

### 3. ANALISA DAN DESAIN SISTEM

Pada bab ini akan dibahas mengenai analisa dan desain sistem yang akan diterapkan dalam pengembangan program analisis sentimen pada *review* merek *skincare* lokal yang ditulis di Twitter. Selain itu, akan dijelaskan pula mengenai struktur data yang digunakan serta desain dan cara kerja dari program aplikasi yang dibuat.

#### 3.1 Analisa Permasalahan

Pasar *skincare* Indonesia saat ini mengalami pertumbuhan yang pesat, hal ini karena didorong oleh tren penggunaan produk kecantikan yang meningkat. Peningkatan ini berdampak pada munculnya berbagai merek *skincare* lokal yang bersaing dalam penjualan. Persaingan ini selain berdampak bagi perusahaan juga menimbulkan dilema bagi konsumen yang ingin membeli. Variasi merek *skincare* yang beragam menjadikan adanya kebutuhan akan ulasan atau *review* produk dari konsumen lain, yang menjadi faktor penting dalam pengambilan keputusan pembelian. Di lain sisi, ulasan konsumen juga memiliki dampak signifikan terhadap *brand image* suatu produk. *Review* buruk dapat membuat konsumen mengurungkan niatnya untuk membeli produk, sementara *review* positif meningkatkan minat beli. Oleh karena itu, penting bagi perusahaan untuk memantau dan merespons ulasan konsumen dengan baik guna menjaga reputasi merek mereka. Twitter atau sekarang memiliki nama X menjadi salah satu *platform* utama untuk diskusi tentang produk *skincare*. Interaksi di Twitter dapat mendorong terjadinya *electronic word of mouth (e-WOM)* yang mempengaruhi minat beli produk *skincare* lokal. Oleh karena itu, analisis sentimen terhadap ulasan di Twitter dapat menjadi salah satu solusi untuk memahami persepsi konsumen terhadap merek *skincare*.

Namun, analisis sentimen di Twitter memiliki beberapa tantangan, dimana jumlah *tweet* setiap harinya memiliki jumlah yang banyak, termasuk *tweet review skincare* dan apabila membaca *review* secara keseluruhan dengan cara manual, tentunya akan menghabiskan waktu yang cukup lama. Oleh karena itu, dibutuhkan suatu aplikasi yang dapat membantu dalam mempercepat pengolahan *review* yang dituliskan pada Twitter. Aplikasi ini dapat membantu konsumen dan produsen dalam pengambilan keputusan.

#### 3.2 Analisa Kebutuhan

Berdasarkan permasalahan yang dituliskan pada Subbab 3.1, maka dapat disimpulkan bahwa adanya kebutuhan untuk pengembangan *platform* atau sistem analisis sentimen

terhadap *review* merek *skincare* lokal di Twitter yang dapat membantu proses pengolahan *review* menjadi lebih cepat. Beberapa fitur yang akan disediakan pada aplikasi yaitu:

1. Aplikasi yang dapat mengevaluasi sentimen dari *tweet* berisi *review skincare* lokal, apakah *tweet* tersebut merupakan pendapat Positif, Negatif, atau Netral.
2. Evaluasi didukung dengan deteksi sarkasme, dimana pendapat dengan sentimen Positif akan dicek kembali, apakah merupakan Sarkasme atau bukan.
3. Aplikasi memberikan fitur bagi konsumen untuk mengunggah atau mengambil data *tweet* yang ada pada Twitter untuk dilakukan analisis sentimen.

Dengan memahami dan memenuhi kebutuhan-kebutuhan tersebut, pengembangan sistem analisis sentimen terhadap merek *skincare* lokal di Twitter diharapkan dapat memberikan manfaat yang signifikan bagi konsumen dan produsen.

### 3.3 Pengambilan Data

*Dataset tweet* yang digunakan dalam penelitian ini berasal dari proses *scraping* menggunakan *library Tweet Harvest* dengan referensi kode untuk *scraping* milik Satria (2023). Proses *scraping* dilakukan dengan memasukkan *keyword* pencarian di Twitter, *keyword* yang digunakan adalah merek *skincare*. Merek yang digunakan dalam penelitian ini adalah Avoskin, Somethinc, dan Skingame dan data *tweet* yang diambil merupakan *tweet* berbahasa Indonesia yang ditulis pada Januari 2023 - Desember 2023. Data *tweet* yang bukan merupakan sebuah *review* akan dihapus, misalnya *tweet* yang merupakan iklan dan *tweet* jual beli tanpa *review*. Hasil dari *dataset* yang telah disortir akan digunakan untuk penelitian. *Dataset* dari *scraping* memiliki kolom *created\_at*, *id\_str*, *full\_text*, *quote\_count*, *reply\_count*, *retweet\_count*, *favorite\_count*, *lang*, *user\_id\_str*, *conversation\_id\_str*, *username*, *tweet\_url*. Untuk *dataset* dalam melatih model deteksi sarkasme, dibuat *file* baru dengan nama "*dataset sarkas*" yang isi data nya diambil dari data sentimen Positif hasil dari metode *Lexicon-based* dengan tambahan dari *dataset* lain yang diambil dengan proses *scraping* dengan kata kunci "#sarkas" dan data kalimat sarkasme yang diambil melalui penyebaran Google Form, hasil respon *form* dapat dilihat pada Lampiran 1. Selanjutnya, data dari hasil metode *Lexicon-based*, diberikan label sarkasme secara manual. Hasil akhir *dataset sarkas* akan memiliki kolom yang berisi label apakah data termasuk ke dalam "Sarcasm" atau "NonSarcasm". Hasil dari pemberian label diverifikasi kebenarannya oleh ahli, bukti verifikasi ada pada Lampiran 4. Pada Tabel 3.1 terdapat nama penyebutan *dataset* yang digunakan dalam penelitian ini beserta keterangan

singkat untuk penjelasan dari *dataset* tersebut. Penyebutan nama dari *dataset* kedepannya akan digunakan pada bab-bab kedepannya.

