BAB IV

HASIL DAN PEMBAHASAN

Bab ini berisi penjelasan menganai pengujian model dan implemestasi dari analisis sentimen tentang isu pemindahan ibu kota pada Twitter dengan metode Naïve Bayes.

4.1 Kebutuhan Sistem

Untuk mengimplementasikan analisis sentimen pemindahan ibu kota pada *tweet* di Twitter dengan mengunakan *metode* Naïve Bayes, ada beberapa spesifikasi perangkat keras dan perangkat lunak yang dibutuhkan.

4.1.1 Perangkat Keras (*Hardware*)

Perangkat keras adalah komponen fisik yang disusun sehingga membentuk suatu komputer, fungsinya untuk menjalankan perintah dari pengguna. Dalam hal ini perangkat keras yang dapat dijadikan acuan untuk melakukan implementasi dari analisis sentimen pemindahan ibu kota pada *tweet* di *Twitter* dengan mengunakan *metode* Naïve Bayes adalah:

2. Model Notebook : ASUS VivoBook S200E
3. Prosesor : Intel Celeron
4. Memory : 4 GB

4.1.2 Perangkat Lunak (*Software*)

Perangkat keras adalah komponen non fisik, gunanya untuk memroses data atau intruksi hingga mendapat hasil atau menjalankan sebuah perintah tertentu. Dalam hal ini, perangkat lunak yang dibutuhkan antara lain adalah:

1. *Web Browser* untuk menjalankan aplikasi berbasis web.
2. Anaconda Navigator untuk menjalankan program yang menggunakan bahasa pemrogaman Python.

4.2 Pembangunan Model

Pada sub bab pembangunan model akan dijelaskan lebih detail langkah langkah yang telah dipaparkan dalam sub bab 3.3.

4.2.1 Pengumpulan Data

Pengumpulan data menggunakan *package* Twitterscraper dari *library* Python, berikut *source code* yang digunakan :

```
twitterscraper "ibu kota" --output
data_ibukota.csv --limit 20000 -begindate 2019-
8-1 --enddate 2019-9-5 --csv
```

Gambar 4.1 *Source Code* Pengumpulan Data

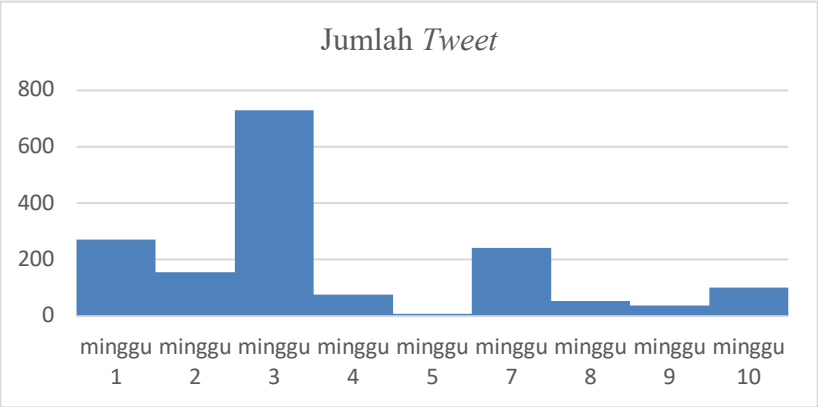
Pada gambar 4.1 menunjukkan *source code* pengumpulan data. Proses pengumpulan data pada Twitter dilakukan dengan menggunakan package Twitterscraper dari *library* Python. Yang mana dapat digunakan tanpa menggunakan API Twitter. Dari *source code* di atas dapat diketahui bahwa data yang diambil berdasarkan kata kunci “ibu kota”, *file output* yang disimpan dalam format csv, banyaknya data yang ditentukan yakni 20.000 data, serta batasan waktu

yakni sejak tanggal 1 Agustus 2019 hingga 5 September 2019. *Syntax* tersebut dapat dijalankan di *command prompt*. Berikut adalah contoh hasil dari pengumpulan data:

```
irfanwahidi60;HM.Irfan
Wahidi;230692266;1164325959069081600;/irfanwahidi60/status/1164
325959069081600;2019-08-21 23:58:35;1566431915;0;0;3;0;;;;Terlalu
nyinyir untuk hal2 sepele (pertamanan)  padahal masalah bangsa ini
demikian besarnya, coba pikirin solusi Papua mungkin ibukota lebih
tepat dipindahkan ke Papua yakin 100% masalah selesai.;"<p
class=""TweetTextSize js-tweet-text tweet-text"" data-aria-label-
part=""0"" lang=""in"">Terlalu nyinyir untuk hal2 sepele (pertamanan)
padahal masalah bangsa ini demikian besarnya, coba pikirin solusi
Papua mungkin <strong>ibukota</strong> lebih tepat dipindahkan ke
```

Gambar 4.2 Contoh Hasil Pengumpulan Data

Data yang dikumpulkan menggunakan *package* Twitterscraper dari *library* Python berupa data *tweet* yang acak. Sehingga apabila digambarkan dengan grafik banyaknya *tweet* yang berhasil didapat pada tiap-tiap minggunya pada Bulan Agustus-September 2019 yakni sebagai berikut:



Gambar 4.3 Grafik Jumlah *Tweet*

4.2.2 Penyaringan Data

Penyaringan dilakukan dengan menghapus kolom yang tidak dibutuhkan dalam membangun model. Penghapusan kolom dalapat dilakukan di Microsoft

Excel. Sehingga hanya tersisa kolom “username”, “timestamp”, dan “text”. Namun data yang harus diolah lebih lanjut yakni yang berada pada kolom “text”.

Tabel 4.1 Contoh Hasil Penyaringan Data

No.	Data
1.	@KompasTV segala ibukota d pindahin buang"dana ....mending dananya buat mensejahtrakan masyarakat... Kalubukan cukong"yg menang proyek nanti siapa lagi pakkkk
2.	Calon ibukota terpapar asap! Kalimantan jg paling banyak titik apinya, banyak lahan gambut+ batubara. Waduh! Terus, klo di sana di tengah hutan mau ngurus apa? Problem yg krusial itu manusia, bukan yg lain. Penguasa hrs lbh dekat dgn mayoritas rakyatnya. https://twitter.com/conan_idn/status/1160912693865275393Â â€¦
3.	Ibukota pindah ! Jakarta akan tetap seperti sekarang, tetap menjadi pusat bisnis dan ekonomi , pusat hiburan dan magnet bagi orang dari luar jakarta untuk mencari penghidupan yang layak .....dan pastinya Jakarta akan tetap macet . #kumparanGiveaway #MembayangkanJakarta
4.	Waowwwww 500t buat Jakarta. Mending 500t buat ibukota baru coy
5.	Kalau aku sih mikir positifnya aja , kapan lagi akhirnya kalimantan akan maju . Selama ini sumber daya alam kaya kita men,etapi pembangun

No.	Data
	infrastruktur dll larinya kepulau jawa semua . Nahhh dgn jd ibukota kita bakalan maju yakan . Mudahan berjalan seperti yang diharapkan

Tabel 4.1 di atas menunjukkan contoh hasil data yang telah dilakukan penyaringan. Data yang diambil hanya data *tweet* dengan label 'text' saja.

4.2.3 Pelabelan Data

Pelabelan data dilakukan dengan manual yakni membaca satu per satu, kemudian memberi label apakah kalimat yang terkandung dalam kolom 'text' memiliki sentimen positif, negatif, atau netral.

Tabel 4.2 Contoh Hasil Pelabelan Data

No.	Data	Sentimen
1.	@KompasTV segala ibukota d pindahin buang"dana ....mending dananya buat mensejahtrakan masyarakat... Kalubukan cukong"yg menang proyek nanti siapa lagi pakkkk	Negatif
2.	Calon ibukota terpapar asap! Kalimantan jg paling banyak titik apinya, banyak lahan gambut+ batubara. Waduh! Terus, klo di sana di tengah hutan mau ngurus apa? Problem yg krusial itu manusia, bukan yg lain. Penguasa hrs lbh dekat dgn mayoritas rakyatnya. https://twitter.com/conan_idn/status/1160912693865275393Â â€¦	Negatif
3.	Ibukota pindah ! Jakarta akan tetap seperti sekarang, tetap menjadi pusat bisnis dan ekonomi , pusat hiburan dan magnet bagi orang dari luar	Netral

No.	Data	Sentimen
	jakarta untuk mencari penghidupan yang layak .....dan pastinya Jakarta akan tetap macet . #kumparanGiveaway #MembayangkanJakarta	
4.	Waowwwww 500t buat Jakarta. Mending 500t buat ibukota baru coy	Positif
5.	Kalau aku sih mikir positifnya aja , kapan lagi akhirnya kalimantan akan maju . Selama ini sumber daya alam kaya kita men,etapi pembangun infrastruktur dll larinya kepulau jawa semua . Nahhh dgn jd ibukota kita bakalan maju yakan . Mudahan berjalan seperti yang diharapkan	Positif

Tabel 4.2 di atas menunjukkan contoh hasil pelabelan data yang dilakukan secara manual. Setiap *tweet* dibaca satu persatu kemudian diberi label sesuai sentimennya.

A. Case Folding

Source code yang digunakan untuk proses *case folding* adalah sebagai berikut:

```
tweet = tweet.lower()
```

Gambar 4.4 *Source Code Case Folding*

Gambar 4.4 di atas menunjukkan *source code* untuk melakukan *proses case folding*. *Source code* tersebut berfungsi untuk meyeragamkan semua huruf yang terdapat pada data *tweet* di dalam variabel *tweet* kemudian hasilnya ke dalam variabel *tweet*.

Hasil dari proses *case folding* adalah data teks *tweet* yang semula hurufnya ada yang kapital berubah menjadi huruf kecil semua. Berikut adalah contoh hasil dari proses *case folding* :

Tabel 4.3 Contoh Hasil *Case Folding*

No.	Data Sebelum Case Folding	Data Setelah Case Folding
1.	@KompasTV segala ibukota d pindahin buang"dana ....mending dananya buat mensejahtrakan masyarakat... Kalubukan cukong"yg menang proyek nanti siapa lagi pakkkk	@kompastv segala ibukota d pindahin buang"dana ....mending dananya buat mensejahtrakan masyarakat... kalubukan cukong"yg menang proyek nanti siapa lagi pakkkk
2.	Calon ibukota terpapar asap! Kalimantan jg paling banyak titik apinya, banyak lahan gambut+ batubara. Waduh! Terus, klo di sana di tengah hutan mau ngurus apa? Problem yg krusial itu manusia, bukan yg lain. Penguasa hrs lbh dekat dgn mayoritas rakyatnya. https://twitter.com/conan_idn/status/1160912693865275393Â â€¦	calon ibukota terpapar asap! kalimantan jg paling banyak titik apinya, banyak lahan gambut+ batubara. waduh! terus, klo di sana di tengah hutan mau ngurus apa? problem yg krusial itu manusia, bukan yg lain. penguasa hrs lbh dekat dgn mayoritas rakyatnya. https://twitter.com/conan_idn/status/1160912693865275393â â€¦
3.	Ibukota pindah ! Jakarta akan tetap seperti sekarang, tetap menjadi pusat bisnis dan	ibukota pindah ! jakarta akan tetap seperti sekarang, tetap menjadi pusat bisnis dan

No.	Data Sebelum Case Folding	Data Setelah Case Folding
	ekonomi , pusat hiburan dan magnet bagi orang dari luar jakarta untuk mencari penghidupan yang layak .....dan pastinya Jakarta akan tetap macet . #kumparanGiveaway #MembayangkanJakarta	ekonomi , pusat hiburan dan magnet bagi orang dari luar jakarta untuk mencari penghidupan yang layak .....dan pastinya jakarta akan tetap macet . #kumparangiveaway #membayangkanjakarta
4.	Waowwwww 500t buat Jakarta. Mending 500t buat ibukota baru coy	waowwwww 500t buat jakarta. mending 500t buat ibukota baru coy
5.	Kalau aku sih mikir positifnya aja , kapan lagi akhirnya kalimantan akan maju . Selama ini sumber daya alam kaya kita men,etapi pembangun infrastruktur dll larinya kepulau jawa semua . Nahhh dgn jd ibukota kita bakalan maju yakan . Mudahan berjalan seperti yang diharapkan	kalau aku sih mikir positifnya aja , kapan lagi akhirnya kalimantan akan maju . selama ini sumber daya alam kaya kita men,etapi pembangun infrastruktur dll larinya kepulau jawa semua . nahhh dgn jd ibukota kita bakalan maju yakan . mudahan berjalan seperti yang diharapkan

Tabel 4.3 menunjukkan data yang telah melalui proses penyeragaman huruf menjadi huruf kecil. Semua data dalam Tabel yang semula huruf kapital diseragamkan menjadi huruf kecil.

