

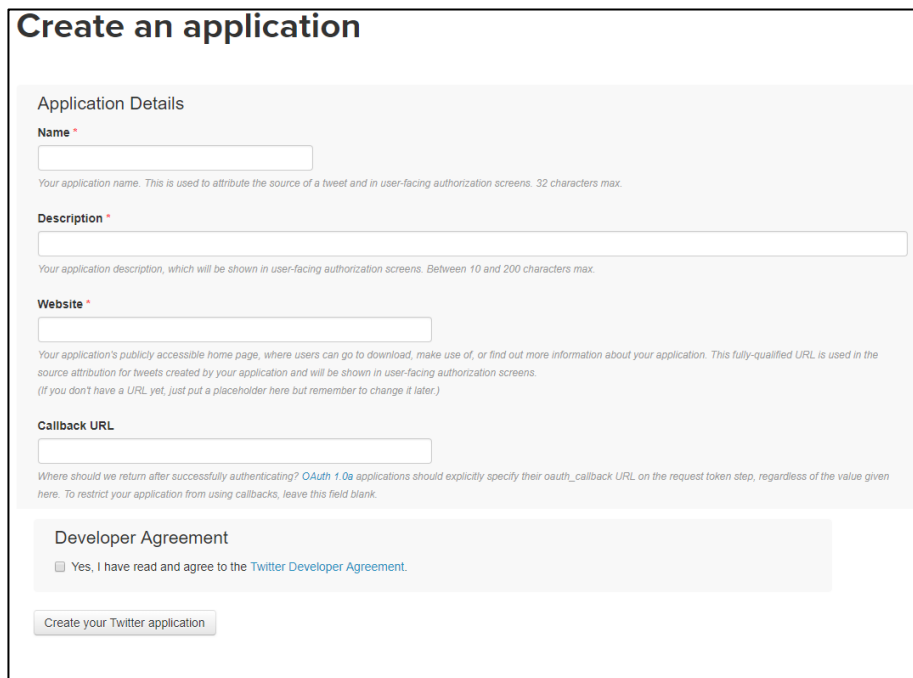
BAB V

PEMBAHASAN

Bab ini akan menjelaskan mengenai hasil dari penelitian yang telah dilakukan oleh peneliti.

5.1 Authentication

Untuk memulai pengambilan data dari *Twitter*, hal pertama yang dibutuhkan adalah membuat kode API pada *Twitter*. Kode API didapatkan dengan mengakses *Twitter* API, yaitu sebuah aplikasi yang bertujuan untuk mempermudah *developer* untuk mengakses informasi *web Twitter*. Kemudian, peneliti diharuskan untuk melakukan registrasi terlebih dahulu.



The image shows a web form titled "Create an application" for registering a Twitter application. The form is divided into several sections:

- Application Details:**
 - Name ***: A text input field with a placeholder. Below it, a note states: "Your application name. This is used to attribute the source of a tweet and in user-facing authorization screens. 32 characters max."
 - Description ***: A larger text input field. Below it, a note states: "Your application description, which will be shown in user-facing authorization screens. Between 10 and 200 characters max."
 - Website ***: A text input field. Below it, a note states: "Your application's publicly accessible home page, where users can go to download, make use of, or find out more information about your application. This fully-qualified URL is used in the source attribution for tweets created by your application and will be shown in user-facing authorization screens. (If you don't have a URL yet, just put a placeholder here but remember to change it later.)"
 - Callback URL**: A text input field. Below it, a note states: "Where should we return after successfully authenticating? OAuth 1.0a applications should explicitly specify their oauth_callback URL on the request token step, regardless of the value given here. To restrict your application from using callbacks, leave this field blank."
- Developer Agreement:**
 - A checkbox labeled "Yes, I have read and agree to the [Twitter Developer Agreement](#)."
- Create your Twitter application**: A button at the bottom of the form.

Gambar 5.1 Form Registrasi

Gambar 5.1 merupakan *form* yang harus diisi pada saat melakukan registrasi. Pendaftaran ini bertujuan untuk menginformasikan kepada pihak *Twitter* tentang tujuan peneliti dalam mengakses atau mengambil data pada *Twitter*. Setelah berhasil melakukan registrasi maka didapatkan beberapa kode, yaitu *consumer key*, *consumer secret*, *access token*, dan *access key* dari *Twitter*. Kode-kode

tersebut merupakan penghubung antara *Twitter* dan aplikasi lainnya, dalam hal ini peneliti menggunakan aplikasi *R Studio*.

5.2 Kebakaran Hutan

5.2.1 *Scrapping Data Twitter*

Pada proses *scrapping* atau pengambilan data pada *Twitter* kata kunci yang digunakan adalah “kebakaranhutan”, “saveriau”, dan “kabutasapriau” dengan menggunakan *syntax* `“tweets = searchTwitter("KebakaranHutan", "saveriau", "kabutasapriau" n = 10000, retryOnRateLimit = 1000”`.