Tabel 3.1

Skema *Dataset* Penelitian

Nama <i>Dataset</i>	Jumlah Data	Penjelasan
<i>Dataset A</i> ( <i>Dataset</i> Utama Tweet Review Skincare)	24150 data	<i>Dataset A</i> adalah <i>dataset</i> utama yang dikumpulkan dari proses <i>scraping</i> pada Twitter dengan <i>keyword</i> Avoskin, Somethinc, Skingame dan <i>tweet</i> yang diambil adalah <i>tweet</i> yang ditulis pada Januari 2023 sampai Desember 2023.
<i>Dataset B</i> ( <i>Dataset</i> Sarkasme)	10904 data dengan 296 data Sarcasm dan 10608 data NonSarcasm	<i>Dataset B</i> adalah <i>dataset</i> yang digunakan untuk melatih model deteksi sarkasme. Data yang ada pada <i>dataset</i> ini berasal dari data dengan sentimen Positif dari <i>Dataset A</i> setelah dilakukan proses analisis sentimen dengan metode <i>Lexicon-based</i> . Selain itu, data yang ada pada <i>Dataset B</i> juga berasal dari pengumpulan data melalui Google Form, hasil <i>scraping</i> dengan <i>keyword</i> “#sarkas” dan data dari <i>dataset</i> sarkas dari penelitian lainnya yang diunggah di GitHub ( <a href="https://github.com/yazid29/Deteksi_Sarkasme/tree/main/uploads">https://github.com/yazid29/Deteksi_Sarkasme/tree/main/uploads</a> ).
<i>Dataset C</i> ( <i>Dataset</i> Sarcasm baru untuk uji coba model)	23 data dengan 10 data NonSarcasm dan 13 data Sarcasm	<i>Dataset C</i> adalah <i>dataset</i> sarkasme di luar dari <i>Dataset B</i> . Dimana terdapat data kalimat-kalimat yang merupakan Sarcasm maupun NonSarcasm. <i>Dataset</i> ini digunakan untuk mencoba performa model yang dilatih ketika dihadapkan dengan <i>dataset</i> baru. Data yang ada pada <i>dataset</i> ini dikumpulkan dari kalimat yang dibikin oleh orang lain maupun penulis.
<i>Dataset D</i> ( <i>Dataset</i> untuk uji coba pengaruh deteksi sarkasme)	9 data dengan seluruh data memiliki label sentimen Negatif	<i>Dataset D</i> adalah <i>dataset</i> baru yang digunakan untuk menguji pengaruh deteksi sarkasme. Seluruh data pada <i>dataset</i> ini merupakan data kalimat-kalimat sarkas dengan label

		sentimen seluruh data adalah Negatif. Pengambilan data dari <i>dataset</i> ini diambil dari sampel <i>dataset</i> sarkasme ( <i>Dataset B</i> ) dan digabungkan dengan <i>dataset</i> yang dikumpulkan penulis dari beberapa mahasiswa Universitas Kristen Petra.
--	--	---

Selain data *tweet*, diambil juga *source* yang digunakan untuk keperluan *preprocessing*, yaitu *file* kamus yang berisi data terjemahan *slang-word* dan *colloquial-word* yang diambil dari *preprocessing source* dari penelitian sebelumnya yang diunggah pada *GitHub* ([https://github.com/onpilot/sentimen-bahasa/tree/master/kamus/nasalsabila\\_kamus-alay](https://github.com/onpilot/sentimen-bahasa/tree/master/kamus/nasalsabila_kamus-alay) dan [https://github.com/evanmartua34/Twitter-COVID19-Indonesia-Sentiment-Analysis---Lexicon-Based/tree/master/cleaning\\_source](https://github.com/evanmartua34/Twitter-COVID19-Indonesia-Sentiment-Analysis---Lexicon-Based/tree/master/cleaning_source)). Kemudian, pada penelitian ini, terdapat juga *file* kamus berisi *list* masalah keluhan kulit yang diambil dari beberapa sumber artikel kecantikan dari *website* berikut.

- *Beauty Haul*  
(<https://www.beautyhaul.com/blog/12-macam-masalah-kulit-wajah-yang-paling-sering-dialami>)
- *Klik Dokter*  
(<https://www.klikdokter.com/info-sehat/kulit/masalah-kulit-yang-sering-dialami-remaja>)
- *CNN Indonesia*  
(<https://www.cnnindonesia.com/gaya-hidup/20230405130746-282-933880/10-jenis-masalah-kulit-penyebab-dan-cara-mengatasinya>)
- *Female Daily*  
(<https://editorial.femaledaily.com/blog/2021/12/07/7-permasalahan-kulit-paling-umum-pada-remaja>)

Analisis kata yang digunakan dari *tweet review* pada Twitter akan digunakan saat analisis sentimen dengan *Lexicon-based* dan *file* ini juga merupakan kontribusi yang dihasilkan dari penelitian ini. Hasil dari pengumpulan kamus kata permasalahan kulit dapat dilihat pada Tabel 3.2.

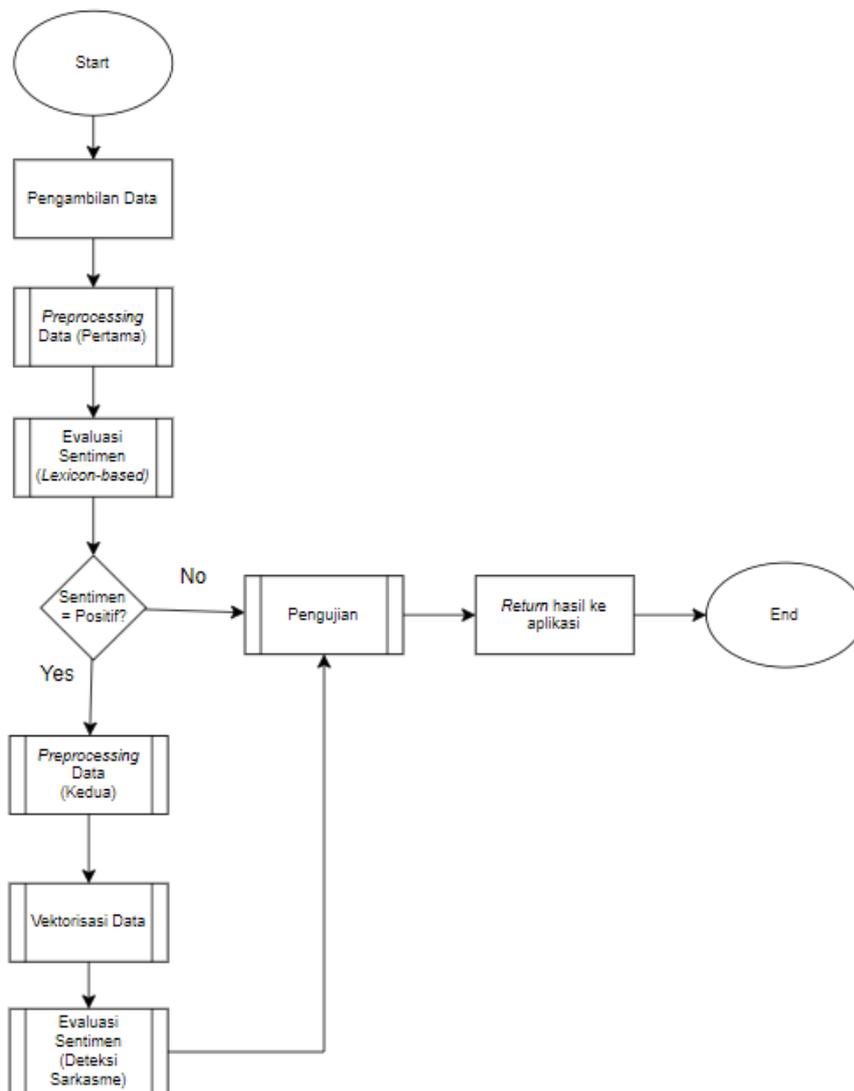
Tabel 3.2

Kata pada kamus permasalahan kulit

jerawat	milia	eksim	kerutan
komedo	kusam	psoriasis	breakout
flek hitam	kering	berminyak	pori-pori besar
rosacea	bruntusan	selulit	
bercak	kudis	stretch mark	
kerutan	scabies	stretchmark	
garis halus	kurap	varises	
milia	herpes	kutil	
kusam	dermatitis seborois	sunburn	