B. *Cleansing*

Pada proses *cleansing* disini akan dijelaskan lebih rinci sebagaimana yang telah disebutkan pada bab III sebelumnya yang mana terdapat beberapa langkah langkah. Berikut tahapan proses *cleansing*:

1. Menghilangkan URL Situs Lain dan URL Gambar & Video

   Untuk menghilangkan atau menghapus URL situs lain maupun gambar dan video, *source code* yang digunakan yakni sebagai berikut:

```
text = re.sub(r'\w+:\/{2}[\d\w-]+(\.[\d\w-]+)*(?:(?:\/[^\s/]*))*', ' ', text)
text = re.sub('\[vid]', " ",text)
text = re.sub('\[vid]', " ",text)
```

Gambar 4.5 *Source Code* Menghilangkan URL Situs Lain dan URL Gambar & Video

Gambar 4.5 menunjukkan *source code* untuk melakukan penghapusan atau menghilangkan URL situs lain serta URL gambar. Dan video. Dalam *source code* tersebut terlihat bahwa untuk menghilangkan URL digunakan fungsi sub dari *package library* re kemudian memasukkan pola yang menggambarkan URL situs lain dan URL Gambar yang diawali kata ‘http’, ‘www’, ‘pic’, dan ‘vid’, karakter untuk menggantikan URL tersebut serta variabel data yang akan diproses. maka keseluruhan di awal katanya akan dihilangkan. Contoh hasil dari proses menghilangkan URL situs lain serta URL gambar dan video adalah sebagai berikut:

Tabel 4.4 Contoh Hasil Menghilangkan URL Situs Lain dan URL Gambar & Video

No.	Data Sebelum URL dihilangkan	Data Setelah URL dihilangkan
1.	@kompastv segala ibukota d pindahin buang"dana ....mending dananya buat mensejahtrakan masyarakat... kalubukan cukong"yg menang proyek nanti siapa lagi pakkkk	@kompastv segala ibukota d pindahin buang"dana ....mending dananya buat mensejahtrakan masyarakat... kalubukan cukong"yg menang proyek nanti siapa lagi pakkkk
2.	calon ibukota terpapar asap! kalimantan jg paling banyak titik apinya, banyak lahan gambut+ batubara. waduh! terus, klo di sana di tengah hutan mau ngurus apa? problem yg krusial itu manusia, bukan yg lain. penguasa hrs lbh dekat dgn mayoritas rakyatnya. https://twitter.com/conan_idn/status/1160912693865275393â â€¦	calon ibukota terpapar asap! kalimantan jg paling banyak titik apinya, banyak lahan gambut+ batubara. waduh! terus, klo di sana di tengah hutan mau ngurus apa? problem yg krusial itu manusia, bukan yg lain. penguasa hrs lbh dekat dgn mayoritas rakyatnya. â€¦
3.	ibukota pindah ! jakarta akan tetap seperti sekarang, tetap menjadi pusat bisnis dan ekonomi , pusat hiburan dan magnet bagi orang dari luar jakarta untuk mencari penghidupan yang layak	ibukota pindah ! jakarta akan tetap seperti sekarang, tetap menjadi pusat bisnis dan ekonomi , pusat hiburan dan magnet bagi orang dari luar jakarta untuk mencari penghidupan yang layak

No.	Data Sebelum URL dihilangkan	Data Setelah URL dihilangkan
	.....dan pastinya jakarta akan tetap macet . #kumparangiveaway #membayangkanjakarta	.....dan pastinya jakarta akan tetap macet . #kumparangiveaway #membayangkanjakarta
4.	waowwwww 500t buat jakarta. mending 500t buat ibukota baru coy	waowwwww 500t buat jakarta. mending 500t buat ibukota baru coy
5.	kalau aku sih mikir positifnya aja , kapan lagi akhirnya kalimantan akan maju . selama ini sumber daya alam kaya kita men,etapi pembangun infrastruktur dll larinya kepulau jawa semua . nahhh dgn jd ibukota kita bakalan maju yakan . mudahan berjalan seperti yang diharapkan	kalau aku sih mikir positifnya aja , kapan lagi akhirnya kalimantan akan maju . selama ini sumber daya alam kaya kita men,etapi pembangun infrastruktur dll larinya kepulau jawa semua . nahhh dgn jd ibukota kita bakalan maju yakan . mudahan berjalan seperti yang diharapkan

Tabel 4.4 menunjukkan semua data telah melalui proses menghilangkan URL situs lain dan Gambar. Jika pada data ditemukan kata ‘http’, ‘www’, ‘pic’, dan ‘vid’ maka kata tersebut akan dihilangkan. Pada tabel di atas, ‘https://twitter.com/conan_idn/status/1160912693865275393â’ di baris nomor 1 telah dihilangkan.

2. Menghilangkan Nama Akun atau *Mention*

Untuk menghilangkan atau menghapus nama akun atau *mention*, maka *source code* yang digunakan adalah sebagai berikut:

```
text = re.sub('@[^\s]+', ' ', text)
```

Gambar 4.6 *Source Code* Menghilangkan Nama Akun Atau *Mention*

Gambar 4.6 menunjukan *source code* untuk menghilangkan nama akun aatau *mention*. Dalam *source code* tersebut terlihat bahwa untuk menghilangkan nama akun digunakan fungsi sub dari *package library* re kemudian memasukkan pola yang menggambarkan nama akun yang selalu diawali karakter at (@), karakter yang menggantikan nama akun, dan variabel data yang akan diproses. Contoh hasil dari proses menghilangkan nama akun adalah sebagai berikut:

Tabel 4.5 Contoh Hasil Menghilangkan Nama Akun atau *Mention*

No.	Data Sebelum Mention dihilangkan	Data Setelah Mention dihilangkan
1.	@kompastv segala ibukota d pindahin buang"dana ....mending dananya buat mensejahtrakan masyarakat... kalubukan cukong"yg menang proyek nanti siapa lagi pakkkk	segala ibukota d pindahin buang"dana ....mending dananya buat mensejahtrakan masyarakat... kalubukan cukong"yg menang proyek nanti siapa lagi pakkkk
2.	calon ibukota terpapar asap! kalimantan jg paling banyak titik apinya, banyak lahan gambut+ batubara. waduh! terus, klo di sana di	calon ibukota terpapar asap! kalimantan jg paling banyak titik apinya, banyak lahan gambut+ batubara. waduh! terus, klo di sana di

No.	Data Sebelum Mention dihilangkan	Data Setelah Mention dihilangkan
	tengah hutan mau ngurus apa? problem yg krusial itu manusia, bukan yg lain. penguasa hrs lbh dekat dgn mayoritas rakyatnya. â€¦	tengah hutan mau ngurus apa? problem yg krusial itu manusia, bukan yg lain. penguasa hrs lbh dekat dgn mayoritas rakyatnya. â€¦
3.	ibukota pindah ! jakarta akan tetap seperti sekarang, tetap menjadi pusat bisnis dan ekonomi , pusat hiburan dan magnet bagi orang dari luar jakarta untuk mencari penghidupan yang layak .....dan pastinya jakarta akan tetap macet . #kumparangiveaway #membayangkanjakarta	ibukota pindah ! jakarta akan tetap seperti sekarang, tetap menjadi pusat bisnis dan ekonomi , pusat hiburan dan magnet bagi orang dari luar jakarta untuk mencari penghidupan yang layak .....dan pastinya jakarta akan tetap macet . #kumparangiveaway #membayangkanjakarta
4.	waowwwww 500t buat jakarta. mending 500t buat ibukota baru coy	waowwwww 500t buat jakarta. mending 500t buat ibukota baru coy
5.	kalau aku sih mikir positifnya aja , kapan lagi akhirnya kalimantan akan maju . selama ini sumber daya alam kaya kita men,etapi pembangun infrastruktur dll larinya kepulau jawa semua . nahhh dgn jd ibukota kita	kalau aku sih mikir positifnya aja , kapan lagi akhirnya kalimantan akan maju . selama ini sumber daya alam kaya kita men,etapi pembangun infrastruktur dll larinya kepulau jawa semua . nahhh dgn jd ibukota kita

No.	Data Sebelum Mention dihilangkan	Data Setelah Mention dihilangkan
	bakalan maju yakan . mudahan berjalan seperti yang diharapkan	bakalan maju yakan . mudahan berjalan seperti yang diharapkan

Tabel 4.5 di atas menunjukan data yang telah melewati proses menghilangkan nama akun atau *mention*. Semua data *tweet* tersebut sudah tidak memuat nama akun yang diawali dengan karakter @ yang diikuti dengan huruf. Pada tabel di atas, '@kompastv' di data nomor 1 telah dihilangkan.

3. Menghilangkan Tagar atau *Hashtag*

Untuk menghilangkan atau menghapus tagar atau *hashtag*, maka *source code* yang digunakan adalah sebagai berikut:

```
text = re.sub('#[^\s]+', ' ', text)
```

Gambar 4.7 *Source Code* Menghilangkan Tagar atau *Hashtag*

Gambar 4.7 di atas menunjukkan *source code* untuk melakukan penghilangan tagar atau *hashtag*. Dalam *source code* tersebut terlihat bahwa untuk menghilangkan hashtag digunakan fungsi sub dari *package library* re kemudian memasukkan pola yang menggambarkan tagar atau *hashtag* yang selalu diawali oleh karakter tanda pagar (#) yang kemudian diikuti dengan sebuah kata setelahnya, karakter yang menggantikan tagar, dan variabel data yang akan diproses. Contoh hasil dari proses menghilangkan hashtag adalah sebagai berikut:

Tabel 4.6 Contoh Hasil Menghilangkan Tagar atau *Hashtag*

No.	Data Sebelum *Hashtag* dihilangkan	Data Setelah *Hasgtag* dihilangkan
1.	segala ibukota d pindahin buang"dana ....mending dananya buat mensejahtrakan masyarakat... kalubukan cukong"yg menang proyek nanti siapa lagi pakkkk	segala ibukota d pindahin buang"dana ....mending dananya buat mensejahtrakan masyarakat... kalubukan cukong"yg menang proyek nanti siapa lagi pakkkk
2.	calon ibukota terpapar asap! kalimantan jg paling banyak titik apinya, banyak lahan gambut+ batubara. waduh! terus, klo di sana di tengah hutan mau ngurus apa? problem yg krusial itu manusia, bukan yg lain. penguasa hrs lbh dekat dgn mayoritas rakyatnya. â€¦	calon ibukota terpapar asap! kalimantan jg paling banyak titik apinya, banyak lahan gambut+ batubara. waduh! terus, klo di sana di tengah hutan mau ngurus apa? problem yg krusial itu manusia, bukan yg lain. penguasa hrs lbh dekat dgn mayoritas rakyatnya. â€¦
3.	ibukota pindah ! jakarta akan tetap seperti sekarang, tetap menjadi pusat bisnis dan ekonomi , pusat hiburan dan magnet bagi orang dari luar jakarta untuk mencari penghidupan yang layak .....dan pastinya jakarta akan tetap macet .	ibukota pindah ! jakarta akan tetap seperti sekarang, tetap menjadi pusat bisnis dan ekonomi , pusat hiburan dan magnet bagi orang dari luar jakarta untuk mencari penghidupan yang layak .....dan pastinya jakarta akan tetap macet .

	#kumparangiveaway #membayangkanjakarta	
4.	waowwwww 500t buat jakarta. mending 500t buat ibukota baru coy	waowwwww 500t buat jakarta. mending 500t buat ibukota baru coy
5.	kalau aku sih mikir positifnya aja , kapan lagi akhirnya kalimantan akan maju . selama ini sumber daya alam kaya kita men,etapi pembangun infrastruktur dll larinya kepulau jawa semua . nahhh dgn jd ibukota kita bakalan maju yakan . mudahan berjalan seperti yang diharapkan	kalau aku sih mikir positifnya aja , kapan lagi akhirnya kalimantan akan maju . selama ini sumber daya alam kaya kita men,etapi pembangun infrastruktur dll larinya kepulau jawa semua . nahhh dgn jd ibukota kita bakalan maju yakan . mudahan berjalan seperti yang diharapkan

Tabel 4.6 menunjukkan contoh hasil dari proses setelah *hashtag* dihilangkan. Semua data tersebut sudah tidak memuat hashtag yang diawali karakter '#' yang diikuti dengan sebuah kata setelahnya. Pada tabel di atas, '#kumparangiveaway' dan '#membayangkanjakarta' di data nomor 3 telah dihilangkan.

