BadWord maka kalimat tersebut menjadi negatif bullying. Demikian juga kalimat yang mengandung unsur pertanyaan disertai pronoun dan BadWord maka kalimat tesebut bernilai negatif bullying. Tabel dibawah merupakan pola kalimat bullying yang disertai negasi dan Question.

Tabel 3. 4 Pola Negasi

Negasi	Badword	…	…
Question	Pronoun	BadWord	…

Setelah dilakukan pelabelan secara manual untuk data training, selanjutnya adalah mengubah file menjadi ARFF. Proses perubahan data bisa dilakukan secara manual maupun otomatis. Perubahan secara manual dilakukan dengan cara data diubah ke.txt terlebih dahulu kemudian menambahkan @relation untuk nama datanya, @attribute berupa text type data string, @attribute @@class@@ {positif,negatif} merupakan kelas atribut berupa positif, dan negatif kemudian @data yang berisi datanya yang ditambahkan single quote dan dilabeli “pos” untuk kalimat positif, “neg” untuk kalimat negatif lalu file .txt di save dengan ekstension ARFF. Sementara pengubahan secara otomatis dilakukan dengan cara mengubah file ke bentuk .CSV, lalu buka tools WEKA, open file, setelah data terbuka save as kembali dengan mengubah type data .CSV menjadi ARFF.

3.4.6 Pengolaan Data Menggunakan WEKA

Weka adalah aplikasi data mining open source berbasis Java. Aplikasi ini dikembangkan pertama kali oleh Universitas Waikato di Selandia Baru sebelum menjadi bagian dari Pentaho. Weka terdiri dari koleksi algoritma machine learning yang dapat digunakan untuk melakukan generalisasi / formulasi dari sekumpulan data sampling. Walaupun kekuatan Weka terletak pada algoritma yang makin lengkap dan canggih, kesuksesan data mining tetap terletak pada faktor pengetahuan manusia implementornya. Tugas pengumpulan data yang berkualitas tinggi dan pengetahuan pemodelan dan penggunaan algoritma yang tepat diperlukan untuk menjamin keakuratan formulasi yang diharapkan (susanto 2012). Penelitian ini menggunakan WEKA versi 3.8.0 Gambar dibawah adalah *Interface* dari WEKA 3.8:

Gambar 3. 13 Interface WEKA

Preprocessing menggunakan WEKA dilakukan dengan cara sebagai berikut:

a. Mengubah Data ke dalam bentuk Vektor

Pada tahap ini, data yang berupa kalimat yang sudah dilabeli dengan positif dan negatif akan diubah kedalam bentuk vector. Adapun caranya adalah pada aplikasi WEKA open file dan arahkan ke file .arff yang sudah diolah sebelumnya, setelah itu tekan tombol choose pada filter dan pilih StringToWordVector. Kemudian filters, Unsupervised, attribute kemudian StringToWordVector lalu Apply. Hal ini dilakukan pada data training maupun data testing.

Pada bentuk vector ini, masing-masing token mewakili satu attribute, contoh untuk data training 50%, data yang sudah diubah kedalam vector dengan jumlah data positif bullying sebanyak 226 dan data negatif bullying sebanyak 66 sehingga totalnya menjadi 292 data training.

b. Pembobotan TF-IDF

Proses pengubahan data teks menjadi data vektor dilakukan dengan membaca kata satu persatu dan menghitung nilai tf-idf. Nilai tf-idf adalah kemunculan kata (*term frequency*) dalam kalimat dikalikan log jumlah dokumen/*record* dibagi jumlah dokumen/*record* yang mengandung kata yang dimaksud.