### 3.4 Desain Sistem Analisis Sentimen

Dalam membangun sistem analisis sentimen pada aplikasi ini, terdapat beberapa proses yang dilakukan hingga aplikasi akhirnya dapat mencapai hasil. Proses tersebut meliputi *preprocessing* data, vektorisasi data, evaluasi sentimen, dan proses *testing*. Alur kerjanya adalah data diambil dengan proses *scraping* dari Twitter berdasarkan *keyword* dan *range* tanggal, kemudian dilakukan *preprocessing* sebelum akhirnya dilakukan analisis sentimen dengan metode *Lexicon-based*. Data yang mendapatkan sentimen positif dari analisis sentimen akan diproses kembali untuk dilihat, apakah merupakan sarkasme atau bukan. Cara kerjanya adalah dengan mengambil *tweet* awal dan dilakukan *preprocessing* kembali karena berbeda dengan *preprocessing* awal, selanjutnya dilakukan deteksi sarkasme. Apabila merupakan sarkasme, maka sentimen akan diubah dari yang awalnya positif menjadi negatif. Selanjutnya, akan dilakukan pengujian dan hasil dari analisis sentimen akan ditampilkan pada aplikasi. Alur kerja dalam sistem analisis sentimen dapat dilihat pada *flowchart* pada Gambar 3.1.



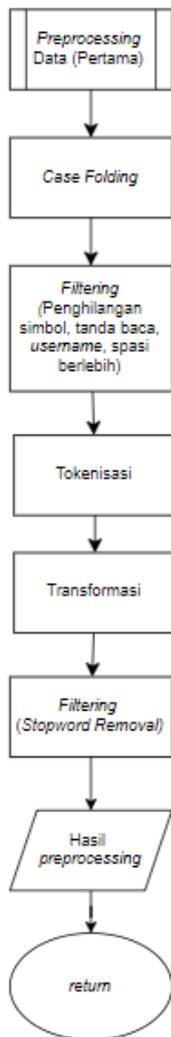
**Gambar 3.1** Alur Kerja Sistem Analisis Sentimen

### 3.4.1 Preprocessing Data

Dalam analisis sentimen, setiap *tweet* yang masuk ke dalam model akan melalui tahap *preprocessing* terlebih dahulu dimana akan terdapat beberapa perlakuan terhadap *tweet* tersebut. Pada model yang dibangun, akan dilakukan dua kali *preprocessing* dengan perlakuan yang berbeda, yaitu *preprocessing* untuk analisis sentimen dengan *Lexicon-based* dan *preprocessing* untuk deteksi sarkasme. *Preprocessing* dilakukan secara berbeda karena penggunaan fitur yang berbeda. Namun, sebelum dilakukan perlakuan, *dataset* mentah yang didapatkan dari hasil *scraping* akan dibersihkan terlebih dahulu dengan menghapus data-data yang bukan merupakan sebuah *review*, misalnya *tweet* iklan, *tweet* jual beli tanpa *review*, dan

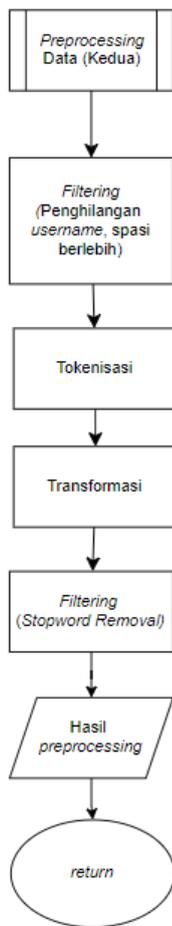
*tweet* yang tidak berhubungan yang diambil dalam proses *scraping*. Setelah dibersihkan, maka akan langsung dilakukan proses *preprocessing* berikutnya dengan Python. *Dataset* yang digunakan dalam bentuk *.xlsx* dimana pada program Python akan digunakan *library Pandas* untuk membaca file *excel* dari *dataset*.

*Preprocessing* pertama, yaitu untuk analisis sentimen dengan *Lexicon-based* akan dimulai dengan *case folding* atau mengubah seluruh huruf pada *tweet* menjadi huruf kecil atau *lowercase*. Proses ini menggunakan *string.lower* yang telah disediakan pada modul *string* pada Python. Kedua, akan dilakukan *filtering* dimana akan dilakukan penghapusan terhadap simbol, tanda baca, *username* pada *tweet*, dan spasi berlebih. Proses ini juga menggunakan fungsi yang sudah disediakan pada modul *string* dan juga *regular expression operations* pada Python seperti *string.strip*, *re.sub*, *string.punctuation*, dan *string.join*. Kemudian, akan dilakukan tokenisasi, dimana *tweet* akan dipecah menjadi kata-kata agar dapat lebih mudah diproses di tahap selanjutnya, proses tokenisasi menggunakan *library nltk* pada Python. Contoh dari tokenisasi adalah “ini bagus banget” akan berubah menjadi “ini,bagus,banget”. Setelah itu, akan dilakukan transformasi, dimana pada proses ini kata-kata hasil tokenisasi sebelumnya akan diproses dengan dicocokkan pada kamus *slang-word* dan *colloquial-word* dan kata yang dikenali oleh kamus tersebut akan diubah menjadi Bahasa Indonesia yang dikenali oleh *lexicon* sentimen agar perhitungan *polarity score* akan lebih akurat. Contoh dari proses transformasi adalah “bangedh” akan menjadi “banget”. Setelah transformasi kata, akan dilakukan *stopword removal*, yaitu menghilangkan kata-kata yang tidak relevan di dalam suatu kalimat. Kata-kata ini misalnya kata sambung, artikel, dan preposisi. Proses *stopword removal* juga menggunakan *library nltk* pada Python. Setelah semua proses sebelumnya dilakukan, maka hal terakhir yang akan dilakukan adalah menggabungkan kembali token-token menjadi satu kalimat dengan menggunakan *string.join* yang merupakan fungsi pada modul *string* pada Python. Penggambaran *flowchart* pada *preprocessing* data ini dapat dilihat pada Gambar 3.2.



**Gambar 3.2** Alur *Preprocessing* Pertama (untuk *Lexicon-based*)

*Preprocessing* kedua adalah *preprocessing* sebelum dilakukan deteksi sarkasme. Pada *preprocessing* ini, akan digunakan *tweet* asli sebelum *preprocessing* pertama. Secara keseluruhan, proses yang dilakukan mirip dengan *preprocessing* pertama. Perbedaannya adalah, pada *preprocessing* kali ini, tidak dilakukan *case folding* dan penghapusan karakter-karakter unik seperti emotikon dan tanda baca. Hal ini dikarenakan, besar kecil huruf dan emotikon dikonversi dan emotikon juga menjadi fitur yang digunakan dalam proses deteksi sarkasme, dikarenakan beberapa hal ini dapat menjadi faktor yang membantu dalam mengenali suatu *tweet* apakah merupakan suatu sarkasme atau bukan. Penggambaran *flowchart* pada *preprocessing* data yang kedua ini dapat dilihat pada Gambar 3.3.