4. Menghilangkan Angka

Untuk menghilangkan atau menghapus angka, maka *source code* yang digunakan adalah sebagai berikut:

```
text = re.sub('\w*\d\w*',' ',text)
```

Gambar 4.8 *Source Code* Menghilangkan Angka

Gambar 4.8 menunjukan *source code* untuk menghilangkan karakter angka. Dalam *source code* tersebut terlihat bahwa untuk menghilangkan karakter angka digunakan fungsi sub dari *package library re* kemudian memasukkan pola yang menggambarkan karakter angka, karakter yang menggantikan angka, dan variabel data yang akan diproses. Contoh hasil dari proses menghilangkan karakter angka adalah sebagai berikut:

Tabel 4.7 Contoh Hasil Menghilangkan Angka

No.	Data Sebelum Angka dihilangkan	Data Setelah Angka dihilangkan
1.	segala ibukota d pindahin buang"dana ....mending dananya buat mensejahtrakan masyarakat... kalubukan cukong"yg menang proyek nanti siapa lagi pakkkk	segala ibukota d pindahin buang"dana ....mending dananya buat mensejahtrakan masyarakat... kalubukan cukong"yg menang proyek nanti siapa lagi pakkkk
2.	calon ibukota terpapar asap! kalimantan jg paling banyak titik apinya, banyak lahan gambut+ batubara. waduh! terus, klo di sana di tengah hutan mau ngurus apa? problem yg krusial itu manusia, bukan yg lain. penguasa hrs lbh dekat dgn mayoritas rakyatnya. â€¦	calon ibukota terpapar asap! kalimantan jg paling banyak titik apinya, banyak lahan gambut+ batubara. waduh! terus, klo di sana di tengah hutan mau ngurus apa? problem yg krusial itu manusia, bukan yg lain. penguasa hrs lbh dekat dgn mayoritas rakyatnya. â€¦

3.	ibukota pindah ! jakarta akan tetap seperti sekarang, tetap menjadi pusat bisnis dan ekonomi , pusat hiburan dan magnet bagi orang dari luar jakarta untuk mencari penghidupan yang layak .....dan pastinya jakarta akan tetap macet .	ibukota pindah ! jakarta akan tetap seperti sekarang, tetap menjadi pusat bisnis dan ekonomi , pusat hiburan dan magnet bagi orang dari luar jakarta untuk mencari penghidupan yang layak .....dan pastinya jakarta akan tetap macet .
4.	waowwwww 500t buat jakarta. mending 500t buat ibukota baru coy	waowwwww t buat jakarta. mending t buat ibukota baru coy
5.	kalau aku sih mikir positifnya aja , kapan lagi akhirnya kalimantan akan maju . selama ini sumber daya alam kaya kita men,etapi pembangun infrastruktur dll larinya kepulau jawa semua . nahhh dgn jd ibukota kita bakalan maju yakan . mudahan berjalan seperti yang diharapkan	kalau aku sih mikir positifnya aja , kapan lagi akhirnya kalimantan akan maju . selama ini sumber daya alam kaya kita men,etapi pembangun infrastruktur dll larinya kepulau jawa semua . nahhh dgn jd ibukota kita bakalan maju yakan . mudahan berjalan seperti yang diharapkan

Tabel 4.7 menunjukan data yang telah melewati proses setelah karakter angka dihilangkan. Dalam tabel tersebut, karakter angka ‘500’ pada baris nomor 4 telah dihilangkan.

5. Menghilangkan Tanda Baca

Untuk menghilangkan atau menghapus tanda baca, maka *source code* yang digunakan adalah sebagai berikut:

```
text = re.sub('\[.*?\]',' ',text)
```

Gambar 4.9 *Source Code* Menghilangkan Tanda Baca

Gambar 4.9 menunjukan *source code* untuk menghilangkan karakter tanda baca. Dalam *source code* tersebut terlihat bahwa untuk menghilangkan karakter tanda baca digunakan fungsi sub dari *package library re* kemudian memasukkan pola yang menggambarkan karakter tanda baca, karakter yang menggantikan tanda baca, dan variabel data yang akan diproses. Contoh hasil dari proses menghilangkan karakter tanda baca adalah sebagai berikut:

Tabel 4.8 Contoh Hasil Menghilangkan Tanda Baca

No.	Data Sebelum Tanda Baca dihilangkan	Data Setelah Tanda Baca dihilangkan
1.	segala ibukota d pindahin buang"dana ....mending dananya buat mensejahtrakan masyarakat... kalubukan cukong"yg menang proyek nanti siapa lagi pakkkk	segala ibukota d pindahin buang dana mending dananya buat mensejahtrakan masyarakat kalubukan cukong yg menang proyek nanti siapa lagi pakkkk
2.	calon ibukota terpapar asap! kalimantan jg paling banyak titik apinya, banyak lahan gambut+ batubara. waduh! terus, klo di sana di	calon ibukota terpapar asap kalimantan jg paling banyak titik apinya banyak lahan gambut batubara waduh terus klo di sana di

No.	Data Sebelum Tanda Baca dihilangkan	Data Setelah Tanda Baca dihilangkan
	tengah hutan mau ngurus apa? problem yg krusial itu manusia, bukan yg lain. penguasa hrs lbh dekat dgn mayoritas rakyatnya. â€¦	tengah hutan mau ngurus apa problem yg krusial itu manusia bukan yg lain penguasa hrs lbh dekat dgn mayoritas rakyatnya â€¦
3.	ibukota pindah ! jakarta akan tetap seperti sekarang, tetap menjadi pusat bisnis dan ekonomi , pusat hiburan dan magnet bagi orang dari luar jakarta untuk mencari penghidupan yang layak .....dan pastinya jakarta akan tetap macet .	ibukota pindah jakarta akan tetap seperti sekarang tetap menjadi pusat bisnis dan ekonomi pusat hiburan dan magnet bagi orang dari luar jakarta untuk mencari penghidupan yang layak dan pastinya jakarta akan tetap macet
4.	waowwwww t buat jakarta. mending t buat ibukota baru coy	waowwwww t buat jakarta mending t buat ibukota baru coy
5.	kalau aku sih mikir positifnya aja , kapan lagi akhirnya kalimantan akan maju . selama ini sumber daya alam kaya kita men,etapi pembangun infrastruktur dll larinya kepulau jawa semua . nahhh dgn jd ibukota kita	kalau aku sih mikir positifnya aja kapan lagi akhirnya kalimantan akan maju selama ini sumber daya alam kaya kita men etapi pembangun infrastruktur dll larinya kepulau jawa semua nahhh dgn jd ibukota kita

No.	Data Sebelum Tanda Baca dihilangkan	Data Setelah Tanda Baca dihilangkan
	bakalan maju yakan . mudahan berjalan seperti yang diharapkan	bakalan maju yakan mudahan berjalan seperti yang diharapkan

Tabel 4.8 menunjukan data yang telah melewati proses setelah karakter tanda baca dihilangkan. Dalam tabel tersebut, karakter tanda baca terdapat pada baris 1-5 seperti '....', '!', '+', ',' dan telah dihilangkan.

6. Menghilangkan Emotikon

Dari data yang didapatkan, emotikon terbentuk dari karakter-karakter tanda baca. Sehingga ketika melakukan pembersihan tanda baca, maka karakter yang membentuk emotikon otomatis terhapus. Oleh karena itu, tidak ada perlakuan khusus atau *souce code* untuk menghilangkan tanda baca.

7. Menghilangkan Huruf Berulang

Untuk menghilangkan atau menghapus huruf berulang, maka *source code* yang digunakan adalah sebagai berikut:

```
text = re.sub(r'([a-z])\1+', r'\1', text)
```

Gambar 4.10 *Source Code* Menghilangkan Huruf Berulang

Gambar 4.10 menunjukan *source code* untuk menghilangkan huruf berulang. Dalam *source code* tersebut terlihat bahwa untuk menghilangkan karakter huruf berulang digunakan fungsi sub dari *package library re* kemudian memasukkan pola yang menggambarkan huruf berulang, karakter yang

menggantikan huruf berulang, dan variabel data yang akan diproses. Contoh hasil dari proses menghilangkan huruf berulang adalah sebagai berikut:

Tabel 4.9 Contoh Hasil Menghilangkan Huruf Berulang

No.	Data Sebelum Huruf Berulang dihilangkan	Data Setelah Huruf Berulang dihilangkan
1.	segala ibukota d pindahin buang dana mending dananya buat mensejahtrakan masyarakat kalubukan cukong yg menang proyek nanti siapa lagi pakkkk	segala ibukota d pindahin buang dana mending dananya buat mensejahtrakan masyarakat kalubukan cukong yg menang proyek nanti siapa lagi pak
2.	calon ibukota terpapar asap kalimantan jg paling banyak titik apinya banyak lahan gambut batubara waduh terus klo di sana di tengah hutan mau ngurus apa problem yg krusial itu manusia bukan yg lain penguasa hrs lbh dekat dgn mayoritas rakyatnya â€¦	calon ibukota terpapar asap kalimantan jg paling banyak titik apinya banyak lahan gambut batubara waduh terus klo di sana di tengah hutan mau ngurus apa problem yg krusial itu manusia bukan yg lain penguasa hrs lbh dekat dgn mayoritas rakyatnya â€¦
3.	ibukota pindah jakarta akan tetap seperti sekarang tetap menjadi pusat bisnis dan ekonomi pusat hiburan dan magnet bagi orang dari luar jakarta untuk	ibukota pindah jakarta akan tetap seperti sekarang tetap menjadi pusat bisnis dan ekonomi pusat hiburan dan magnet bagi orang dari luar jakarta untuk

No.	Data Sebelum Huruf Berulang dihilangkan	Data Setelah Huruf Berulang dihilangkan
	mencari penghidupan yang layak dan pastinya jakarta akan tetap macet	mencari penghidupan yang layak dan pastinya jakarta akan tetap macet
4.	waowwwww t buat jakarta mending t buat ibukota baru coy	waow t buat jakarta mending t buat ibukota baru coy
5.	kalau aku sih mikir positifnya aja kapan lagi akhirnya kalimantan akan maju selama ini sumber daya alam kaya kita men etapi pembangun infrastruktur dll larinya kepulau jawa semua nahhh dgn jd ibukota kita bakalan maju yakan mudahan berjalan seperti yang diharapkan	kalau aku sih mikir positifnya aja kapan lagi akhirnya kalimantan akan maju selama ini sumber daya alam kaya kita men etapi pembangun infrastruktur dl larinya kepulau jawa semua nah dgn jd ibukota kita bakalan maju yakan mudahan berjalan seperti yang diharapkan

Tabel 4.9 menunjukan data yang telah melewati proses setelah huruf berulang dihilangkan. Dalam tabel tersebut, huruf berulang terdapat pada baris 1 seperti ‘pakkkk’ menjadi ‘pak’, baris 4 seperti ‘waowwwww’ menjadi ‘waow’ dan baris 5 seperti ‘dll’ menjadi ‘dl’, ‘nahhh’ menjadi ‘nah’.

8. Menghilangkan Huruf Tunggal

Untuk menghilangkan atau menghapus huruf tunggal, maka *source code* yang digunakan adalah sebagai berikut:

```
text = re.sub(r'\b[a-zA-Z]\b', ' ', text)
```

Gambar 4.11 *Source Code* Menghilangkan Huruf Tunggal

Gambar 4.11 menunjukan *source code* untuk menghilangkan huruf tunggal. Dalam *source code* tersebut terlihat bahwa untuk menghilangkan karakter huruf tunggal digunakan fungsi sub dari *package library re* kemudian memasukkan pola yang menggambarkan huruf tunggal, karakter yang menggantikan huruf tunggal, dan variabel data yang akan diproses. Contoh hasil dari proses menghilangkan huruf tunggal adalah sebagai berikut:

Tabel 4.10 Contoh Hasil Menghilangkan Huruf Tunggal

No.	Data Sebelum Huruf Tunggal dihilangkan	Data Setelah Huruf Tunggal dihilangkan
1.	segala ibukota d pindahin buang dana mending dananya buat mensejahtrakan masyarakat kalubukan cukong yg menang proyek nanti siapa lagi pak	segala ibukota pindahin buang dana mending dananya buat mensejahtrakan masyarakat kalubukan cukong yg menang proyek nanti siapa lagi pak
2.	calon ibukota terpapar asap kalimantan jg paling banyak titik apinya banyak lahan gambut batubara waduh terus klo di sana di tengah hutan mau ngurus apa problem yg krusial itu manusia bukan yg lain	calon ibukota terpapar asap kalimantan jg paling banyak titik apinya banyak lahan gambut batubara waduh terus klo di sana di tengah hutan mau ngurus apa problem yg krusial itu manusia bukan yg lain

No.	Data Sebelum Huruf Tunggal dihilangkan	Data Setelah Huruf Tunggal dihilangkan
	penguasa hrs lbh dekat dgn mayoritas rakyatnya â€¦	penguasa hrs lbh dekat dgn mayoritas rakyatnya â€¦
3.	ibukota pindah jakarta akan tetap seperti sekarang tetap menjadi pusat bisnis dan ekonomi pusat hiburan dan magnet bagi orang dari luar jakarta untuk mencari penghidupan yang layak dan pastinya jakarta akan tetap macet	ibukota pindah jakarta akan tetap seperti sekarang tetap menjadi pusat bisnis dan ekonomi pusat hiburan dan magnet bagi orang dari luar jakarta untuk mencari penghidupan yang layak dan pastinya jakarta akan tetap macet
4.	waow t buat jakarta mending t buat ibukota baru coy	waow buat jakarta mending buat ibukota baru coy
5.	kalau aku sih mikir positifnya aja kapan lagi akhirnya kalimantan akan maju selama ini sumber daya alam kaya kita men etapi pembangun infrastruktur dl larinya kepulau jawa semua nah dgn jd ibukota kita bakalan maju yakan mudahan berjalan seperti yang diharapkan	kalau aku sih mikir positifnya aja kapan lagi akhirnya kalimantan akan maju selama ini sumber daya alam kaya kita men etapi pembangun infrastruktur dl larinya kepulau jawa semua nah dgn jd ibukota kita bakalan maju yakan mudahan berjalan seperti yang diharapkan

Tabel 4.10 menunjukan data yang telah melewati proses setelah huruf tunggal dihilangkan. Dalam tabel tersebut, huruf tunggal terdapat pada baris 1 seperti ‘d’ dan baris 4 seperti ‘t’ telah dihilangkan.