Tabel 5.1 Data *Tweet*

No.	Text	Created	Retweet Count
1	Stay save entah itu Riau kalimantan	9/14/2019	0
2	udah sesak gw anjir .#save riau	9/12/2019	0
3	Kami dari Riau hanya bisa Minta Tlg Bantu Doa dan Tolong Sampaikan ke Seluruh Dunia	9/14/2019	9867
4	Asap tebal akibat kebakaran hutan dan lahan menyelimuti Palembang dan Pekanbaru	9/12/2019	95
5	Asap Pekanbaru Kian Parah.	9/12/2019	1

Tabel 5.1 adalah beberapa hasil dari *scrapping* data pada *Twitter* mengenai kebakaran hutan yang terjadi di provinsi Riau, hasil yang didapatkan berjumlah 13258 tweets mulai tanggal 5 september 2019 sampai 14 september 2019. Pada hasil tersebut terdapat data *text*, *created*, dan *retweet count*. Data *text* berisikan kicauan atau *tweet* yang dituliskan oleh pengguna *Twitter*. *Created* adalah data tanggal kapan *tweet* dibuat, sedangkan *retweet count* merupakan data yang berisikan banyaknya pengguna *Twitter* lain ikut menyebarkan ulang *tweet* aslinya.

5.2.2 *Proses Cleaning Data*

Tahap ini merupakan tahap yang bertujuan untuk membersihkan data-data dari *noise* (singkatan, kata yang tidak diperlukan, serta tanda baca ataupun bahasa gaul). Dari proses *cleaning* data didapatkan hasil seperti pada **Tabel 5.2**

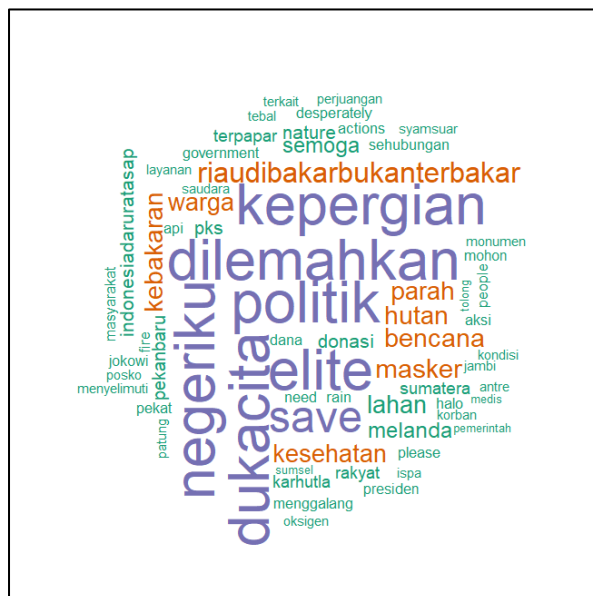
Tabel 5.2 Data Hasil *Cleaning*

No	Sebelum <i>Cleaning</i>	Sesudah <i>Cleaning</i>
1	Stay save entah itu Riau kalimantan	“stay” “save” “riau” “kalimantan”
2	udah sesak gw anjir .#save riau	“udah” “sesak” “save” “riau”
3	Kami dari Riau hanya bisa Minta Tlg Bantu Doa dan Tolong Sampaikan ke Seluruh Dunia	“kami” “riau” “hanya” “minta” “bantu” “doa” “tolong” “sampaikan” “seluruh” “dunia”
4	Asap tebal akibat kebakaran hutan dan lahan menyelimuti Palembang dan Pekanbaru	“asap” “tebal” “akibat” “kebakaran” “hutan” “lahan” “menyelimuti” “palembang” “pekanbaru”
5	Asap Pekanbaru Kian Parah.	“asap” “pekanbaru” “parah”

Pada **Tabel 5.2** dapat dilihat perubahan antara data sesudah dan sebelum dilakukan *cleaning* tanda baca, singkatan, dan kata yang tidak memiliki makna dihilangkan, serta pengubahan huruf kapital menjadi huruf kecil.

5.2.3 Word Cloud

Pada tahap ini didapatkan hasil berupa infografis. Infografis yang berisikan kata-kata yang sering muncul pada data teks. *Syntax* yang digunakan untuk membuat atau memunculkan *word cloud* yaitu `wordcloud(words = names(word.freq), freq = word.freq, min.freq = 5, random.order = F, colors = brewer.pal(2, "Dark2"))`.