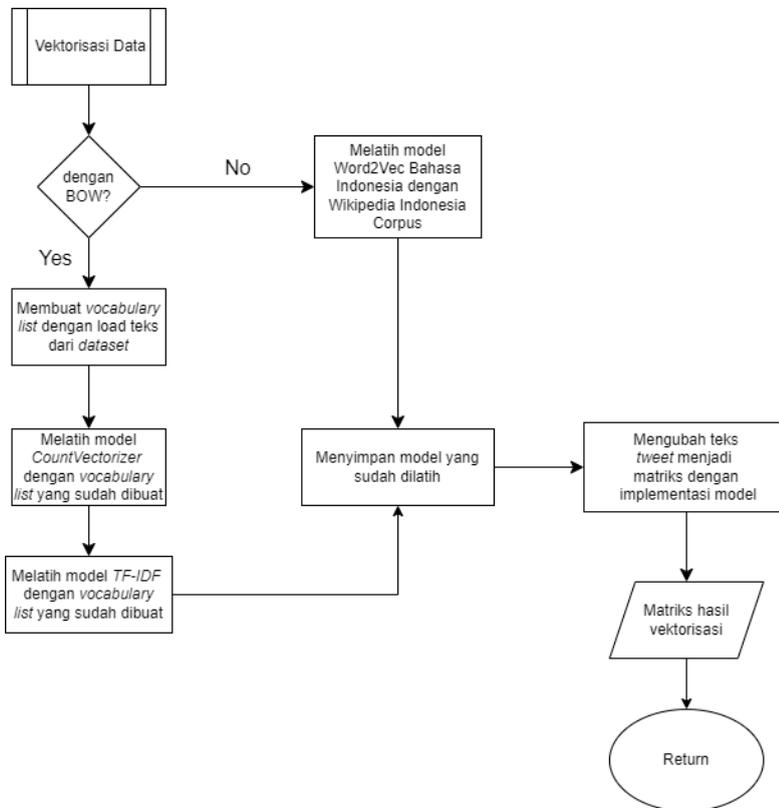


**Gambar 3.3** Alur *Preprocessing* Kedua (untuk deteksi sarkasme)

### 3.4.2 Vektorisasi Data

Pada deteksi sarkasme, digunakan dua metode *Machine Learning* yaitu *Support Vector Machine* dan *Random Forest*. Namun, sebelum masuk ke kedua metode tersebut, walaupun sudah dilakukan *preprocessing* data, model tidak dapat langsung menerima data dalam bentuk teks. Perlu dilakukan vektorisasi data, dimana teks hasil *preprocessing* diubah menjadi matriks. Vektorisasi dilakukan dengan *feature extraction*, dimana akan dilakukan pengambilan fitur dari data dengan menghitung jumlah kapital, melihat inkonsisten kapital, keberadaan kata *slang* sarkas, keberadaan emoji dan perhitungan tanda baca. Selain itu, dilakukan juga proses vektorisasi data dengan memanfaatkan *library* yang sudah ada, yaitu menggunakan *Bag of Words* (BOW) yaitu *CountVectorizer* dan *TF-IDF* dan juga menggunakan *word-embedding* dengan *Word2Vec*. Sebelum menggunakan model BOW, perlu untuk membuat suatu *vocabulary list* yang dimana adalah *raw data* yang digunakan sebagai acuan model *CountVectorizer* untuk mengubah kata menjadi bentuk matriks. Selain itu, untuk *Word2Vec* juga

diperlukan untuk pelatihan dengan Wikipedia Indonesia *Corpus* yang diubah menjadi teks. Proses dari vektorisasi data digambarkan pada *flowchart* yang dapat dilihat pada Gambar 3.4.



**Gambar 3.4** Alur Vektorisasi Data

### 3.4.3 Evaluasi Sentimen

Dalam melakukan evaluasi sentimen dari data *tweet*, terdapat proses yang melibatkan beberapa metode analisis sentimen. Setelah dilakukan data *preprocessing* dan vektorisasi data, maka selanjutnya akan dilakukan evaluasi sentimen. Evaluasi sentimen adalah proses analisis sentimen dari sebuah data *tweet*. Evaluasi pertama adalah dengan menggunakan metode *Lexicon-based* dimana data hasil *preprocessing* akan dihitung *polarity score* berdasarkan *lexicon* atau kamus yang sudah ada, pada penelitian kali ini, akan menggunakan *lexicon* dari *InSet* dengan gabungan dari *lexicon Sentistrength* untuk kata yang tidak terdapat pada *lexicon Inset*. Bobot pada *lexicon* ini memiliki skor dari rentang -5 sampai dengan +5 yang merupakan bawaan dari *lexicon*. Total dari gabungan *Inset Lexicon* dan *Sentistrength* menghasilkan jumlah data pada kamus sebanyak 10044 data kata beserta bobotnya. Skema yang berisi kata dan bobot dari *Inset Lexicon* dan *Sentistrength Lexicon* dapat dilihat pada Tabel 3.3 dan Tabel 3.4.

Tabel 3.3

Skema *InSet Lexicon*

<b>Word</b>	<b>Weight</b>
bagus	2
cocok	4
jerawat	-3

Tabel 3.4

Skema *Sentistrength Lexicon*

<b>Word</b>	<b>Weight</b>
kecewa	-3
memukau	4
lumayan	2

Untuk perhitungan *polarity score* pada penelitian ini, tidak langsung menghitung kata hasil *preprocessing*, namun dilakukan pengecekan, apakah data *tweet* mengandung kata keluhan kulit dan apabila mengandung, dicek kembali apakah terdapat kata yang termasuk kata dengan *review* baik. Apabila terdapat kata *review* baik pada data *tweet* yang mengandung permasalahan kulit, maka *score* untuk kata permasalahan kulit tidak akan dimasukkan ke dalam perhitungan, misalnya pada kalimat “merek ini bagus dalam menyembuhkan jerawat di wajahku” maka untuk bobot dari kata “Jerawat” tidak akan dimasukkan ke dalam perhitungan karena kalimat tersebut mengandung kata “Bagus”. Hal ini dikarenakan kata permasalahan kulit termasuk dalam *lexicon* berbobot negatif. Skema untuk kata dan bobot dari kata permasalahan kulit yang ada pada *lexicon* dapat dilihat pada Tabel 3.5.

Tabel 3.5

Skema kata Permasalahan Kulit pada *Lexicon*

<b>Word</b>	<b>Weight</b>
jerawat	-3
bercak	-5

kusam	-3
kering	-3
berkurap	-4
kutil	-4

Pengecualian ini bertujuan untuk menghindari kekeliruan dalam evaluasi sentimen, terutama ketika dimana ada kasus pengguna menceritakan keluhan kulit terlebih dahulu sebelum memberikan *review* baik terhadap suatu merek, apabila tidak dilakukan penanganan perhitungan, maka *polarity score* akan cenderung negatif karena keluhan-keluhan kulit memiliki nilai negatif. Contoh *tweet* yang bersentimen positif namun terhitung sebagai negatif dapat dilihat dari Tabel 3.6.