9. Menghilangkan Spasi Berlebih

Untuk menghilangkan atau menghapus spasi berlebih, maka *source code* yang digunakan adalah sebagai berikut:

```
text = re.sub('[\s]+', " ", text)
```

Gambar 4.12 *Source Code* Menghilangkan Spasi Berlebih

Gambar 4.12 menunjukan *source code* untuk menghilangkan spasi berlebih. Dalam *source code* tersebut terlihat bahwa untuk menghilangkan karakter spasi berlebih digunakan fungsi sub dari *package library re* kemudian memasukkan pola yang menggambarkan spasi berlebih, karakter yang menggantikan spasi berlebih, dan variabel data yang akan diproses. Contoh hasil dari proses menghilangkan spasi berlebih adalah sebagai berikut:

Tabel 4.11 Contoh Hasil Menghilangkan Spasi Berlebih

No.	Data Sebelum Spasi Berlebih dihilangkan	Data Seteleh Spasi Berlebih dihilangkan
1.	segala ibukota pindahin buang dana mending dananya buat mensejahtrakan masyarakat kalubukan cukong yg menang proyek nanti siapa lagi pak	segala ibukota pindahin buang dana mending dananya buat mensejahtrakan masyarakat kalubukan cukong yg menang proyek nanti siapa lagi pak

No.	Data Sebelum Spasi Berlebih dihilangkan	Data Seteleh Spasi Berlebih dihilangkan
2.	calon ibukota terpapar asap kalimantan jg paling banyak titik apinya banyak lahan gambut batubara waduh terus klo di sana di tengah hutan mau ngurus apa problem yg krusial itu manusia bukan yg lain penguasa hrs lbh dekat dgn mayoritas rakyatnya â€¦	calon ibukota terpapar asap kalimantan jg paling banyak titik apinya banyak lahan gambut batubara waduh terus klo di sana di tengah hutan mau ngurus apa problem yg krusial itu manusia bukan yg lain penguasa hrs lbh dekat dgn mayoritas rakyatnya â€¦
3.	ibukota pindah jakarta akan tetap seperti sekarang tetap menjadi pusat bisnis dan ekonomi pusat hiburan dan magnet bagi orang dari luar jakarta untuk mencari penghidupan yang layak dan pastinya jakarta akan tetap macet	ibukota pindah jakarta akan tetap seperti sekarang tetap menjadi pusat bisnis dan ekonomi pusat hiburan dan magnet bagi orang dari luar jakarta untuk mencari penghidupan yang layak dan pastinya jakarta akan tetap macet
4.	waow buat jakarta mending buat ibukota baru coy	waow buat jakarta mending buat ibukota baru coy
5.	kalau aku sih mikir positifnya aja kapan lagi akhirnya kalimantan akan maju selama ini sumber daya alam	kalau aku sih mikir positifnya aja kapan lagi akhirnya kalimantan akan maju selama ini sumber daya alam

No.	Data Sebelum Spasi Berlebih dihilangkan	Data Seteleh Spasi Berlebih dihilangkan
	kaya kita men etapi pembangun infrastruktur dl larinya kepulau jawa semua nah dgn jd ibukota kita bakalan maju yakan mudahan berjalan seperti yang diharapkan	kaya kita men etapi pembangun infrastruktur dl larinya kepulau jawa semua nah dgn jd ibukota kita bakalan maju yakan mudahan berjalan seperti yang diharapkan

Tabel 4.11 menunjukan data yang telah melewati proses setelah spasi berlebih dihilangkan. Dalam tabel tersebut, spasi berlebih terdapat pada baris 1-5 dan telah dihilangkan.

10. Menghilangkan Baris Baru

Untuk menghilangkan atau menghapus baris baru atau (*newline*), maka *source code* yang digunakan adalah sebagai berikut:

```
text = re.sub(' +', " ", text)
```

Gambar 4.13 *Source Code* Menghilangkan Baris Baru

Gambar 4.13 menunjukan *source code* untuk menghilangkan baris baru. Dalam *source code* tersebut terlihat bahwa untuk menghilangkan karakter baris baru digunakan fungsi sub dari *package library re* kemudian memasukkan pola yang menggambarkan baris baru, karakter yang menggantikan baris baru, dan variabel data yang akan diproses. Contoh hasil dari proses menghilangkan baris baru adalah sebagai berikut:

Tabel 4.12 Contoh Hasil Menghilangkan Baris Baru

No.	Data Sebelum Baris Baru dihilangkan	Data Seteleh Baris Baru dihilangkan
1.	segala ibukota pindahin buang dana mending dananya buat mensejahtrakan masyarakat kalubukan cukong yg menang proyek nanti siapa lagi pak	segala ibukota pindahin buang dana mending dananya buat mensejahtrakan masyarakat kalubukan cukong yg menang proyek nanti siapa lagi pak
2.	calon ibukota terpapar asap kalimantan jg paling banyak titik apinya banyak lahan gambut batubara waduh terus klo di sana di tengah hutan mau ngurus apa problem yg krusial itu manusia bukan yg lain penguasa hrs lbh dekat dgn mayoritas rakyatnya â€¦	calon ibukota terpapar asap kalimantan jg paling banyak titik apinya banyak lahan gambut batubara waduh terus klo di sana di tengah hutan mau ngurus apa problem yg krusial itu manusia bukan yg lain penguasa hrs lbh dekat dgn mayoritas rakyatnya â€¦
3.	ibukota pindah jakarta akan tetap seperti sekarang tetap menjadi pusat bisnis dan ekonomi pusat hiburan dan magnet bagi orang dari luar jakarta untuk mencari penghidupan yang layak dan pastinya jakarta akan tetap macet	ibukota pindah jakarta akan tetap seperti sekarang tetap menjadi pusat bisnis dan ekonomi pusat hiburan dan magnet bagi orang dari luar jakarta untuk mencari penghidupan yang layak dan pastinya jakarta akan tetap macet

No.	Data Sebelum Baris Baru dihilangkan	Data Seteleh Baris Baru dihilangkan
4.	waow buat jakarta mending buat ibukota baru coy	waow buat jakarta mending buat ibukota baru coy
5.	kalau aku sih mikir positifnya aja kapan lagi akhirnya kalimantan akan maju selama ini sumber daya alam kaya kita men etapi pembangun infrastruktur dl larinya kepulau jawa semua nah dgn jd ibukota kita bakalan maju yakan mudahan berjalan seperti yang diharapkan	kalau aku sih mikir positifnya aja kapan lagi akhirnya kalimantan akan maju selama ini sumber daya alam kaya kita men etapi pembangun infrastruktur dl larinya kepulau jawa semua nah dgn jd ibukota kita bakalan maju yakan mudahan berjalan seperti yang diharapkan

Tabel 4.12 menunjukan data yang telah melewati proses setelah baris baru dihilangkan. Dalam tabel tersebut, spasi berlebih terdapat pada baris 1, 3, dan 4 telah dihilangkan.

C. Mengubah Kata Singkatan (Abreviasi)

Sesuai dengan langkah yang telah dijelaskan pada bab III, yakni mengubah kata singkatan atau abreviasi. Untuk mengubah kata singkatan, maka *source code* yang digunakan adalah sebagai berikut:

```
def translator(user_string):
    user_string = user_string.split(" ")
    j = 0
    for _str in user_string:
        fileName = "colloquial-indonesian-lexicon.csv"
        accessMode = "r"
        with open(fileName, accessMode) as myCSVfile:
            dataFromFile = csv.reader(myCSVfile,
delimiter=",")

            str = re.sub('[^a-zA-Z0-9-.]', '', _str)
            for row in dataFromFile:


                if _str.lower() == row[0]:
                    user_string[j] = row[1]
            myCSVfile.close()
        j = j + 1

    a = (' '.join(user_string))
    print(a)
    return a

abreviasi = lambda x: translator(x)
```

Gambar 4.14 *Source Code* Mengubah Kata Singkatan

Gambar 4.14 menunjukan *source code* untuk mengubah kata singkatan. Proses mengubah kata singkatan (abreviasi) dilakukan dengan menggunakan kamus 'colloquial-indonesian-lexicon.csv' yang telah diunggah ke anaconda. Cara kerja dari code tersebut yakni dengan membaca satu persatu data '*tweet*' kemudian disesuaikan dengan data yang terdapat pada kamus 'colloquial-indonesian-lexicon.csv'. Apabila kata pada data *tweet* terdapat pada kamus, maka singkatan yang ada pada data *tweet* digantikan dengan kata yang sebenarnya (bukan singkatan). Contoh hasil dari proses mengubah kata singkatan adalah sebagai berikut:

Tabel 4.13 Contoh Hasil Mengubah Kata Singkatan

No.	Data Sebelum Singkatan diubah	Data Seteleh Singkatan diubah
1.	segala ibukota pindahin buang dana mending dananya buat mensejahtrakan	segala ibukota pindahin buang dana mending dananya buat mensejahtrakan

No.	Data Sebelum Singkatan diubah	Data Seteleh Singkatan diubah
	masyarakat kalubukan cukong yg menang proyek nanti siapa lagi pak	masyarakat kalubukan cukong yang menang proyek nanti siapa lagi pak
2.	calon ibukota terpapar asap kalimantan jg paling banyak titik apinya banyak lahan gambut batubara waduh terus klo di sana di tengah hutan mau ngurus apa problem yg krusial itu manusia bukan yg lain penguasa hrs lbh dekat dgn mayoritas rakyatnya â€¦	calon ibukota terpapar asap kalimantan juga paling banyak titik apinya banyak lahan gambut batubara waduh terus kalo di sana di tengah hutan mau ngurus apa problem yang krusial itu manusia bukan yang lain penguasa harus lebih dekat dengan mayoritas rakyatnya â€¦
3.	ibukota pindah jakarta akan tetap seperti sekarang tetap menjadi pusat bisnis dan ekonomi pusat hiburan dan magnet bagi orang dari luar jakarta untuk mencari penghidupan yang layak dan pastinya jakarta akan tetap macet	ibukota pindah jakarta akan tetap seperti sekarang tetap menjadi pusat bisnis dan ekonomi pusat hiburan dan magnet bagi orang dari luar jakarta untuk mencari penghidupan yang layak dan pastinya jakarta akan tetap macet
4.	waow buat jakarta mending buat ibukota baru coy	waow buat jakarta mending buat ibukota baru coy
5.	kalau aku sih mikir positifnya aja kapan lagi akhirnya kalimantan akan maju selama ini sumber daya alam kaya kita men etapi pembangun	kalau aku sih mikir positifnya aja kapan lagi akhirnya kalimantan akan maju selama ini sumber daya alam kaya kita men etapi pembangun

No.	Data Sebelum Singkatan diubah	Data Seteleh Singkatan diubah
	infrastruktur dl larinya kepulau jawa semua nah dgn jd ibukota kita bakalan maju yakan mudahan berjalan seperti yang diharapkan	infrastruktur dulu larinya kepulau jawa semua nah dengan jadi ibukota kita bakalan maju yakan mudahan berjalan seperti yang diharapkan

Tabel 4.13 di atas menunjukan data yang telah melewati proses pengubahan kata singkatan atau abreviasi. Dalam tabel tersebut, perubahan kata singkatan dapat dilihat pada baris 1 dan 2 seperti ‘yg’ menjadi ‘yang’, baris 5 seperti ‘dgn’ menjadi ‘dengan’, ‘dl’ menjadi ‘dulu’, dan ‘jd’ menjadi ‘jadi’.

