BAB IV

HASIL DAN PEMBAHASAN

4.1 Implementasi dan Hasil

4.1.1 Pengumpulan Data

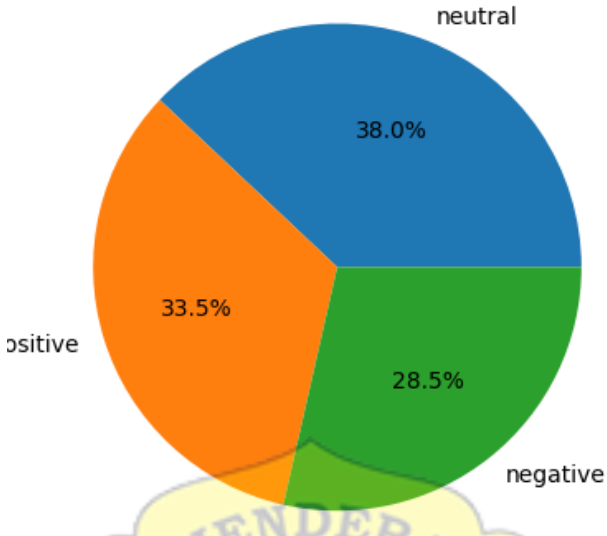
Untuk memenuhi kebutuhan penarikan data latih, penulis menggunakan sumber data yaitu data cuitan twitter, dan untuk alur pengambilan data nya adalah dengan menggunakan API yang disediakan oleh twitter sendiri. Alur pengambilan data melalui twitter adalah memanggil API twitter untuk mengambil tweet - tweet terbaru menggunakan API *get recent tweets* (***GET*** */2/tweets/search/recent*). API ini menyediakan balikkan dalam bentuk JSON yang akan sangat mudah untuk kemudian diproses dalam bentuk CSV. Dengan menggunakan API ini penulis menarik kurang lebih 1000 data tweet yang nantinya akan dibagi menjadi 20% (202) data uji dan 80% (807) data testing. Data - data yang diambil dalam bentuk JSON kemudian diproses dan di *compile* ke dalam bentuk CSV dengan header ***username, tweet_id, text, dan referenced_tweet***.

Data tweets yang didapat kemudian diberi label positif maupun negatif, positif berarti tweet tersebut berpihak/setuju/sependapat dengan pinjaman online dan negatif apabila tweet tersebut tidak sejalan/tidak setuju/tidak sependapat terhadap pinjaman online, yang akan dinilai berdasarkan konteks tweet tersebut. Berikut beberapa contoh tweets yang sudah diberi label pada Tabel 3.

Tabel 3. Tweets Yang Telah Diberi Label

tweet_id	text	label
1591022377819058176	Waspada dc pinjol menagih utang menyamar menjadi kurir paket.	**NEGATIVE**
1591782857571196929	Tak dapat dipungkiri, layanan paylater yang ditawarkan oleh perusahaan fintech P2P lending menerima sambutan hangat masyarakat dari berbagai kalangan. https://t.co/6SrZUxCdPr	**POSITIVE**
1591454819851833348	Ada yg tau pinjaman uang yg bisa dicicil? Lagi urgent bgt, pake jaminan gppa deh, tp jgn pinjol	**POSITIVE**
1591323926378213376	nomor tlp pribadi saya masih rutin di tlp pihak julo terkait tagihan an Sri Rahayu Ningsih, sya tidak kenal dengan org tsb dan sya tidak pernah sekalipun melakukan pinjaman online, sudah 2 minggu sya di teror oleh tlp dari pihak julo	**NEGATIVE**
1591319222474649600	ada berapa banyak nyawa manusia dinegara ini yang meninggal karena pinjol setiap harinya? apa pinjol gak mau dihapuskan? jujur gue salah satu diantara banyaknya populasi manusia di negara ini yg selalu coba berbagai cara untuk bunuh diri	**NEGATIVE**

Setelah melakukan proses labelling pada keseluruhan data, penulis mendapat perbandingan jumlah label positif, negatif, dan netral pada Gambar 3. Gambar 3 menunjukan perbedaan jumlah polaritas yang diperoleh dari proses pelabelan data. Dari 1009 data yang diambil, terdapat 338 data dengan label positif dan 288 data dengan label negatif dan juga 383 data netral. Dari perhitungan lanjutan, dapat diketahui bahwa data dengan sentimen netral merupakan sentimen terbesar dari keseluruhan data.



Gambar 3. Jumlah Polaritas Data

4.1.2 Preprocessing Data

Preprocessing data merupakan tahapan awal yang dilakukan setelah pengambilan dataset. Terdapat beberapa tahapan yang dilakukan.secara berurutan, yaitu *cleansing* data, *case folding*, *stemming*, *tokenizing*, dan *filtering*.

4.1.2.1 *Normalization of informal text*

Normalization of informal text merupakan tahapan dalam *preprocessing* data yang bertujuan mengubah kata - kata non formal berupa singkatan, kata - kata yang tidak baku menjadi kata - kata yang baku dan formal. Berikut contoh sebuah kalimat tweet yang sudah melalui.

Tabel 4. *Normalization of informal text*

Sebelum	Sesudah
Waspada dc pinjol menagih utang menyamar menjadi kurir paket.	Waspada debt collector pinjaman online menagih utang menyamar menjadi kurir paket.
Tak dapat dipungkiri, layanan paylater yang ditawarkan oleh perusahaan fintech P2P lending menerima sambutan hangat masyarakat dari berbagai kalangan. https://t.co/6SrZUxCdPr	Tidak dapat dipungkiri layanan paylater yang ditawarkan oleh perusahaan fintech p2p lending menerima sambutan hangat masyarakat dari berbagai kalangan. https://t.co/6SrZUxCdPr
Ada yg tau pinjaman uang yg bisa dicicil? Lagi urgent bgt, pake jaminan gppa deh, tp jgn pinjol	Ada yang tau pinjaman uang yang bisa dicicil? Lagi Ada yang tau pinjaman uang bisa dicicil? Lagi urgent banget, pake jaminan tidak apa deh, tapi jangan online banget, pake jaminan tidak apa apa deh, tapi jangan pinjaman online
nomor tlp pribadi saya masih rutin di tlp pihak julo terkait tagihan an Sri Rahayu Ningsih, sya tidak kenal dengan org tsb dan sya tidak pernah sekalipun melakukan pinjaman online, sudah 2 minggu sya di teror oleh tlp dari pihak julo	nomor telepon pribadi saya masih rutin di telepon pihak julo terkait tagihan atas nama Sri Rahayu Ningsih, saya tidak kenal dengan orang tersebut dan saya tidak pernah sekalipun melakukan pinjaman online, sudah 2 minggu saya di teror oleh telepon dari pihak julo
ada berapa banyak nyawa manusia dinegara ini yang meninggal karena pinjol setiap harinya? apa pinjol gak mau dihapuskan? jujur gue salah satu diantara banyaknya populasi manusia di negara ini yg selalu coba berbagai cara untuk bunuh diri	ada berapa banyak nyawa manusia dinegara ini yang meninggal karena pinjaman online setiap harinya? apa pinjaman online tidak mau dihapuskan? jujur saya salah satu diantara banyak populasi manusia di negara yang selalu coba berbagai cara untuk bunuh diri

Implementasi Pada Kode:

```
def replace_alay(sentence, alay_dictionary):
```

```
    alay_dict = dict(zip(alay_dictionary['original'],
alay_dictionary['replacement']))
    return ' '.join([alay_dict[w] if w in alay_dict else w for w in
sentence.split()])
```

Gambar 4. Potongan kode perubahan informal text

4.1.2.2 *Cleansing Data*

Cleansing data merupakan tahapan dalam *preprocessing* data yang bertujuan untuk menghilangkan dan atau mengurangi kata - kata yang tidak akan digunakan pada data tweet yang sebelumnya diambil. Langkah yang dilakukan yaitu penghapusan data yang sama atau mengalami redudansi data, menghapus *link* atau *url*, dan menghapus simbol seperti !@#$%^&*()_+. Berikut contoh sebuah kalimat tweet yang sudah melalui proses *cleansing data*.

Tabel 5. *Cleansing Data*

Sebelum	Sesudah
Waspada debt collector pinjaman online menagih utang menyamar menjadi kurir paket.	Waspada debt collector pinjaman online menagih utang menyamar menjadi kurir paket
Tidak dapat dipungkiri layanan paylater yang ditawarkan oleh perusahaan fintech p2p lending menerima sambutan hangat masyarakat dari berbagai kalangan. https://t.co/6SrZUxCdPr	Tidak dapat dipungkiri layanan paylater yang ditawarkan oleh perusahaan fintech p2p lending menerima sambutan hangat masyarakat dari berbagai kalangan
Ada yang tau pinjaman uang yang bisa dicicil? Lagi urgent banget, pake jaminan tidak apa apa deh, tapi jangan pinjaman online	Ada yang tau pinjaman uang bisa dicicil Lagi urgent banget pake jaminan tidak apa deh tapi jangan online
nomor telepon pribadi saya masih rutin di telepon pihak julo terkait tagihan atas nama Sri Rahayu Ningsih, saya tidak kenal dengan orang tersebut dan saya tidak pernah sekalipun melakukan pinjaman online, sudah 2	nomor telepon pribadi saya masih rutin di pihak terkait tagihan atas nama Sri Rahayu Ningsih tidak kenal dengan orang tersebut dan pernah sekalipun melakukan pinjaman online sudah 2 minggu teror oleh dari julo

Sebelum	Sesudah
minggu saya di teror oleh telepon dari pihak julo	
ada berapa banyak nyawa manusia dinegara ini yang meninggal karena pinjaman online setiap harinya? apa pinjaman online tidak mau dihapuskan? jujur saya salah satu diantara banyak populasi manusia di negara yang selalu coba berbagai cara untuk bunuh diri	ada berapa banyak nyawa manusia dinegara ini yang meninggal karena pinjaman online setiap harinya apa tidak mau dihapuskan jujur saya salah satu diantara banyak populasi di negara yang selalu coba berbagai cara untuk bunuh diri

```
def cleansing(sentence):
    sentence = re.sub(r'\\n', ' ', sentence)
    sentence = re.sub(r'\\t', ' ', sentence)
    sentence = re.sub(r'\"', '', sentence)
    sentence = re.sub(r'\[username\]', '', sentence)
    sentence = re.sub(r'\[user\]', '', sentence)
    sentence = re.sub(r'\[url\]', '', sentence)
    sentence = re.sub(r'\\t', ' ', sentence)
    sentence = re.sub(r'@[A-Za-z0-9]+', '', sentence)
    sentence = re.sub(r'\$\w*', '', sentence)
    sentence = re.sub(r'(^(rt)|\s)+(rt)+\s', ' ', sentence)
    #url
    sentence =
re.sub(r'((www\.[^\s]+)|(https?://[^\s]+)|(http?://[^\s]+))', ' ',
sentence)
    sentence = re.sub(r'#', '', sentence)
    sentence = re.sub(r'\\x[*0-9a-zA-Z]+', ' ', sentence)
    sentence = re.sub(r'user', '', sentence)
    sentence = re.sub(r'url', ' ', sentence)
    sentence = re.sub(r'ssl', ' ', sentence)
    sentence = re.sub(r'[^0-9a-zA-Z]+', ' ', sentence)
    sentence = re.sub(r'\d+', '', sentence)
    sentence = re.sub(r'  +', ' ', sentence)
    return sentence
```

Gambar 5. Potongan kode *cleansing data tweet*

4.1.2.3 *Case Folding*

Case folding merupakan tahapan dalam *preprocessing* data yang bertujuan untuk penyeragaman pada dataset tweet dengan cara mengubah huruf besar menjadi huruf kecil. Berikut contoh sebuah kalimat tweet yang sudah melalui proses *case folding*.