Tabel 3.6

Contoh teks bersentimen Positif yang terhitung menjadi Negatif

Teks	Score	Sentimen
@ohmybeautybank avoskin bagus bgtt bikin jerawat mendem gajadi numbuh	-5	Negatif
aku pemula retinol, dan st aku combi oily, pake retinol tonernya avoskin yg mini size, bagus bgtt parah bikin kulit cerah dan bantu keringin jerawat, klo ada jerawat tumbuh langsung mateng besoknya! tekstur kulit jadi rata juga.	-3	Negatif
avoskin vit c serius. di cumbo sama ss azarine di pake nya pagi/ siang. muka gue asalnya kusam grgr pake serum itu muka gue bagus bgt	-2	Negatif

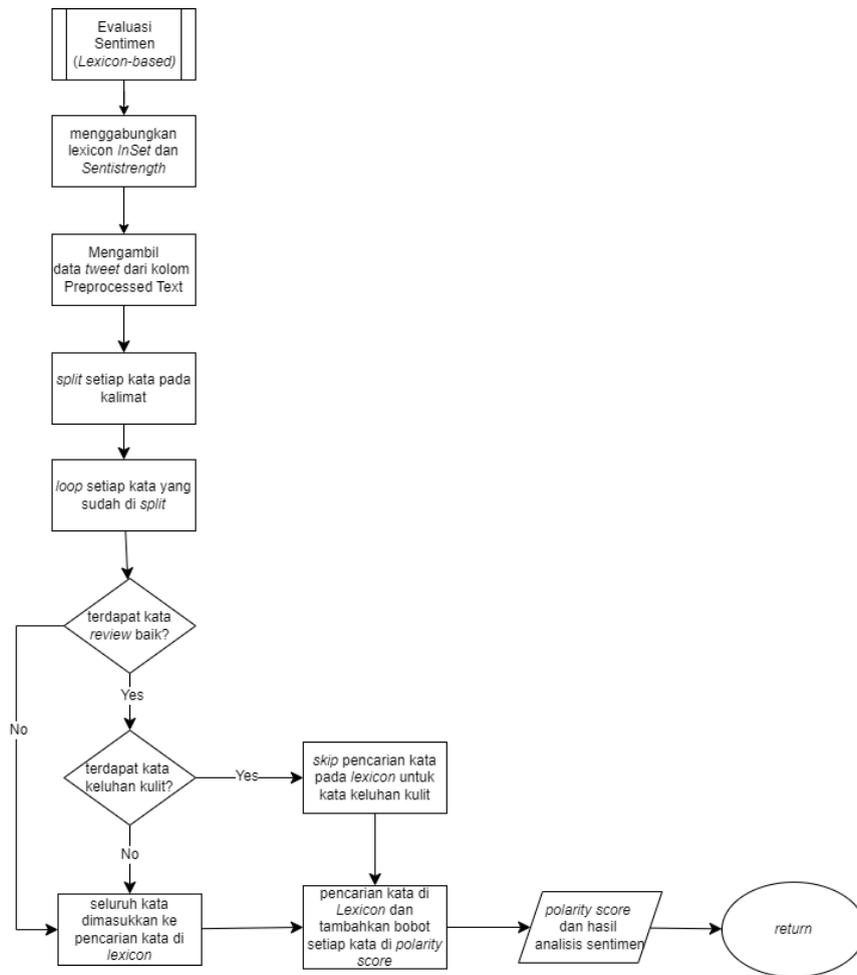
Setelah dilakukan pengecualian perhitungan terhadap kata yang merupakan permasalahan kulit apabila terdeteksi kata yang digunakan untuk *review* hal baik, didapatkan hasil seperti pada Tabel 3.7.

Tabel 3.7

Teks setelah dilakukan pengecualian pada perhitungan

<b>Teks</b>	<b>Score</b>	<b>Sentimen</b>
@ohmybeautybank avoskin bagus bgtt bikin jerawat mendem gajadi numbuh	1	Positif
aku pemula retinol, dan st aku combi oily, pake retinol tonernya avoskin yg mini size, bagus bgtt parah bikin kulit cerah dan bantu keringin jerawat, klo ada jerawat tumbuh langsung mateng besoknya! tekstur kulit jadi rata juga.	3	Positif
avoskin vit c serius. di cumbo sama ss azarine di pake nya pagi/ siang. muka gue asalnya kusam grgr pake serum itu muka gue bagus bgt	1	Positif

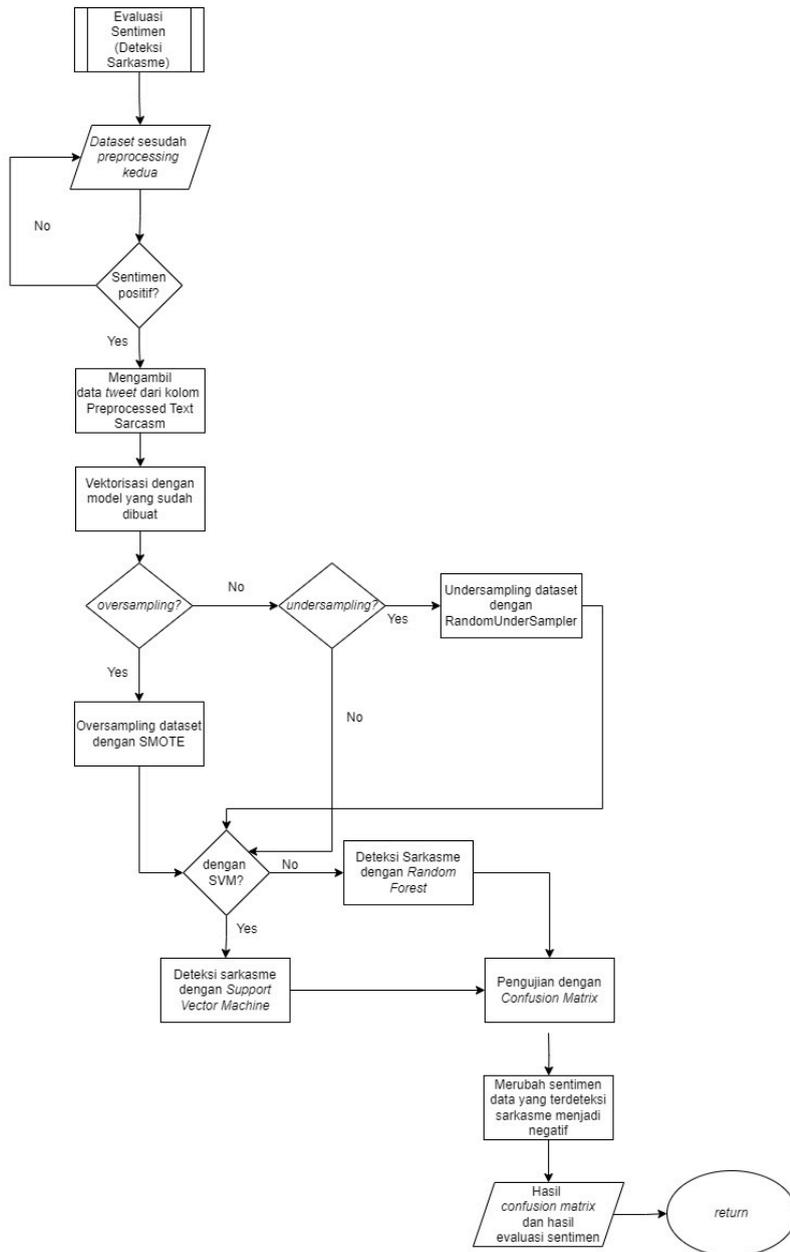
Penggambaran *flowchart* pada evaluasi sentimen dengan metode *lexicon-based* dapat dilihat pada Gambar 3.5.