Gambar 5.2 Tampilan *Wordcloud*

Gambar 5.2 merupakan *word cloud* dari topik kebakaran hutan yang ada di *Twitter*. Dapat dilihat kata “politik”, “kepergian”, “dilemahkan”, dan lain-lain menjadi kata yang paling banyak diucapkan oleh pengguna *Twitter*. Hal ini ditandai dengan ukuran kata yang memiliki ukuran paling besar pada *word cloud*. Selain menarik *word cloud* juga memudahkan pembaca dalam mencari informasi terkait.

5.2.4 *Statistical Term Frequency-Invers Document Frequency (TF-IDF)*

Statistical Term Frequency-Invers Document Frequency (TF-IDF) merupakan teknik pembobotan kata sebelum melakukan klasifikasi. Nilai dari TF-IDF dipengaruhi oleh banyaknya kemunculan kata dalam suatu dokumen dan jumlah kata ataupun frekuensi kemunculan kata secara keseluruhan.

Tahapan awal yang dilakukan adalah mencari *Invers Document Frequency (IDF)* yaitu jumlah dokumen dibagi dengan jumlah kata yang muncul. Contoh kata yang akan dihitung adalah kata “kebakaran”, “Politik”, “Hutan”, dan “Parah”. Untuk hasil dari jumlah kemunculan data dan perhitungan IDF dapat dilihat pada **Tabel 5.3**.

Tabel 5.3 Perhitungan IDF

Kata	Dokumen				Jumlah	IDF = $\log\left(\frac{N}{df}\right)$
	1	2	...	13256		
Kebakaran	1	0	...	0	857	$\log\left(\frac{13256}{857}\right) = 1.19$
Politik	0	0	...	1	2415	$\log\left(\frac{13256}{2415}\right) = 0.74$
⋮	⋮	⋮	⋮	⋮	⋮	⋮
Hutan	0	0	...	0	873	$\log\left(\frac{13256}{873}\right) = 1.18$
Parah	0	1	...	0	838	$\log\left(\frac{13256}{838}\right) = 1.2$

Berdasarkan **Tabel 5.3** dapat diketahui nilai IDF, nilai IDF untuk kata “Kebakaran” pada seluruh dokumen adalah sebesar 1.19. Perhitungan yang sama juga dilakukan untuk kata-kata yang lainnya sehingga didapatkan nilai IDF dari masing-masing kata. Setelah mendapatkan nilai IDF masing-masing kata, selanjutnya adalah melakukan perhitungan TF, yaitu dengan cara menjumlahkan banyaknya kata dalam sebuah dokumen dibagi dengan jumlah kata keseluruhan yang terdapat dalam sebuah dokumen.

Tabel 5.4 Perhitungan TF

Kata	Dokumen				
	1	...	6	...	13256
Kebakaran	1/10= 0.1	...	0	...	0
Politik	0	...	0	...	0
⋮	⋮	⋮	⋮	⋮	⋮
Hutan	1/10= 0.1	...	0	...	0
Parah	0	...	0	...	0
Jumlah	10	...	0	...	0

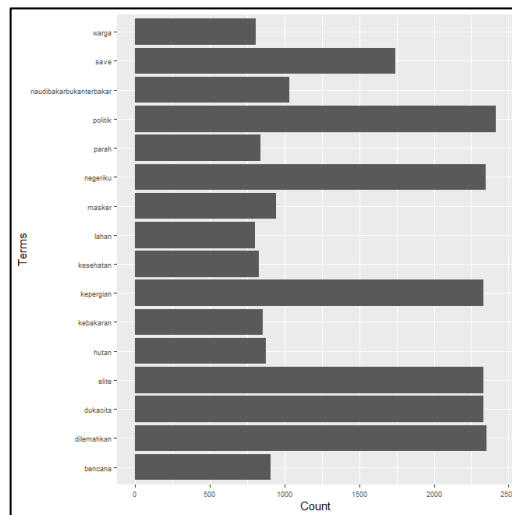
Berdasarkan **Tabel 5.4** diketahui bahwa kata “kebakaran” pada dokumen 6 berjumlah 1 dan jumlah keseluruhan kata pada dokumen 6 berjumlah 10 kata. Sehingga didapatkan nilai TF untuk kata “kebakaran” yaitu $1/10 = 0.1$. Untuk kata yang lain, nilai TF juga didapatkan dengan perhitungan yang sama. Setelah mendapatkan nilai TF dan IDF, selanjutnya melakukan perhitungan untuk TF-IDF. Nilai TF-IDF diperoleh dengan mengalikan nilai TF dengan nilai IDF. Perhitungan TF-IDF dapat dilihat pada **Tabel 5.5**.

