BAB IV
HASIL DAN PEMBAHASAN

4.1 *Preprocessing Data*

Setelah dilakukan perancangan maka dilakukan implementasi dari tiap Langkah-langkah tersebut yang dimulai dari *data preparation*, *cleaning data* dan *labeling data*.

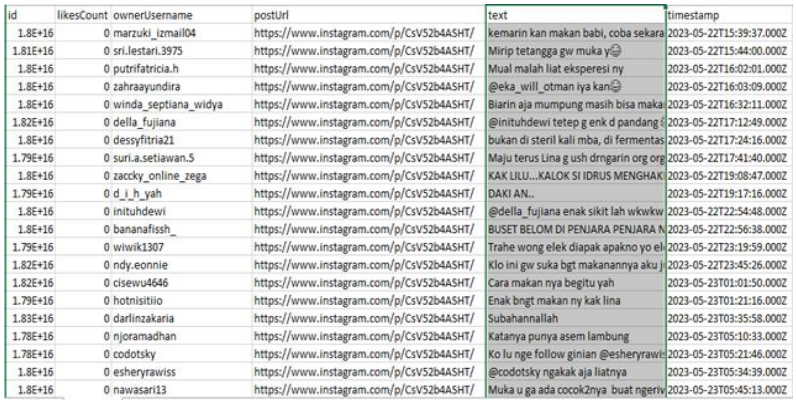
4.1.1 Hasil *Data Preparation*

Berdasarkan penjelasan yang diberikan pada sub bab sebelumnya, pengumpulan *data* collection atau scraping *data* dilakukan melalui beberapa tahap berikut:

1. Mengumpulkan *Link*

   Pada tahap ini, langkah awal adalah mengumpulkan *link posting*an dari berbagai pengguna *Instagram*. *Link* tersebut merujuk pada *posting*an yang mengandung gambar, video, atau *reels*. Sebagai contoh, peneliti telah mengumpulkan tiga *link posting*an seperti yang terlihat di atas.
2. *Scraping Data*

   Setelah terkumpulnya *link* dari setiap *posting*an tersebut kemudian hasil dari scraping *data* komentar dapat dilihat pada gambar 4.1 berikut:



id	likesCount	ownerUsername	postUrl	text	timestamp
1.8E+16	0	marzuki_izmail04	https://www.instagram.com/p/CsV52b4ASHT/	kemarin kan makan babi, coba sekara	2023-05-22T15:39:37.000Z
1.81E+16	0	sri.lestari.3975	https://www.instagram.com/p/CsV52b4ASHT/	Mirip tetangga gw muka y	2023-05-22T15:44:00.000Z
1.8E+16	0	putrifatricia.h	https://www.instagram.com/p/CsV52b4ASHT/	Mual malah liat eksperesi ny	2023-05-22T16:02:01.000Z
1.8E+16	0	zahraayundira	https://www.instagram.com/p/CsV52b4ASHT/	@eka_will_otman iya kan	2023-05-22T16:03:09.000Z
1.8E+16	0	winda_septiana_widya	https://www.instagram.com/p/CsV52b4ASHT/	Biarin aja mumpung masih bisa maka	2023-05-22T16:32:11.000Z
1.82E+16	0	della_fujiana	https://www.instagram.com/p/CsV52b4ASHT/	@inituhdewi tetep g enk d pandang	2023-05-22T17:12:49.000Z
1.8E+16	0	dessyfitria21	https://www.instagram.com/p/CsV52b4ASHT/	bukan di steril kali mba, di fermentas	2023-05-22T17:24:16.000Z
1.79E+16	0	suri.a.setiawan.5	https://www.instagram.com/p/CsV52b4ASHT/	Maju terus Lina g ush drngarin org org	2023-05-22T17:41:40.000Z
1.8E+16	0	zaccky_online_zega	https://www.instagram.com/p/CsV52b4ASHT/	KAK LILU...KALOK SI IDRUS MENGHAK	2023-05-22T19:08:47.000Z
1.79E+16	0	d_i_h_yah	https://www.instagram.com/p/CsV52b4ASHT/	DAKI AN..	2023-05-22T19:17:16.000Z
1.8E+16	0	inituhdewi	https://www.instagram.com/p/CsV52b4ASHT/	@della_fujiana enak sikit lah wkwkw	2023-05-22T22:54:48.000Z
1.8E+16	0	bananafissh_	https://www.instagram.com/p/CsV52b4ASHT/	BUSET BELOM DI PENJARA PENJARA N	2023-05-22T22:56:38.000Z
1.79E+16	0	wiwik1307	https://www.instagram.com/p/CsV52b4ASHT/	Trahe wong elek diapak apakno yo el	2023-05-22T23:19:59.000Z
1.82E+16	0	ndy.eonnie	https://www.instagram.com/p/CsV52b4ASHT/	Klo ini gw suka bgt makanannya aku	2023-05-22T23:45:26.000Z
1.82E+16	0	cisewu4646	https://www.instagram.com/p/CsV52b4ASHT/	Cara makan nya begitu yah	2023-05-23T01:01:50.000Z
1.79E+16	0	hotnisitiio	https://www.instagram.com/p/CsV52b4ASHT/	Enak bngt makan ny kak lina	2023-05-23T01:21:16.000Z
1.83E+16	0	darlinzakaria	https://www.instagram.com/p/CsV52b4ASHT/	Subahannallah	2023-05-23T03:35:58.000Z
1.78E+16	0	njoramadhan	https://www.instagram.com/p/CsV52b4ASHT/	Katanya punya asem lambung	2023-05-23T05:10:33.000Z
1.78E+16	0	codotsky	https://www.instagram.com/p/CsV52b4ASHT/	Ko lu nge follow ginian @esheryrawis	2023-05-23T05:21:46.000Z
1.8E+16	0	esheryrawiss	https://www.instagram.com/p/CsV52b4ASHT/	@codotsky ngakak aja liatnya	2023-05-23T05:34:39.000Z
1.8E+16	0	nawasari13	https://www.instagram.com/p/CsV52b4ASHT/	Muka u ga ada cocok2nya buat ngeriv	2023-05-23T05:45:13.000Z

Gambar 4. 1 Hasil Pengumpulan *Data*

Pada tahap ini, *data* komentar diambil dari setiap *link postingan* yang telah dikumpulkan sebelumnya. Dalam contoh di atas, terdapat tiga *postingan* dengan masing-masing memiliki beberapa komentar. *Data* komentar tersebut dapat digunakan untuk analisis lebih lanjut atau proses *cleaning* dan *labeling* dalam penelitian yang sedang dilakukan. Dimana jumlah data yang diperoleh dari hasil pengumpulan data ini sebanyak 8.734 data komentar.

Berikut ini adalah yang akan contoh konteks pengambilan *data* komentar *Instagram* berdasarkan *posting*an yang akan diambil *data*nya yang bertujuan untuk memperoleh *data* komentar yang dapat diguankan dalam menganalisis dan membuat model untuk mendeteksi unsur kalimat *toxic* yang ada adalam *posting*an tersebut atau kalimat yang digunakan apakah memiliki unsur kalimat *toxic*. Terdapat 2 poin utama yang dijadikan sebagai acuan dalam pengambilan *data* komentar *posting*an *Instagram* seperti berikut:

1. Konsep atau konteks dari Konten yang di *posting*

   Gambar 4.2 berikut merupakan contoh dari *postingan instagram* yang akan diambil *data*nya

Gambar 4. 2 Contoh Postingan

Gambar 4.2 diatas merupakan *postingan* yang telah di *publish* oleh *user* lainnya (bukan pemilik asli video atau objek yang ada dalam video) dari gambar diatas deskripsi atau konteks video yang dapat dijelaskan adalah melibatkan seorang subjek perempuan yang terlibat dalam praktik *lipsyncing*, dimana gerakan bibirnya disinkronkan dengan suara yang berasal dari sumber eksternal. Namun, yang menonjol dalam presentasi visual ini adalah ekspresi wajah yang dibentuk oleh elemen riasan bibir yang menonjolkan aspek *non*-konvensional dan alis yang

dipersentasikan dengan tampilan yang sengaja kurang teratur. Lebih jauh, kualitas estetika yang diajukan oleh elemen riasan ini kemudian diperkuat oleh bahasa tubuh subjek, yang secara tersirat mengandung intonasi mengkritik atau mengejek pihak lain.

Hasil analisa dari konsep video ini mungkin bertujuan untuk mengangkat isu-isu yang terkait dengan norma estetika dan konformitas sosial. Secara implisit, penyajian visual ini dapat dianggap sebagai suatu bentuk penolakan terhadap paradigma kecantikan yang konvensional serta penggambaran yang lebih eksperimental mengenai estetika individu. Selain itu, bahasa tubuh yang mengandung sentimen sindiran memberikan dimensi interpretatif tambahan kepada penonton, yang dapat menggiring refleksi terkait citra diri, persepsi kelompok, atau bahkan representasi sosial yang lebih luas.

Selain hal-hal yang telah dijelasakan sebelumnya penting untuk dicatat bahwa interpretasi terhadap karya seni, termasuk video yang dijelaskan, dapat bervariasi secara signifikan antara individu. Intensi dari pemilik video mungkin tidak selalu sepenuhnya tercermin dalam persepsi yang dimiliki oleh setiap penonton atau audiens.

2. Komentar

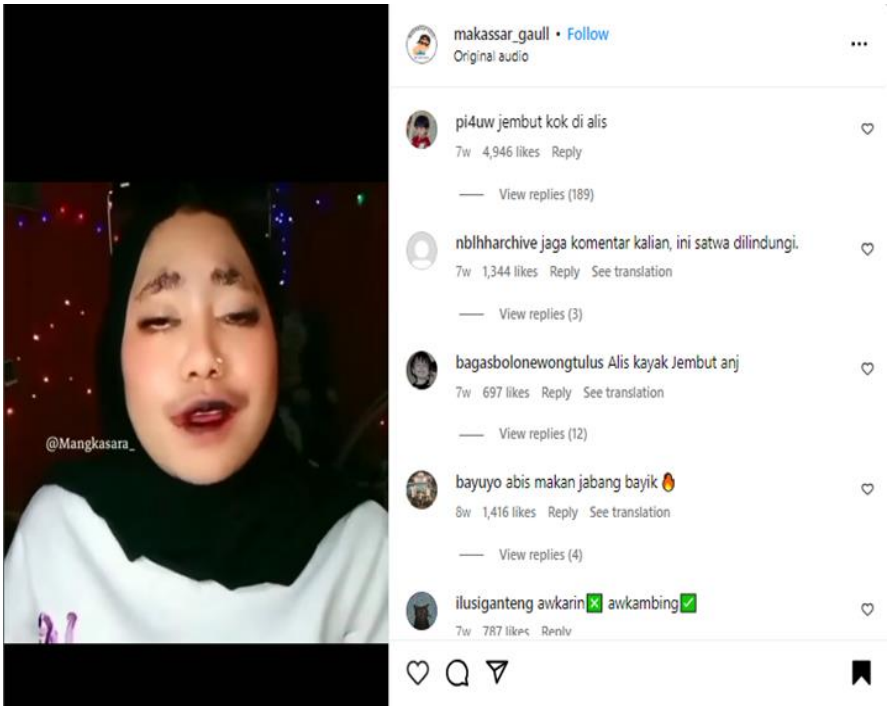
   Setelah menguraikan konteks dan deskripsi dari konten yang dijelaskan sebelumnya, langkah selanjutnya yang akan diambil oleh peneliti adalah melakukan analisis terhadap respon dan tanggapan dari para audiens yang telah meninjau atau menyaksikan video yang di*posting*. Analisis ini dilakukan dengan memfokuskan perhatian pada kolom komentar yang menyertainya, dengan tujuan mengidentifikasi kemungkinan adanya penggunaan bahasa, kalimat, atau kata-kata yang bersifat *toxic*. Pada tahap ini, penting untuk melihat keragaman reaksi yang tercermin dalam komentar yang diberikan oleh setiap pengguna.