Pada penelitia ini Pembobotan TF-IDF dan preprocessing dengan menggunakan WEKA dilakukan dengan cara klik text box yang berisikan StringToWordVector. Setelah muncul gambar (weka.gui.genericobjecteditor), lakukan pilihan sesuai dengan preprocessing yang akan dilakukan, seperti *casefolding,Token N-gram,Penggunaan Stopword* dan penghapusan emoticon lalu apply.

c. Stopword Removal

Stopwords removal adalah proses menghilangkan kata-kata yang umum digunakan dan tidak mempunyai informasi yang berharga pada suatu konteks. Kamus stopwords yang digunakan berasal dari (Tala 2003) yang diunduh disitus http://hikaruyuuki.lecture.ub.ac.id/kamus-kata-dasar-dan-stopword-list-bahasa-indonesia/. Contohnya dapat dilihat pada table dibawah:

Tabel 3. 5 Tabel Stopword Tala

Ada
adalah
adanya
adapun
agak
agaknya
agar
akan
akankah
akhir
akhiri
akhirnya
aku
akulah
amat
amatlah
sampaikan
sana
sangat
sangatlah
satu
…dan seterusnya

Untuk menggunakan Stopword Tala Bahasa Indonesia dilakukan dengan cara melakukan klik pada tulisan “weka-3-8-0” kemudian memilih stopword yang akan digunakan. untuk mengubah semua huruf kecil dengan memilih “true” pada lowercasetoken. Untuk melakukan normalisasi panjang dokumen terhadap seluruh data dengan memilih “normalize all data” pada normalizeDocLength.

d. N-Gram

Penelitian ini mengimplementasikan tokenisasi N-Gram yang tidak terikat dengan satu aturan bahasa apapun, Tokenisasi menggunakan N-Gram adalah tahap pemrosesan dimana teks input dibagi menjadi unit-unit kecil yang disebut *token* sepanjang n karakter. Dalam bahasa Indonesia, frasa dengan satu kesatuan arti memiliki maksimal 3 kata, pembagian *token* dibagi menjadi Unigram, Bigram, Trigram dan N-Gram, berikut contoh pemecahan pada kalimat “orang pada buta semua pendukung semu”.

Unigram: yaitu *token* yang terdiri dari hanya satu kata, menghasilkan: “orang”,”pada”,”buta”,”semua”,”pendukung”,”semu”.

Bigram: yaitu *token* yang terdiri dari dua kata, menghasilkan: “orang pada”,”pada buta”,”buta semua”,”semua pendukung”,”pendukung semu”.

Trigram: yaitu *token* yang terdiri dari tiga kata, menghasilkan: “orang pada buta”,”pada buta semua”,”buta semua pendukung”,”semua pendukung semu”.

Proses N-gram pada penelitian ini juga menggunakan *machine learning* WEKA.

Cara penggunaan N-Gram pada WEKA adalah pilih NGramTokenizer dengan cara klik pada tombol “choose” pada tokenizer, kemudian pilih Ngramtokenizer. Selanjutnya memecah kata dengan mengubah angka pada NgramMaxSize dan NgramMinSize yang terdapat pada gambar. Untuk ungram NgramMaxSize diubah menjadi 1 dan NGramMinSize menjadi 2. Kemudian Ngram dengan mengubah angka Ngrammaxsize menjadi 3 dan Ngrammaxsize menjadi 1. Selanjutnya mengubah delimeter, dan menghapus emoticon.

e. Validation Data

Penerapan untuk classifier akan diuji sesuai dengan pilihan yang ditetapkan dan sesuai kebutuhan penelitian. Ada beberapa test option yang bias dipilih pada WEKA sebelum melakukan klasifikasi yaitu:

1. Use training set

   Pengetesan dilakukan dengan menggunakan data training itu sendiri.

2. Supplied test set

Pengetesan dilakukan dengan menggunakan data lain. Dengan menggunakan option inilah, bisa dilakukan prediksi terhadap data tes.

3. Cross-validation

   Pada cross-validation, akan ada pilihan berapa fold yang akan digunakan. Nilai default-nya adalah 10. Mekanisme-nya adalah sebagai berikut : Data training dibagi menjadi k buah subset (subhimpunan). Dimana k adalah nilai dari fold. Selanjutnya, untuk tiap dari subset, akan dijadikan data tes dari hasil klasifikasi yang dihasilkan dari k-1 subset lainnya. Jadi, akan ada 10 kali tes. Dimana, setiap datum akan menjadi data tes sebanyak 1 kali, dan menjadi data training sebanyak k-1 kali. Kemudian, error dari k tes tersebut akan dihitung rata-ratanya.