Tabel 6. *Case Folding*

Sebelum	Sesudah
Waspada debt collector pinjaman online menagih utang menyamar menjadi kurir paket	waspada debt collector pinjaman online menagih utang menyamar menjadi kurir paket
Tidak dapat dipungkiri layanan paylater yang ditawarkan oleh perusahaan fintech p2p lending menerima sambutan hangat masyarakat dari berbagai kalangan	tidak dapat dipungkiri layanan paylater yang ditawarkan oleh perusahaan fintech p2p lending menerima sambutan hangat masyarakat dari berbagai kalangan
Ada yang tau pinjaman uang bisa dicicil Lagi urgent banget pake jaminan tidak apa deh tapi jangan online	ada yang tau pinjaman uang bisa dicicil lagi urgent banget pake jaminan tidak apa deh tapi jangan online
nomor telepon pribadi saya masih rutin di pihak terkait tagihan atas nama Sri Rahayu Ningsih tidak kenal dengan orang tersebut dan pernah sekalipun melakukan pinjaman online sudah 2 minggu teror oleh dari julo	nomor telepon pribadi saya masih rutin di pihak terkait tagihan atas nama sri rahayu ningsih tidak kenal dengan orang tersebut dan pernah sekalipun melakukan pinjaman online sudah 2 minggu teror oleh dari julo
ada berapa banyak nyawa manusia dinegara ini yang meninggal karena pinjaman online setiap harinya apa tidak mau dihapuskan jujur saya salah satu diantara banyak populasi di negara yang selalu coba berbagai cara untuk bunuh diri	ada berapa banyak nyawa manusia dinegara ini yang meninggal karena pinjaman online setiap harinya apa tidak mau dihapuskan jujur saya salah satu diantara banyak populasi di negara yang selalu coba berbagai cara untuk bunuh diri

Implementasi pada kode:

```
def to_lowercase(text):
    return text.lower()
```

Gambar 6. Potongan kode *case folding data tweet*

4.1.2.4 *Stemming*

Stemming merupakan tahapan dalam *preprocessing* data yang bertujuan

untuk mempersingkat kata yang digunakan pada tiap dataset yang di ambil. Langkah yang dilakukan yaitu mengubah kata yang berimbuhan menjadi sebuah kata dasar. Berikut contoh sebuah kalimat tweet yang sudah melalui proses *stemming*.

Tabel 7. *Stemming*

Sebelum	Sesudah
waspada debt collector pinjaman online menagih utang menyamar menjadi kurir paket	waspada debt collector pinjam online tagih utang samar jadi kurir paket
tidak dapat dipungkiri layanan paylater yang ditawarkan oleh perusahaan fintech p2p lending menerima sambutan hangat masyarakat dari berbagai kalangan	tidak dapat mungkir layan paylater yang tawar oleh perusahaan fintech p2p lending terima sambut hangat masyarakat dari bagai kalangan
ada yang tau pinjaman uang bisa dicicil lagi urgent banget pake jaminan tidak apa deh tapi jangan online	ada yang tau pinjam uang bisa cicil lagi urgent banget pake jamin tidak apa deh tapi jangan online
nomor telepon pribadi saya masih rutin di pihak terkait tagihan atas nama sri rahayu ningsih tidak kenal dengan orang tersebut dan pernah sekalipun melakukan pinjaman online sudah 2 minggu teror oleh dari julo	nomor telepon pribadi saya masih rutin di pihak kait tagih atas nama sri rahayu ningsih tidak kenal dengan orang sebut dan pernah sekali laku pinjam online sudah 2 minggu teror oleh dari julo
ada berapa banyak nyawa manusia dinegara ini yang meninggal karena pinjaman online setiap harinya apa tidak mau dihapuskan jujur saya salah satu diantara banyak populasi di negara yang selalu coba berbagai cara untuk bunuh diri	ada berapa banyak nyawa manusia di negara ini yang meninggal karena pinjam online setiap hari apa tidak mau hapus jujur saya salah satu diantara banyak populasi di negara yang selalu coba bagai cara untuk bunuh diri

Implementasi pada kode:

```
def stemming_words(sentence):
    factory = StemmerFactory()
    stemmer = factory.create_stemmer()
    return stemmer.stem(sentence)
```

Gambar 7. Potongan kode *stemming* data tweet

4.1.2.5 *Tokenizing*

Tokenizing merupakan tahapan dalam *preprocessing* data yang bertujuan untuk memotong atau memecah data tweet berdasarkan pada spasi yang ada di antara kata. Berikut contoh sebuah kalimat tweet yang sudah melalui proses *tokenizing*.

Tabel 8. *Tokenizing*

Sebelum	Sesudah
waspada debt collector pinjam online tagih utang samar jadi kurir paket	['waspada', 'debt' ,'collector' ,'pinjam' ,'online' ,'tagih' ,'utang' ,'samar' ,'jadi' ,'kurir', 'paket']
tidak dapat mungkir layan paylater yang tawar oleh perusahaan fintech p2p lending terima sambut hangat masyarakat dari bagai kalangan	['tidak', 'dapat', 'mungkir', 'layan', 'paylater', 'yang', 'tawar', 'oleh', 'perusahaan', 'fintech', 'p2p', 'lending', 'terima', 'sambut', 'hangat', 'masyarakat', 'dari', 'bagai', 'kalangan']
ada yang tau pinjam uang bisa cicil lagi urgent banget pake jamin tidak apa deh tapi jangan online	['ada', 'yang', 'tau', 'pinjam', 'uang', 'bisa', 'cicil', 'lagi', 'urgent', 'banget', 'pake', 'jamin', 'tidak', 'apa', 'deh', 'tapi', 'jangan', 'online']
nomor telepon pribadi saya masih rutin di pihak kait tagih atas nama sri rahayu ningsih tidak kenal dengan orang sebut dan pernah sekali laku pinjam online sudah 2 minggu teror oleh dari julo	['nomor', 'telepon', 'pribadi', 'saya', 'masih', 'rutin', 'di', 'pihak', 'kait', 'tagih', 'atas', 'nama', 'sri', 'rahayu', 'ningsih', 'tidak', 'kenal', 'dengan', 'orang', 'sebut', 'dan', 'pernah', 'sekali', 'laku', 'pinjam', 'online', 'sudah', '2', 'minggu', 'teror', 'oleh', 'dari', 'julo']

Sebelum	Sesudah
ada berapa banyak nyawa manusia di negara ini yang meninggal karena pinjam online setiap hari apa tidak mau hapus jujur saya salah satu diantara banyak populasi di negara yang selalu coba bagai cara untuk bunuh diri	['ada', 'berapa', 'banyak', 'nyawa', 'manusia', 'di', 'negara', 'ini', 'yang', 'meninggal', 'karena', 'pinjam', 'online', 'setiap', 'hari', 'apa', 'tidak', 'mau', 'hapus', 'jujur', 'saya', 'salah', 'satu', 'diantara', 'banyak', 'populasi', 'di', 'negara', 'yang', 'selalu', 'coba', 'bagai', 'cara', 'untuk', 'bunuh', 'diri']

4.1.2.6 *Filtering*

Filtering merupakan tahapan dalam *preprocessing* data yang bertujuan untuk menyaring kalimat dengan mengambil kata yang penting dan membuang kata yang dirasa kurang berguna seperti "dan", "atau", "dll" atau kata-kata ini disebut juga dengan *stopwords*. Berikut contoh sebuah kalimat tweet yang sudah melalui proses *filtering*.

Tabel 9. *Filtering*

Sebelum	Sesudah
['waspada', 'debt' ,'collector' ,'pinjam' ,'online' ,'tagih' ,'utang' ,'samar' ,'jadi' ,'kurir', 'paket']	['waspada', 'debt', 'collector', 'pinjam', 'online', 'tagih', 'utang', 'samar', 'kurir', 'paket']
['tidak', 'dapat', 'mungkir', 'layan', 'paylater', 'yang', 'tawar', 'oleh', 'perusahaan', 'fintech', 'p2p', 'lending', 'terima', 'sambut', 'hangat', 'masyarakat', 'dari', 'bagai', 'kalangan']	['mungkir', 'layan', 'paylater', 'tawar', 'perusahaan', 'fintech', 'p2p', 'lending', 'terima', 'sambut', 'hangat', 'masyarakat', 'kalangan']
['ada', 'yang', 'tau', 'pinjam', 'uang', 'bisa', 'cicil', 'lagi', 'urgent', 'banget', 'pake', 'jamin', 'tidak', 'apa', 'deh', 'tapi', 'jangan', 'online']	['tau', 'pinjam', 'uang', 'cicil', 'urgent', 'banget', 'pake', 'jamin', 'deh', 'online']

Sebelum	Sesudah
['nomor', 'telepon', 'pribadi', 'saya', 'masih', 'rutin', 'di', 'pihak', 'kait', 'tagih', 'atas', 'nama', 'sri', 'rahayu', 'ningsih', 'tidak', 'kenal', 'dengan', 'orang', 'sebut', 'dan', 'pernah', 'sekali', 'laku', 'pinjam', 'online', 'sudah', '2', 'minggu', 'teror', 'oleh', 'dari', 'julo']	['nomor', 'telepon', 'pribadi', 'rutin', 'kait', 'tagih', 'nama', 'sri', 'rahayu', 'ningsih', 'kenal', 'orang', 'laku', 'pinjam', 'online', '2', 'minggu', 'teror', 'julo']
['ada', 'berapa', 'banyak', 'nyawa', 'manusia', 'di', 'negara', 'ini', 'yang', 'meninggal', 'karena', 'pinjam', 'online', 'setiap', 'hari', 'apa', 'tidak', 'mau', 'hapus', 'jujur', 'saya', 'salah', 'satu', 'diantara', 'banyak', 'populasi', 'di', 'negara', 'yang', 'selalu', 'coba', 'bagai', 'cara', 'untuk', 'bunuh', 'diri']	['nyawa', 'manusia', 'negara', 'meninggal', 'pinjam', 'online', 'hapus', 'jujur', 'salah', 'populasi', 'negara', 'coba', 'bunuh']

4.2 Ekstraksi Fitur

Ekstraksi fitur merupakan kunci dalam tahap analisis sentimen agar algoritma klasifikasi dapat menjalankan prosesnya dengan lancar. Dalam penelitian ini dilakukan ekstraksi fitur dengan metode TF-IDF kemudian dilakukan pemrosesan lebih lanjut dengan BM25. Proses ini akan dilakukan kepada masing-masing dataset, yaitu dataset stemming dan non stemming. Untuk tahapan dari perhitungan metode ini, pertama akan dilakukan perhitungan Term Frequency.