   Langkah pertama dalam analisis ini adalah mengidentifikasi adanya elemen kalimat *toxic* dalam kolom komentar. Toksikitas ini dapat mencakup berbagai bentuk, seperti penghinaan, pelecehan, ujaran

kebencian, atau komentar merendahkan. Setiap komentar perlu dianalisis secara mendalam untuk memahami konteks dan niat di balik kata-kata yang digunakan. Pada titik ini, *data* komentar yang mengandung elemen kalimat *toxic* akan menjadi fokus penelitian lebih lanjut.

Berdasarkan temuan tersebut, peneliti akan mempertimbangkan untuk menggunakan *data* komentar yang mengandung unsur kalimat *toxic* sebagai *dataset* yang akan diproses lebih lanjut dalam pembuatan model deteksi kalimat *toxic*. *Data*set ini akan mengandung contoh-contoh kalimat yang bersifat *toxic* dari berbagai reaksi komentar yang diberikan oleh para pengguna.

Berdasarkan penjelasan diatas berikut merupakan contoh komentar yang mengandung unsur *toxic* yang dapat dilihat pada gambar 4.3 berikut:



Gambar 4. 3 Isi Komentar

Berdarakan gambar 4.3 diatas dapat dilihat beberapa analisa awal yang dilakukan terkait makna dari komentar yang diberikan terhadapa beberapa *data* komentar berikut:

- "jembut kok di alis"

  Komentar ini mengandung bahasa kasar yang dapat dianggap menghina atau merendahkan. Penggunaan kata yang tidak pantas ("jembut") menunjukkan karakter *toxic* dalam komentar ini. Komentar ini tampaknya berusaha menjadikan riasan alis yang aneh dalam video sebagai subjek lelucon yang merendahkan.
- "jaga komentar kalian, ini satwa dilindungi."

  Komentar ini menunjukkan respons yang lebih positif dan menggunakan bahasa yang tidak *toxic* ("jaga komentar kalian"). Meskipun menggunakan bahasa lelucon ("ini satwa dilindungi"), komentar ini memiliki makna yang menyiratkan bahwa perempuan yang merupakan objek dalam video tersebut merupakan hewan atau satwa. Komentar ini bisa diartikan sebagai upaya untuk memperingatkan pengguna lain untuk tidak memberikan komentar negatif namun dalam kalimat tersebut tetap memiliki unsur penghinaan di dalamnya yang ditujukan kepada objek yang ada dalam video.
- "Alis kayak Jembut anj"

  Komentar ini terdapat kemiripan dengan komentar pertama yaitu menggunakan bahasa kasar dan merendahkan atau penghinaan. Komentar ini membandingkan tampilan alis dengan kata yang kasar ("Jembut" dan "anj;anjing"), menciptakan pernyataan yang tidak sopan dan mungkin meremehkan penampilan objek wanita dalam video.
- "abis makan jabang bayik"

  Komentar ini mungkin mengandung unsur humor dalam bentuk lelucon yang lebih halus. Meskipun tidak sepenuhnya *toxic*, ada

kemungkinan bahwa komentar ini mengandung referensi yang tidak sepenuhnya dikenali oleh semua penonton, yang dapat mengganggu pemahaman komentar ini. Dimana komentar ini merujuk atau memiliki makna "habis makan bayi." Dengan konotasi yang dihubungkan dengan "makan bayi," komentar ini bisa saja mencoba menciptakan reaksi yang bisa diartikan sebagai bentuk humor gelap.

- "awkarin❎ awkambing✅"

  Komentar ini menggunakan perbandingan antara "awkarin" dan "awkambing" untuk mengekspresikan preferensi atau opini terhadap keduanya. Komentar ini bisa dianggap sebagai komentar yang mengejek atau membandingkan tokoh tertentu. Komentar ini menggunakan perbandingan antara "Awkarin" sebagai seorang artis dan "Awkambing" sebagai representasi hewan kambing. Penggunaan emoji "❎" dan "✅" menambahkan dimensi visual pada komentar, yang mengindikasikan bahwa objek wanita dalam video tersebut tidak cocok (❎) sebagai "awkarin" (artis), tetapi cocok (✅) sebagai "awkambing" (kambing).

Selain penjelasan yang telah dipaparkan diatas berikut ini merupakan beberapa *data* komentar yang memiliki ragam atau variasi yang berebeda baik *toxic* maupun *non-toxic* yang diambil apada kolom komentar instagarm yang sesuai dengan karektristik *user* atau pengguna instagam yang dapat dilihat pada tabel 4.1 berikut:

Tabel 4.1 Contoh Komentar

Komentar *toxic*	**Komentar *non-toxic***
jembut kok di taro d situ mbak..?	Ada yg knal kk nya gk ???pen tau kk ny gimana skrng

Komentar *toxic*	**Komentar *non-toxic***
Jwmbut nya pindah ke atas	loh? ini pacar u yg kemarin u ceritain kan ke i?
kesannya rada angker gitu yak	bayi baru lahir
Alis ❌ jembut✅	masih utamakan salam🔥🖐
ALIS LO KRITING □	Setidaknya Dia Mengucapkan Salam Terlebih Dahulu 🤐
Mulut nya kaya tai kring😷😷	Sebaiknya kita tidak boleh mentertawakan orang yg punya kekurangan
Bibirnya gosong😷😷😷🤦‍♀️	Tapi bagaimana pun ini ciptaan Tuhan 😟
perempuan berbulu	Soundnya lebih ngarah ke sabar aja sih
lagi sange ya kak?	pasti dia capek bangett,kasiann😟
serem juga ya liatnya	Sepertinya paket internetku habis

Berdasarkan tabel diatas dan penjelasan-penjelsan sebelumnya setalah mempertimbangkan karakteristik dari setiap *posting*an yang telah di*publish* maka *data- data* tersebut siap di lakukan scraping yang hasilnya dapat dilihat pada gambar (Gambar scraping) yang kemudian akan diproses untuk proses selanjutnya yaitu *cleaning data*

4.1.2 Hasil *Cleaning Data*

Data Cleaning dilakukan dengan tujuan membersihkan dan mempersiapkan *data* teks untuk analisis atau pemrosesan selanjutnya.

Berikut merupakan contoh kalimat yang digunakan sebagai ilustrasi dari tahap *cleaning data* “Teks awal: "Hello @john! How are you? ☺"” kemudian setalah tahap *cleaning data* maka susunan kalimat dalam teks tersebut dapat dilihat pada tabel 4.2 berikut

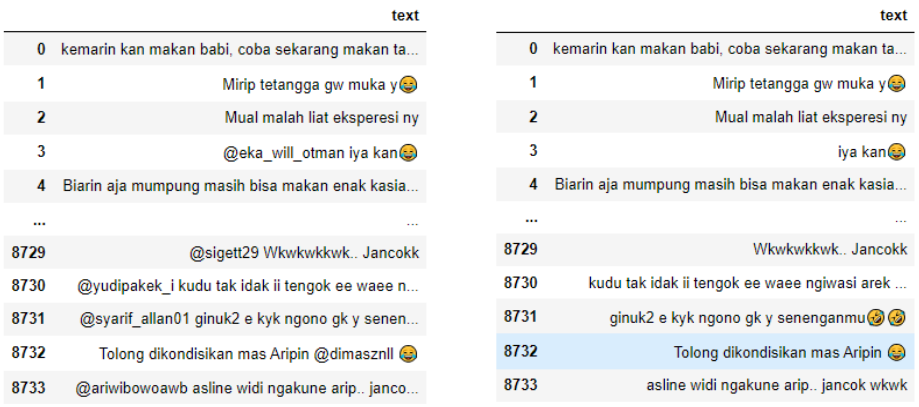
Tabel 4. 2 Contoh Hasil *Cleaning*

Remove Username	"Hello! How are you? ☺"
Remove Emoticon	"Hello! How are you?"
Case Folding	"hello! how are you?"
Remove Special Character	"hello how are you"
Remove Number	"hello how are you"
***Remove* Punctuation**	"hello how are you"
Remove Single Character	"hello how are you"
Tokenizing	["hello", "how", "are", "you"]
Stopword Removal	["hello"]
Stemming	["hello"]

Berdasrakan ilustrasi pada tabel 4.2 diatas maka dapat dijelaskan sebagai berikut:

1. *Remove Username*

   Langkah ini bertujuan untuk menghapus *user*name atau tag pengguna dalam teks, seperti "@john". Biasanya, *username* tidak memberikan informasi yang signifikan dalam analisis teks, sehingga dapat dihapus. Penerapan *Remove Username* ini dapat dilihat pada gambar berikut:



	text
0	kemarin kan makan babi, coba sekarang makan ta...
1	Mirip tetangga gw muka y😂
2	Mual malah liat eksperesi ny
3	@eka_will_otman iya kan😂
4	Biarin aja mumpung masih bisa makan enak kasia...
...	...
8729	@sigett29 Wkwkwkkwk.. Jancokk
8730	@yudipakek_i kudu tak idak ii tengok ee waee n...
8731	@syarif_allan01 ginuk2 e kyk ngono gk y senen...
8732	Tolong dikondisikan mas Aripin @dimasznll 😂
8733	@ariwibowoawb asline widi ngakune arip.. janco...

	text
0	kemarin kan makan babi, coba sekarang makan ta...
1	Mirip tetangga gw muka y😂
2	Mual malah liat eksperesi ny
3	iya kan😂
4	Biarin aja mumpung masih bisa makan enak kasia...
...	...
8729	Wkwkwkkwk.. Jancokk
8730	kudu tak idak ii tengok ee waee ngiwasi arek ...
8731	ginuk2 e kyk ngono gk y senenganmu🧐🧐
8732	Tolong dikondisikan mas Aripin 😂
8733	asline widi ngakune arip.. jancok wkwk

Gambar 4. 4 *Remove Username*

Gambar pertama (sebelumnya) merupakan gambar *dataset* sebelum dilalukan proses *cleaning data*, kemudian gamabr setelahnya merupakan gambar dari *cleaning data* dengan menggunakan teknik *remove user*name. Contoh ilustarsi hasil dari penerapan teknik *remove user*name ini dapat dilihat pada tabel 4.3 berikut:

Tabel 4. 3 *Remove Username*

Teks Awal	***Remove Username***
@codotsky ngakak aja liatnya	ngakak aja liatnya
@linamukherjee_ skippp baperan	skippp baperan
@petradnr mesti langsung viral	mesti langsung viral
@tenkuts @achmd.andik @abdi.pramono langsung di check out guys	langsung di check out guys
@rivaldwesley gk jelas njing	gk jelas njing

Berdasarkan tabel 4.3 diatas dapat dlihat bawa teks awal yang belum di lakukan *remove user*name memiliki nama pengguna yang diawali dengan symbol @ yang diikuti dengan nama yang berebeda-beda dari setiap *user* kemudian, diterapkanya teknik *remove user*name yang bertujuan untuk menghapus *user*name yang ada dalam *dataset* tersebut maka, hasilnya adalah seluruh *user*name yang ada di dalam *dataset* telah di hapus seperti pada tabel di atas di kolom "*Remove Username*"

2. *Remove Emoticon*

   Emoticon adalah simbol grafis yang digunakan untuk menyampaikan emosi atau ekspresi dalam teks. Langkah ini menghapus *emoticon* dari teks, seperti wajah senyum ":)" atau ikon hati "<3" dll. Penerapan *Remove Emoticon* ini dapat dilihat pada gambar 4.5 berikut:

	text
0	kemarin kan makan babi, coba sekarang makan ta...
1	Mirip tetangga gw muka y
2	Mual malah liat eksperesi ny
3	iya kan
4	Biarin aja mumpung masih bisa makan enak kasia...
...	...
8729	Wkwkwkkwk.. Jancokk
8730	kudu tak idak ii tengok ee waee ngiwasi arek ...
8731	ginuk2 e kyk ngono gk y senenganmu
8732	Tolong dikondisikan mas Aripin
8733	asline widi ngakune arip.. jancok wkwk

Gambar 4. 5 *Remove Emoticon*

Berdasarkan gambar diatas dapat dilihat bahwa seluruh emoticon yang ada di dalam *dataset* telah di hapus. Tabel 4.4 berikut ini adalah ilustrasi dari teknik *Remove Emoticon* yang dilakukan seperti pada gambar diatas

Tabel 4. 4 *Remove Emoticon*

Teks Awal	*Remove Emoticon*
Yg mulet pasti kepelet😂	Yg mulet pasti kepelet

Teks Awal	*Remove Emoticon*
Mirip tetangga gw muka y😂	Mirip tetangga gw muka y
Maju terus Lina g ush drngarin org org sirik ,👐👐👐👐👐	Maju terus Lina g ush drngarin org org sirik ,
Astaga gagal fokus sma lobang hidungnyaa🔥 gede amat uda kek penyedot debu😂😍	Astaga gagal fokus sma lobang hidungnyaa gede amat uda kek penyedot debu
Perut dah buncit tuh pak..kebanyakan makan ya...😊	Perut dah buncit tuh pak..kebanyakan makan ya...

Tabel 4.4 diatas pada kolom ‘Teks Awal’ merupakan teks sebelum di terapkannya *remove* emoticon dimana di dalam *dataset* tersebut masih terdapat berbagai emoticon yang digunakan, yang kemudian setalah diterapkanya teknik *remove* emoticon maka seluruh emoticon yang ada didalam *dataset* akan di hapus seperti pada gambar 4.5 di atas pada kolom ‘*Remove Emoticon*’

3. *Case Folding*

   Langkah ini melibatkan mengubah semua huruf dalam teks menjadi huruf kecil atau huruf kapital. Hal ini dilakukan untuk menghindari perbedaan yang tidak perlu dalam analisis teks, misalnya antara "Hello" dan "hello". Penerapan *Case Folding* ini dapat dilihat pada gambar 4.6 berikut:

	text
0	kemarin kan makan babi, coba sekarang makan ta...
1	mirip tetangga gw muka y
2	mual malah liat eksperesi ny
3	iya kan
4	biarin aja mumpung masih bisa makan enak kasia...
...	...
8729	wkwkwkkwk.. jancokk
8730	kudu tak idak ii tengok ee waee ngiwasi arek ...
8731	ginuk2 e kyk ngono gk y senenganmu
8732	tolong dikondisikan mas aripin
8733	asline widi ngakune arip.. jancok wkwk

Gambar 4. 6 *Case Folding*

Beradasarkan gambar diatas dapat dilihat bahwa seluruh teks yang ada dalam *dataset* terdiri dari susunan huruf kecil untuk setiap kata yang ada dalam dokumen *dataset*. Sebagai ilustrasi dari teknik *case fold*ing yang dilakukan ini dapat di lihat pada tabel 4.5 berikut:

Tabel 4. 5 *Case Folding*

Teks Awal	*Case Folding*
Cara makan nya begitu yah	cara makan nya begitu yah
Enak bngt makan ny kak lina	enak bngt makan ny kak lina
Subahannallah	subahannallah
Katanya punya asem lambung	katanya punya asem lambung
BUSET BELOM DI PENJARA PENJARA NIH NENE NENE	buset belom di penjara penjara nih nene nene

Berdasarkan tabel 4.5 diatas dimana pada kolom ‘Teks Awal’ menunjukan bahwa teks dalam dokumen *dataset* masih memiliki kombinasi susuan huruf *capital* dan huruf kecil yang kemudian setelah di terapkannya teknik *case fold*ing maka hasil dari penerapan teknik ini

adalah semua huruf yang ada dalam dokumen *dataset* diubah menjadi huruf kecil/lower*case* seperti pada tabel 4.5 diatas pada kolom '*Case Folding*'

4. *Remove Special Character*

   Langkah ini melibatkan penghapusan karakter khusus yang tidak diperlukan dalam teks, seperti tanda baca khusus, simbol, atau karakter *non*-alfanumerik. Penerapan *Remove Special Character* ini dapat dilihat pada gambar 4.7 berikut:

	text
0	kemarin kan makan babi coba sekarang makan tai...
1	mirip tetangga gw muka y
2	mual malah liat eksperesi ny
3	iya kan
4	biarin aja mumpung masih bisa makan enak kasia...
...	...
8729	wkwkwkkwk jancokk
8730	kudu tak idak ii tengok ee waee ngiwasi arek ...
8731	ginuk2 e kyk ngono gk y senenganmu
8732	tolong dikondisikan mas aripin
8733	asline widi ngakune arip jancok wkwk

Gambar 4. 7 *Remove Special Character*

Dalam kumpulan dokumen *dataset* banyak di temukan dokumen yang mengandung unsur karakter-karakter tertentu atau special *character* yaitu susuan karakter *non*-alfanumerik atau biasa di sebut dengan tanda baca (*Punctuation*) yaitu penghapusan tanda baca seperti tanda titik, koma, tanda tanya, dan lainnya. Ilustrasi yang dapat diberikan dalam penerapan teknik ini dapat di lihat pada tabel 4.6 berikut:

Tabel 4. 6 *Remove Special Character*

Teks Awal	*Remove Special Character*
ini beli dmna ka lina??	ini beli dmna ka lina
ngak di campur ba..	ngak di campur ba

Teks Awal	*Remove Special Character*
#biadab!	biadab
#copotkadiskeslampung	copotkadiskeslampung
"ibu tips biar jambul tinggi kaya gimana bu"	ibu tips biar jambul tinggi kaya gimana bu
di lampung 20% stor proyek..!!□ fakta	di lampung 20 stor proyek fakta

Berdasarkan tabel 4.6 diatas dapat di lihat pada kolom ‘Teks Awal’ dokumen *dataset* masih memiliki karakter-karakter *non*-alfanumerik seperti tanda tanya, tanda seru, tanda titik dan lainya. Karakter-karakter tersebut nantinya akan dihapus atau dihilangkan di seluruh dokumen *dataset*. Dapat dilihat pada kolom ‘*Remove Special Character*’ susuan dokumen *dataset* sudah tidak memiliki karakter *non*-alfanumerik yang menandakan bahwa penerapan teknik *remove* special *character* sudah berhasil diterapkan dalam dokumen *dataset*.

5. *Remove Number*

   Langkah ini bertujuan untuk menghapus angka atau digit dalam teks. Angka sering kali tidak memberikan kontribusi signifikan dalam analisis teks, kecuali jika konteksnya memang relevan. Penerapan *Remove Number* ini dapat dilihat pada gambar 4.8 berikut:

	text
0	kemarin kan makan babi coba sekarang makan tai...
1	mirip tetangga gw muka y
2	mual malah liat eksperesi ny
3	iya kan
4	biarin aja mumpung masih bisa makan enak kasia...
...	...
8729	wkwkwkkwk jancokk
8730	kudu tak idak ii tengok ee waee ngiwasi arek ...
8731	ginuk e kyk ngono gk y senenganmu
8732	tolong dikondisikan mas aripin
8733	asline widi ngakune arip jancok wkwk

Gambar 4. 8 *Remove Number*

Gambar diatas menunjukan kumpulan dokumen *dataset* yang di dalamnya tidak terdapat susuan angka karena sudah dihialngkan/dihapus dengan menggunakan teknik *remove number*, untuk ilustrasi dari penerapan teknik *remove number* ini dapat dilihat pada tabel 4.7 berikut:

Tabel 4. 7 *Remove Number*

Teks Awal	*Remove Number*
di lampung 20 stor proyek fakta	di lampung stor proyek fakta
prabowo dan mahfud md presiden 2024	prabowo dan mahfud md presiden
ginuk2 e kyk ngono gk y senenganmu	ginuk e kyk ngono gk y senenganmu
slide 4 dipikir rk mau videonya kesebar elahhhhhhhhhh	slide dipikir rk mau videonya kesebar elahhhhhhhhhh
100k aja bang free kandang	k aja bang free kandang

Berdasarkan tabel 4.7 diatas pada kolom ‘Teks Awal’ dokumen *dataset* masih memiliki susunan angka di dalamnya kemudain pada kolom ‘*Remove Number*’ merupakan hasil dari diterapkannya teknik *remove number* yang menghapus seluruh angka yang terdapat dalam dokumen *dataset*.

6. *Remove Single Character*

   Langkah ini melibatkan penghapusan karakter tunggal dalam teks. Karakter tunggal sering kali tidak memberikan makna yang signifikan dalam analisis teks, kecuali jika konteksnya memang relevan. Penerapan *Remove Single Character* ini dapat dilihat pada gambar 4.9 berikut:

	text
0	kemarin kan makan babi coba sekarang makan tai...
1	mirip tetangga gw muka
2	mual malah liat eksperesi ny
3	iya kan
4	biarin aja mumpung masih bisa makan enak kasia...
...	...
8729	wkwkwkkwk jancokk
8730	kudu tak idak ii tengok ee waee ngiwasi arek ...
8731	ginuk kyk ngono gk senenganmu
8732	tolong dikondisikan mas aripin
8733	asline widi ngakune arip jancok wkwk

Gambar 4. 9 *Remove Single Character*

Penerapan teknik *remove single* charcter ini seperti pada gambar diatas adalah dengan menghapus karakter tunggal yang ada dalam dokumen datset seperti a, i, u, e, o dan seterusnya. Sebagai ilustrasinya dapat di lihat pada tabel 4.8 berikut:

Tabel 4. 8 *Remove Single Character*

Teks Awal	*Remove Single Character*
mirip tetangga gw muka y	mirip tetangga gw muka

Teks Awal	*Remove Single Character*
ginuk e kyk ngono gk y senenganmu	Ginuk kyk ngono gk senenganmu
i love tv one i love dewi persikk	love tv one love dewi persikk
p minimal mandi	minimal mandi
a tolol	tolol

Berdasarkan ilustrasi pada tabel 4.8 diatas dapat dijelaskan bahwa dalam kolom 'Teks Awal' kumpulan dokumen *dataset* memiliki karakter tunggal seperti y, e, i, dan p kemudian, setelah diterapkannya teknik *remove single character* maka maka karakter-karakter *single* tersebut dihilangkan dari *dataset* sperti pada kolom '*Remove Single Character*' pada tabel diatas.