D. Tokenisasi

Sesuai dengan langkah yang telah dijelaskan pada bab III, setelah mengubah kata singkatan selanjutnya yakni tokenisasi. Untuk melakukan tokenisasi, maka *source code* yang digunakan adalah sebagai berikut:

```
tokens = word_tokenize(text)
```

Gambar 4.15 *Source Code* Tokenisasi

Gambar 4.15 menunjukan *source code* untuk melakukan tokenisasi. Proses tokenisasi digunakan untuk memisahkan kata dalam sebuah *tweet* dengan koma. Dalam melakukan tokenisasi, digunakan fungsi word_tokenize dari *package library* NLTK kemudian memasukkan variabel data yang akan diproses, dan hasilnya ditampung pada variabel text. Contoh hasil dari proses tokenisasi adalah sebagai berikut:

Tabel 4.14 Contoh Hasil Tokenisasi

No.	Data Sebelum dilakukan Tokenisasi	Data Seteleh dilakukan Tokenisasi
1.	segala ibukota pindahin buang dana mending dananya buat mensejahtrakan masyarakat kalubukan cukong yang menang proyek nanti siapa lagi pak	'segala', 'ibukota', 'pindahin', 'buang', 'dana', 'mending', 'dananya', 'buat', 'mensejahtrakan', 'masyarakat', 'kalubukan', 'cukong', 'yang', 'menang', 'proyek', 'nanti', 'siapa', 'lagi', 'pak'
2.	calon ibukota terpapar asap kalimantan juga paling banyak titik apinya banyak lahan gambut batubara waduh terus kalo di sana di tengah hutan mau ngurus apa problem yang krusial itu manusia bukan yang lain penguasa harus lebih dekat dengan mayoritas rakyatnya â€¦	'calon', 'ibukota', 'terpapar', 'asap', 'kalimantan', 'juga', 'paling', 'banyak', 'titik', 'apinya', 'banyak', 'lahan', 'gambut', 'batubara', 'waduh', 'terus', 'kalo', 'di', 'sana', 'di', 'tengah', 'hutan', 'mau', 'ngurus', 'apa', 'problem', 'yang', 'krusial', 'itu', 'manusia', 'bukan', 'yang', 'lain', 'penguasa', 'harus', 'lebih', 'dekat', 'dengan', 'mayoritas', 'rakyatnya', 'â€¦'
3.	ibukota pindah jakarta akan tetap seperti sekarang tetap menjadi pusat bisnis dan ekonomi pusat hiburan dan magnet bagi orang dari luar jakarta untuk mencari penghidupan yang layak dan pastinya jakarta akan tetap macet	'ibukota', 'pindah', 'jakarta', 'akan', 'tetap', 'seperti', 'sekarang', 'tetap', 'menjadi', 'pusat', 'bisnis', 'dan', 'ekonomi', 'pusat', 'hiburan', 'dan', 'magnet', 'bagi', 'orang', 'dari', 'luar', 'jakarta', 'untuk', 'mencari', 'penghidupan', 'yang', 'layak', 'dan',

No.	Data Sebelum dilakukan Tokenisasi	Data Seteleh dilakukan Tokenisasi
		'pastinya', 'jakarta', 'akan', 'tetap', 'macet'
4.	waow buat jakarta mending buat ibukota baru coy	'waow', 'buat', 'jakarta', 'mending', 'buat', 'ibukota', 'baru', 'coy'
5.	kalau aku sih mikir positifnya aja kapan lagi akhirnya kalimantan akan maju selama ini sumber daya alam kaya kita men etapi pembangun infrastruktur dulu larinya kepulau jawa semua nah dengan jadi ibukota kita bakalan maju yakan mudahan berjalan seperti yang diharapkan	'kalau', 'aku', 'sih', 'mikir', 'positifnya', 'aja', 'kapan', 'lagi', 'akhirnya', 'kalimantan', 'akan', 'maju', 'selama', 'ini', 'sumber', 'daya', 'alam', 'kaya', 'kita', 'men', 'etapi', 'pembangun', 'infrastruktur', 'dulu', 'larinya', 'kepulau', 'jawa', 'semua', 'nah', 'dengan', 'jadi', 'ibukota', 'kita', 'bakalan', 'maju', 'yakan', 'mudahan', 'berjalan', 'seperti', 'yang', 'diharapkan',

Tabel 4.14 menunjukkan contoh hasil dari proses tokenisasi. Data yang telah ditokenisasi akan terpisah kata per kata.

E. Stopword Removal

Setelah melakukan tokenisasi, langkah selanjutnya adalah melakukan *stopword removal*, *source code* yang digunakan adalah sebagai berikut:

```
def get_stopword(stopwordsfile):
    stopwords=[]
    file_stopwords = open(stopwordsfile,'r')
    row = file_stopwords.readline()
    while row:
        word = row.strip()
        stopwords.append(word)
        row = file_stopwords.readline()
    file_stopwords.close()
    return stopwords

stop_words_indo = get_stopword('stopwordlist.txt')

def stopwords(text):
    tokens = word_tokenize(text)
    filtered = []

    for w in tokens:
        if w not in stop_words_indo:
            filtered.append(w)

    hasil = ' '.join(filtered)
    return hasil


st = lambda x: stopwords(x)
```

Gambar 4.16 *Source Code Stopword Removal*

Gambar 4.16 di atas merupakan *source code* untuk melakukan *stopword removal*. Setiap kata dalam data akan diperiksa apakah memuat stopwords atau tidak, apabila kata tersebut adalah mengandung *stopword* maka kata tersebut akan dihilangkan. Setelah semua kata yang bukan stopword terkumpul, maka kata-kata tersebut ditampung dalam variabel text kemudian digabungkan kembali menjadi kalimat dengan fungsi join dan hasilnya ditampung dalam variabel hasil. Contoh hasil dari proses *stopword removal* adalah sebagai berikut:

Tabel 4.15 Contoh Hasil *Stopword Removal*

No.	Data Sebelum dilakukan Stopword Removal	Data Seteleh dilakukan Stopword Removal
1.	'segala', 'ibukota', 'pindahin', 'buang', 'dana', 'mending', 'dananya', 'buat',	ibukota pindahin buang dana mending dananya buat mensejahtrakan

No.	Data Sebelum dilakukan Stopword Removal	Data Seteleh dilakukan Stopword Removal
	'mensejahtrakan', 'masyarakat', 'kalubukan', 'cukong', 'yang', 'menang', 'proyek', 'nanti', 'siapa', 'lagi', 'pak'	masyarakat kalubukan cukong menang proyek
2.	'calon', 'ibukota', 'terpapar', 'asap', 'kalimantan', 'juga', 'paling', 'banyak', 'titik', 'apinya', 'banyak', 'lahan', 'gambut', 'batubara', 'waduh', 'terus', 'kalo', 'di', 'sana', 'di', 'tengah', 'hutan', 'mau', 'ngurus', 'apa', 'problem', 'yang', 'krusial', 'itu', 'manusia', 'bukan', 'yang', 'lain', 'penguasa', 'harus', 'lebih', 'dekat', 'dengan', 'mayoritas', 'rakyatnya', 'â€¦'	calon ibukota terpapar asap kalimantan titik apinya banyak lahan gambut batubara kalo hutan ngurus apa problem yang krusial itu manusia penguasa mayoritas rakyatnya
3.	'ibukota', 'pindah', 'jakarta', 'akan', 'tetap', 'seperti', 'sekarang', 'tetap', 'menjadi', 'pusat', 'bisnis', 'dan', 'ekonomi', 'pusat', 'hiburan', 'dan', 'magnet', 'bagi', 'orang', 'dari', 'luar', 'jakarta', 'untuk', 'mencari', 'penghidupan', 'yang', 'layak', 'dan', 'pastinya', 'jakarta', 'akan', 'tetap', 'macet'	ibukota pindah jakarta pusat bisnis ekonomi pusat hiburan magnet orang jakarta mencari penghidupan layak jakarta macet

No.	Data Sebelum dilakukan Stopword Removal	Data Seteleh dilakukan Stopword Removal
4.	'waow', 'buat', 'jakarta', 'mending', 'buat', 'ibukota', 'baru', 'coy'	waow jakarta mending ibukota baru
5.	'kalau', 'aku', 'sih', 'mikir', 'positifnya', 'aja', 'kapan', 'lagi', 'akhirnya', 'kalimantan', 'akan', 'maju', 'selama', 'ini', 'sumber', 'daya', 'alam', 'kaya', 'kita', 'men', 'etapi', 'pembangun', 'infrastruktur', 'dulu', 'larinya', 'kepulau', 'jawa', 'semua', 'nah', 'dengan', 'jadi', 'ibukota', 'kita', 'bakalan', 'maju', 'yakan', 'mudahan', 'berjalan', 'seperti', 'yang', 'diharapkan',	mikir positifnya kalimantan maju sumber daya alam kaya pembangun infrastruktur larinya kepulau jawa ibukota maju mudahan berjalan diharapkan

Tabel 4.15 menunjukkan contoh hasil dari proses *stopword removal*. Data *tweet* yang mengandung kata pada *stopword list* dihilangkan. Apabila tidak ada di *stopword list*, maka data dibiarkan.

F. Stemming

Sesuai pada bab III, setelah melakukan *stopword removal* yakni melakukan *stemming*. Untuk melakukan *stemming*, *source code* yang digunakan adalah sebagai berikut:

```
def stemming(text):

    factory_stem = StemmerFactory()
    stemmer = factory_stem.create_stemmer()
    text = stemmer.stem(text)
    return text

stem = lambda x: stemming(x)
```

Gambar 4.17 *Source Code Stemming*

Gambar 4.17 di atas merupakan *source code* untuk melakukan *stemming*. Prosesnya yakni mendeklarasikan objek dari StemmerFactory() dan memanggil fungsi create_stemmer(). Kemudian ditampung pada objek yang bernama stemmer, dan data diproses.

Tabel 4.16 Contoh Hasil *Stemming*

No.	Data Sebelum dilakukan Stemming	Data Seteleh dilakukan Stemming
1.	ibukota pindahin buang dana mending dananya buat mensejahtrakan masyarakat kalubukan cukong menang proyek	ibukota pindah buang dana mending dana sejahtra masyarakat kalubuk cukong menang proyek
2.	calon ibukota terpapar asap kalimantan titik apinya banyak lahan gambut batubara kalo hutan ngurus apa problem yang krusial itu manusia penguasa mayoritas rakyatnya	calon ibukota papar asap kalimantan titik api lahan gambut batubara kalo hutan urus problem krusial manusia kuasa mayoritas rakyat
3.	ibukota pindah jakarta pusat bisnis ekonomi pusat hiburan magnet orang jakarta mencari penghidupan layak jakarta macet	ibukota pindah jakarta pusat bisnis ekonomi pusat hibur magnet orang jakarta cari hidup layak jakarta macet

No.	Data Sebelum dilakukan Stemming	Data Seteleh dilakukan Stemming
4.	waow jakarta mending ibukota baru	waow jakarta mending ibukota baru
5.	mikir positifnya kalimantan maju sumber daya alam kaya etapi pembangun infrastruktur larinya kepulau jawa ibukota maju mudahan berjalan diharapkan	mikir positif kalimantan maju sumber daya alam kaya bangun infrastruktur lari pulau jawa ibukota maju yakan mudah jalan harap

Tabel 4.16 menunjukkan contoh hasil dari proses *stemming*. Data *tweet* akan diubah menjadi kata dasar sesuai dengan kamus yang telah ditentukan.

G. Pembobotan TF-IDF

Setelah melakukan *stemming*, selanjutnya yakni melakukan pembobotan. Untuk melakukan pembobotan, *source code* yang digunakan adalah sebagai berikut:

```
def get_stopword(stopwordsfile):
    stopwords=[]
    file_stopwords = open(stopwordsfile,'r')
    row = file_stopwords.readline()
    while row:
        word = row.strip()
        stopwords.append(word)
        row = file_stopwords.readline()
    file_stopwords.close()
    return stopwords

stop_words_indo = get_stopword('stopwordlist.txt')

vectorizer = TfidfVectorizer(use_idf=True, lowercase=True,
strip_accents='ascii', stop_words=stop_words_indo)
```

Gambar 4.18 *Source Code* Pembobotan

Gambar 4.18 di atas merupakan *source code* untuk melakukan pembobotan. Sehingga akan dihasilkan rincian matriks sebagai berikut:

```
TfidfVectorizer(analyzer='word', binary=False, decode_error='strict',
                dtype=<class 'numpy.float64'>, encoding='utf-8',
                input='content', lowercase=True, max_df=1.0, max_features=None,
                min_df=1, ngram_range=(1, 1), norm='l2', preprocessor=None,
                smooth_idf=True,
                stop_words=['ada', 'adalah', 'adanya', 'adapun', 'agak',
                            'agaknya', 'agar', 'akan', 'akankah', 'akhir',
                            'akhiri', 'akhirnya', 'aku', 'akulah', 'amat',
                            'amatlah', 'anda', 'andalah', 'antar', 'antara',
                            'antaranya', 'apa', 'apaan', 'apabila', 'apakah',
                            'apalagi', 'apatah', 'artinya', 'asal', 'asalkan', ...],
                strip_accents='ascii', sublinear_tf=False,
                token_pattern='(?u)\\b\\w\\w+\\b', tokenizer=None, use_idf=True,
                vocabulary=None)
```

Gambar 4.19 Rincian Data Matriks Pembobotan

Gambar 4.19 Menunjukan rincian data matriks yang digunakan untuk proses pembobotan data. Di dalam matriks tersebut memuat semua informasi terkait proses pembobotan data. Contoh fitur dan nilai TF-IDF ditunjukan pada gambar 4.19 di bawah ini:

```
warga asli pontianak tuju pontianak ibukota negara budaya bahasa adat istiadat tergangu alhamdulilah pontianak kandid
at cocok ibukota ri

        [('pontianak', 0.6345742700888499), ('istiadat', 0.26141124739424043), ('adat', 0.2468203842188006), ('kandi
dat', 0.2468203842188006), ('bahasa', 0.22843807058975715), ('tergangu', 0.22843807058975715), ('alhamdulilah', 0.228
43807058975715), ('asli', 0.22187713886982205), ('budaya', 0.22187713886982205), ('cocok', 0.20728627569438218), ('r
i', 0.19138670479169506), ('warga', 0.18234303034540364), ('tuju', 0.12734032009771995), ('ibukota', 0.11136233561165
3), ('negara', 0.10327481329656686)]


calon ibukota negara indonesia miris

        [('miris', 0.7383858499928031), ('calon', 0.47383510762688624), ('indonesia', 0.3470370803153911), ('negar
a', 0.2917114759175955), ('ibukota', 0.15727780204075087)]
```

Gambar 4.20 Contoh Hasil Pembobotan Data

Gambar 4.20 menunjukkan nilai atau frekuensi kemunculan kata dalam dokumen.