4.2.1 Pembobotan Term Frequency

Berikut disajikan contoh dokumen untuk melakukan perhitungan bobot term frequency. Dokumen ditunjukkan pada Tabel 10.

Tabel 10. Contoh Data latih

Doc #	Tweet	Label
1.	waspada debt collector pinjam online tagih utang samar kurir paket	**NEGATIVE**
2.	mungkir layan paylater tawar perusahaan fintech p2p lending terima sambut hangat masyarakat kalangan	**POSITIVE**
3.	tau pinjam uang cicil urgent banget pake jamin deh online	**POSITIVE**
4.	nomor telepon pribadi rutin kait tagih nama sri rahayu ningsih kenal orang laku pinjam online 2 minggu teror julo	**NEGATIVE**
5.	nyawa manusia negara meninggal pinjam online hapus jujur salah populasi negara coba bunuh	**NEGATIVE**

Langkah dalam melakukan perhitungan adalah menghimpun kata menjadi id yg urut untuk mengubah identitas string menjadi sebuah angka. Kemudian frekunsi term akan dihitung pada tiap dokumen, selanjutnya dibuat sebuah vektor dari dokumen tersebut, berikut Tabel yang menggambarkan proses tersebut ditunjukkan pada Tabel 11.

Tabel 11. Pembobotan Term Frequency

Kata	Term Frequency (TF)				
	Doc1	Doc2	Doc3	Doc4	Doc5
banget	0	0	1	0	0
bunuh	0	0	0	0	1
cicil	0	0	1	0	0
coba	0	0	0	0	1
collector	1	0	0	0	0
debt	1	0	0	0	0

Kata	Term Frequency (TF)				
	Doc1	Doc2	Doc3	Doc4	Doc5
deh	0	0	1	0	0
fintech	0	1	0	0	0
hangat	0	1	0	0	0
hapus	0	0	0	0	1
jamin	0	0	1	0	0
jujur	0	0	0	0	1
julo	0	0	0	1	0
kait	0	0	0	1	0
kalangan	0	1	0	0	0
kenal	0	0	0	1	0
kurir	1	0	0	0	0
laku	0	0	0	1	0
layan	0	1	0	0	0
lending	0	1	0	0	0
manusia	0	0	0	0	1
masyarakat	0	1	0	0	0
meninggal	0	0	0	0	1
minggu	0	0	0	1	0
mungkir	0	1	0	0	0
nama	0	0	0	1	0
negara	0	0	0	0	2
ningsih	0	0	0	1	0
nomor	0	0	0	1	0
nyawa	0	0	0	0	1
online	1	0	1	1	1
orang	0	0	0	1	0
p2p	0	1	0	0	0
pake	0	0	1	0	0
paket	1	0	0	0	0
paylater	0	1	0	0	0
perusahaan	0	1	0	0	0
pinjam	1	0	1	1	1

Kata	Term Frequency (TF)				
	Doc1	Doc2	Doc3	Doc4	Doc5
populasi	0	0	0	0	1
pribadi	0	0	0	1	0
rahayu	0	0	0	1	0
rutin	0	0	0	1	0
salah	0	0	0	0	1
samar	1	0	0	0	0
sambut	0	1	0	0	0
sri	0	0	0	1	0
tagih	1	0	0	1	0
tau	0	0	1	0	0
tawar	0	1	0	0	0
telepon	0	0	0	1	0
terima	0	1	0	0	0
teror	0	0	0	1	0
uang	0	0	1	0	0
urgent	0	0	1	0	0
utang	1	0	0	0	0
waspada	1	0	0	0	0

4.2.2 Pembobotan TF-IDF

Berikut adalah contoh perhitungan pembobotan TF-IDF dari data tweet sample yang sebelumnya digunakan.

4.2.2.1 Perhitungan *Document Frequency (DF)*

Untuk menghitung Document Frequency sebuah term, hanya perlu dijumlahkan nilai dari *Term Frequency* sebuah kata atau term tersebut pada seluruh dokumen yang ada. Pada Tabel sebelumnya berikut ini term “bunuh” hanya muncul satu kali pada dokumen 5, sehingga nilai Document Frequency dari term “bunuh” adalah 1.

Tabel 12. Perhitungan Document Frequency(DF)

Kata	Term Frequency (TF)					Document Frequency (DF)
	Doc1	Doc2	Doc3	Doc4	Doc5	
banget	0	0	1	0	0	1
bunuh	0	0	0	0	1	1
cicil	0	0	1	0	0	1
coba	0	0	0	0	1	1
collector	1	0	0	0	0	1
debt	1	0	0	0	0	1
deh	0	0	1	0	0	1
fintech	0	1	0	0	0	1
hangat	0	1	0	0	0	1
hapus	0	0	0	0	1	1
jamin	0	0	1	0	0	1
jujur	0	0	0	0	1	1
julo	0	0	0	1	0	1
kait	0	0	0	1	0	1
kalangan	0	1	0	0	0	1
kenal	0	0	0	1	0	1
kurir	1	0	0	0	0	1
laku	0	0	0	1	0	1
layan	0	1	0	0	0	1
lending	0	1	0	0	0	1
manusia	0	0	0	0	1	1
masyarakat	0	1	0	0	0	1
meninggal	0	0	0	0	1	1
minggu	0	0	0	1	0	1
mungkir	0	1	0	0	0	1
nama	0	0	0	1	0	1
negara	0	0	0	0	2	2
ningsih	0	0	0	1	0	1
nomor	0	0	0	1	0	1
nyawa	0	0	0	0	1	1

Kata	Term Frequency (TF)					Document Frequency (DF)
	Doc1	Doc2	Doc3	Doc4	Doc5	
online	1	0	1	1	1	4
orang	0	0	0	1	0	1
p2p	0	1	0	0	0	1
pake	0	0	1	0	0	1
paket	1	0	0	0	0	1
paylater	0	1	0	0	0	1
perusahaan	0	1	0	0	0	1
pinjam	1	0	1	1	1	4
populasi	0	0	0	0	1	1
pribadi	0	0	0	1	0	1
rahayu	0	0	0	1	0	1
rutin	0	0	0	1	0	1
salah	0	0	0	0	1	1
samar	1	0	0	0	0	1
sambut	0	1	0	0	0	1
sri	0	0	0	1	0	1
tagih	1	0	0	1	0	2
tau	0	0	1	0	0	1
tawar	0	1	0	0	0	1
telepon	0	0	0	1	0	1
terima	0	1	0	0	0	1
teror	0	0	0	1	0	1
uang	0	0	1	0	0	1
urgent	0	0	1	0	0	1
utang	1	0	0	0	0	1
waspada	1	0	0	0	0	1

4.2.2.2 Menghitung *Inverse Document Frequency*

Rumus perhitungan *Inverse Document Frequency* adalah log dari jumlah

dokumen yang ada dibagi jumlah term yang muncul di seluruh dokumen. Pada Tabel 13 berikut ini nilai Document Frequency kata ambil adalah 1, dan total jumlah dokumen yang ada berjumlah 4. Sehingga untuk menghitung *Inverse Document Frequency* adalah.

$$log_1^5 = 0{,}602$$

Tabel 13. Menghitung *Inverse Document Frequency*

Kata	Term Frequency (TF)					Document Frequency (DF)	Inverse Document Frequency ($\log \frac{df}{\Sigma document}$)
	Doc 1	Doc 2	Doc 3	Doc 4	Doc 5		
banget	0	0	1	0	0	1	0.69897
bunuh	0	0	0	0	1	1	0.69897
cicil	0	0	1	0	0	1	0.69897
coba	0	0	0	0	1	1	0.69897
collector	1	0	0	0	0	1	0.69897
debt	1	0	0	0	0	1	0.69897
deh	0	0	1	0	0	1	0.69897
fintech	0	1	0	0	0	1	0.69897
hangat	0	1	0	0	0	1	0.69897
hapus	0	0	0	0	1	1	0.69897
jamin	0	0	1	0	0	1	0.69897
jujur	0	0	0	0	1	1	0.69897
julo	0	0	0	1	0	1	0.69897
kait	0	0	0	1	0	1	0.69897
kalangan	0	1	0	0	0	1	0.69897
kenal	0	0	0	1	0	1	0.69897
kurir	1	0	0	0	0	1	0.69897
laku	0	0	0	1	0	1	0.69897
layan	0	1	0	0	0	1	0.69897
lending	0	1	0	0	0	1	0.69897
manusia	0	0	0	0	1	1	0.69897
masyarakat	0	1	0	0	0	1	0.69897

Kata	Term Frequency (TF)					Document Frequency (DF)	Inverse Document Frequency ($\log \frac{df}{\Sigma document}$)
	Doc 1	Doc 2	Doc 3	Doc 4	Doc 5		
meninggal	0	0	0	0	1	1	0.69897
minggu	0	0	0	1	0	1	0.69897
mungkir	0	1	0	0	0	1	0.69897
nama	0	0	0	1	0	1	0.69897
negara	0	0	0	0	2	2	0.39794
ningsih	0	0	0	1	0	1	0.69897
nomor	0	0	0	1	0	1	0.69897
nyawa	0	0	0	0	1	1	0.69897
online	1	0	1	1	1	4	0.09691
orang	0	0	0	1	0	1	0.69897
p2p	0	1	0	0	0	1	0.69897
pake	0	0	1	0	0	1	0.69897
paket	1	0	0	0	0	1	0.69897
paylater	0	1	0	0	0	1	0.69897
perusahaan	0	1	0	0	0	1	0.69897
pinjam	1	0	1	1	1	4	0.09691
populasi	0	0	0	0	1	1	0.69897
pribadi	0	0	0	1	0	1	0.69897
rahayu	0	0	0	1	0	1	0.69897
rutin	0	0	0	1	0	1	0.69897
salah	0	0	0	0	1	1	0.69897
samar	1	0	0	0	0	1	0.69897
sambut	0	1	0	0	0	1	0.69897
sri	0	0	0	1	0	1	0.69897
tagih	1	0	0	1	0	2	0.39794
tau	0	0	1	0	0	1	0.69897
tawar	0	1	0	0	0	1	0.69897
telepon	0	0	0	1	0	1	0.69897
terima	0	1	0	0	0	1	0.69897
teror	0	0	0	1	0	1	0.69897
uang	0	0	1	0	0	1	0.69897