7. *Tokenizing*

Langkah ini melibatkan pemisahan teks menjadi unit-unit yang lebih kecil, yang disebut token. Token dapat berupa kata-kata individual dalam teks. Penerapan *Tokenizing* ini dapat dilihat pada gambar 4.10 berikut:

	text
0	[kemarin, kan, makan, babi, coba, sekarang, ma...
1	[mirip, tetangga, gw, muka]
2	[mual, malah, liat, eksperesi, ny]
3	[iya, kan]
4	[biarin, aja, mumpung, masih, bisa, makan, ena...
...	...
8442	[wkwkwkkwk, jancokk]
8443	[kudu, tak, idak, ii, tengok, ee, waee, ngiwas...
8444	[ginuk, kyk, ngono, gk, senenganmu]
8445	[tolong, dikondisikan, mas, aripin]
8446	[asline, widi, ngakune, arip, jancok, wkwk]

Gambar 4. 10 *Tokenizing*

Gambar 4.10 diatas menunjukan bahwa dokumen *dataset* telah di pisahkan menjadi token-token kata secara individual. Ilustrasi dari teknik tokenizing ini dapat dilihat pada tabel 4.9 berikut:

Tabel 4. 9 *Tokenizing*

Teks Awal	*Tokenizing*
kemarin kan makan babi coba sekarang makan tai mbak	[kemarin, kan, makan, babi, coba, sekarang, makan, tai, mbak]
mirip tetangga gw muka	[mirip, tetangga, gw, muka]
mual malah liat eksperesi ny	[mual, malah, liat, eksperesi, ny]
iya kan	[iya, kan]
bukan di steril kali mba di fermentasi	[bukan, di, steril, kali, mba, di, fermentasi]

Berdasarkan tabel 4.9 diatas kolom'Teks Awal' menunjukan dokumen *dataset* yang belum di tokenisasi dimana setiap dokumen masih berupa susunan kalimat kata. Kemudian setelah dilakukan tokenizing seperti pada kolom '*Tokenizing*' maka dokumen *dataset* yang aalnya berupa susunan kalimat kata menjadi susunan kata per kata atau telah diubah menjadi token untuk setiap kata yang ada dalam dokumen.

8. *Stopword Removal*

   Stopword adalah kata-kata umum yang sering muncul dalam teks tetapi tidak memberikan makna yang signifikan. Langkah ini menghapus *stopword* dari teks, seperti kata-kata penghubung "dan", "atau", "di", dll. Penerapan *Stopword Removal* ini dapat dilihat pada gambar 4.11 berikut:

	text
0	[kemarin, makan, babi, coba, makan, tai, mbak]
1	[tetangga, gw, muka]
2	[mual, liat, eksperesi, ny]
3	[iya]
4	[biarin, aja, mumpung, makan, enak, kasian, ka...
...	...
8729	[wkwkwkkwk, jancokk]
8730	[kudu, idak, ii, tengok, ee, waee, ngiwasi, ar...
8731	[ginuk, kyk, ngono, gk, senenganmu]
8732	[tolong, dikondisikan, mas, aripin]
8733	[asline, widi, ngakune, arip, jancok, wkwk]

Gambar 4. 11 *Stopword Removal*

Berdasarkan gambar 4.11 di atas maka untuk memberikan ilustrasi dari hasil penerapan teknik *stopword removal* dapat dilihat pada tabel 4.10 berikut:

Tabel 4. 10 *Stopword Removal*

Teks Awal	*Stopword Removal*
kemarin kan makan babi coba sekarang makan tai mbak	kemarin makan coba babi makan tai mbak
mirip tetangga gw muka	tetangga gw muka
mual malah liat eksperesi ny	mual liat eksperesi ny
iya kan	iya

Berdasarkan tabel 4.10 diatas kolom ‘Teks Awal’ menunjukan teks yang ada dalam dokumen *dataset* seblum diimplementasikannya teknik *stopword removal* dimana pada teks awal terdapat beberapa *stoplist* kata yaitu (sekarang, kan mirip, malah dan seterusnya) setiap kata yang

termasuk dalam stoplist kata maka akan dihilangkan atau di hapus dalam *dataset* seperti pada kolom ’*Stopword Removal*’

9. *Stemming*

   Stemming adalah proses mengubah kata-kata dalam teks menjadi bentuk dasarnya atau kata dasar. Langkah ini bertujuan untuk mengurangi variasi kata yang memiliki akar yang sama. Contohnya, seperti "berlari dapat diubah menjadi kata dasar "lari". Penerapan *Stemming* ini dapat dilihat pada gambar 4.12 berikut:

	text
0	[kemarin, makan, babi, coba, makan, tai, mbak]
1	[tetangga, gw, muka]
2	[mual, liat, eksperesi, ny]
3	[]
4	[biarin, aja, mumpung, makan, enak, kasian, ka...
...	...
8442	[wkwkwkkwk, jancokk]
8443	[kudu, idak, ii, tengok, ee, waee, ngiwasi, ar...
8444	[ginuk, kyk, ngono, gk, senenganmu]
8445	[tolong, dikondisikan, ma, aripin]
8446	[aslin, widi, ngakun, arip, jancok, wkwk]

Gambar 4. 12 *Stemming*

Gambar 4.12 diatas merupakan hasil dari penerapan teknik *stemming* dengan mengubah bentuk kata menjadi kata dasar serta penambahan additional kata yang akan di hapus contohnya sperti kata ‘iya’ yang terdapat dalam *dataset* di proses sebelumnya yaitu *stopword* filtering yang kemudian pada tahap *stemming* ini akan di hapus kata ‘iya’ tersebut. Contoh ilustrasi dari penerapan teknik *stemming* ini dapat dilihat pada tabel 4.11 berikut:

Tabel 4. 11 *Stemming*

Teks Awal	*Stemming*
kemarin makan coba babi makan tai mbak	kemarin makan coba babi makan tai mbak
tetangga gw muka	tetangga gw muka
mual liat eksperesi ny	mual liat eksperesi ny
iya	[]

Berdasarkan tabel 4.11 diatas antara kedua kolom'Teks Awal' dan '*Stemming*' tidak memiliki perbedaan yang signifikan dari segi susunan kalimat dalam dokumen *dataset* dikarenakan semua kata yang digunakan dalam dokumen- dokumen pada tabel diatas sudah dalam bentuk dasar sehingga tidak mengalami perubahan namun, terdapat perebdaan pada kata 'iya' dimaan dalam kolom 'Teks Awal' kata 'iya' masih termasuk dalam dokumen dan setelah implementasi teknik *stemming* kata 'iya' sudah di hapus dari seluruh dokumen yang ada dalam *dataset*.

10. *Null Values*

    Null values merupakan teknik yang di terapkan untuk menghapus dokumen dalam datset yang tidak memiliki nilai atau kosng, dokumen kosong ini dapat secara signifikan memepengaruhi kualitas dari *dataset* yang dimiliki, oleh karena itu penerapan teknik *remove null value* ini penting untuk dilakukan. Hasil dari implementasi *remove null value* ini dapat dilihat pada gambar 4.13 berikut:

	text
0	kemarin makan babi coba makan tai mbak
1	tetangga gw muka
2	mual liat eksperesi ny
3	biarin aja mumpung makan enak kasian kalo penj...
4	tetep enk pandang
...	...
7503	wkwkwkkwk jancokk
7504	kudu idak ii tengok ee waee ngiwasi arek ee ko...
7505	ginuk kyk ngono gk senenganmu
7506	tolong dikondisikan ma aripin
7507	aslin widi ngakun arip jancok wkwk

Gambar 4. 13 *Null Values*

Jika dilihat pada gambar sebelumnya yaitu gambar 4.10 terdapat dokumen *dataset* yang tidak memiliki nilai atau *null*, lebih tepatnya pada baris ke tiga dokumen yang merupakan hasil dari penghapusan/penghilangan kata ‘iya’, dikarenakan dokemen tiga tersebut hanya memiliki 1 kata yaitu ‘iya’ yang kemudian pada tahap *stemming* kata ‘iya’ tersebut dihapus maka, secara otomatis dokumen tiga menjadi kosong atu *null* atau dalam artian bahwa dokumen baris ketiga tidak memiliki nilai. Dokuemn-dokumen yang tidak memiliki nilai tersebut dapat mempengaruhi performa dari model yang akan dibuat. Oleh karena itu seluruh dokumen yang ada dalam *dataset* yang kosong atau tidak memiliki nilai akan dihapus seperti pada gambar diatas.

4.1.3 Hasil *Labeling Data*

Hasil dari *labeling data* dapat dilihat pada penjelasan berikut:

1. Membuat Kamus Kata *Toxic* dan *Non-Toxic*

   Tahapan pertama yang dilakukan dalam melakukan *labeling data* adalah dengan membuat kamus kata *toxic* dan non *toxic* yang nantinya digunakan untuk membedakan manakah kalimat yang

menggunakan kalimat *toxic* atau tidak. Pembuatan kamus ini dilakukan berdasrkan trend atau pola yang ada dalam *dataset* yang digunakan seperti pada tabel 4.12 berikut:

Tabel 4. 12 Kamus *Toxic* dan *Non-Toxic*

Kamus Non-*Toxic*	**Kamus *Toxic***
langsung	aborsi
pagi	bodoh
teriak	cecurut
tolong	dungu
marah	enek

Penjelasan lebih lanjut terkait hasil dari pembuatan kamus kata *toxic* dan *non-toxic* diatas adalah setelah *data* siap, konten diurai untuk mengidentifikasi kata-kata dan frasa yang digunakan dalam setiap *postingan*, video, atau *reel*. Konteks yang lebih luas juga diperhitungkan, termasuk keterangan, *hashtag*, dan komentar. Kata-kata dan frasa di klasifikasikan menjadi dua kategori: *toxic* dan *non-toxic*. Proses ini dilakukan berdasarkan konteks bahasa dan konteks sosial untuk membedakan antara penggunaan kata dalam situasi yang merendahkan atau tidak merendahkan.Hasil klasifikasi kata-kata atau frasa, sebuah kamus kata *toxic* dan *non-toxic* dibuat. Kamus ini berisi daftar kata-kata yang telah diidentifikasi dan dikategorikan sesuai dengan sifat toksisitasnya.