Tabel 5.5 Perhitungan TF-IDF Kasus Kebakaran Hutan

Kata	Dokumen (TF)					IDF	TF-IDF				
	1	...	100	...	13256		1	...	100	...	13256
Kebakaran	0.1	...	0	...	0	1.19	0.12	...	0	...	0
Politik	0	...	0	...	0	0.74	0	...	0	...	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
Hutan	0.1	0	...	0	1.18	0.11	0	...	0
Parah	0	0	...	0	1.2	0	0	...	0

Setelah mendapatkan TF-IDF dapat diketahui bahwa semakin besar nilai TF-IDF dari suatu kata maka semakin besar pula hubungan kata tersebut pada sebuah dokumen.

Selanjutnya untuk mengetahui frekuensi kata yang sering muncul pada data *tweet* dapat dilihat pada **Gambar 5.3**.

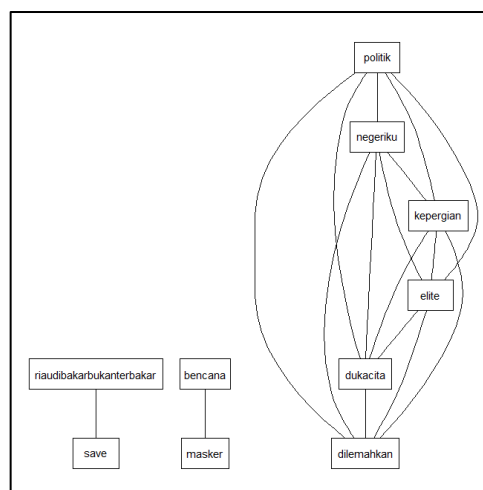


Gambar 5.3 Tampilan *Statistical Term Frequency*

Pada **Gambar 5.3** dapat dilihat hasil dari *Statistical Term Frequency*. Kata yang paling banyak disebutkan dalam kasus kebakaran hutan adalah kata “politik” kemudian kata “dilemahkan”, “negeriku”, dan diikuti oleh kata-kata lainnya. Sedangkan untuk kasus banjir kata yang sering digunakan adalah kata “anies”, “jokowi”, “hujan” dan “macet”.

5.2.5 Asosiasi Kata

Asosiasi kata digunakan untuk melihat keterkaitan kata dengan kata lainnya sehingga dapat dihubungkan satu dengan yang lainnya.



Gambar 5.4 Tampilan Asosiasi Kata

Dari asosiasi kata yang ditampilkan menampilkan kata yang memiliki kemunculan lebih dari 900 kemunculan pada data *tweet*. Menurut (Sarwono,

2006) pada penelitian (Burhanuddin, 2012) hasil asosiasi dapat diketahui kekuatan hubungan antar dua kata yang saling berhubungan dengan kisaran nilai -1 s/d 1. Ada beberapa kategori nilai korelasi yang digunakan sebagai berikut :

- 0 : Tidak ada korelasi antara dua variabel
- >0 - 0,25 : Korelasi lemah
- >0,25 - 0,5 : Korelasi cukup
- >0,5 – 0,75 : Korelasi kuat
- 1 : Korelasi sangat kuat.

Berikut beberapa kata dan asosiasinya dengan kata lain, yaitu kata “politik” seperti pada **Tabel 5.6**

Tabel 5.6 Asosiasi Kata “Politik” Kasus Kebakaran Hutan

Kata	Nilai
Dukacita	0.98
Elite	0.98
Kepergian	0.98
Dilemahkan	0.97
Negeriku	0.97
Pekerjaan	0.17
Partai	0.16

Berdasarkan nilai asosiasi menggunakan nilai korelasi minimal 0,1 dan didapatkan kata yang berasosiasi dengan kata “politik”. Kata yang memiliki nilai asosiasi paling besar adalah kata “dukacita”, “elite”, dan “kepergian”. Ada kemungkinan kata tersebut berasosiasi dikarenakan pemberitaan bahwa adanya unsur politik dalam bencana kebakaran ini, serta perginya salah satu *elite* politik yang pergi melakukan kunjungan kerja ke luar negeri pada saat bencana ini berlangsung.

5.2.6 *Sentiment Analysis*

Memasuki tahap analisis sentiment, pada tahap ini setiap *tweet* akan diberikan label yaitu, sentimen positif dan sentimen negatif. Penilaian kelas sentiment didasarkan pada kumpulan kata Bahasa Indonesia yang berisikan kumpulan kata positif dan kata negatif. Kemudian dilakukan pelabelan otomatis dengan menggunakan aplikasi *R* dengan cara menghitung selisih jumlah kata

Pada **Gambar 5.6** dapat dilihat kata “kesehatan” menjadi kata yang sering muncul dalam *tweet* kelas positif terkait dengan kasus kebakaran hutan yang ada di Provinsi Riau.