Kata	Term Frequency (TF)					Document Frequency (DF)	Inverse Document Frequency ($\log \frac{df}{\Sigma document}$)
	Doc 1	Doc 2	Doc 3	Doc 4	Doc 5		
urgent	0	0	1	0	0	1	0.69897
utang	1	0	0	0	0	1	0.69897
waspada	1	0	0	0	0	1	0.69897

4.2.2.3 Menghitung *TF-IDF*

Untuk menghitung nilai TF-IDF, nilai IDF akan dikalikan dengan nilai kemunculan kata pada tiap dokumen. Kata "Waspada" pada doc 1 memiliki *Term Frequency* yaitu 1 kali kemunculan, dan memiliki IDF senilai 0.699 Maka dengan rumus:

$$Wdt = tfdt \times idft$$

$$Wdt = 1 \times 0.699 = 0.699$$

Maka nilai bobot TF-IDF dari kata "Waspada" pada doc 1 adalah senilai 0.699.

Selanjutnya weight atau bobot sebuah dokumen tersebut dihitung dengan menjumlahkan nilai TF-IDF per dokumen. Pada Tabel 14 dokumen 1 atau D1 memiliki bobot sebesar 4,940

Tabel 14. Perhitungan nilai TF-IDF

Kata	Term Frequency (TF)					DF	IDF	TF-IDF				
	Doc 1	Doc 2	Doc3	Doc 4	Doc 5			Doc1	Doc2	Doc3	Doc4	Doc5
banget	0	0	1	0	0	1	0.69897	0.00000	0.00000	0.69897	0.00000	0.00000
bunuh	0	0	0	0	1	1	0.69897	0.00000	0.00000	0.00000	0.00000	0.69897
cicil	0	0	1	0	0	1	0.69897	0.00000	0.00000	0.69897	0.00000	0.00000
coba	0	0	0	0	1	1	0.69897	0.00000	0.00000	0.00000	0.00000	0.69897
collector	1	0	0	0	0	1	0.69897	0.69897	0.00000	0.00000	0.00000	0.00000
debt	1	0	0	0	0	1	0.69897	0.69897	0.00000	0.00000	0.00000	0.00000
deh	0	0	1	0	0	1	0.69897	0.00000	0.00000	0.69897	0.00000	0.00000
fintech	0	1	0	0	0	1	0.69897	0.00000	0.69897	0.00000	0.00000	0.00000
hangat	0	1	0	0	0	1	0.69897	0.00000	0.69897	0.00000	0.00000	0.00000
hapus	0	0	0	0	1	1	0.69897	0.00000	0.00000	0.00000	0.00000	0.69897
jamin	0	0	1	0	0	1	0.69897	0.00000	0.00000	0.69897	0.00000	0.00000
jujur	0	0	0	0	1	1	0.69897	0.00000	0.00000	0.00000	0.00000	0.69897
julo	0	0	0	1	0	1	0.69897	0.00000	0.00000	0.00000	0.69897	0.00000

Kata	Term Frequency (TF)					DF	IDF	TF-IDF				
	Doc 1	Doc 2	Doc3	Doc 4	Doc 5			Doc1	Doc2	Doc3	Doc4	Doc5
kait	0	0	0	1	0	1	0.69897	0.00000	0.00000	0.00000	0.69897	0.00000
kalangan	0	1	0	0	0	1	0.69897	0.00000	0.69897	0.00000	0.00000	0.00000
kenal	0	0	0	1	0	1	0.69897	0.00000	0.00000	0.00000	0.69897	0.00000
kurir	1	0	0	0	0	1	0.69897	0.69897	0.00000	0.00000	0.00000	0.00000
laku	0	0	0	1	0	1	0.69897	0.00000	0.00000	0.00000	0.69897	0.00000
layan	0	1	0	0	0	1	0.69897	0.00000	0.69897	0.00000	0.00000	0.00000
lending	0	1	0	0	0	1	0.69897	0.00000	0.69897	0.00000	0.00000	0.00000
manusia	0	0	0	0	1	1	0.69897	0.00000	0.00000	0.00000	0.00000	0.69897
masyarakat	0	1	0	0	0	1	0.69897	0.00000	0.69897	0.00000	0.00000	0.00000
meninggal	0	0	0	0	1	1	0.69897	0.00000	0.00000	0.00000	0.00000	0.69897
minggu	0	0	0	1	0	1	0.69897	0.00000	0.00000	0.00000	0.69897	0.00000
mungkir	0	1	0	0	0	1	0.69897	0.00000	0.69897	0.00000	0.00000	0.00000
nama	0	0	0	1	0	1	0.69897	0.00000	0.00000	0.00000	0.69897	0.00000
negara	0	0	0	0	2	2	0.39794	0.00000	0.00000	0.00000	0.00000	0.79588
ningsih	0	0	0	1	0	1	0.69897	0.00000	0.00000	0.00000	0.69897	0.00000

Kata	Term Frequency (TF)					DF	IDF	TF-IDF				
	Doc 1	Doc 2	Doc3	Doc 4	Doc 5			Doc1	Doc2	Doc3	Doc4	Doc5
nomor	0	0	0	1	0	1	0.69897	0.00000	0.00000	0.00000	0.69897	0.00000
nyawa	0	0	0	0	1	1	0.69897	0.00000	0.00000	0.00000	0.00000	0.69897
online	1	0	1	1	1	4	0.09691	0.09691	0.00000	0.09691	0.09691	0.09691
orang	0	0	0	1	0	1	0.69897	0.00000	0.00000	0.00000	0.69897	0.00000
p2p	0	1	0	0	0	1	0.69897	0.00000	0.69897	0.00000	0.00000	0.00000
pake	0	0	1	0	0	1	0.69897	0.00000	0.00000	0.69897	0.00000	0.00000
paket	1	0	0	0	0	1	0.69897	0.69897	0.00000	0.00000	0.00000	0.00000
paylater	0	1	0	0	0	1	0.69897	0.00000	0.69897	0.00000	0.00000	0.00000
perusahaan	0	1	0	0	0	1	0.69897	0.00000	0.69897	0.00000	0.00000	0.00000
pinjam	1	0	1	1	1	4	0.09691	0.09691	0.00000	0.09691	0.09691	0.09691
populasi	0	0	0	0	1	1	0.69897	0.00000	0.00000	0.00000	0.00000	0.69897
pribadi	0	0	0	1	0	1	0.69897	0.00000	0.00000	0.00000	0.69897	0.00000
rahayu	0	0	0	1	0	1	0.69897	0.00000	0.00000	0.00000	0.69897	0.00000
rutin	0	0	0	1	0	1	0.69897	0.00000	0.00000	0.00000	0.69897	0.00000
salah	0	0	0	0	1	1	0.69897	0.00000	0.00000	0.00000	0.00000	0.69897

Kata	Term Frequency (TF)					DF	IDF	TF-IDF				
	Doc 1	Doc 2	Doc3	Doc 4	Doc 5			Doc1	Doc2	Doc3	Doc4	Doc5
samar	1	0	0	0	0	1	0.69897	0.69897	0.00000	0.00000	0.00000	0.00000
sambut	0	1	0	0	0	1	0.69897	0.00000	0.69897	0.00000	0.00000	0.00000
sri	0	0	0	1	0	1	0.69897	0.00000	0.00000	0.00000	0.69897	0.00000
tagih	1	0	0	1	0	2	0.39794	0.39794	0.00000	0.00000	0.39794	0.00000
tau	0	0	1	0	0	1	0.69897	0.00000	0.00000	0.69897	0.00000	0.00000
tawar	0	1	0	0	0	1	0.69897	0.00000	0.69897	0.00000	0.00000	0.00000
telepon	0	0	0	1	0	1	0.69897	0.00000	0.00000	0.00000	0.69897	0.00000
terima	0	1	0	0	0	1	0.69897	0.00000	0.69897	0.00000	0.00000	0.00000
teror	0	0	0	1	0	1	0.69897	0.00000	0.00000	0.00000	0.69897	0.00000
uang	0	0	1	0	0	1	0.69897	0.00000	0.00000	0.69897	0.00000	0.00000
urgent	0	0	1	0	0	1	0.69897	0.00000	0.00000	0.69897	0.00000	0.00000
utang	1	0	0	0	0	1	0.69897	0.69897	0.00000	0.00000	0.00000	0.00000
waspada	1	0	0	0	0	1	0.69897	0.69897	0.00000	0.00000	0.00000	0.00000

4.2.2.4 Menghitung *Best-Match25 (BM-25)*

Perhitungan rumus metode BM25 dapat dilihat pada Persamaan berikut.

$$BM25(dj, q1:N) = \sum_{\epsilon=1}^{N} IDF_{(qi)} . \frac{TF(qi, dj) . (k+1)}{TF(qi, dj) + k . (1 - b + b . \frac{|dj|}{L})}$$

Keterangan:

- N : Jumlah dokumen di dalam korpus
- $IDFqi$: Inverse document frequency dari kata qi
- $TFqi,dj$: Term frequency dari kata qi di dokumen dj
- k, b : Nilai konstan untuk evaluasi
- $|dj|$: Panjang dokumen
- dj L : Rata-rata panjang dokumen di dalam korpus

Dengan mengetahui nilai k = 1.3 dan nilai b = 0.75, berikut contoh perhitungan untuk kata "waspada" pada doc 1.

$$BM25(dj, q1:N) = \sum_{\epsilon=1}^{N} IDF_{(qi)} . \frac{TF(qi, dj) . (k+1)}{TF(qi, dj) + k . (1 - b + b . \frac{|dj|}{L})}$$

$$= 0.69897 . \frac{0.69897 . (1.3+1)}{0.69897 + 1.3 . (1 - 0.75 + 0.75 . \frac{10}{12.8})} = 0.7716714925$$

Maka nilai ekstraksi fitur BM25 untuk kata waspada pada *document 1* adalah 0.7716714925.