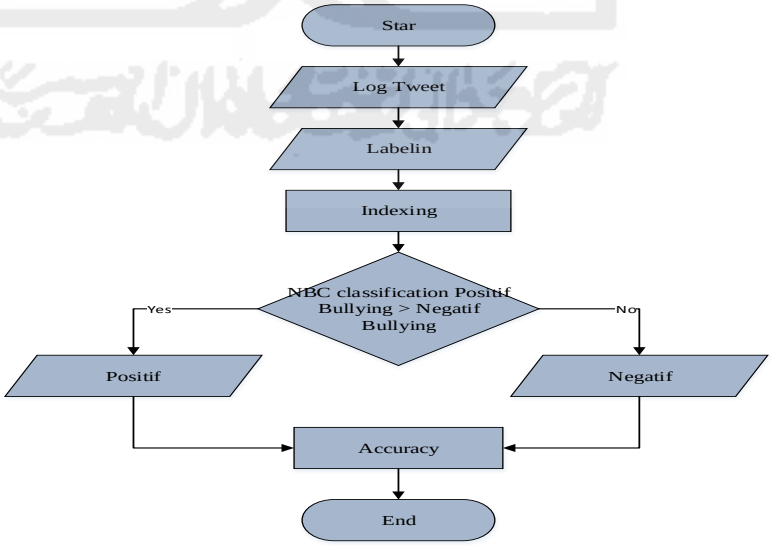
4. Percentage split

   Hasil klasifikasi akan dites dengan menggunakan k% dari data tersebut. k merupakan masukan dari user.

Dalam penelitian ini, digunakan Cross-validation dengan nilai default 10 sehingga disebut juga *10 cross validation folds*. Tujuan penggunaan model ini adalah untuk menentukan pola untuk data testing terhadap data training tujuannya untuk membatasi masalah *overfitting* dan mengetahui bahwa model ini mengenaralisasi data pada data testing untuk mendapatkan hasil klasifikasi.

3.5 Klasifikasi

Dalam menentukan akurasi dengan menggunakan metode Naïve bayes, dilakukan berdasarkan probabilitas kemunculan kata. Gambar 3.14 dibawah adalah alur dari metode Naive Bayes Classifier:



Gambar 3. 14 Flowchart Naive Bayes Classifier

Data teks yang digunakan adalah data bersih yang telah melalui preprocessing, selanjutnya diberi label secara manual pada data training, setelah itu dilakukan pembobotan TF-IDF, dan validasi data menggunakan *10 fold cross validation* lalu klasifikasi menggunakan Naïve Bayes, untuk teks yang positif cyberbullying akan di klasifikasikan ke class positif bullying dan teks yang negatif bullying akan ke class negatif bullying. Demikian pula untuk jenis cyberbullying akan diklasifikasikan ke class masing-masing Seperti jenis bullying yang *related psychology* akan diklasifikasikan ke *class related psychology* dan seterunya untuk jenis cyberbullying yang lain. Keseluruhan proses ini dilakukan pada *machine learning* WEKA.

Penentuan probabiltas positif bullying dan negatif bullying secara manual dapat dilihat pada Contoh, untuk jumlah data training adalah 292, untuk data positif bullying sebanyak 226 dan negatif bullying sebanyak 66:

- Probabilitas data positif P(Y=positif)=226/292 = 0,77
- Probabilitas data negatif P(Y=negatif)=60/292 = 0,20
- 292 merupakan jumlah seluruh data (226+60).

Proses selanjutnya yaitu set data testing pada WEKA, data testing ini juga merupakan data bersih, dari pola yang telah di proses pada data training akan mengikuti pola untuk data testing sehingga hasil klasifikasi dapat diprediksi. Untuk mengetahui hasl klasifikasi keseluruhan dapat dilihat pada hasil klasifikasi data training kemudian tambahkan pada hasil prediksi.