Gambar 5.7 Tampilan *Wrodcloud* Kelas Negatif

Gambar 5.7 menunjukkan kata “parah” merupakan kata yang sering digunakan dalam *tweet* kelas negatif diketahui dari besarnya ukuran huruf dan diikuti oleh kata “bencana”, “kebakaran”, dan lain-lain.

5.2.7 *Naïve Bayes Classifier*

Setelah mengetahui klasifikasi sentimen pada topik, hal berikutnya yang dilakukan oleh peneliti adalah mengetahui tingkat akurasi dari klasifikasi sentimen tersebut. Dalam hal ini peneliti menggunakan metode *naïve bayes classifier*.

Naïve bayes classifier memerlukan data latih dan data uji. Data latih dapat mempengaruhi tingkat akurasi yang dihasilkan. Data uji merupakan data yang digunakan untuk menguji akurasi dari model yang telah dibuat oleh data latih. Dari total 6011 data keseluruhan digunakan 80% sebagai data latih dan 20% digunakan untuk data uji.

Tabel 5.7 Pembagian Data *Training* dan *Testing*

	<i>Data Training</i>	<i>Data Testing</i>
<i>Ratio</i>	80%	20%

Jumlah	4809	1202
--------	------	------

Analisis *naïve bayes classifier* menghasilkan *confusion matrix*. Klasifikasi pada analisis NBC sebelumnya harus memiliki *prior probabilitas*, yang merupakan komponen utama pada konsep *Naïve Bayes*. Nilai *prior probabilitas* dapat dilihat pada **Table 5.8**

Tabel 5.8 Prior Probabilitas

Positif	Negatif
0.29	0.71

Nilai *prior probabilitas* didapatkan dari jumlah pada kategori kelas positif dan negatif dibagi dengan total dari seluruh data klasifikasi. Selanjutnya yaitu mendapatkan *confusion matrix*.

Tabel 5.9 Data Prediksi dan Aktual

No.	Kalimat	Data	Prediksi	Aktual
1	terpapar warga pekanbaru antre oksigen posko kesehatan pks layanan kesehatan pks	1	Pos	Pos
2	kekhawatiran yg menyelimuti sibolgatapnuli qadarullah	2	Neg	Neg
3	subhanallah pekat karhutla menyelimuti permukiman warga sesak napas ya api	3	Pos	Pos
4	foto patung monumen perjuangan rakyat dipasang masker medis aksi bentuk sindiran terkait penangana	4	Neg	Neg
...
53	ganggu penerbangan penumpang gagal terbang	53	Neg	Neg
54	tebal bandara apt pranoto samarinda ditutup	54	Neg	Neg
55	ayolah fiz capek ni taun teriak save save udara bersih melulu ujungnya masker	55	Neg	Neg
56	mohon saudara desak mempercepat proses penanggulangan bencana	56	Neg	Pos
57	yadiluar ntar sakit dirumah istirahat ya	57	Neg	Neg

Tabel 5.9 merupakan tabel perbandingan antara data prediksi dan aktual pada kolom 1 dan 2 memiliki prediksi benar, sentimen positif diprediksi sentimen positif dan sentimen negatif diprediksi sentimen negatif. Tapi pada kolom 3 sentimen positif diprediksi sebagai sentimen negatif sehingga dapat disebut sebagai *false negative* (FN).

Tabel 5.10 *Confusion Matrix*

Prediksi	Aktual	
	Positif	Negatif
Positif	311	21
Negatif	44	826

Dari **Tabel 5.10** dapat diketahui 311 data masuk dalam *True Positive*, 826 *True Negative*, 44 *False Negative*, dan 21 *False Positive*. Setelah mendapatkan *confusion matrix* langkah selanjutnya adalah menentukan nilai akurasi yang didapatkan.

Dengan menggunakan aplikasi *R Studio* didapat nilai akurasi *Naïve bayes classifier* sebesar 0.94. Dengan nilai akurasi yang cukup tinggi maka dapat dikatakan bahwa klasifikasi sudah tepat. Untuk nilai prediksi dan nilai-nilai lainnya bisa dihitung secara manual sebagai berikut :

$$Recall = \frac{TP}{TP + FN} 100 = \frac{311}{311 + 44} = 0.876 = 87.6\%$$

$$Precision = \frac{TP}{TP + FP} 100 = \frac{311}{311 + 21} = 0.936 = 93.6\%$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} 100 = \frac{311 + 826}{311 + 21 + 826 + 44} = 0.945 = 94.5\%$$

$$Spesificity = \frac{TN}{TN + FP} 100 = \frac{826}{826 + 21} = 0.975 = 97.5\%$$

$$FPR = 1 - Spesificity$$

$$= 1 - 0.975 = 0.025$$

$$AUC = \frac{1 + recall - FPR}{2} = \frac{1 + 0.876 - 0.025}{2} = 0.9255$$

5.3 Banjir

5.3.1 Scapping Data Twitter

Pada proses scraping atau pengambilan data pada *Twitter* kata kunci yang digunakan adalah “Bajir Jakarta” dengan menggunakan *syntax* “`tweets = searchTwitter("Banjir Jakarta" n = 10000, retryOnRateLimit = 1000`”.