2. Menghitung Jumlah kemunculan Kata

Setelah membuat kamus untuk kata *toxic* dan non *toxic* langkah selanjutnya adalah menghitung kemuculan kata berdasarkan kamus yang dibuat seperti pada penjelasan berikut:

Contoh dokumen *text*:

“lina lutfiawati banyuwangi nik abdul mukhit banyuwangi nik sismarwati banyuwangi nik lpkirjt januari tindak pidana penipuan media elektronik ditangani sat reskrim re metro jakarta timur pelapor siti nuraeni korban siti nuraeni saksi heru ira satria suryani terlapor lina lutfiawati kronolog pelapor diajak terlapor nonton konser new delhi india biaya rp pelapor bawa mumbai pelapor dibelikan tiket indonesia sedsngksn perjanjian terlapor biaya tiketnya pulang pergi kejadian pelapor dirugikan lpvyanspkt pmj may tindak pidana pencemaran nama ditangani sat reskrim re metro jakarta pusat pelapor bambang sri pujo sh korban ajay kumar nahar saksi miranti terlapor lina mukherjeelina lutfiawati kronolog pelapor kuasa korban menerangkan korban *instagram* akun linamukherj menyebarkan mengupload foto korban korban botak berkalikali korban dicemarkan nama baiknya terlapor menyebarkan nomor telepon korban kejadian korban dicemarkan nama baiknya pelapor spkt polda metro jaya laporan pengaduan penyelidikan sesuai hukum berlaku lpbvispktpolda sumut juni tindak pidana penipuan ditangani ditkrimum sumut pelapor fauzi ramadhan nasut korban fauzi ramadhan nasut saksi mariam angriama terlapor lina mukherjeelina lutfiawati kronolog”

Berdasarkan contoh dokumen *text* diatas maka hasil yang didapatkan dari menghitung kemunculan jumalh kata ada sebagai berikut:

Jumlah kata	= 169 kata
Jumlah *Toxic*	= 69 kata
Jumlah *Non-toxic*	= 162 kata

Berdasarkan penjelasan diatas dalam ‘Contoh dokumen *Text*’ memiliki jumlah kata sebanyak 169 kata yang miliki unsur kata *toxic* sebnayak 69 kata dan sebanyak 162 kata yang tidak mengandung unsur *toxic*. Perhitungan ini didaptakkan berdasarkan kamus *toxic* dan kamus *non-toxic* yang dibuat sebelumnya, sehingga jika kata dalam ‘contoh dokumen *text*’ muncul pada kamus *toxic* maka kata tersebut akan terhitung sebagai kata *toxic* begitupun sebaliknya jika, kata dalam ‘contoh dokumen *text*’ muncul pada kamus *non-toxic* maka kata tersebut akan terhitung sebagai kata *non-toxic*. Apabila kata dalam ‘contoh dokumen *text*’ muncul pada kamus *toxic* dan *non-toxic* maka kata tersebut akan masuk atau dihitung sebagai kata *toxic* dan *non-toxic* dimaan yang menjadi pembeda dari kedua hal tersebut adalah keseluruhan jumlah kata yang dihitung dalam satu dokumen.

3. Menghitung Persentase

   Setelah menetahui jumlah kemunculan kata *toxic* dan non *toxic* dalam dokumem maka kemudian akan dihitung persentase dari kemunculan kata tersebut pada masing-masing dokumen seperti pada ilustrasi berikut.

   Berdasarkan penjelasan sebelumnya pada tahap menghitung kemunculan jumlah kata maka setelah mengetahui jumlah kata yang muncul dalam ‘contoh dokumen *text*’ akan menjadi kunci utama dalam menghitung persentase dokumen *text* apakah masuk dalam kelas *toxic* atau *non-toxic*. Dalam ‘contoh dokumen *text*’ sebelumnya diperoleh *data* sebagai berikut:

   Jumlah kata = 169 kata

   Jumlah *Toxic* = 69 kata

   Jumlah *Non-toxic* = 162 kata

   Berdasarkan *data* diatas maka untuk mengetahui persentase kelas dari ‘contoh dokumen *text*’ adalah sebagai berikut:

Persentase *toxic* (Jumlah *Toxic*/Jumlah Kata):

$$\frac{69}{169} = 0.42592592592592593$$

Persentase *non-toxic* (Jumlah *non-toxic*/Jumlah Kata)

$$\frac{162}{169} = 0.958579882$$

Berdasarkan penjelasan diatas dapat dilihat bahwa ‘contoh dokumen *text*’ memiliki persentase *non-toxic* yang lebih besar dibandingkan dengan persentase *toxic* dimana, persentase *non-toxic* dalam ‘contoh dokumen teks’ sebesar 0.9585% sedangkan persentase *toxic* hanya

sebesar 0.4259%. setalah mengetahui persentase untuk kalimat *toxic* dan *non-toxic* dalam ‘contoh dokumen *text*’ maka langkah terkahir yang akan dilakukan dalam proses *labeling data* adalah dengan memberikan label pada ‘contoh dokumen *text*’

4. Memberikan Label

   Dokumen akan diberikan label jika, persentase berdasrkan kemunculan kata *toxic* lebih besar dari kata *non-toxic* maka, dokumen *dataset* tersebut akan diberikan label *toxic* dan sebaliknya. Setelah menghitung dan mengetahui persentase dari ‘contoh dokumen *text*’ maka langkah yang dilakukan selanjutnya adalah dengan memberikan label sesuai dengan persentase kata pada dokumen. Berdasarkan pemaparan yang telah dijelaskan seblumnya maka dalam ‘contoh dokumen *text*’ didapatkan *data* sebagai berikut:

Jumlah kata	= 169 kata
Jumlah *Toxic*	= 69 kata
Jumlah *Non-toxic*	= 162 kata
Persentase *toxic*	= 0.42592592592592593
Persentase *non-toxic*	= 0.958579882

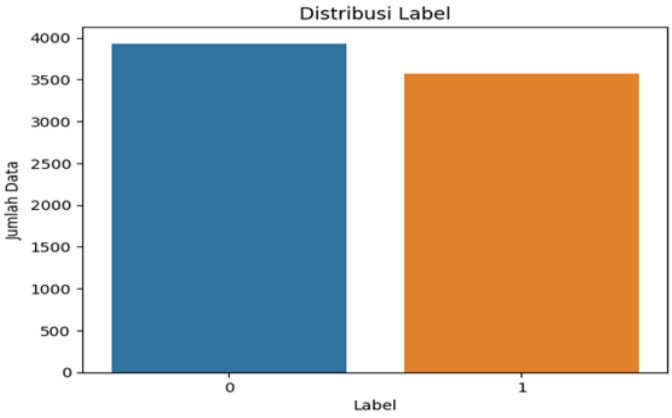
Beradasarkan *data* diatas maka didapatkan kesimpulan bahwa ‘contoh dokumen *text*’ akan diberikan label kelas ‘*non-toxic*’ karena persentase *non-toxic* lebih besar dari persentase *toxic*.

Proses finalisai *labeling data* ini adalah dengan mengubah label kela yang sebelumnya berupa tipe *data* string atau teks yaitu ‘*toxic* dan *non-toxic*’ akan diubah menjadi tipe *data* integer atau angka yaitu ‘0 dan 1’ dimana label kelas *non-toxic* diubah menjadi 0 dan label kelas *toxic* diubah menjadi 1 seperti pada gambar 4.14 berikut:

	text	label
0	kemarin makan babi coba makan tai mbak	1
1	tetangga gw muka	1
2	mual liat eksperesi ny	1
3	biarin aja mumpung makan enak kasian kalo penj...	1
4	tetep enk pandang	1
...	...	...
7503	wkwkwkkwk jancokk	1
7504	kudu idak ii tengok ee waee ngiwasi arek ee ko...	0
7505	ginuk kyk ngono gk senenganmu	0
7506	tolong dikondisikan ma aripin	0
7507	aslin widi ngakun arip jancok wkwk	1

Gambar 4. 14 *Labeling Data*

Berdasarkan gambar 4.14 diatas dapat dilihat bahwa semua dokumen yang ada dalam dokumen *dataset* telah di berikan label dalam bentuk angka atau array. Secara keseluruhan hasil dari *labeling data* ini memiliki distribusi *data* yang dapa dilihat pada gambar 4.15 berikut:



Gambar 4. 15 Distribusi Label

Berdasarkan gambar 5.15 diatas label kelas 0 memiliki jumlah *data* yang lebih banyak dibandingkan dengan label 1 dimana label 0 memiliki distribusi *data* sebanyak 3.932 *data* dan untuk label 1 memiliki *data* kelas sebanyak 3.576 *data*.

4.2 Hasil *Feature Extraction*

Tahap selanjutnya yang dilakukan adalah *feature extraction* yang bertujuan untuk memeriksa kerja fitur dengan model yang dibuat dan meningkatkan fitur untuk direpresentasikan dalam bentuk numeric *data* atau *data matrix*.

Sebelum diimplementasikannya *feature extraction*, hal pertama yang dilakukan terlebih dahulu adalah dengan membagi *dataset* menjadi *data training* dan *data testing* dengan rasio pembagian *data training* dan *data testing* adalah 80:20.

Setelah pembagian *data training* dan *data testing* telah diterapkan maka *feature* extrction siap diimplementasikan dengan mengimplementasikan pembobotan kata atau *TF-IDF* seperti berikut:

Untuk menghitung *TF-IDF* (*Term Frequency-Inverse Document Frequency*), Anda memerlukan dua tahap utama: menghitung *Term Frequency*

(*TF*) dan menghitung *Inverse Document Frequency* (*IDF*). Berikut adalah penjelasan dan perhitungan untuk setiap tahap:

1. Hitung *Term Frequency* (*TF*):

   Term Frequency (*TF*) mengukur seberapa sering suatu kata muncul dalam dokumen tertentu. Dalam hal ini perlu untuk menghitung jumlah kemunculan kata di dokumen tersebut, kemudian membaginya dengan total jumlah kata dalam dokumen tersebut.

   Contoh:

   Terdapat dokumen sebagai berikut:

   "maju lina ush drngarin org org sirik".

   Kata yang akan hitung adalah: "maju".

   Jumlah kemunculan kata "maju" = 1

   Total jumlah kata dalam dokumen = 6

   Maka, *Term Frequency* (*TF*) untuk kata "maju" dalam dokumen tersebut adalah:

   TF = Jumlah kemunculan kata / Total jumlah kata

   = 1 / 6

   = 0.1667

   Maka term frequency dari kata maju adalah 0.1667

2. Hitung *Inverse Document Frequency* (*IDF*):

   Inverse Document Frequency (*IDF*) mengukur seberapa penting suatu kata dalam kumpulan dokumen.

   Dalam hal ini perlu untuk menghitung jumlah dokumen dalam kumpulan dokumen yang mengandung kata tersebut, kemudian membagi jumlah total dokumen dengan hasil tersebut.

   Contoh:

   Terdapat kumpulan dokumen dengan total 5 dokumen. Kata yang akan dihitung adalah: "maju".