Formulasi diatas penulis lakukan kepada data sampel sehingga

mendapatkan hasil tabulasi yang dapat dilihat pada tabel 15. Berikut adalah hasil perhitungan dari BM-25 yang kemudian sebagai tahap ekstraksi fitur untuk selanjutnya dimasukkan kedalam perhitungan multinomial Naive Bayes.

Tabel 15. Perhitungan nilai BM-25

Kata	BM25				
	d1	d2	d3	d4	d5
banget	0	0	0.7716714925	0	0
bunuh	0	0	0	0	0.6942977277
cicil	0	0	0.7716714925	0	0
coba	0	0	0	0	0.6942977277
collector	0.7716714925	0	0	0	0
debt	0.7716714925	0	0	0	0
deh	0	0	0.7716714925	0	0
fintech	0	0.6942977277	0	0	0
hangat	0	0.6942977277	0	0	0
hapus	0	0	0	0	0.6942977277
jamin	0	0	0.7716714925	0	0
jujur	0	0	0	0	0.6942977277
julo	0	0	0	0.594884794	0
kait	0	0	0	0.594884794	0
kalangan	0	0.6942977277	0	0	0
kenal	0	0	0	0.594884794	0
kurir	0.7716714925	0	0	0	0
laku	0	0	0	0.594884794	0
layan	0	0.6942977277	0	0	0
lending	0	0.6942977277	0	0	0
manusia	0	0	0	0	0.6942977277
masyarakat	0	0.6942977277	0	0	0
meninggal	0	0	0	0	0.6942977277

Kata	BM25				
	d1	d2	d3	d4	d5
minggu	0	0	0	0.594884794	0
mungkir	0	0.6942977277	0	0	0
nama	0	0	0	0.594884794	0
negara	0	0	0	0	0.5555963598
ningsih	0	0	0	0.594884794	0
nomor	0	0	0	0.594884794	0
nyawa	0	0	0	0	0.6942977277
online	0.1069898478	0	0.1069898478	0.08247892293	0.09626221641
orang	0	0	0	0.594884794	0
p2p	0	0.6942977277	0	0	0
pake	0	0	0.7716714925	0	0
paket	0.7716714925	0	0	0	0
paylater	0	0.6942977277	0	0	0
perusahaan	0	0.6942977277	0	0	0
pinjam	0.1069898478	0	0.1069898478	0.08247892293	0.09626221641
populasi	0	0	0	0	0.6942977277
pribadi	0	0	0	0.594884794	0
rahayu	0	0	0	0.594884794	0
rutin	0	0	0	0.594884794	0
salah	0	0	0	0	0.6942977277
samar	0.7716714925	0	0	0	0
sambut	0	0.6942977277	0	0	0
sri	0	0	0	0.594884794	0
tagih	0.4393306701	0	0	0.3386818584	0
tau	0	0	0.7716714925	0	0
tawar	0	0.6942977277	0	0	0
telepon	0	0	0	0.594884794	0

Kata	BM25				
	d1	d2	d3	d4	d5
terima	0	0.6942977277	0	0	0
teror	0	0	0	0.594884794	0
uang	0	0	0.7716714925	0	0
urgent	0	0	0.7716714925	0	0
utang	0.7716714925	0	0	0	0
waspada	0.7716714925	0	0	0	0

4.3 Klasifikasi

Data yang telah melewati proses perhitungan untuk vektorisasi melalui BM-25, selanjutnya data akan melalui tahap pengklasifikasian menggunakan algoritma Multinomial Naïve Bayes. Sedangkan untuk tahapan klasifikasi menggunakan algoritma ini contoh perhitungannya secara manual adalah sebagai berikut: Berikut diberikan data latih dan data uji yang akan dihitung, dalam data latih diberikan masing-masing satu tweet dengan kelas berbeda, data uji kemudian diberikan tanpa memberitahu label data tersebut ditunjukkan pada Tabel berikut.

Tabel 16. Contoh data latih dan data uji

Text	label
Waspada dc pinjol menagih utang menyamar menjadi kurir paket.	**NEGATIVE**
Tak dapat dipungkiri, layanan paylater yang ditawarkan oleh perusahaan fintech P2P lending menerima sambutan hangat masyarakat dari berbagai kalangan. https://t.co/6SrZUxCdPr	**POSITIVE**
Ada yg tau pinjaman uang yg bisa dicicil? Lagi urgent bgt, pake jaminan gppa deh, tp jgn pinjol	**POSITIVE**

Text	label
nomor tlp pribadi saya masih rutin di tlp pihak julo terkait tagihan an Sri Rahayu Ningsih, sya tidak kenal dengan org tsb dan sya tidak pernah sekalipun melakukan pinjaman online, sudah 2 minggu sya di teror oleh tlp dari pihak julo	**NEGATIVE**
ada berapa banyak nyawa manusia dinegara ini yang meninggal karena pinjol setiap harinya? apa pinjol gak mau dihapuskan? jujur gue salah satu diantara banyaknya populasi manusia di negara ini yg selalu coba berbagai cara untuk bunuh diri	**NEGATIVE**
pinjam online teror dan bunuh banyak orang	**?**

4.3.1 Hasil perhitungan probabilitas

Pada tahap ini dilakukan klasifikasi data uji dengan menghitung probabilitas suatu dokumen masuk ke dalam suatu kelas tertentu berdasarkan nilai bobot BM-25 dan dengan melakukan laplace smoothing agar menghilangkan nilai 0 dalam data.' Berikut perhitungan untuk menentukan dokumen 5 pada Tabel 17 masuk kedalam kelas negatif, netral atau positif. Untuk menghitung probabilitas doc 5, digunakan rumus perhitungan Multinomial Naïve Bayes dengan menggunakan input weight dari perhitungan BM-25 sebelumnya.

Tabel 17. Hasil perhitungan probabilitas *(Laplace smoothing)*

Term	**positif**	**negative**
pinjam	(0.1069898478 + 1) / (15.4132221 + 37.89194486) = 0.02076702018	(0.2857309871 + 1) / (22.47872277 + 37.89194486) = 0.02129727958
membuat	(0 + 1) / (15.4132221 + 37.89194486) = 0.01875990747	(0 + 1) / (22.47872277 + 37.89194486) = 0.01656433562
online	(0.1069898478 + 1) / (15.4132221 + 37.89194486) = 0.02076702712	(0.2857309871 + 1) / (22.47872277 + 37.89194486) = 0.02129727958

Term	positif	negative
rugi	(0 + 1) / (15.4132221 + 37.89194486) = 0.01875990747	(0 + 1) / (22.47872277 + 37.89194486)= 0.01656433562
bunuh	(0 + 1) / (15.4132221 + 37.89194486) = 0.01875990747	(0.6942977277 +1) / (22.47872277 + 37.89194486) = 0.02806491619
banyak	(0 + 1) / (15.4132221 + 37.89194486) = 0.01875990747	(0 + 1) / (22.47872277 + 37.89194486) = 0.009869921148
orang	(0+1) / (15.4132221 + 37.89194486)= 0.01875990747	(0.594884794 + 1) / (22.47872277 + 37.89194486) = 0.026418207

$$P(d6 \mid positive) = \frac{2}{5} \times 1.2130 \times 10^{-12} = 4.008316127751272 \times 10^{-13}$$

$$P(d6 \mid negative) = \frac{3}{5} x\ 9.1070\, x\, 10^{-13} = 5.464221917315121 \times 10^{-13}$$

4.3.2 Penentuan Kelas

Tahap terakhir adalah menentukan hasil akhir kelas dari sebuah dokumen dengan memilih nilai probabilitas tertinggi. Hasil perhitungan probabilitas doc 6 atau kalimat uji dengan menggunakan Multinomial Naïve Bayes ditambah dengan weight dari perhitungan BM-25 menunjukkan bahwa doc 6 cenderung masuk kedalam sentimen negatif dikarenakan nilai probabilitas kelas negatif yang paling besar dibanding probabilitas kelas lainya.

$$P(d6 \mid negative) > P(d6 \mid positive)$$

4.4 Hasil dan Evaluasi

Hasil penelitian yang diharapkan pada penelitian skripsi dengan judul "Rancang Bangun Aplikasi Untuk Menganalisa Sentimen Masyarakat Terhadap Pinjaman Online Di Media Sosial Twitter Menggunakan Natural Language Processing Dan Naive Bayes Classifier" adalah hasil prediksi sentimen dari sebuah data tweet serta nilai akurasi dari perhitungan yang dilakukan oleh metode yang digunakan. Dalam mengoreksi keberhasilan juga dihitung Confusion matrix untuk mengetahui mana data yang diprediksi benar dan kelas aslinya memang benar, serta data yang diprediksi benar namun kelas aslinya salah, begitu sebaliknya. Software Jupyter Notebook digunakan untuk menjalankan proses penelitian analisis sentimen. Dalam melakukan penelitian ini data akan dibagi menjadi dua skenario, yaitu data hasil crawling yang tidak di stemming serta data hasil crawling yang di stemming. Nilai akurasi akan dibandingkan antara kedua skenario tersebut untuk mengetahui mana yang lebih baik digunakan. Berikut adalah bentuk dataset yang digunakan, ditunjukkan pada Gambar 8.

number	username	tweet_id	text	referenced_twee	label
500	1371650588	1591666599437	titipan saran pinjol yg aman guys		neutral
127	1331650559518	1591757788987	sender kalo ktp hilang disalahgun		negative
1678	1108258395679	1591268362767	nanya kalo pinjam uang pinjol dm		negative
1490	1477650310818	1591358103681	time is belajar coba negara orang		positive
1167	2421711883	1591464550809	pungkas pinjol jaya buka subuh d		neutral
628	1035957565006	1591642887904	menulis surel noam chomsky sela		positive
283	1046084008742	1591718124893	cm sender bayar ukt pake pinjol k		negative
220	2508650070	1591741434994	populer terjebak pinjol ilegal baha		neutral
1829	2508650070	1591151078187	populer ratusan software aplikasi		positive
339	2508650070	1591700658943	populer yes pinjol legal yg blokir c		positive
1820	2508650070	1591174456843	populer yes pinjol legal yg blokir c		positive
931	564951196	1591588674009	software aplikasi ratusan software		neutral
1847	564951196	1591135340374	software aplikasi yess software ap		neutral
1604	9015471770249	1591321500505	osamudazai pinj	1591306765110	positive
1741	1317750858256	1591250162080	angrywolf udh st	1579673583089	negative
551	1591044281837	1591656608530	inspektur iyap be	1591647559604	positive
1801	7650352008961	1591210883459	w t t maaf ya kal	1591205102823	positive
1663	1248517914983	1591276129499	bantu orang yg p	1591275950163	positive
1683	120043643	1591264530948	riyanto didu insti	1591263156680	positive
584	1175344309261	1591652107031	fesss ngga pinjo	1591651419546	negative
555	1412711288745	1591656272864	fesss ga pinjol n	1591651419546	positive
311	7726293770497	1591708125262	fesss gara pinjol	1591651419546	negative
561	1553763567429	1591655056520	fesss gw pinjol k	1591651419546	positive
447	1590051767957	1591673448316	fesss kali pinjem	1591651419546	positive
455	507021732	1591671372459	fesss pinjol ga d	1591651419546	negative
579	4134026114	1591652456479	fesss ngga si mi	1591651419546	neutral
1956	1567336696450	1591054649997	fessss nder plea	1590707807996	positive
2074	7426029592723	1591024964672	fessss ya u data	1590707807996	neutral
1913	1579139192415	1591076268564	riskaa yu bantu p	1590513558944	positive
1432	1429439161698	1591388815419	via kl liat ktp mal	1591384559761	negative
802	7346539681863	1591618395505	hmm nagih pinjo	1591617745996	neutral
1916	11736070721113	1591073212317	daftarin pinjol	1591073002804	neutral