Tabel 5.11 Data *Tweet*

No.	Text	Created	Retweet Count
1	Ketika ditanyakan hal ini, Anies hanya tersenyum dan enggan menjawab sambil berlalu. https://t.co/bg9LPM40e5	12/20/2019	4
2	Saat ditanya Banjir di Jakarta, Jokowi sempat terdiam kurang lebih 3 Menit sebelum menjawab	12/20/2019	222
3	Anggaran yang harusnya untuk kepentingan rakyat di pangkas hampir 50%, lihat saja sampah, banjir, kemacetan, polusi udara....	12/20/2019	20
4	Ini bukan salah hujan apalagi salah gubernurnya. Ini murni salah warga Jakarta dalam memilih kendaraan.	12/20/2019	15823
5	Agar tidak kesetrum, ada beberapa tips dari PLN Unit Induk Distribusi Jakarta Raya apabila rumah terendam banjir.... https://t.co/BkAODynZod	12/20/2019	2

Pada **Tabel 5.11** adalah beberapa hasil dari *scraping* data pada *Twitter* mengenai banjir yang terjadi di DKI Jakarta. Data yang didapat adalah 10000 data *tweet*.

5.3.2 Proses *Cleaning* Data

Pada **Tabel 5.12** dapat dilihat hasil proses *cleaning* data dari bencana banjir di DKI Jakarta.

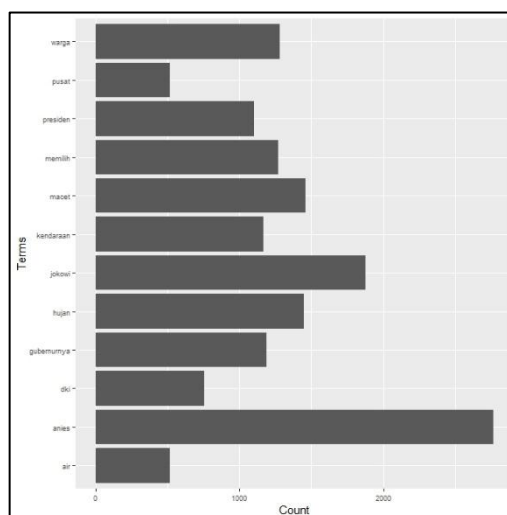
Tabel 5.12 Data Hasil *Cleaning*

No	Sebelum <i>Cleaning</i>	Sesudah <i>Cleaning</i>
1	Ketika ditanyakan hal ini, Anies hanya tersenyum dan enggan menjawab sambil berlalu. https://t.co/bg9LPM40e5	ditanyakan anies tersenyum enggan menjawab sambil berlalu
2	Saat ditanya Banjir di Jakarta, Jokowi sempat terdiam kurang lebih 3 Menit sebelum menjawab	ditanya jakarta jokowi sempat terdiam kurang lebih menit sebelum menjawab
3	Anggaran yang harusnya untuk kepentingan rakyat di pangkas	anggaran harusnya kepentingan rakyat pangkas hampir lihat

Tabel 5.13 Perhitungan TF-IDF

Kata	Dokumen (TF)					IDF	TF-IDF				
	1	...	13	...	10000		1	...	13	...	10000
jokowi	0	...	0.07	...	0	0.72	0	...	0.05	...	0
anies	0	...	0.15	...	0	0.56	0	...	0.084	...	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
macet	0	0	...	0	0.83	0	0	...	0
hujan	0	0	...	0	0.84	0	0	...	0

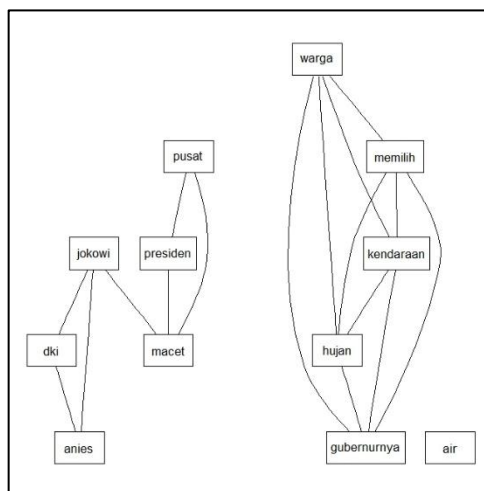
Semakin kecil nilai TF-IDF maka akan semakin besar nilai TF-IDF makan semakin besar similaritas antara kata dan dokumen yang ada. Selanjutnya pada **Gambar** dapat dilihat frekuensi atau jumlah kata yang paling banyak digunakan pada dokumen.