   Jumlah dokumen dalam kumpulan dokumen yang mengandung kata "maju" = 3

   Total jumlah dokumen dalam kumpulan dokumen = 5

Maka, *Inverse Document Frequency* (*IDF*) untuk kata "maju" dalam kumpulan dokumen tersebut adalah:

IDF = log(Total dokumen / Jumlah dokumen dengan term)
= log(5 / 3)
= 0.1761

3. Hitung *TF-IDF*:

   TF-IDF menggabungkan nilai *TF* dan *IDF* untuk mengukur pentingnya suatu kata dalam dokumen tertentu dalam konteks kumpulan dokumen yang lebih besar.

   Untuk mendapatkan nilai *TF-IDF* dilakuan dengan mengalikan nilai *TF* dengan nilai *IDF* yang didaptakan sebelumnya.

   Contoh:

   Dengan menggunakan nilai *TF* = 0.1667 (dari perhitungan sebelumnya) dan nilai *IDF* = 0.1761 (dari perhitungan sebelumnya), sehingga dapat menghitung *TF-IDF* untuk kata "maju" dalam dokumen tersebut.

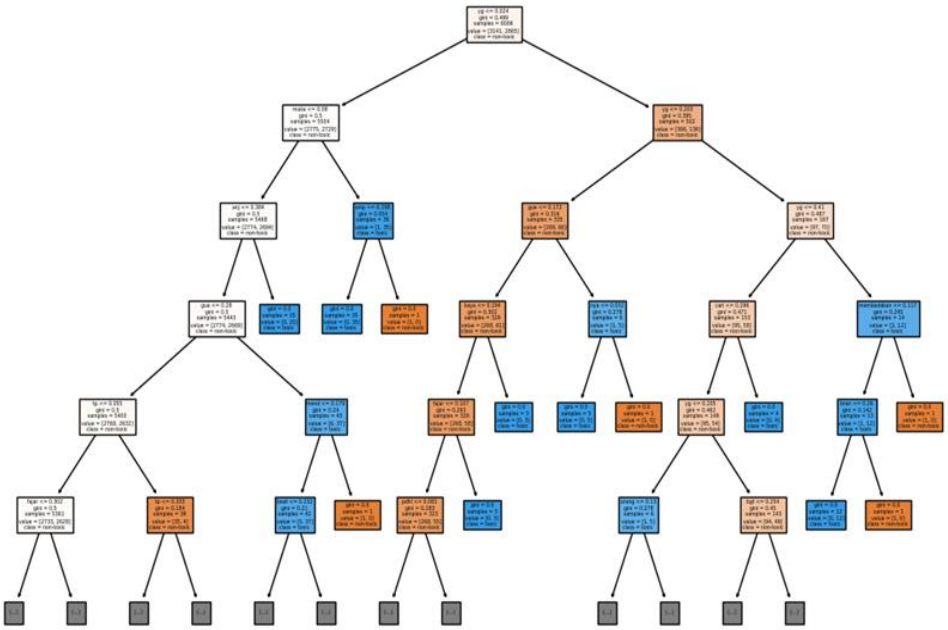
   TF-IDF = *TF* * *IDF*
   = 0.1667 * 0.1761
   = 0.0293

   Jadi, nilai *TF-IDF* untuk kata "maju" dalam dokumen tersebut adalah 0.0293.

   Dalam penerapan *TF-IDF* ini akan dilakukan dengan mengulang semua proses untuk setiap kata dalam dokumen dan kumpulan dokumen yang dimiliki untuk mendapatkan nilai *TF-IDF* untuk masing-masing kata.

4.3 Hasil *Model Training* dan *Testing Model*

Hasil dari proses *training* dan testing dapat dilihat pada gambar 4.16 berikut:



Gambar 4. 16 Pohon Keputusan *Decision Tree*

Gambar diatas merupakan hasil atau struktur pohon keputusan dari algoritma decision tree yang dihasilkan dari proses pengklasifikasian yang dilakukan. Namun untuk dapat mengilustrasikan gambar diatas dapat dikatakan terlalu kompleks sehingga peneliti memutuskan untuk menggunakan sample *data* untuk dapat membuat pohon keputusan pada penjelasan berikutnya

Dalam proses *training* dan testing untuk menghitung dan meengklasifikasikan dokumen menggunakan decision tree untuk klasifikasi teks *toxic* dan *non-toxic*, terdapat langkah-langkah yang perlu lakukan dengan ilustrasi berikut: Dalam ilustrasi ini, dengan menggunakan sample *data* dari dokumen *dataset* sebagai berikut:

"maju lina ush drngarin org org sirik" (label *non-toxic*)

"kemarin makan babi coba makan tai mbak" (label *toxic*)

"enak sikit wkwkwk" (label *non-toxic*)

"tetangga gw muka" (label *toxic*)

"trahe elek diapak apakno yo elek" (label *non-toxic*)

Untuk menghitung pohon keputusan dari sample dokumen diatas, perlu untuk memutuskan fitur mana yang akan digunakan sebagai *root node*. Dalam penelitian ini, akan menggunakan fitur kata-kata sebagai dasar pembagian. Berikut adalah langkah-langkah untuk menghitung pohon keputusan:

1. Hitung Jumlah Teks *Toxic* dan *Non-toxic*:
   - Terdapat 2 teks *toxic* dan 3 teks *non-toxic* dalam *data training*.
2. Hitung *Entropi Data Training*:
   - Proporsi teks *toxic*: 2/5 = 0.4
   - Proporsi teks *non-toxic*: 3/5 = 0.6
   - *Entropi* = -0.4 * log2(0.4) - 0.6 * log2(0.6) = 0.971
3. Hitung Information *Gain* (IG) untuk Setiap Fitur:
   - Untuk setiap kata dalam dokumen, hitung IG dengan menggunakan rumus IG = *Entropy* sebelum pemisahan - (*Weighted Entropy* setelah pemisahan oleh fitur)
     a) Fitur "maju":
        Jumlah teks *toxic* dengan fitur "maju": 0
        Jumlah teks *non-toxic* dengan fitur "maju": 1
        Entropi setelah pemisahan dengan fitur "maju": 0 (kelompok *non-toxic* homogen)
        IG = 0.971 - (1/5 * 0) = 0.971
     b) Fitur "lina":
        Jumlah teks *toxic* dengan fitur "lina": 0
        Jumlah teks *non-toxic* dengan fitur "lina": 1
        Entropi setelah pemisahan dengan fitur "lina": 0 (kelompok *non-toxic* homogen)
        IG = 0.971 - (1/5 * 0) = 0.971
     c) Fitur "ush":
        Jumlah teks *toxic* dengan fitur "ush": 0
        Jumlah teks *non-toxic* dengan fitur "ush": 1

Entropi setelah pemisahan dengan fitur "ush": 0 (kelompok *non-toxic* homogen)

IG = 0.971 - (1/5 * 0) = 0.971

d) Fitur "drngarin":

Jumlah teks *toxic* dengan fitur "drngarin": 0

Jumlah teks *non-toxic* dengan fitur "drngarin": 1

Entropi setelah pemisahan dengan fitur "drngarin": 0 (kelompok *non-toxic* homogen

IG = 0.971 - (1/5 * 0) = 0.971

e) Fitur "org":

Jumlah teks *toxic* dengan fitur "org": 0

Jumlah teks *non-toxic* dengan fitur "org": 2

Entropi setelah pemisahan dengan fitur "org": 0 (kelompok *non-toxic* homogen)

IG = 0.971 - (2/5 * 0) = 0.971

f) Fitur "sirik":

Jumlah teks *toxic* dengan fitur "sirik": 0

Jumlah teks *non-toxic* dengan fitur "sirik": 1

Entropi setelah pemisahan dengan fitur "sirik": 0 (kelompok *non-toxic* homogen)

IG = 0.971 - (1/5 * 0) = 0.971

g) Fitur "kemarin":

Jumlah teks *toxic* dengan fitur "kemarin": 1

Jumlah teks *non-toxic* dengan fitur "kemarin": 0

Entropi setelah pemisahan dengan fitur "kemarin": 0 (kelompok *toxic* homogen)

IG = 0.971 - (1/5 * 0) = 0.971

h) Fitur "makan":

Jumlah teks *toxic* dengan fitur "makan": 1

Jumlah teks *non-toxic* dengan fitur "makan": 0

Entropi setelah pemisahan dengan fitur "makan": 0 (kelompok *toxic* homogen)

IG = 0.971 - (1/5 * 0) = 0.971

i) Fitur "babi":

Jumlah teks *toxic* dengan fitur "babi": 1

Jumlah teks *non-toxic* dengan fitur "babi": 0

Entropi setelah pemisahan dengan fitur "babi": 0 (kelompok *toxic* homogen)

IG = 0.971 - (1/5 * 0) = 0.971

j) Fitur "coba":

Jumlah teks *toxic* dengan fitur "coba": 1

Jumlah teks *non-toxic* dengan fitur "coba": 0

Entropi setelah pemisahan dengan fitur "coba": 0 (kelompok *toxic* homogen)

IG = 0.971 - (1/5 * 0) = 0.971

k) Fitur "tai":

Jumlah teks *toxic* dengan fitur "tai": 1

Jumlah teks *non-toxic* dengan fitur "tai": 0

Entropi setelah pemisahan dengan fitur "tai": 0 (kelompok *toxic* homogen)

IG = 0.971 - (1/5 * 0) = 0.971

l) Fitur "mbak":

Jumlah teks *toxic* dengan fitur "mbak": 1

Jumlah teks *non-toxic* dengan fitur "mbak": 0

Entropi setelah pemisahan dengan fitur "mbak": 0 (kelompok *toxic* homogen)

IG = 0.971 - (1/5 * 0) = 0.971

m) Fitur "enak":

Jumlah teks *toxic* dengan fitur "enak": 0

Jumlah teks *non-toxic* dengan fitur "enak": 1

Entropi setelah pemisahan dengan fitur "enak": 0 (kelompok *non-toxic* homogen)

IG = 0.971 - (1/5 * 0) = 0.971

n) Fitur "sikit":

Jumlah teks *toxic* dengan fitur "sikit": 0

Jumlah teks *non-toxic* dengan fitur "sikit": 1

Entropi setelah pemisahan dengan fitur "sikit": 0 (kelompok *non-toxic* homogen)

IG = 0.971 - (1/5 * 0) = 0.971

o) Fitur "wkwkwk":

Jumlah teks *toxic* dengan fitur "wkwkwk": 0

Jumlah teks *non-toxic* dengan fitur "wkwkwk": 1

Entropi setelah pemisahan dengan fitur "wkwkwk": 0 (kelompok *non-toxic* homogen)

IG = 0.971 - (1/5 * 0) = 0.971

p) Fitur "tetangga":

Jumlah teks *toxic* dengan fitur "tetangga": 1

Jumlah teks *non-toxic* dengan fitur "tetangga": 0

Entropi setelah pemisahan dengan fitur "tetangga": 0 (kelompok *toxic* homogen)

IG = 0.971 - (1/5 * 0) = 0.971

q) Fitur "gw":