4.2.4 Pembagian Data

Dalam melakukan pembagian data, ada 2 metode yang digunakan yakni *hold out* dan *cross validation*. *Source code* yang digunakan untuk melakukan pembagian data menggunakan metode *hold out* adalah sebagai berikut:

```
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.2)
```

Gambar 4.21 *Source Code* Pembagian Data menggunakan metode *Hold Out*

Gambar 4.21 di atas menunjukkan *source code* pembagian data mengunakan metode *Hold Out* yang mana pada *source code* di atas menggunakan data tes sebanyak 20% data. Variabel x_train adalah data teks yang digunakan sebagai data latih sedangkan variabel x_test adalah data teks yang digunakan sebagai data uji. Variabel y_train adalah data sentimen yang digunakan sebagai data latih sedangkan variabel y_test adalah data sentimen yang digunakan sebagai data uji. Sedangkan untuk pembagian data menggunakan metode *cross validation* adalah sebagai berikut:

```
(cross_val_score(mnb, xvec, y, cv=10, scoring='accuracy').mean())
```

Gambar 4.22 *Source Code* Pembagian Data menggunakan metode *Cross Validation*

Gambar 4.22 menunjukkan *source code* proses pembagian data menggunakan metode *Cross Validation* yang mana menggunakan *fold* sebanyak 10.

4.2.5 *Synthetic Minority Oversampling Technique* (SMOTE)

Setelah melaukan pembobotan TF-IDF, langkah selanjutnya adalah melakukan SMOTE untuk mengatasi ketidak seimbangan data. Karena data yang diolah sebanyak 1679 data dengan jumlah sentimen positif 459 data, negatif 680 data, dan netral 540 data. *Source code* yang digunakan adalah sebagai berikut:

```
from imblearn.over_sampling import SMOTE

sm = SMOTE(random_state=1)
xtrain, ytrain = sm.fit_sample(Xv, ytrain)
```

Gambar 4.23 *Source Code* SMOTE

Setelah melakukan SMOTE seperti pada gambar 4.23 di atas, maka data yang semula tidak seimbang menjadi seimbang dengan mengikuti jumlah data kelas mayoritas.

4.2.6 Klasifikasi Data

A. Skenario 1

Pada skenario 1, pembagian data dilakukan dengan menggunakan metode *hold out*. Kemudian klasifikasi data menggunakan algoritma Naïve Bayes dengan jenis Multinomial. Berikut adalah *source code* yang digunakan:

```
mnb = MultinomialNB()
y_train = y_train.astype('int')
Xv = vectorizer.fit_transform(X_train)

mnb.fit(Xv, y_train)
x_testdf = vectorizer.transform(X_test)

pred = mnb.predict(x_testdf)
actual = np.array(y_test)
accuracy_score(actual, pred2)
```

Gambar 4.24 *Source Code* Klasifikasi Data Multinomial Naïve Bayes dengan Metode Pembagian Data *Hold Out*

B. Skenario 2

Pada skenario 2, pembagian data dilakukan dengan menggunakan metode *hold out*. Kemudian klasifikasi data menggunakan algoritma Naïve Bayes dengan jenis Bernoulli. Berikut adalah *source code* yang digunakan:

```
bnb = BernoulliNB(binarize = True)
Xv = vectorizer.fit_transform(X_train)
y_train = y_train.astype('int')

bnb = BernoulliNB(binarize = True)
bnb.fit(X, y_train)
x_testdf = vectorizer.transform(X_test)

pred2 = bnb.predict(x_testdf)
actual = np.array(y_test)

accuracy_score(actual, pred2)
```

Gambar 4.25 *Source Code* Klasifikasi Data Bernoulli Naïve Bayes dengan Metode Pembagian Data *Hold Out*

C. Skenario 3

Pada skenario 3, pembagian data dilakukan dengan menggunakan metode *hold out*. Kemudian klasifikasi data menggunakan algoritma Naïve Bayes dengan jenis Gaussian. Berikut adalah *source code* yang digunakan:

```
gnb = GaussianNB()
y_train = y_train.astype('int')

x_traindf =
vectorizer.transform(X_train.astype('U')).toarray()
x_testdf =
vectorizer.transform(X_test.astype('U')).toarray()
gnb.fit(x_traindf, y_train)

pred3 = gnb.predict(x_testdf)
actual = np.array(y_test)
```

Gambar 4.26 *Source Code* Klasifikasi Data Gaussian Naïve Bayes dengan Metode Pembagian Data *Hold Out*

D. Skenario 4

Pada skenario 4, pembagian data dilakukan dengan menggunakan metode *hold out*, kemudian dilakukan penyeimbangan data pada masing-masing kelas menggunakan SMOTE. Setelah itu melakukan klasifikasi data menggunakan algoritma Naïve Bayes dengan jenis Multinomial. Berikut adalah *source code* yang digunakan:

```
mnb = MultinomialNB()

mnb.fit(xtrain,ytrain)

pred = mnb.predict(xtestdf)
actual = np.array(ytest)
```

Gambar 4.27 *Source Code* Klasifikasi Data Multinomial Naïve Bayes menggunakan SMOTE dengan Metode Pembagian Data *Hold Out*

E. Skenario 5

Pada skenario 5, pembagian data dilakukan dengan menggunakan metode *hold out*, kemudian dilakukan penyeimbangan data pada masing-masing kelas menggunakan SMOTE. Setelah itu melakukan klasifikasi data menggunakan algoritma Naïve Bayes dengan jenis Bernoulli. Berikut adalah *source code* yang digunakan:

```
bnb = BernoulliNB(binarize = True)
bnb.fit(xtrain, ytrain)

pred2 = bnb.predict(xtestdf)
actual = np.array(ytest)

accuracy_score(actual, pred2)
```

Gambar 4.28 *Source Code* Klasifikasi Data Bernoulli Naïve Bayes menggunakan SMOTE dengan Metode Pembagian Data *Hold Out*

F. Skenario 6

Pada skenario 6, pembagian data dilakukan dengan menggunakan metode *hold out*, kemudian dilakukan penyeimbangan data pada masing-masing kelas menggunakan SMOTE. Setelah itu melakukan klasifikasi data menggunakan algoritma Naïve Bayes dengan jenis Gaussian. Berikut adalah *source code* yang digunakan:

```
X_train, X_test, y_train, y_test = train_test_split(xdata, ydata, test_size=0.2)

gnb = GaussianNB()
y_train = y_train.astype('int')

x_traindf = vectorizer.transform(X_train.astype('U')).toarray()
x_testdf = vectorizer.transform(X_test.astype('U')).toarray()
gnb.fit(x_traindf, y_train)

pred3 = gnb.predict(x_testdf)
actual = np.array(y_test)

accuracy_score(actual, pred3)
```

Gambar 4.29 *Source Code* Klasifikasi Data Gaussian Naïve Bayes menggunakan SMOTE dengan Metode Pembagian Data *Hold Out*

G. Skenario 7

Pada skenario 7, pembagian data dilakukan dengan menggunakan metode *cross validation*. Kemudian klasifikasi data menggunakan algoritma Naïve Bayes dengan jenis Multinomial. Berikut adalah *source code* yang digunakan:

```
mnb = MultinomialNB()

xvec = vectorizer.fit_transform(x)
print(cross_val_score(mnb, xvec, y, cv=10, scoring='accuracy').mean())
```

Gambar 4.30 *Source Code* Klasifikasi Data Multinomial Naïve Bayes dengan Metode Pembagian Data *Cross Validation*

H. Skenario 8

Pada skenario 8, pembagian data dilakukan dengan menggunakan metode *cross validation*. Kemudian klasifikasi data menggunakan algoritma Naïve Bayes dengan jenis Bernoulli. Berikut adalah *source code* yang digunakan:

```
bnb = BernoulliNB()

xvec = vectorizer.fit_transform(x)
print(cross_val_score(bnb, xvec, y, cv=10, scoring='accuracy').mean())
```

Gambar 4.31 *Source Code* Klasifikasi Data Bernoulli Naïve Bayes dengan Metode Pembagian Data *Cross Validation*

I. Skenario 9

Pada skenario 9, pembagian data dilakukan dengan menggunakan metode *cross validation*. Kemudian klasifikasi data menggunakan algoritma Naïve Bayes dengan jenis Gaussian. Berikut adalah *source code* yang digunakan:

```
gnb = GaussianNB()

xvec = vectorizer.fit_transform(x.astype('U')).toarray()
print(cross_val_score(gnb, xvec, y, cv=10, scoring='accuracy').mean())
```

Gambar 4.32 *Source Code* Klasifikasi Data Gaussian Naïve Bayes dengan Metode Pembagian Data *Cross Validation*

J. Skenario 10

Pada skenario 10, pembagian data dilakukan dengan menggunakan metode *cross validation*, kemudian dilakukan penyeimbangan data pada masing-masing kelas menggunakan SMOTE. Setelah itu melakukan klasifikasi data menggunakan algoritma Naïve Bayes dengan jenis Multinomial. Berikut adalah *source code* yang digunakan:

```
mnb = MultinomialNB()

print(cross_val_score(mnb, xsm, ysm, cv=10, scoring="accuracy").mean())
```

Gambar 4.33 *Source Code* Klasifikasi Data Multinomial Naïve Bayes menggunakan SMOTE dengan Metode Pembagian Data *Cross Validation*

K. Skenario 11

Pada skenario 11, pembagian data dilakukan dengan menggunakan metode *cross validation*, kemudian dilakukan penyeimbangan data pada masing-masing kelas menggunakan SMOTE. Setelah itu melakukan klasifikasi data menggunakan algoritma Naïve Bayes dengan jenis Bernoulli. Berikut adalah *source code* yang digunakan:

```
bnb = BernoulliNB()

print(cross_val_score(bnb, xsm, ysm, cv=10, scoring='accuracy').mean())
```

Gambar 4.34 *Source Code* Klasifikasi Data Bernoulli Naïve Bayes menggunakan SMOTE dengan Metode Pembagian Data *Cross Validation*

L. Skenario 12

Pada skenario 12, pembagian data dilakukan dengan menggunakan metode *cross validation*, kemudian dilakukan penyeimbangan data pada masing-masing kelas menggunakan SMOTE. Setelah itu melakukan klasifikasi data menggunakan algoritma Naïve Bayes dengan jenis Multinomial. Berikut adalah *source code* yang digunakan:

```
gnb = GaussianNB()

xvec = vectorizer.fit_transform(x.astype('U')).toarray()
xsm, ysm = sm.fit_sample(xvec, y)

print(cross_val_score(gnb, xsm, ysm, cv=10, scoring="accuracy").mean())
```

Gambar 4.35 *Source Code* Klasifikasi Data Gaussian Naïve Bayes menggunakan SMOTE dengan Metode Pembagian Data *Cross Validation*

Gambar 4.22 sampai dengan gambar 4.34 menunjukkan *source code* dengan berbagai macam skenario untuk memperoleh hasil akurasi terbaik. Contoh hasil klasifikasi data yakni sebagai berikut:

Tabel 4.17 Contoh Hasil Klasifikasi Data

No.	Data	Sentimen
1.	ibukota pindahin buang dana mending dana mensejahtrakan masyarakat kalubukan cukong menang proyek	Negatif
2.	calo calon ibukota papar asap kalimantan titik api lahan gambut batubara kalo hutan urus problem krusial manusia kuasa mayoritas rakyat	Negatif
3.	ibukota pindah jakarta pusat bisnis ekonomi pusat hibur magnet orang jakarta cari hidup layak jakarta macet	Netral
4.	waow jakarta mending ibukota baru	Positif
5.	mikir positif kalimantan maju sumber daya alam kayak men etapi bangun infrastruktur lari pulau jawa ibukota maju yakan mudah jalan harap	Positif

Tabel 4.17 di atas menunjukkan contoh hasil data *tweet* yang telah melewati berbagai tahap pemrosesan sebelumnya. Sedangkan untuk melakukan klasifikasi Naïve Bayes secara manual dapat menggunakan langkah-langkah sebagai berikut:

1. Menentukan pada kelas mana yang memiliki probabilitas paling besar.

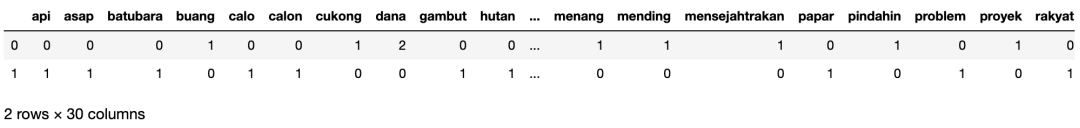
$$\text{P(negatif)} = \frac{\text{Banyak dokumen pada kelas negatif}}{\text{Total dokumen pada dataset}}$$

$$\text{P(positif)} = \frac{\text{Banyak dokumen pada kelas positif}}{\text{Total dokumen pada dataset}}$$

$$\text{P(netral)} = \frac{\text{Banyak dokumen pada kelas netral}}{\text{Total dokumen pada dataset}}$$

Sehingga didapatkan nilai sebagai berikut:

Negatif:

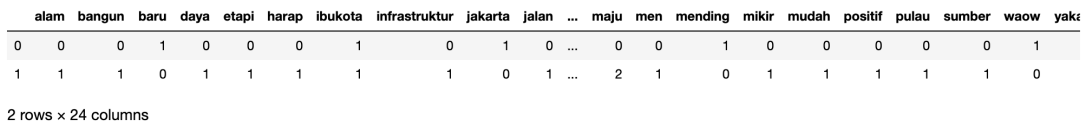


	api	asap	batubara	buang	calo	calon	cukong	dana	gambut	hutan	...	menang	mending	mensejahtrakan	papar	pindahin	problem	proyek	rakyat
0	0	0	0	1	0	0	1	2	0	0	...	1	1	1	0	1	0	1	0
1	1	1	1	0	1	1	0	0	1	1	...	0	0	0	1	0	1	0	1

2 rows × 30 columns

Gambar 4.36 Probabilitas Kelas Negatif

Positif:



	alam	bangun	baru	daya	etapi	harap	ibukota	infrastruktur	jakarta	jalan	...	maju	men	mending	mikir	mudah	positif	pulau	sumber	waow	yaka
0	0	0	1	0	0	0	1	0	1	0	...	0	0	1	0	0	0	0	0	1	
1	1	1	0	1	1	1	1	1	0	1	...	2	1	0	1	1	1	1	1	0	

2 rows × 24 columns

Gambar 4.37 Probabilitas Kelas Positif

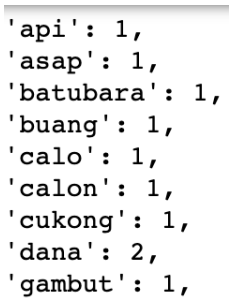
Netral:

	bisnis	cari	ekonomi	hibur	hidup	ibukota	jakarta	layak	macet	magnet	orang	pindah	pusat
0	1	1	1	1	1	1	3	1	1	1	1	1	2

Gambar 4.38 Probabilitas Kelas Netral

2. Menghitung banyak kata dalam suatu kelas.

Negatif:



```
'api': 1,
'asap': 1,
'batubara': 1,
'buang': 1,
'calo': 1,
'calon': 1,
'cukong': 1,
'dana': 2,
'gambut': 1,
```

Gambar 4.39 Banyak Kata Kelas Negatif

Positif:

```
'alam': 1,
'bangun': 1,
'baru': 1,
'daya': 1,
'etapi': 1,
'harap': 1,
'ibukota': 2,
'infrastruktur': 1,
'jakarta': 1,
'jalan': 1,
```

Gambar 4.40 Banyak Kata Kelas Positif

Netral:

```
'bisnis': 1,
'cari': 1,
'ekonomi': 1,
'hibur': 1,
'hidup': 1,
'ibukota': 1,
'jakarta': 3,
'layak': 1,
'macet': 1,
'magnet': 1,
```

Gambar 4.41 Banyak Kata Kelas Netral

3. Menghitung probabilitas kemunculan kata dalam suatu kelas.

Negatif:

```
'api': 0.03333333333333333,
'asap': 0.03333333333333333,
'batubara': 0.03333333333333333,
'buang': 0.03333333333333333,
'calo': 0.03333333333333333,
'calon': 0.03333333333333333,
'cukong': 0.03333333333333333,
'dana': 0.06666666666666667,
'gambut': 0.03333333333333333,
'hutan': 0.03333333333333333,
```

Gambar 4.42 Probabilitas Kata Kelas Negatif

Positif :

```
'alam': 0.04166666666666664,
'bangun': 0.04166666666666664,
'baru': 0.04166666666666664,
'daya': 0.04166666666666664,
'etapi': 0.04166666666666664,
'harap': 0.04166666666666664,
'ibukota': 0.08333333333333333,
'infrastruktur': 0.04166666666666664,
'jakarta': 0.04166666666666664,
'jalan': 0.04166666666666664,
```

Gambar 4.43 Probabilitas Kata Kelas Positif

Netral:

```
'bisnis': 0.07692307692307693,
'cari': 0.07692307692307693,
'ekonomi': 0.07692307692307693,
'hibur': 0.07692307692307693,
'hidup': 0.07692307692307693,
'ibukota': 0.07692307692307693,
'jakarta': 0.23076923076923078,
'layak': 0.07692307692307693,
'macet': 0.07692307692307693,
'magnet': 0.07692307692307693,
```

Gambar 4.44 Probabilitas Kata Kelas Netral

4. Menghitung Probabilitas dengan *inplace* suatu kelas.

Probabilitas dengan perataan *inplace* untuk semua kata di kelas negatif:

```
[0.02127659574468085, 0.02127659574468085, 0.010638297872340425, 0.010638297872340425, 0.010638297872340425]
```

Gambar 4.45 Probabilitas dengan *Inplace* Kelas Negatif

Probabilitas dengan perataan *inplace* untuk semua kata di kelas positif:

```
[0.011363636363636364, 0.011363636363636364, 0.011363636363636364, 0.022727272727272728, 0.022727272727272728]
```

Gambar 4.46 Probabilitas dengan *Inplace* Kelas Positif

Probabilitas dengan perataan *inplace* untuk semua kata di kelas netral:

```
[0.01282051282051282, 0.01282051282051282, 0.02564102564102564, 0.01282051282051282, 0.01282051282051282]
```

Gambar 4.47 Probabilitas dengan *Inplace* Kelas Netral

5. Kemudian dikalikan seluruh probabilitas pada masing-masing sentimen:

Positif:

```
5.450303960617726e-10
```

Gambar 4.48 Perkalian Kelas Positif

Negatif:

```
7.579605994374453e-10
```

Gambar 4.49 Perkalian Kelas Negatif

Netral:

```
6.927188126103506e-10
```

Gambar 4.50 Perkalian Kelas Netral

4.2.7 Evaluasi Performa Model Klasifikasi

Seperti tahapan yang ada pada bab 3, langkah selanjutnya setelah dilakukan klasifikasi data dengan menggunakan metode Naïve Bayes, tujuannya adalah untuk mencari akurasi yang paling tinggi. Data yang digunakan yakni 1679 data yang sudah diolah sebelumnya. Untuk melakukan evaluasi model, maka perlu dilakukan dengan beberapa skenario. Skenario dilakukan dengan mengubah parameter pada pembobotan TF-IDF, pembagian data, dan model Naïve Bayes. Untuk informasi lebih rinci dapat dilihat pada tabel di bawah ini:

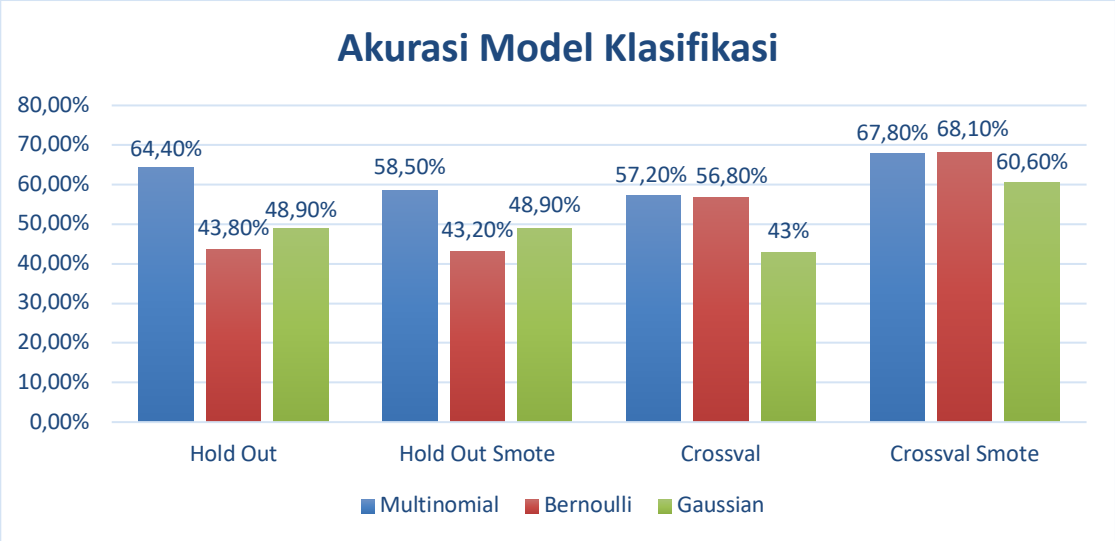
Tabel 4.18 Hasil Skenario

Percobaan Skenario		Akurasi	Presisi	Recall	F1-score	Support
Skenario 1	-1	64,40%	68%	85%	75%	150
	0		52%	49%	51%	96

	1		74%	47%	58%	89
Skenario 2	-1 0 1	43,80%	44% 0% 0%	100% 0% 0%	61% 0% 0%	147 100 88
Skenario 3	-1 0 1	48,90%	62% 47% 41%	46% 47% 55%	53% 47% 47%	132 108 95
Skenario 4	-1 0 1	58,50%	69% 50% 52%	66% 41% 66%	68% 45% 58%	145 104 86
Skenario 5	-1 0 1	43,20%	43% 0% 0%	100% 0% 0%	60% 0% 0%	145 104 86
Skenario 6	-1 0 1	48,90%	60% 46% 43%	47% 44% 58%	53% 45% 49%	130 110 95
Skenario 7	-1 0 1	57,20%	58% 26% 16%	83% 40% 40%	68% 30% 79%	680 538 456
Skenario 8	-1 0 1	56,80%	62% 48% 57%	76% 45% 42%	68% 47% 48%	680 538 456
Skenario 9	-1 0	43%	57% 38%	41% 34%	48% 36%	680 538

	1		37%	57%	45%	456
Skenario 10	-1 0 -1	67,80%	73% 64% 66%	68% 59% 77%	71% 62% 71%	680 680 680
Skenario 11	-1 0 1	68,10%	73% 64% 66%	68% 59% 77%	71% 44% 71%	680 680 680
Skenario 12	-1 0 1	60,60%	70% 60% 57%	40% 57% 85%	51% 58% 68%	680 680 680

Tabel 4.19 di atas menunjukkan hasil dari beberapa percobaan skenario yang dilakukan. Sehingga didapatkan hasil akurasi paling tinggi pada skenario 11, yakni 68,10%. Apabila digambarkan dalam bentuk diagram, maka seperti berikut:



Gambar 4.51 Diagram Hasil Akurasi Model Klasifikasi

Pada gambar 4.51 di atas yakni diagram hasil akurasi model klasifikasi menunjukkan bahwa yang menduduki akurasi tertinggi sebesar 68,10% terdapat

pada klasifikasi Bernoulli Naïve Bayes yang mana pembagian data dilakukan dengan metode *Cross Validation* dan menggunakan SMOTE untuk menyeminbangkan data. Sehingga menghasilkan *confusion matrix* seperti gambar berikut:

```
array([[462, 118, 100],
       [115, 399, 166],
       [ 52, 105, 523]])
```

Gambar 4.52 *Confusion Matrix* Skenario 11

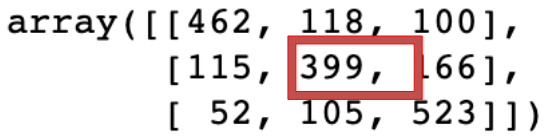
Pada gambar 4.52 di atas menunjukkan bahwa kelas dengan prediksi yang tepat paling banyak di duduki oleh sentimen positif yakni sebanyak 523 data, selanjutnya sentimen negatif sebanyak 462 data, kemuadian disusul dengan sentimen netral sejumlah 399 data.