Gambar 8. Sample data *stemming*

	A	B	C	D	E	F	G
1	number	username	tweet_id	text	referenced_twee	label2	label
2	500	1371650588	1591666599437	titipan saran aman guys pinjam bi		neutral	positive
3	127	1331650559518	1591757788987	sender kartu tanda penduduk hila		negative	negative
4	1678	1108258395679	1591268362767	pinjam uang ya tau butuh banget		negative	negative
5	1490	1477650310818	1591358103681	time is belajar coba negara orang		positive	positive
6	1167	2421711883	1591464550809	pungkas jaya buka subuh direct n		neutral	neutral
7	628	1035957565006	1591642887904	menulis surel noam chomsky sela		positive	positive
8	283	1046084008742	1591718124893	sender bayar ukt pakai orang tua		negative	positive
9	220	2508650070	1591741434994	populer terjebak ilegal bahayanya		neutral	neutral
10	1829	2508650070	1591151078187	populer ratusan software aplikasi		positive	positive
11	339	2508650070	1591700658943	populer iya legal blokir ojk layak i		positive	positive
12	1820	2508650070	1591174456843	populer iya legal blokir ojk layak i		positive	positive
13	931	564951196	1591588674009	software aplikasi ratusan software		neutral	neutral
14	1847	564951196	1591135340374	software aplikasi yess software ap		neutral	negative
15	1604	9015471770249	1591321500505	osamudazai lega	1591306765110	positive	positive
16	1741	1317750858256	1591250162080	angrywolf stasiu	1579673583089	negative	neutral
17	551	1591044281837	1591656608530	inspektur iya kak	1591647559604	positive	positive
18	1801	7650352008961	1591210883459	gue t t maaf ya k	1591205102823	positive	positive
19	1663	1248517914983	1591276129499	bantu orang kak	1591275950163	positive	positive
20	1683	120043643	1591264530948	riyanto didu insti	1591263156680	positive	positive
21	584	1175344309261	1591652107031	fesss bank jns k	1591651419546	negative	negative
22	555	1412711288745	1591656272864	fesss namanya u	1591651419546	positive	positive
23	311	7726293770497	1591708125262	fesss gara rumal	1591651419546	negative	negative
24	561	1553763567429	1591655056520	fesss gue kali dc	1591651419546	positive	positive
25	447	1590051767957	1591673448316	fesss kali pinjam	1591651419546	positive	positive
26	455	507021732	1591671372459	fesss dasar huku	1591651419546	negative	negative
27	579	4134026114	1591652456479	fesss sih pinjam	1591651419546	neutral	positive
28	1956	1567336696450	1591054649997	fessss nder plea	1590707807996	positive	positive
29	2074	7426029592723	1591024964672	fessss ya datan	1590707807996	neutral	neutral
30	1913	1579139192415	1591076268564	riskaa yu bantu	1590513558944	positive	positive
31	1432	1429439161698	1591388815419	via lihat kartu tar	1591384559761	negative	neutral
32	802	7346539681863	1591618395505	hem menagih	1591617745996	neutral	negative
33	1916	1173607072111	1591073212317	daftarin	1591073002804	neutral	neutral
34	962	466155027	1591579168076	coba cek kakak	1584507903868	positive	positive
35	1764	1241397809003	1591235189073	aly gaya bahas k	1591231873954	negative	negative
36	885	2351576791	1591600904205	makan istri wkwk	1591362861066	positive	positive
37	2018	2656561256	1591042617735	sjsksjsk bantu ba	1591042008453	positive	positive

Gambar 9. Sample data non stemming

Ekstraksi fitur dari data yang telah dilakukan preprocessing tersebut dilakukan dengan metode BM25. Tahap - tahap awal dalam menghitung BM-25 adalah yang pertama mencari nilai *Term Frequency* atau jumlah frekuensi kemunculan suatu kata pada setiap dokumen.

1. Perhitungan *Term Frequency*

Berikut potongan kode untuk mendapatkan nilai *Term Frequency* (TF) dari keseluruhan kata.

```
import numpy as np
import pandas as pd
import sklearn
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
from lib.feature_extractor import TfidfTransformer


df = pd.read_csv('../output_3.csv')

print("COUNT:", df['label'].count())
bow_transformer = CountVectorizer().fit(df['text'])

text_bow = bow_transformer.transform(df['text'])

print(text_bow)
print(text_bow.shape)
```

Gambar 10. Potongan kode Perhitungan *Term Frequenc y*

Nilai *Term Frequency* yang dihasilkan untuk beberapa kata dari keseluruhan data training dapat dilihat dari gambar 11 dan 12.

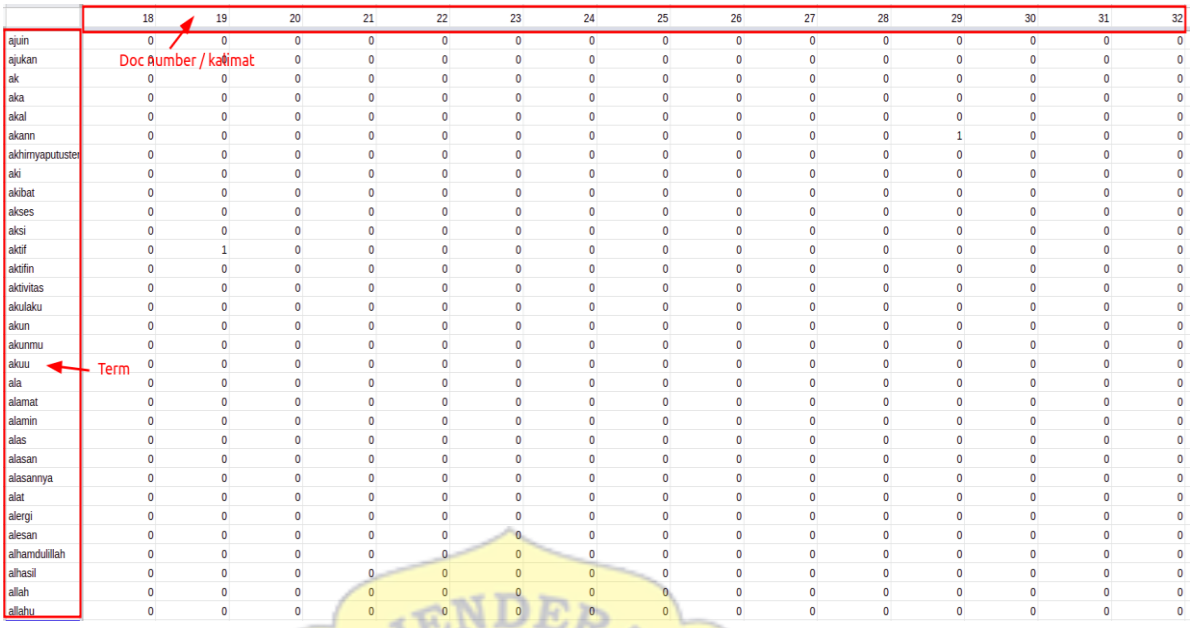
```
COUNT: 1009
  (0, 80)       1
  (0, 217)      1
  (0, 921)      1
  (0, 950)      1
  (0, 1023)     1
  (0, 1092)     1
  (0, 1691)     1
  (0, 1862)     1
  (0, 1934)     1
  (0, 2349)     1
  (0, 2567)     1
  (0, 2973)     1
  (0, 3048)     1
  (0, 3174)     1
  (1, 269)      1
  (1, 337)      1
  (1, 694)      1
  (1, 722)      1
  (1, 899)      1
  (1, 986)      1
  (1, 1088)     2
  (1, 1089)     1
  (1, 1323)     1
  (1, 1397)     1
  (1, 1532)     2
```

Gambar 11. Keluaran Perhitungan *Term Frequency Data Stemming*

```
COUNT: 1009
  (0, 64)       1
  (0, 188)      1
  (0, 797)      1
  (0, 851)      1
  (0, 904)      1
  (0, 1101)     1
  (0, 1648)     1
  (0, 1949)     1
  (0, 2143)     1
  (0, 2435)     1
  (0, 2488)     1
  (1, 178)      1
  (1, 231)      1
  (1, 592)      1
  (1, 616)      1
  (1, 900)      2
  (1, 901)      1
  (1, 1096)     2
  (1, 1147)     1
  (1, 1293)     1
  (1, 1479)     1
  (1, 1865)     2
  (1, 2211)     3
  (1, 2376)     1
  (1, 2380)     2
  :     :
  (1006, 2315)  2
  (1006, 2451)  2
  (1006, 2532)  1
  (1007, 92)    1
  (1007, 371)   1
  (1007, 426)   1
  (1007, 1467)  1
  (1007, 1543)  1
  (1007, 1585)  1
  (1007, 1717)  1
  (1007, 1792)  1
  (1007, 2100)  1
  (1007, 2190)  1
  (1007, 2343)  1
  (1008, 92)    1
  (1008, 371)   1
  (1008, 426)   1
  (1008, 1467)  1
  (1008, 1543)  1
  (1008, 1585)  1
  (1008, 1717)  1
  (1008, 1792)  1
  (1008, 2100)  1
  (1008, 2190)  1
  (1008, 2343)  1
(1009, 2656)
```