Jumlah teks *toxic* dengan fitur "gw": 1

Jumlah teks *non-toxic* dengan fitur "gw": 0

Entropi setelah pemisahan dengan fitur "gw": 0 (kelompok *toxic* homogen)

IG = 0.971 - (1/5 * 0) = 0.971

r) Fitur "muka":

Jumlah teks *toxic* dengan fitur "muka": 1

Jumlah teks *non-toxic* dengan fitur "muka": 0

Entropi setelah pemisahan dengan fitur "muka": 0 (kelompok *toxic* homogen)

IG = 0.971 - (1/5 * 0) = 0.971

s) Fitur "trahe":

Jumlah teks *toxic* dengan fitur "trahe": 0

Jumlah teks *non-toxic* dengan fitur "trahe": 1

Entropi setelah pemisahan dengan fitur "trahe": 0 (kelompok *non-toxic* homogen)

IG = 0.971 - (1/5 * 0) = 0.971

t) Fitur "elek":

Jumlah teks *toxic* dengan fitur "elek": 0

Jumlah teks *non-toxic* dengan fitur "elek": 2

Entropi setelah pemisahan dengan fitur "elek": 0 (kelompok *non-toxic* homogen)

IG = 0.971 - (2/5 * 0) = 0.971

u) Fitur "diapak":

Jumlah teks *toxic* dengan fitur "diapak": 0

Jumlah teks *non-toxic* dengan fitur "diapak": 1

Entropi setelah pemisahan dengan fitur "diapak": 0 (kelompok *non-toxic* homogen)

IG = 0.971 - (1/5 * 0) = 0.971

v) Fitur "apakno":

Jumlah teks *toxic* dengan fitur "apakno": 0

Jumlah teks *non-toxic* dengan fitur "apakno": 1

Entropi setelah pemisahan dengan fitur "apakno": 0 (kelompok *non-toxic* homogen)

IG = 0.971 - (1/5 * 0) = 0.971

w) Fitur "yo":

Jumlah teks *toxic* dengan fitur "yo": 0

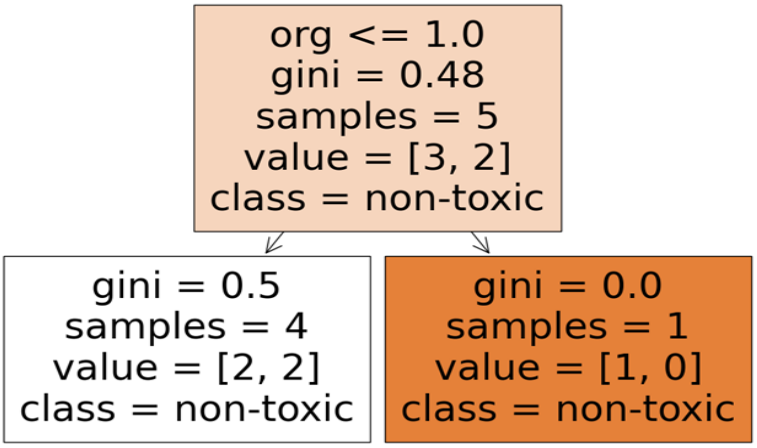
Jumlah teks *non-toxic* dengan fitur "yo": 1

Entropi setelah pemisahan dengan fitur "yo": 0 (kelompok *non-toxic* homogen)

IG = 0.971 - (1/5 * 0) = 0.971

4. Pilih *Root Node*
    - Dari hasil perhitungan IG, semua fitur memiliki IG yang sama yaitu 0.971.
    - Pilih salah satu fitur sebagai *root node*, misalnya "org".
5. Buat *Split Node*:
    - Buat *split node* dengan fitur "org".
6. Melanjutkan Pembagian:
    - Bagi *data training* menjadi dua kelompok berdasarkan nilai fitur "org":
    - Kelompok dengan nilai fitur "org": "maju lina ush drngarin org org sirik"
    - Kelompok tanpa nilai fitur "org": "kemarin makan babi coba makan tai mbak", "enak sikit wkwkwk", "tetangga gw muka", "trahe elek diapak apakno yo elek"
7. Membuat *Leaf Node*:
    - Pada kelompok dengan nilai fitur "org", semua teks memiliki label *non-toxic*. Buat *leaf node* dengan label *non-toxic*.
    - Pada kelompok tanpa nilai fitur "org", semua teks memiliki label *toxic*. Buat *leaf node* dengan label *toxic*.
8. Membangun Pohon Keputusan:

    Hasil dari phon keputusan untuk sample *data* diatas dapat dilihat pada gambar 4.17 berikut:



Gambar 4. 17 Contoh Sampel Pohon Keputusan

Berdasarkan pohon keputusan dari gambar diatas, berikut adalah penjelasan mengenai setiap *node* dalam pohon:

1. *Node* 1:

   Fitur: 'org' <= 1.0

   Gini: 0.48

   Jumlah sampel: 5

   Nilai: [3, 2]

   Kelas: *non-toxic*

   Node 1 adalah *root node* atau simpul akar dari pohon keputusan. Ini berarti *node* ini merupakan titik awal dari pengambilan keputusan dalam pohon. Pada *node* ini, pohon memeriksa apakah nilai fitur 'org' lebih kecil atau sama dengan 1.0. Jika benar, pohon akan beralih ke anak kiri, dan jika tidak, pohon akan beralih ke anak kanan.

2. *Node* 2:

   Gini: 0.5

   Jumlah sampel: 4

   Nilai: [2, 2]

   Kelas: *non-toxic*

   Node 2 adalah anak kiri dari *node* 1. Pada *node* ini, pohon tidak melakukan pemisahan berdasarkan fitur karena tidak ada fitur yang

dipertimbangkan. Sebaliknya, pohon langsung memberikan keputusan bahwa kelasnya adalah *non-toxic*. *Node* ini mewakili suatu kondisi atau hasil di mana tidak ada fitur yang perlu diperiksa lagi dan keputusan langsung diambil berdasarkan hasil yang ada.

3. *Node* 3:

   Gini: 0.0

   Jumlah sampel: 1

   Nilai: [1, 0]

   Kelas: *non-toxic*

   Node 3 adalah anak kanan dari *node* 1. Pada *node* ini, pohon sekali lagi tidak melakukan pemisahan berdasarkan fitur karena hanya ada satu sampel yang tersisa. Pohon langsung memberikan keputusan bahwa kelasnya adalah *non-toxic*.

Berdasarkan pohon keputusan diatas, jika fitur 'org' kurang dari atau sama dengan 1.0, maka keputusan akhir adalah *non-toxic*. Jika fitur 'org' lebih besar dari 1.0, maka juga akan menghasilkan keputusan *non-toxic*. Dalam hal ini, pohon keputusan hanya mempertimbangkan satu fitur, yaitu 'org', dan dengan menggunakan aturan tersebut, pohon memberikan keputusan bahwa semua sampel yang diperiksa adalah *non-toxic*.

Hasil lainya yang diperoleh dari tahap model *training* dan testing ini adalah akurasi atau performa model yang diberikan selama proses *training* dan testing yang dilakukan seperti pada gambar berikut:

	precision	recall	f1-score	support
non-toxic	0.64	0.69	0.67	791
toxic	0.62	0.57	0.59	711
accuracy			0.63	1502
macro avg	0.63	0.63	0.63	1502
weighted avg	0.63	0.63	0.63	1502

Accuracy Score: 0.6338215712383488

Gambar 4. 18 *Classificaton Report*

Berdasarkan hasil gambar diatas, terdapat beberapa metrik yang dapat digunakan untuk melakukan penilaian terhadap hasil *training* dan testing model sebagai berikut:

1. *Precision*

   Pada kelas "*non-toxic*", precision adalah 0.64, yang berarti dari semua *data* yang diprediksi sebagai "*non-toxic*", sekitar 64% benar-benar *non-toxic*. Untuk kelas "*toxic*", precision adalah 0.62, yang berarti dari semua *data* yang diprediksi sebagai "*toxic*", sekitar 62% benar-benar *toxic*.
2. *Recall*

   Pada kelas "*non-toxic*", *recall* adalah 0.69, yang berarti model berhasil menemukan sekitar 69% dari semua *data* yang sebenarnya *non-toxic*. Untuk kelas "*toxic*", *recall* adalah 0.57, yang berarti model berhasil menemukan sekitar 57% dari semua contoh yang sebenarnya *toxic*.
3. *F1-score*

   Pada kelas "*non-toxic*" adalah 0.67, sedangkan untuk kelas "*toxic*" adalah 0.59. Semakin tinggi *F1-score,* semakin baik kinerja model dalam mengklasifikasikan *data non-toxic* dan *toxic*.
4. *Accuracy*

   Akurasi model yang didapatkan adalah 0.6338 atau sekitar 63.38%. Hal ini berarti model secara keseluruhan memprediksi dengan benar sekitar 63.38% dari semua *data* yang ada.

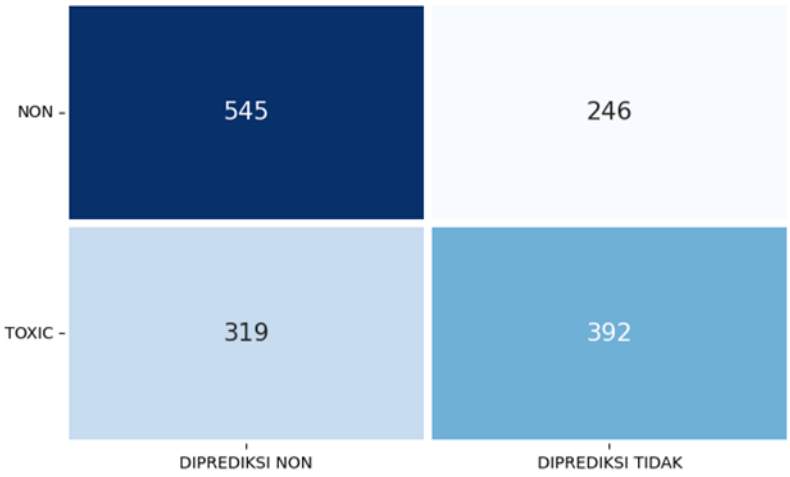
Secara keseluruhan (*macro avg dan weighted avg*), semua metrik (precision, *recall*, dan *F1-score*) memiliki nilai rata-rata sekitar 63%. Hal ini menunjukkan bahwa model memiliki performa yang serupa untuk kedua kelas, tanpa adanya preferensi yang jelas antara kelas "*non-toxic*" dan "*toxic*". Berdasarkan penjelasan sebelumnya menunjukkan adanya ruang untuk peningkatan dalam mengenali *data* yang sebenarnya *toxic*.