**Gambar 3.5** Alur Sistem Analisis Sentimen dengan Metode *Lexicon-based*

Berikutnya untuk evaluasi sentimen kedua, yaitu deteksi sarkasme pada analisis sentimen. Proses ini mengecek kembali sentimen dari *tweet* yang mendapatkan sentimen Positif dari hasil analisis sentimen dengan *Lexicon-based*. Sentimen positif diambil karena sarkasme merupakan kalimat dengan sentimen negatif namun diutarakan dengan kalimat positif. Deteksi sarkasme menggunakan dua metode, yaitu *Support Vector Machine* dan *Random Forest*. Kedua metode akan digunakan dan hasil akhirnya akan dibandingkan, metode manakah yang memiliki akurasi paling tinggi. Penerapan kedua metode dalam penelitian ini akan menggunakan *library sklearn*. Cara kerjanya adalah setelah didapatkan hasil sentimen dari metode *Lexicon-based*, diambil *dataset* untuk hasil sentimen Positif. Data yang diolah adalah *dataset* awal sebelum *preprocessing*. Hal ini dikarenakan pada proses evaluasi sentimen kedua ini, *preprocessing* data berbeda dengan *preprocessing* dengan *Lexicon-based*. Setelah itu, dilakukan *preprocessing* kedua seperti yang sudah dijelaskan pada Sub bab 3.4.1 dan setelah *preprocessing* akan dilakukan vektorisasi data seperti yang dijelaskan pada Sub bab 3.4.2. Kemudian, akan dilakukan proses *training* dan *testing* data. Kedua metode akan mengevaluasi,

apakah data *tweet* dengan label positif tersebut merupakan Sarkasme atau Non Sarkasme. Apabila merupakan Sarkasme, maka sentimen *tweet* akan diubah, dari yang awalnya positif akan menjadi negatif. Model yang sudah dibuat akan disimpan untuk diimplementasikan ke dalam aplikasi. Penggambaran *flowchart* deteksi sarkasme pada evaluasi sentimen dengan metode *Support Vector Machine* dan *Random Forest* dapat dilihat pada Gambar 3.6.



**Gambar 3.6** Alur Sistem Deteksi Sarkasme dengan Metode SVM dan *Random Forest*

### 3.5 Alur Pengujian

Model dari sistem analisis sentimen, sebelum diimplementasikan pada aplikasi, akan melewati tahap pengujian terlebih dahulu. Pengujian untuk hasil dari metode *Lexicon-based* dilakukan dengan membandingkan hasil evaluasi sentimen dari perhitungan *polarity score* dengan data yang sudah dilabeli sebelumnya. Ketepatan hasil perhitungan sentimen akan dihitung dan dicari persentase akurasi. Kemudian untuk pengujian deteksi sarkasme dengan metode *Support Vector Machine* dan metode *Random Forest*, akan digunakan pengujian dengan menggunakan *confusion matrix* dengan 4 parameter yaitu akurasi, presisi, *recall*, dan *f1-score*. Penerapan *confusion matrix* pada program akan menggunakan *library sklearn* yang ada pada Python. Selain itu, untuk menguji jalannya model, dilakukan percobaan implementasi model terhadap *dataset* yang sudah diberi label sebelumnya dan dilihat, seberapa akurat hasil prediksi model dibandingkan dengan label yang sudah diberikan.

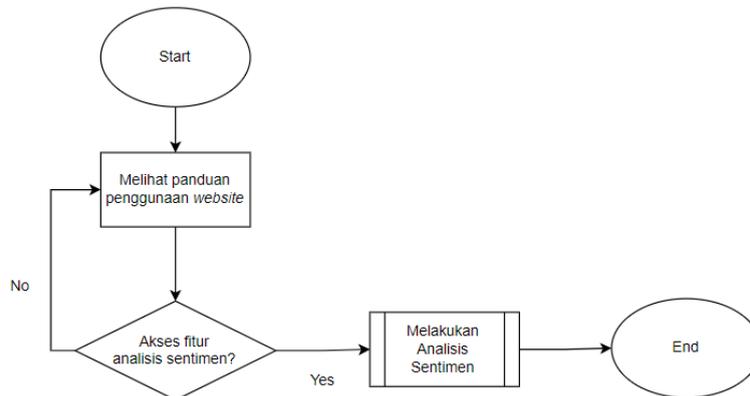
### 3.6 Desain Sistem Aplikasi

Sistem analisis sentimen akan diimplementasikan dalam suatu aplikasi berbasis *website* yang memiliki fungsi utama untuk membantu *user* dalam pengolahan *tweet review* dari merek *skincare* lokal. Pada aplikasi yang dibangun, tentunya terdapat fitur dan alur kerja dalam penggunaannya.

#### 3.6.1 Alur Kerja Aplikasi

Aplikasi yang dibangun memiliki fitur untuk membantu pengolahan *review* yang ada pada Twitter. Alur kerja dari aplikasi ini adalah, ketika membuka *website*, *user* akan diarahkan ke *landing page* yaitu home yang sekaligus merupakan halaman panduan dan halaman panduan ini merupakan halaman yang menjelaskan tata cara menggunakan aplikasi analisis sentimen ini. Kemudian apabila ingin melakukan analisis sentimen, *user* dapat masuk ke halaman Analisis dan untuk memulai analisis sentimen, *user* harus sudah memiliki *dataset tweet* yang ingin dianalisis. Dalam *website*, terdapat fitur yang dapat membantu *user* untuk mendapatkan *dataset tweet*. Analisis sentimen dilengkapi opsi yaitu mengambil data dan mengunggah data. Untuk mengambil data, *user* dapat memasukkan *keyword* berupa merek *skincare* dan *range* dari tanggal *tweet* yang ingin dianalisis. Selain itu, apabila *user* sudah memiliki *dataset*, maka terdapat opsi lain yaitu mengunggah data, dimana *user* dapat mengunggah *dataset* dengan beberapa ketentuan, yaitu penamaan kolom dan letak kolom harus sesuai dengan sistem. Hasil akhir dari analisis sentimen adalah persentase dari sentimen positif, netral, dan negatif dan juga

*word cloud* untuk visualisasi kata-kata dari *dataset* yang dianalisis. Berikut adalah gambaran alur dari penggunaan aplikasi analisis sentimen dapat dilihat pada Gambar 3.7.