Gambar 5.9 Frekuensi Kata

5.3.5 Asosiasi Kata

Pada **Gambar** akan ditampilkan hasil asosiasi kata pada kasus banjir di DKI Jakarta. Kemudian pada **Tabel** diberikan contoh nilai asosiasi kata yang dalam hal ini contoh diambil dari kata “jokowi”.



Gambar 5.10 Asosiasi Kata

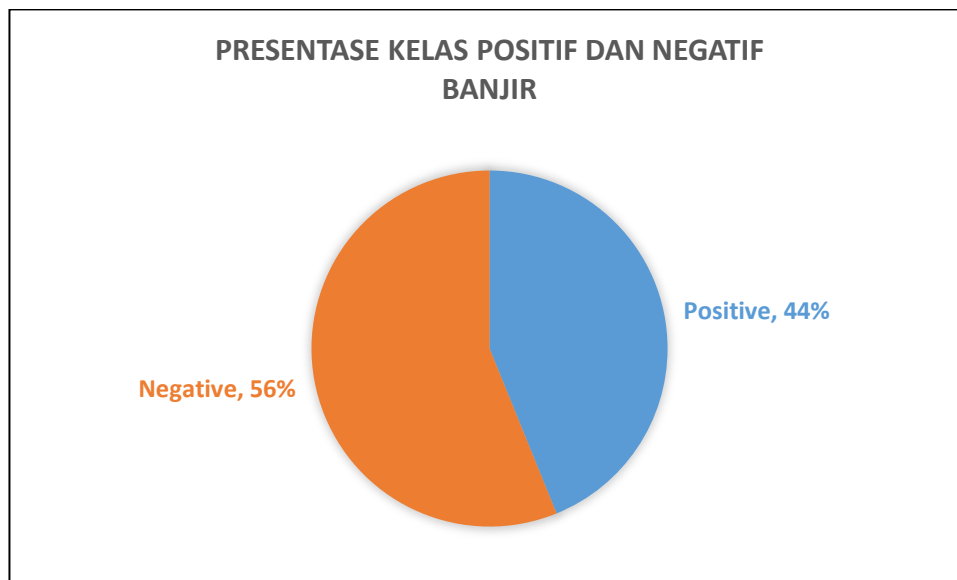
Tabel 5.14 Asosiasi Kata “jokowi” Kasus Banjir

Kata	Nilai
Waduk	0.41
Dikeruk	0.35
Pemprov	0.31
Bersihkan	0.27
Macet	0.27
kota	0.19

Berdasarkan nilai asosiasi kata untuk “Jokowi” dan kata lainnya, kata waduk memiliki nilai asosiasi paling besar. Hal ini berkaitan dengan perencanaan Presiden Jokowi dalam mengatasi banjir di DKI Jakarta dengan membangun waduk di daerah Cirebon.

5.3.6 *Sentiment Analysis*

Hasil *sentiment Analysis* untuk kasus banjir di DKI Jakarta dapat dilihat pada **Gambar 5.11**.



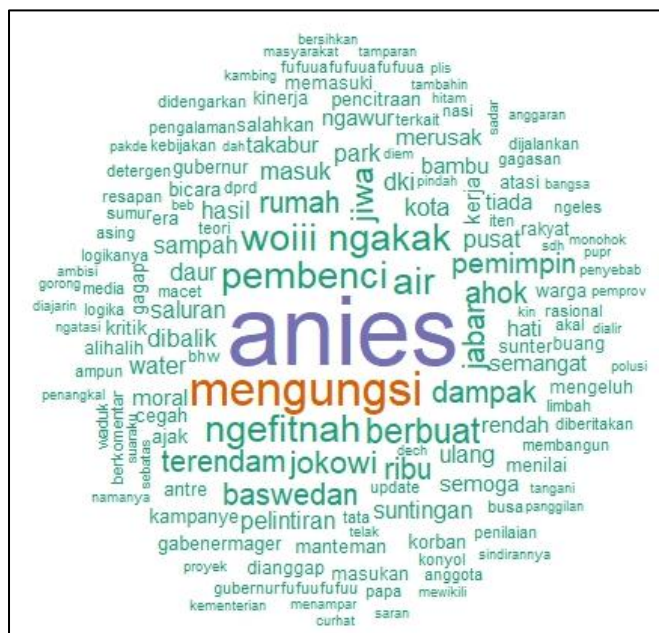
Gambar 5.11 *Sentiment Analysis*

Pada **Gambar 5.11** hasil klasifikasi *sentiment Analysis* didapatkan sebanyak 56% atau sebanyak 3103 *tweet* masuk dalam kelas sentimen negatif dan 44% atau 2424 masuk dalam kelas sentimen positif. Untuk melihat gambaran kata yang sering muncul dalam *tweet* kelas positif dan kelas negatif dapat dilihat pada **Gambar 5.12** dan **Gambar 5.13**.