Sehingga 31,9% sisanya menjadi presentase bagi kelas yang salah prediksi, misalnya:

```
array([[462, 118, 100],
       [115, 399, 166],
       [ 52, 105, 523]])
```

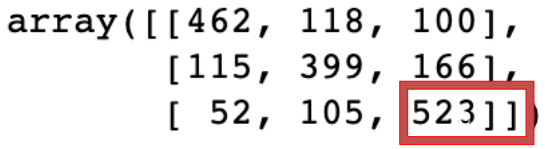
Gambar 4.53 *Confusion Matrix* Skenario 11 (2)

Pada gambar 4.53 di atas, kotak merah menunjukkan data dengan sentimen negatif yang terprediksi dengan tepat. Sedangkan 118 di samping kanannya menunjukkan bahwa data yang seharusnya negatif namun terprediksi netral, dan di posisi paling kanan menunjukkan 100 data yang seharusnya negatif namun terprediksi positif.



Gambar 4.54 *Confusion Matrix* Skenario 11 (3)

Pada gambar 4.54 di atas, kotak merah menunjukkan data dengan sentimen netral yang terprediksi dengan benar. Sedangkan di samping kiri kotak merah menunjukkan 115 data yang artinya data tersebut salah prediksi, yang seharusnya netral namun terprediksi negatif. Untuk 166 data lainnya yang terletak di sebelah kanan kotak merah menunjukkan data yang seharusnya netral namun terprediksi positif.



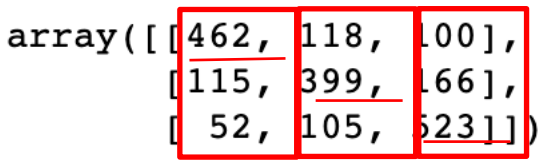
Gambar 4.55 *Confusion Matrix* Skenario 11 (4)

Pada gambar 4.55 di atas, kotak merah menunjukkan data positif yang terprediksi dengan benar. Sedangkan data sebanyak 105 di samping kotak merah menunjukkan data yang salah prediksi, yang mana seharusnya positif namun terprediksi netral. Dan untuk data sebanyak 52 yang terletak di ujung kiri menunjukkan bahwa data yang seharusnya positif namun terprediksi negatif.

Untuk menghitung nilai presisi, *recall*, F1-*score*, dan *support* secara manual dapat dilakukan dengan cara sebagai berikut:

1. Presisi

   Merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positf.



Gambar 4.56 *Confusion Matrix* Skenario 11 (5)

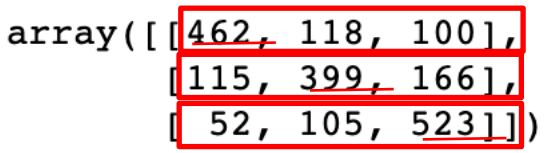
Negatif : $\frac{462}{462+115+52}$ = 0,73

Netral : $\frac{399}{118+399+105}$ = 0,64

Positif : $\frac{523}{100+166+523}$ = 0,66

2. *Recall*

Merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif.



Gambar 4.57 *Confusion Matrix* Skenario 11 (6)

Negatif : $\frac{462}{462+118+100}$ = 0,68

Netral : $\frac{399}{115+399+166}$ = 0,59

Positif : $\frac{523}{52+105+523}$ = 0,77

3. F1-*score*

F1 Score merupakan perbandingan rata-rata presisi dan recall yang dibobotkan dengan rumus berikut:

$$2x\frac{\text{presisi x } recall}{\text{presisi} + recall}$$

Negatif : $2x\frac{0,50}{1,41}$ = 0,71

Netral : $2x\frac{0,27}{1,23}$ = 0,44

Positif : $2x\frac{0,51}{1,43}$ = 0,71

4. *Support*

Merupakan jumlah prediksi benar positif dengan keseluruhan data yang benar positif.

```
array([[462, 118, 100],
       [115, 399, 166],
       [ 52, 105, 523]])
```

Gambar 4.58 *Confusion Matrix* Skenario 11 (7)

Negatif : 462 + 188 + 100 = 680

Netral : 115 + 399 + 166 = 680

Positif : 52 + 105 + 523 = 680

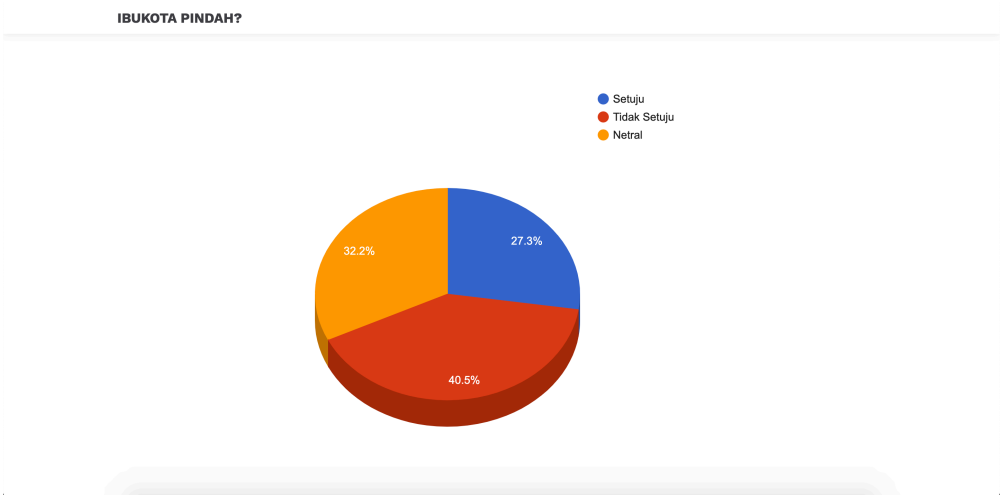
4.3 Implementasi Sistem

Setelah melakukan pembuatan model, selanjutnya adalah pembuatan web yang berisi visualisasi dari hasil klasifikasi yang dibangun dengan menggunakan bahasa pemrograman PHP. Berikut ini adalah tampilan visualisasi yang telah dibangun:

Gambar 4.59 Web Visualisasi

Gambar 4.60 Web Visualisasi (2)

Pada gambar 4.60 di atas menggambarkan *word cloud* mengenai *tweet* tentang pemindahan Ibu Kota Indonesia yang bersentimen positif dan negatif. Sehingga, apabila ingin mengetahui kata yang sering muncul secara garis besar, diharapkan *word cloud* dapat membantu.

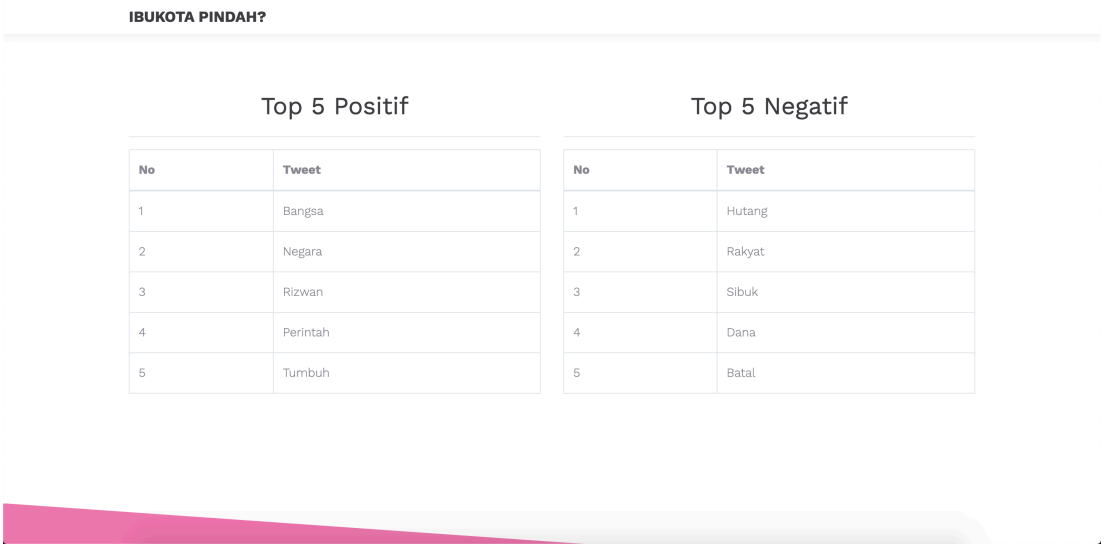


Gambar 4.61 Web Visualisasi (3)

Pada gambar 4.61 di atas menggambarkan sebuah *pie chart* dengan presentase masing-masing sentimen dalam *dataset* yang diolah. Sehingga pada *pie chart* tersebut dapat diketahui sebanyak 27.3% orang bersentimen positif atau setuju, 40.5% orang bersentimen negatif atau tidak setuju, dan sisanya 32.2% orang bersentimen netral.

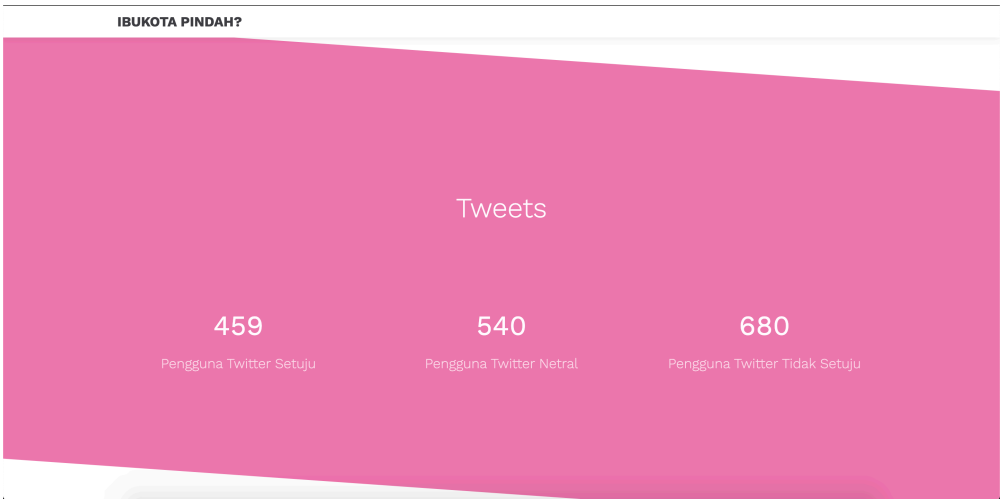
Gambar 4.62 Web Visualisasi (4)

Pada gambar 4.62 di atas, di sana menjelaskan mengenai pembindahan Ibu Kota Indonesia secara singkat.



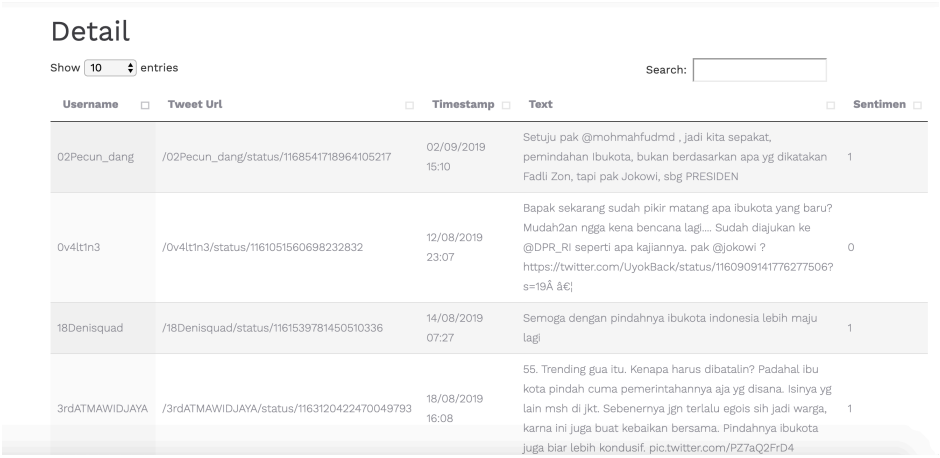
Gambar 4.63 Web Visualisasi (5)

Pada gambar 4.44 di atas, menunjukkan kata-kata yang paling sering muncul pada *tweet*. Kata yang diambil sebanyak 5 saja dan memiliki setimen positif dan negatif. Atau bisa juga disebut rincian dari *word cloud* yang telah dijelaskan sebelumnya.



Gambar 4.64 Web Visualisasi (6)

Pada gambar 4.64 di atas menunjukkan jumlah data *tweet* pada masing-masing kelas. Sejumlah 459 data dengan sentimen positif atau setuju, 540 data dengan sentimen netral, dan 680 data dengan sentimen negatif atau tidak setuju.



Gambar 4.65 Web Visualisasi (7)

Pada gambar 4.65 di atas adalah tabel yang terletak di akhir halaman web yang berisi data *tweet* beserta nama pengguna, URL, waktu *tweet*, beserta sentimennya yang mana 1 adalah positif, 0 dalah netral, dan -1 adalah negatif.

Dari gambar 4.59 sampai dengan 4.65 di atas yang menjelaskan tentang implementasi sistem guna memvisualisasikan hasil klasifikasi, dapat diketahui berapa banyak perbandingan persentase orang-orang yang beropini positif, negatif, atau netral pada isu pemindahan Ibu Kota Indonesia. Dari sentimen positif dan negatif dapat diambil kata kunci yang paling banyak digunakan sehingga dapat diketahui kata yang menjadi pemicu dalam beropini.