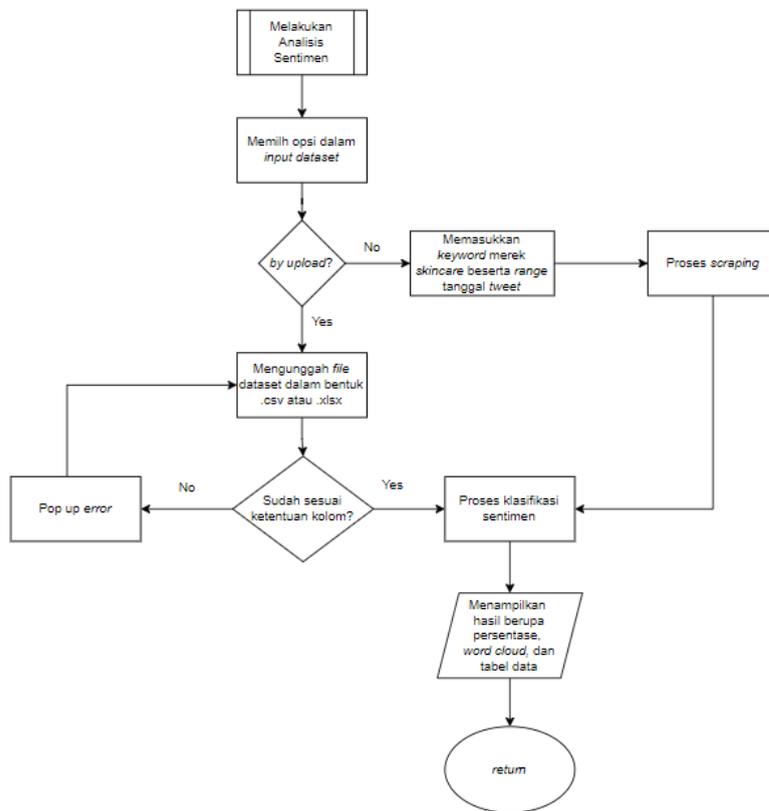
**Gambar 3.7** Alur Kerja Keseluruhan Aplikasi

### 3.6.2 Fitur Melihat Panduan Penggunaan Website

Ketika *user* kebingungan dan membutuhkan panduan dalam penggunaan, *user* dapat masuk ke halaman panduan. Halaman ini akan memberikan tata cara dalam penggunaan aplikasi sehingga nantinya aplikasi dapat digunakan dengan maksimal oleh *user*. Tata cara akan disediakan dalam bentuk tulisan dan gambar penjelasan yang dapat memberikan panduan kepada *user*.

### 3.6.3 Fitur Melakukan Analisis Sentimen

Fitur ini merupakan sistem utama pada aplikasi, dimana fitur ini akan membantu *user* dalam mengolah data *tweet review* merek *skincare* lokal dengan menampilkan hasil pengolahan data kepada *user*. Dalam menggunakan fitur ini, *user* akan diminta untuk memasukkan data dari dua opsi yang sudah dijelaskan pada Sub bab 3.4.1 dan dari data yang dimasukkan maka akan ditampilkan hasil berupa persentase dan *word cloud*. Alur fitur ini dapat dilihat melalui *flowchart* pada Gambar 3.8.



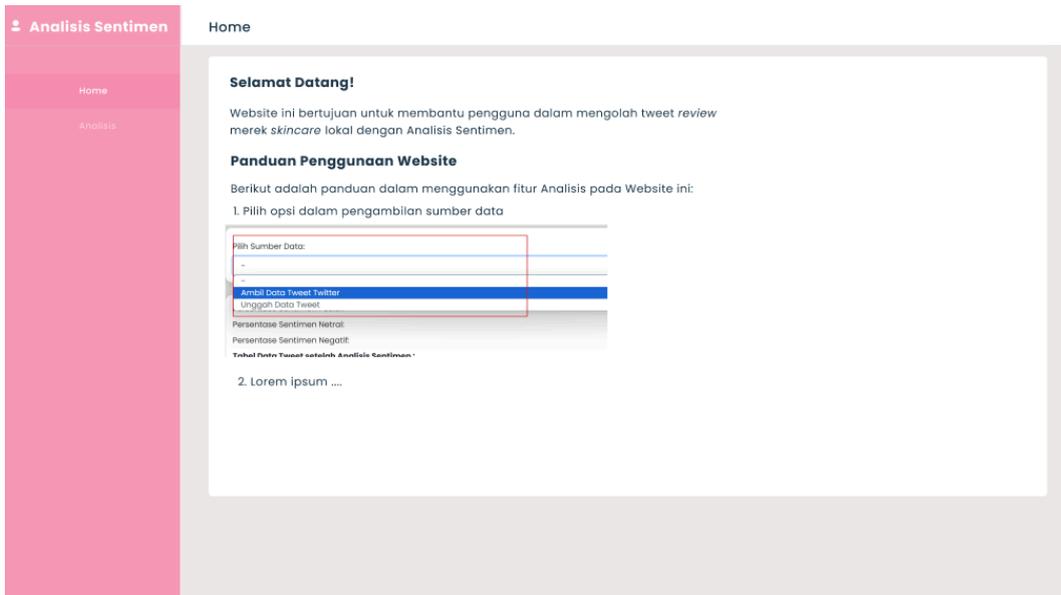
**Gambar 3.8** Flowchart Alur Fitur Melakukan Analisis Sentimen

### 3.7 Desain Tampilan Website

Bagian ini akan menjelaskan rancangan *user interface* dari aplikasi analisis sentimen berbasis *website* yang dibangun. Tampilan aplikasi meliputi halaman Panduan Penggunaan dan halaman Analisis Sentimen.

#### 3.7.1 Desain Halaman Panduan Penggunaan

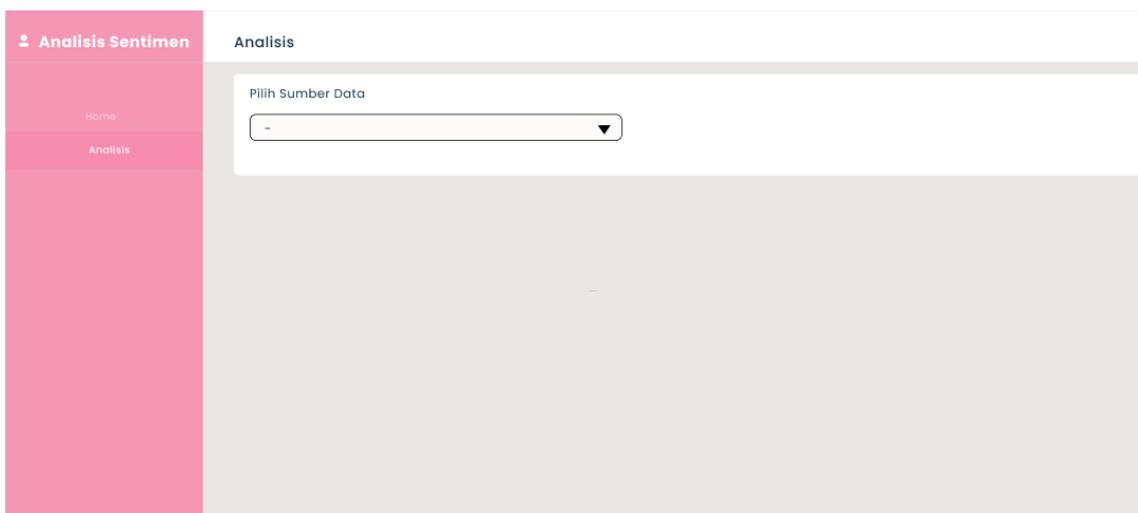
Halaman ini merupakan *landing page* dan pada halaman ini terdapat kalimat sapaan dan panduan dari cara menggunakan *website* analisis sentimen ini. Panduan akan diberikan berupa teks *tutorial* singkat. Desain untuk halaman Panduan Penggunaan dapat dilihat pada Gambar 3.9.