Gambar 5.12 *Tampilan Wordcloud Kelas Positif*

Pada **Gambar 5.12** kata “hujan” menjadi kata yang paling banyak digunakan dalam *tweet* kelas positif.



Gambar 5.13 Tampilan *Wordcloud* Kelas Negatif

Pada **Gambar 5.13** kata “anies” adalah kata yang paling banyak digunakan dalam *tweet* kelas negative oleh pengguna *Twitter*.

5.3.7 *Naïve Bayes Classifier*

Untuk penilaian *naïve bayes classifier* digunakan data *training* dan *testing* sebesar 80% *training* dan 20% *testing* seperti pada **Tabel 5.15**. Kemudian didapatkan hasil untuk *prior* probabilitas seperti pada **Tabel 5.16**.

Tabel 5.15 Pembagian Data *Training* dan *Testing*

	<i>Data Training</i>	<i>Data Testing</i>
<i>Ratio</i>	80%	20%
Jumlah	4422	1105

Tabel 5.16 *Prior* Probabilitas

Positif	Negatif
0.44	0.56

Tabel 5.17 Tabel Perbandingan Nilai Prediksi dan Aktual

No	Kalimat	Data	Prediksi	Aktual
...
18	mdablek aibon jokowi saran atasi reaksi anies ya	18	neg	neg
19	kerjakerja karya ahok diakui karya	19	neg	pos
20	namanya menampar sindirannya monohok mewikili suaraku	20	neg	neg
21	rasional logika rakyat menilai penilaian yg kinerja	21	neg	neg
22	air aksi damai bela	22	neg	pos
23	water park pelintiran daur ulang hasil suntingan dibalik	23	neg	neg
24	hujan gubernurnya warga memilih kendaraan	24	pos	pos

Tabel 5.17 merupakan table perbandingan antara data prediksi dan aktual pada kolom 1 sampai 38 memiliki prediksi benar, sentimen positif diprediksi sentimen positif dan sentimen negatif diprediksi sentimen negatif. Tapi pada kolom 39 sentimen positif diprediksi sebagai sentimen negatif sehingga dapat disebut sebagai *false negative* (FN).

Tabel 5.18 *Confusion Matrix*

Prediksi	Aktual	
	Positif	Negatif
Positif	467	15
Negatif	31	592

Dari **Tabel 5.18** dapat diketahui 292 data masuk dalam *True Positive*, 396 *True Negative*, 18 *False Negative*, dan 12 *False Positive*. Setelah mendapatkan *confusion matrix* langkah selanjutnya adalah menentukan nilai akurasi yang didapatkan.

Dengan menggunakan aplikasi *R Studio* didapat nilai akurasi *Naïve bayes classifier* sebesar 0.958. Dengan nilai akurasi yang cukup tinggi maka dapat dikatakan bahwa klasifikasi sudah tepat. Untuk nilai prediksi dan nilai-nilai lainnya bisa dihitung secara manual sebagai berikut :

$$Recall = \frac{TP}{TP + FN} \times 100 = \frac{467}{467 + 31} \times 100 = 93.7\%$$

$$Precision = \frac{TP}{TP + FP} \times 100 = \frac{467}{467 + 15} \times 100 = 96.8\%$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} 100 = \frac{467 + 592}{467 + 15 + 592 + 31} 100 = 95.8\%$$

$$Spesificity = \frac{TN}{TN + FP} 100 = \frac{592}{592 + 15} = 0.975 = 97.5\%$$

$$FPR = 1 - Spesificity$$

$$= 1 - 0,975 = 0,025$$

$$AUC = \frac{1 + recall - FPR}{2} = \frac{1 + 0.937 - 0.025}{2} = 0.95$$

Tabel 5.19 Tabel Hasil

Statistik	Hasil	
	Kebakaran Hutan	Banjir
Recall	87.6%	93.7%
Precision	93.6%	96.8%
Accuracy	94.5%	95.8%
Spesificity	97.5%	97.5%
AUC	0.92	0.95

Dari **Tabel 5.19** dapat diketahui hasil dari akurasi metode *Naïve Bayes Classifier* untuk klasifikasi kasus kebakaran hutan di Provinsi Riau adalah sebesar 94,5%, sedangkan untuk kasus banjir di DKI Jakarta adalah sebesar 95,8%. Dengan nilai akurasi yang didapat maka dapat disimpulkan bahwa hasil klasifikasi sudah sangat baik.