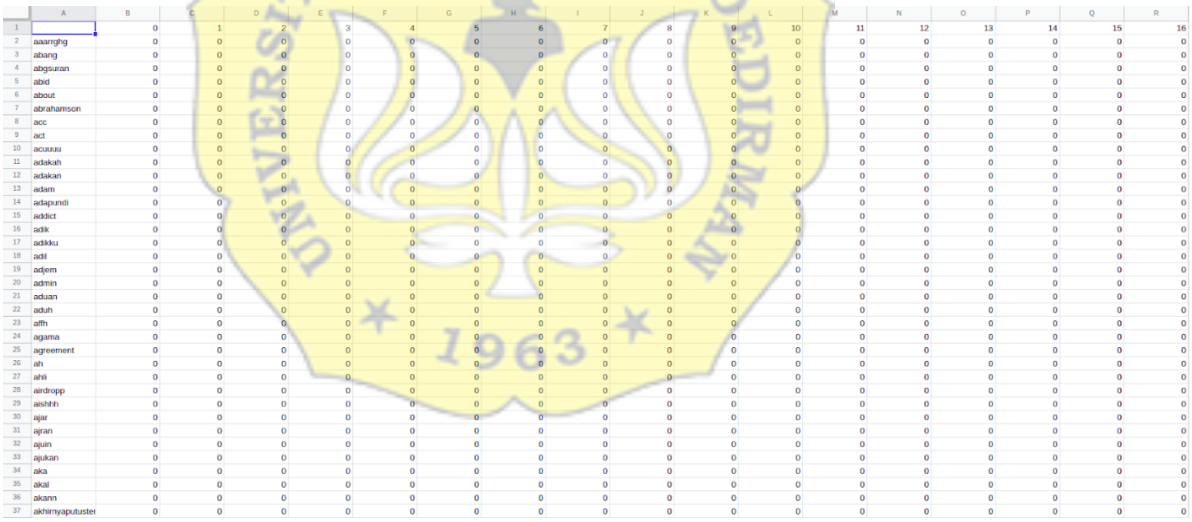
Gambar 12. Keluaran Perhitungan Term Frequency Data Non Stemming

Visualisasi dari data diatas dalam bentuk CSV hasilnya akan dapat dilihat pada gambar berikut, dimana sumbu x adalah semua term yang ada pada keseluruhan data tweet, dan sumbu y adalah kalimat atau document 1 sampai x sejumlah total data tweet, dan isi dari table merupakan *Term Frequency* atau jumlah kemunculan suatu kata pada suatu document.



	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
ajuin	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ajukan	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ak	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
aka	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
akal	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
akann	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
akhirnyaputuster	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
aki	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
akibat	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
akses	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
aksi	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
aktif	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
aktifin	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
aktivitas	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
akulaku	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
akun	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
akunmu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
akuu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ala	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
alamat	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
alamin	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
alas	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
alasan	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
alasannya	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
alat	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
alergi	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
alesan	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
alhamdulillah	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
alhasil	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
allah	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
allahu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Gambar 13. Gambaran Hasil Perhitungan *Term Frequency Data Stemming*



	A	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
2	aaarrghg	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	abang	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	abgpuran	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	abid	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	about	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	abrahamson	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	acc	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	act	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	acuuuu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	adakah	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	adakan	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	adam	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	adapundi	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	addict	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	adik	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	adikku	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	adil	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	adjem	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	admin	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	aduan	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	aduh	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	affh	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	agama	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	agreement	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	ah	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	ahli	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28	airdropp	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29	aishhh	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30	ajar	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	ajran	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32	ajuin	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
33	ajukan	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34	aka	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
35	akal	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
36	akann	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
37	akhirnyaputuster	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Gambar 14. Gambaran Hasil Perhitungan Term Frequency Data Non Stemming

Potongan gambar 13 dan 14 hanya menunjukkan sebagian data, untuk data keseluruhan dapat diakses melalui link berikut https://drive.google.com/open?id=1urVJTNexvt_No56RM7dmNEBfKYwHjoTA

2. Perhitungan Nilai BM-25

Berikut adalah potongan kode untuk mendapatkan nilai BM25 dari masing-masing kata dalam masing - masing dokumen. Nilai BM25 akan digunakan pada tahap selanjutnya ketika membuat model prediksi multinomial, nilai bm25 akan menjadi input data dalam model.

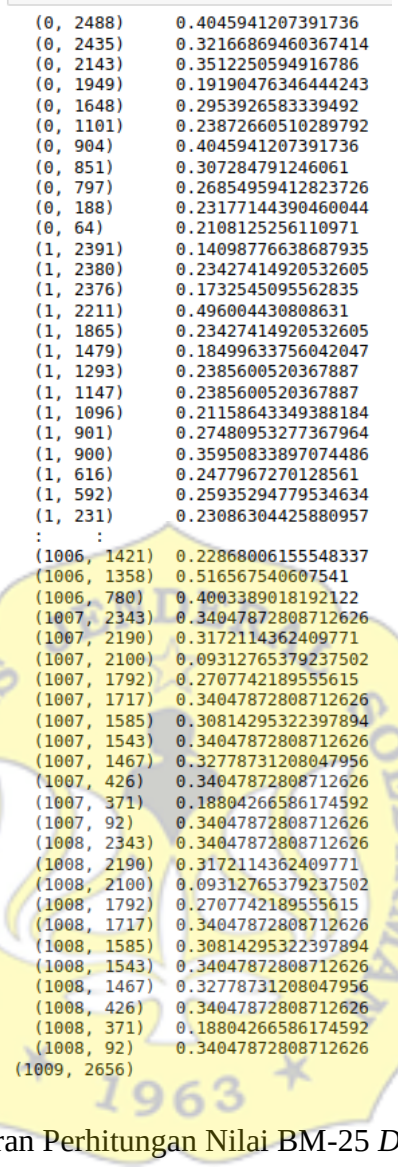
```
bm25_transformer=TfidfTransformer(bm25_tf=True,use_bm25idf=False).fit(t
ext_bow)
tweet_bm25=bm25_transformer.transform(text_bow)
print(tweet_bm25)
print(tweet_bm25.shape)
```

Gambar 15. Potongan kode Perhitungan Nilai BM-25

Berikut adalah hasil output dari nilai BM25 masing - masing kata terhadap masing - masing dokumen uji.

```
  (0, 3174)     0.14394162607400304
  (0, 3048)     0.20706419295218023
  (0, 2973)     0.36584100482237036
  (0, 2567)     0.3194413908750458
  (0, 2349)     0.06051688020398461
  (0, 1934)     0.2795150706993516 6
  (0, 1862)     0.2518389018823334
  (0, 1691)     0.34530884856357 19
  (0, 1092)     0.36584100482237036
  (0, 1023)     0.29564109890195334
  (0, 950)      0.2795150706993516 6
  (0, 921)      0.20085579182332208
  (0, 217)      0.22704890033236083
  (0, 80)       0.21167086989653142
  (1, 2864)     0.1622031490358197
  (1, 2848)     0.18420055137152644
  (1, 2651)     0.465238641140689
  (1, 2349)     0.04472466813816245
  (1, 1779)     0.19028268030163112
  (1, 1578)     0.23608140651538026
  (1, 1532)     0.24524231186194004
  (1, 1397)     0.23608140651538026
  (1, 1323)     0.1395190462186292
  (1, 1089)     0.27037278651610125
  (1, 1088)     0.3858463738833843
  :     :
  (1007, 2629)  0.3040466833410193
  (1007, 2516)  0.1333428153693775
  (1007, 2339)  0.16105169695546573
  (1007, 2177)  0.2786930754073621
  (1007, 2147)  0.18579065173358592
  (1007, 2059)  0.3241709741038131
  (1007, 1853)  0.2963957830783547
  (1007, 1820)  0.3241709741038131
  (1007, 1769)  0.31309577769679797
  (1007, 501)   0.3241709741038131
  (1007, 438)   0.20083857647271164
  (1007, 108)   0.3241709741038131
  (1008, 2814)  0.3241709741038131
  (1008, 2629)  0.3040466833410193
  (1008, 2516)  0.1333428153693775
  (1008, 2339)  0.16105169695546573
  (1008, 2177)  0.2786930754073621
  (1008, 2147)  0.18579065173358592
  (1008, 2059)  0.3241709741038131
  (1008, 1853)  0.2963957830783547
  (1008, 1820)  0.3241709741038131
  (1008, 1769)  0.31309577769679797
  (1008, 501)   0.3241709741038131
  (1008, 438)   0.20083857647271164
  (1008, 108)   0.3241709741038131
(1009, 3199)
```

Gambar 16. Keluaran Perhitungan Nilai BM-25 *Data Stemming*



Gambar 17. Keluaran Perhitungan Nilai BM-25 *Data Non Stemming*

3. Pelatihan data uji

Cara mendapatkan data latih dan data uji dilakukan dengan pembagian total dataset yang ada menjadi dua, yaitu data training (latih) dan data test (uji). Kedua dataset memiliki ukuran yang sama sehingga keduanya memiliki ukuran data test sebesar 20% dari 1000 data. Artinya keduanya memiliki 202 data uji, dan 807 data latih. Data ini akan dibagi menjadi dua yaitu data x dan data y, sehingga total ada 4, x train, x test, y train, y test.

```
x_train, x_test, y_train, y_test = train_test_split(tweet_bm25,
df.label,test_size=0.2, random_state=35)

print(x_train.shape)
print(y_train.shape)
print(x_test.shape)
print(x_test.shape)
```

Gambar 18. Potongan kode Pelatihan data uji

Berikut adalah output dari pembagian data uji dan data latih dari keseluruhan 1009 data *tweet*.

```
(807, 3199)
(807,)
(202, 3199)
(202,)
```

Gambar 19. Keluaran Pelatihan data uji

4. Pembuatan Model Multinomial Naive Bayes dan Prediksi Data Uji

Kemudian dilakukan pembuatan model, menggunakan library scikit-learn, yang merupakan library pemrograman di python. Metode yang digunakan adalah Multinomial Naive Bayes, dengan menggunakan data latih x_train, dan y_train untuk melatih model. Kemudian hasil pelatihan nya diujikan ke data uji x_test dan y_test dengan memberi prediksi label. Berikut adalah potongan kode pembuatan model multinomial naive bayes.

```
from sklearn.preprocessing import MaxAbsScaler
from sklearn.pipeline import Pipeline

p =
Pipeline([('Normalizing',MaxAbsScaler()),('MultinomialNB',MultinomialNB
())])

model = MultinomialNB().fit(x_train,y_train)
prediction = model.predict(x_test)
predict= pd.Series(prediction)
print(predict.to_string())
```

Gambar 20. Kode Pembuatan Model Multinomial Naive Bayes dan Prediksi Data Uji

Berikut adalah hasil output dari prediksi data uji tersebut

0	positive
1	neutral
2	negative
3	positive
4	negative
5	neutral
6	neutral
7	neutral
8	neutral
9	positive
10	positive
11	positive
12	neutral
13	neutral
14	neutral
15	positive
16	neutral
17	neutral
18	neutral
19	neutral
20	neutral
21	negative
22	negative
23	positive
24	neutral
25	negative
26	positive
27	neutral
28	positive
29	neutral
30	neutral
31	negative
32	neutral
33	negative
34	neutral
35	neutral
36	positive
37	neutral
38	neutral
39	negative
40	positive
41	positive
42	negative
43	neutral