4.4 Hasil *Evaluation Model*

Dalam tahap model evaluasi ini dilakukan dengan mengevaluasi performa model dengan menggunakan dua matriks yaitu confusion *matrix* untuk evaluasi model dan *cross validation* untuk validasi model serta melakukan uji coba untuk memperkuat analisa performa yang diberikan oleh model yang dapat silihat pada penjelasan berikut:

1. *Confusion Matrix*

   Hasil evaluasi confusion *matrix* dapat dilihat pada gambar 4.19 berikut:



Gambar 4. 19 *Confusion Matrix*

Confusion matrix terdiri dari empat angka yang mewakili jumlah prediksi yang benar dan salah untuk setiap kelas. seperti pada gambar diatas maka, untuk penejlasanya dapat di lihat pada sebagai berikut:

- Jumlah prediksi benar untuk kelas *non-toxic* (kelas 0): Terdapat 545 prediksi yang benar untuk kelas *non-toxic*. Ini berarti ada 545 *data* pada set *data* yang sebenarnya adalah *non-toxic* dan juga diprediksi dengan benar sebagai *non-toxic*.
- Jumlah prediksi salah untuk kelas *non-toxic* (kelas 0): Terdapat 246 prediksi yang salah untuk kelas *non-toxic*. Ini berarti ada 246 *data* pada set *data* yang sebenarnya adalah *non-toxic*, tetapi salah diprediksi sebagai *toxic*.
- Jumlah prediksi benar untuk kelas *toxic* (kelas 1): Terdapat 392 prediksi yang benar untuk kelas *toxic*. Ini berarti ada 392 *data* pada set *data* yang sebenarnya adalah *toxic* dan juga diprediksi dengan benar sebagai *toxic*.
- Jumlah prediksi salah untuk kelas *toxic* (kelas 1): Terdapat 319 prediksi yang salah untuk kelas *toxic*. Ini berarti ada 319 *data* pada set *data* yang sebenarnya adalah *toxic*, tetapi salah diprediksi sebagai *non-toxic*.

Confusion matrix memberikan gambaran yang jelas tentang performa model dalam melakukan prediksi pada set *data*. Dalam penelitian ini, model cenderung memiliki kecenderungan untuk melakukan prediksi yang benar pada kelas *non-toxic* (kelas 0), dengan jumlah prediksi benar yang lebih tinggi (545) dibandingkan dengan jumlah prediksi salah (246). Namun, model memiliki tingkat kesalahan yang lebih tinggi dalam memprediksi kelas *toxic* (kelas 1), dengan jumlah prediksi benar (392) yang lebih rendah dibandingkan dengan jumlah prediksi salah (319). Hal ini mengindikasikan bahwa model memiliki tingkat keakuratan yang lebih baik dalam memprediksi kelas *non-toxic*, tetapi perlu diperbaiki dalam memprediksi kelas *toxic*.

Berdasarkan matriks daiats juga dapat diperoleh akurasi dari model yang apat dihitung menggunakan persamaan 2.3 sebagai berikut:

Beradasarkan *data* yang diketahui sebelumnya bahwa:

Jumlah prediksi benar = 545 + 392 = 937

Jumlah total *data* = 545 + 246 + 319 + 392 = 1502

$$Akurasi = \frac{(542 + 392)}{(545 + 246 + 319 + 392)} = 0.6238$$

Berdasarkan nilai yang didapatkan diaats maka, akurasi dari model yang dibuat adalah sebesar 0.6238 atau 62.38%

2. *Cross Validation*

   Cross-validation adalah teknik yang berguna untuk menguji performa model dengan membagi *data* menjadi beberapa lipatan dan melakukan evaluasi menggunakan kombinasi lipatan yang berbeda. Hasil dari penerapan teknik *cross validation* dengan menggunakan 5 lipatan atau 5 *fold* untuk memvalidasi performa model akan dijelaskan seperti pada penjelasan berikut:

   - *Fold*-1: Skor akurasi pada lipatan pertama adalah 0.6397670549084858. Ini berarti saat model dievaluasi pada lipatan pertama dari *data* yang telah dibagi menjadi beberapa lipatan, akurasi yang diperoleh adalah sekitar 63.98%.
   - *Fold*-2: Skor akurasi pada lipatan kedua adalah 0.649458784346378. Ini menunjukkan bahwa ketika model dievaluasi pada lipatan kedua dari *data*, akurasi yang diperoleh adalah sekitar 64.95%.
   - *Fold*-3: Skor akurasi pada lipatan ketiga adalah 0.6328059950041632. Saat model dievaluasi pada lipatan ketiga dari *data*, akurasi yang diperoleh adalah sekitar 63.28%.
   - *Fold*-4: Skor akurasi pada lipatan keempat adalah 0.6328059950041632. Ini berarti ketika model dievaluasi pada lipatan keempat dari *data*, akurasi yang diperoleh adalah sekitar 63.28%.
   - *Fold*-5: Skor akurasi pada lipatan kelima adalah 0.6469608659450458. Ini menunjukkan bahwa saat model dievaluasi

pada lipatan kelima dari *data*, akurasi yang diperoleh adalah sekitar 64.70%.

Berdasarkan penjelasan diatas maka diperoleh nilai rata-rata atau mean score dari semua lipatan adalah 0.6403597390416473. Ini merupakan nilai rata-rata dari skor akurasi pada setiap lipatan dalam *cross-validation*. Rata-rata akurasi ini memberikan gambaran tentang performa model secara keseluruhan saat dievaluasi menggunakan *cross-validation*. Dalam penelitian ini, rata-rata akurasi sekitar 64.04%.

Berdasarkan skor akurasi pada setiap lipatan dan skor akurasi rata-rata dari *cross-validation*, dapat disimpulkan bahwa model tidak secara konsisten memberikan hasil yang sama pada setiap lipatan. Skor akurasi pada setiap lipatan memiliki variasi, dengan rentang skor antara 0.6328 hingga 0.6495. Selain itu, skor akurasi rata-rata dari *cross-validation* adalah sekitar 0.6404.

Dengan variasi yang ada pada skor akurasi, dapat disimpulkan bahwa model mungkin tidak konsisten dalam mengklasifikasikan *data* pada lipatan yang berbeda. Ini bisa disebabkan oleh faktor-faktor seperti ukuran *data* yang terbatas, variasi dalam sampel *data* antara lipatan, atau sensitivitas model terhadap perubahan *data*.

3. Uji Coba *Model*

   Pada proses uji coba ini akan dibuat sebuah sistem yang digunkan untuk mengklasifikasikan teks ke dalam kategori *toxic* atau *non-toxic* berdasarkan model yang telah dibuat. Hasil dari uji coba ini dapat dilihat pada tabel 4.13 berikut:

Tabel 4. 13 Uji Coba *Model*

Comment	***Predicted***	***True Class***
setuju bgt ama sony pan si IR itu viral gegara maslah keluarganya,, blm ada prestasi apa2 udah dpt award heyy	*Toxic*	*Toxic*

Comment	***Predicted***	***True Class***
abda yakinnn😂,, kalo emang istri yg sholeha ga nyebarin aib suaminya d sosmed apapun konsekuensinya seberat apapun masalahnya kan katanya tau agama lebih baik curhat disepertiga malam mbakk😂		
Kek odong odong pasar malem	*Toxic*	*Toxic*
Gigi nih bos senggol dong	*Toxic*	*Toxic*
Kurang banyak kisah cinta A-ri dan jun kyoung pdhal d nantikkan bangeett,jlan bareng,santai bareng,makan bareng,kepantai dll□□	*Non-toxic*	*Non-toxic*
Kucing kesayangan Alm Ayah ku juga tau makam ayah ku, tanpa ada yg bawa kesana.. Padahal jenazah ayah saat itu ngak diselenggarakan di rumah karna positif covid.. Jd dari rumah sakit langsung ke pemakaman.. Tapi tu kucing tau aja makam ayah kami	*Non-toxic*	*Non-toxic*
Semoga yg di timpa musibah bisa bersabar nanti tuhan pasti akan menggantinya yg lebih baik dan banyak	*Toxic*	*Non-toxic*
Bang densu emang beda, keren... Di saat yang lain jadi kompor dia memandang kasus ini lebih bijak	*Non-toxic*	*Non-toxic*
Freshgraduate silahkan melamar terus di persyaratan selanjutnya harus mempunyai pengalaman 😂	*Non-toxic*	*Non-toxic*
Baunya pasti bikin hidung trauma	*Toxic*	*Toxic*

Comment	***Predicted***	***True Class***
Nitipin sapi di wilayah pak RT krn ga punya tempat utk sapinya, trus pas idul adha diambil utk di sembelih di tempat lain dan utk warga lain. Begitu kan ya intinya?	*Non-toxic*	*Non-toxic*

Berdasarkan tabel diatas dengan menggunakan sepuluh *data* sebagai *data* uji coba didapatkan hasil bahwa 1 dari 10 *data* yang diganakn memberikan hasil prediksi yang salah dimana hasil prediksi adalah *toxic* namun true classnya adalah *non-toxic* sedangkan 9 *data* uji lainya berhasil diprediksi dengan benar sesuai dengan true classnya.

Berdasarkan penjelasan diatas apabila membandingkan perfomra model berdasarkan akurasi dan hasil dari uji coba dengan menggunakan 10 *data* dapat di simpulkan bahwa jika, model memiliki akurasi sebesar 62.38% dalam memprediksi toksisitas komentar. Meskipun akurasi ini tidak terlalu tinggi, model masih memberikan hasil yang lebih baik daripada tebakan acak atau asumsi nol (50%).

Hasil uji coba *data* baru menunjukkan bahwa model cenderung mampu memprediksi dengan benar sebagian besar komentar. Mayoritas komentar dengan kelas sebenarnya "*toxic*" atau "*non-toxic*" diprediksi dengan benar oleh model.

Dengan akurasi 62.38% dan kemampuan yang relatif baik dalam memprediksi komentar *toxic*, dapat dikatakan bahwa model ini memiliki performa yang cukup baik. Namun, masih terdapat ruang untuk peningkatan performa agar lebih mendekati atau melebihi tingkat akurasi yang lebih tinggi.

4.5 Kelemahan Sistem

Berdasarkan hasil analisa yang dilakukan oleh peneliti ditemukan beberapa kelamahan dalam model atau sistem klasifikasi yang dibuat antara lain adalah:

1. *Model* memiliki akurasi yang relatif rendah, yaitu 62.38%. Hal ini menunjukkan bahwa model tidak dapat memprediksi dengan tepat sebagian besar komentar.
2. *Model* memeiliki kemungkinan tidak dapat secara akurat memprediksi komentar dalam variasi bahasa yang lebih kompleks atau dalam konteks yang tidak ditemukan dalam *data* pelatihan. *Model* memiliki keterbatasan dalam mengenali nuansa bahasa, kata-kata slang, atau konteks keberagaman bahasa daerah.
3. *Model* ini tidak mampu menangkap makna yang lebih dalam atau fitur *non*-teks dalam komentar, seperti konteks sosial, *emoticon*, atau tanda baca. Hal ini dapat menyebabkan keterbatasan dalam memahami niat atau nada yang tersembunyi dalam suatu komentar.
4. *Overfitting* pada model terjadi karena model terlalu dipersonalisasi untuk *data* pelatihan dan tidak dapat menggeneralisasi dengan baik pada *data* baru. Di sisi lain, underfitting terjadi karena model terlalu sederhana dan tidak dapat menangkap pola yang kompleks dalam *data*.