Gambar 21. Keluaran Pembuatan Model Multinomial Naive Bayes dan Prediksi Data Uji

Data Stemming

0	positive
1	neutral
2	negative
3	positive
4	negative
5	neutral
6	neutral
7	neutral
8	neutral
9	positive
10	positive
11	positive
12	neutral
13	neutral
14	neutral
15	positive
16	neutral
17	neutral
18	neutral
19	neutral
20	neutral
21	negative
22	negative
23	positive
24	neutral
25	neutral
26	positive
27	neutral
28	positive
29	neutral
30	neutral
31	negative
32	neutral
33	negative
34	neutral
35	neutral
36	positive
37	neutral
38	neutral
39	negative
40	positive
41	negative
42	negative
43	neutral

Gambar 22. Keluaran Model Multinomial Naive Bayes dan Prediksi Data Uji

Data Non stemming

Setelah berhasil melakukan prediksi data uji, dilakukan perhitungan akurasi, recall dan precision beserta confusion matrix. Berikut kode program ditunjukkan pada Gambar 23.

```
from time import time
from sklearn import metrics
import matplotlib.pyplot as plt
import seaborn as sn
from sklearn.metrics import confusion_matrix
from sklearn.metrics import plot_confusion_matrix
from pandas import DataFrame

t = time()
y_pred = model.predict(x_test)

test_time = time() - t
print("test time:  %0.3fs" % test_time)

score1 = metrics.accuracy_score(y_test, y_pred)
print("accuracy:   %0.3f" % score1)

print(metrics.classification_report(y_test, y_pred,
target_names=['negatif', 'positif']))
```

Gambar 23. Potongan kode penentuan nilai akurasi model

Hasil output dari potongan kode pada gambar 18 adalah sebagai berikut:

```
test time:  0.001s
accuracy:   0.649
              precision    recall  f1-score   support

     negatif       0.70      0.59      0.64        51
     neutral       0.55      0.82      0.66        76
     positif       0.83      0.52      0.64        75

    accuracy                           0.65       202
   macro avg       0.69      0.64      0.65       202
weighted avg       0.69      0.65      0.65       202
```

Gambar 24. Keluaran penentuan nilai akurasi model *data stemming*

```
test time:  0.002s
accuracy:   0.663
              precision    recall  f1-score   support

    negative       0.73      0.56      0.63        57
     neutral       0.52      0.83      0.64        65
    positive       0.89      0.60      0.72        80

    accuracy                           0.66       202
   macro avg       0.71      0.66      0.66       202
weighted avg       0.72      0.66      0.67       202
```

Gambar 25. Keluaran penentuan nilai akurasi model *data non stemming*

Hasil akurasi yang didapat adalah 64,9% dengan waktu test selama 0,001detik, sedangkan nilai precision untuk kelas negatif sebesar 70%, kelas netral 55%, dan kelas positif 65%. Untuk nilai recall pada kelas negatif sebesar 59%, kelas netral sebesar 82% dan kelas positif sebesar 52%. Total data uji yang masuk kedalam kelas negatif adalah 51 tweet, kelas netral 76 tweet dan kelas positif 75 tweet.

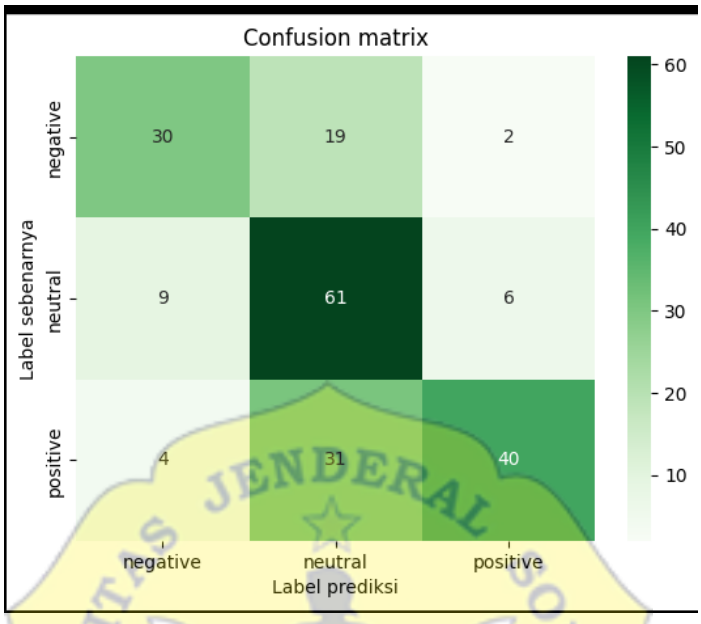
Selanjutnya akan dilakukan pengecekan evaluasi performansi menggunakan tabel confusion matrix. Confusion matrix digunakan untuk menentukan nilai *true positive* dan atau *true negative* untuk menentukan apakah hasil prediksi tersebut benar seutuhnya atau tidak. Berikut potongan kode untuk mendapatkan nilai visualisasi dari *Confusion Matrix* menggunakan library *seaborn.*

```
columns = ['negatif','netral','positif']
confm = confusion_matrix(y_test, y_pred)
df_cm = DataFrame(confm, index=columns, columns=columns)

ax = sn.heatmap(df_cm, cmap='Greens', annot=True)
ax.set_title('Confusion matrix')
ax.set_xlabel('Label prediksi')
ax.set_ylabel('Label sebenarnya')
```

Gambar 26. Kode visualisasi confusion matrix (Evaluasi performansi)

Hasil confusion matrix dari data tersebut adalah sebagai berikut, sebagaimana ditunjukkan pada Gambar 27.

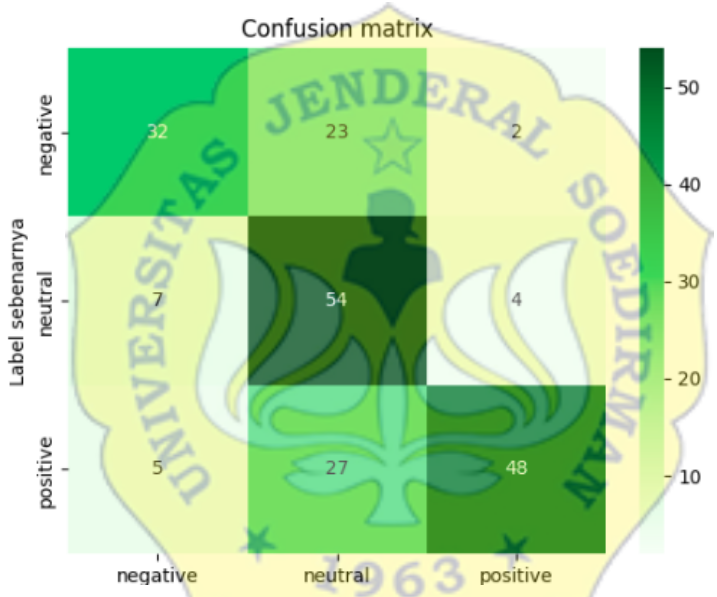


Gambar 27. Visualisasi confusion matrix (Evaluasi performansi) *Data Stemming*

Pada gambar 27 ditunjukkan bahwa jumlah nilai error pada polaritas negatif adalah sebanyak 21 data, dengan rincian 19 data lebih condong ke polaritas netral dan 2 data ke polaritas positif. Sedangkan untuk data berpolaritas netral yaitu pada baris kedua, jumlah data error mencapai 15 data, 9 data condong ke polaritas negatif dan 6 data condong ke polaritas positif. Selanjutnya untuk data berpolaritas positif, yaitu pada baris ketiga. Jumlah error sebanyak 35 data, 4 dari data tersebut masuk kedalam polaritas negatif, dan 31 masuk ke polaritas netral.

Hasil analisa dari confusion matrix yang ada menunjukkan bahwa data negatif cenderung bisa masuk ke polaritas netral dibanding positif, melihat laju error data negatif lebih banyak masuk ke netral, yaitu 19 data dibanding positif, 2 data. Selanjutnya adalah data berpolaritas netral. Dari hasil confusion matrix yang ada data netral cenderung lebih banyak masuk kedalam polaritas negatif dibanding

positif. Hal ini dapat dilihat bahwa data error netral yang menuju polaritas negatif sebanyak 9 data, sedangkan data positif hanya 6 data. Untuk data berpolaritas positif, cenderung masuk kedalam polaritas netral, dibandingkan data negatif, dengan jumlah data yang masuk kedalam polaritas netral 31, dan data yang masuk kedalam polaritas negatif hanya 4. Hal ini menunjukkan bahwa sentimen positif beririsan erat dengan sentimen netral dikarenakan ke-ambiguan kata dalam cuitan yang ditemukan.



Gambar 28. Visualisasi confusion matrix (Evaluasi performansi) *Data non stemming*

Pada data uji non stemming gambar 28 ditunjukkan bahwa jumlah nilai error pada polaritas negatif adalah sebanyak 25 data, dengan rincian 23 data lebih condong ke polaritas netral dan 2 data ke polaritas positif. Sedangkan untuk data berpolaritas netral yaitu pada baris kedua, jumlah data error mencapai 11 data, 7 data condong ke polaritas negatif dan 4 data condong ke polaritas positif.

Selanjutnya untuk data berpolaritas positif, yaitu pada baris ketiga. Jumlah error sebanyak 32 data, 5 dari data tersebut masuk kedalam polaritas negatif, dan 27 masuk ke polaritas netral.

Hasil analisa dari confusion matrix yang ada menunjukkan bahwa data negatif cenderung bisa masuk ke polaritas netral dibanding positif, melihat laju error data negatif lebih banyak masuk ke netral, yaitu 23 data dibanding positif, 2 data. Selanjutnya adalah data berpolaritas netral. Dari hasil confusion matrix yang ada data netral cenderung lebih banyak masuk kedalam polaritas negatif dibanding positif. Hal ini dapat dilihat bahwa data error netral yang menuju polaritas negatif sebanyak 7 data, sedangkan data positif hanya 4 data. Untuk data berpolaritas positif, cenderung masuk kedalam polaritas netral, dibandingkan data negatif, dengan jumlah data yang masuk kedalam polaritas netral 27, dan data yang masuk kedalam polaritas negatif hanya 5. Hal ini menunjukkan bahwa sentimen positif beririsan erat dengan sentimen netral dikarenakan ke-ambiguan kata dalam cuitan yang ditemukan.