**Gambar 3.9** Desain Tampilan Halaman Utama dari *Website*

### 3.7.2 Desain Halaman Analisis Sentimen

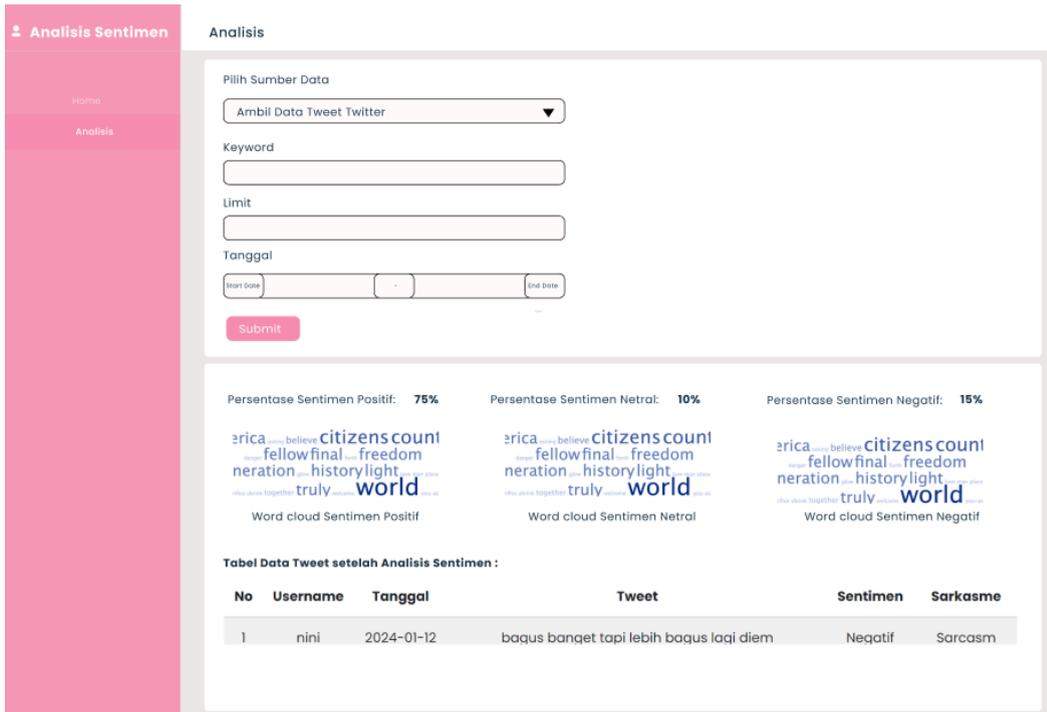
Halaman Analisis Sentimen adalah halaman untuk melakukan analisis sentimen dan merupakan halaman utama fitur program aplikasi. Pada fitur analisis sentimen, pengguna dapat memilih opsi dalam pengambilan sumber data, yaitu Ambil Data Tweet Twitter dan Unggah Data Tweet. Pemilihan sumber data melalui *dropdown* Pilih Sumber Data yang berisi kedua pilihan tersebut, *dropdown* seperti pada Gambar 3.10.



**Gambar 3.10** Desain Tampilan Halaman Analisis dari *Website*

Ketika *user* memilih opsi Ambil Data Tweet, artinya *user* ingin mengambil langsung data *tweet* dari Twitter dengan menggunakan fitur yang telah disediakan. *User* akan memasukkan *keyword* pencarian, limit *tweet* yang akan diambil, *range* tanggal *tweet* dan apabila klik tombol *Submit*,

maka sistem akan melakukan pencarian dan mengambil *tweet* sesuai *input* dari *user*. Hasil akhirnya nanti akan langsung keluar hasil analisis sentimen berupa persentase dan *word cloud* dari masing-masing tipe sentimen. Selain itu, akan ditampilkan tabel yang berisi *tweet review* dan hasil analisis sentimen. Desain tampilan dapat dilihat pada Gambar 3.11.



**Gambar 3.11** Desain Tampilan Opsi Ambil Data *Tweet* Twitter

Kemudian, untuk opsi Unggah Data *Tweet* adalah opsi ketika *user* sudah memiliki *file* data *tweet* yang ingin dianalisis sentimennya. Opsi ini memperbolehkan *user* untuk mengunggah *file* tersebut dengan catatan harus sesuai dengan ketentuan format dari program. *User* dapat melihat ketentuan tersebut dengan klik tulisan biru “Lihat Ketentuan Upload File” yang ada pada tampilan program. Apabila tulisan tersebut di klik, akan keluar *pop up* yang menjelaskan ketentuan-ketentuan *file* yang harus diikuti *user* sebelum diunggah. Apabila sistem mendeteksi adanya ketidaksesuaian format, maka proses analisis sentimen tidak dapat dilakukan dan *user* akan menerima *pop up* pemberitahuan bahwa format dari *file* yang diunggah belum sesuai dengan ketentuan sistem. Setelah berhasil mengunggah, sama seperti opsi sebelumnya, akan ada *output* dari hasil analisis sentimen berupa persentase sentimen dan *word cloud* dari masing-masing tipe sentimen. Selain itu, pada opsi ini juga ditampilkan tabel dari *dataset tweet review* dan hasil analisis sentimen. Desain tampilan dapat dilihat pada Gambar 3.12.

**Analisis Sentimen**

Home

Analisis

### Analisis

Pilih Sumber Data

Unggah Data Tweet
▼

[Lihat Ketentuan Upload File](#)

Unggah File Data Tweet Twitter

Choose File

Submit

---

Persentase Sentimen Positif: **75%**



Word cloud Sentimen Positif

Persentase Sentimen Netral: **10%**



Word cloud Sentimen Netral

Persentase Sentimen Negatif: **15%**



Word cloud Sentimen Negatif

**Tabel Data Tweet setelah Analisis Sentimen :**

No	Username	Tanggal	Tweet	Sentimen	Sarkasme
1	nini	2024-01-12	baqus banqet tapi lebih baqus laqi diem	Negatif	Sarcasm

**Gambar 3.12** Desain Tampilan Opsi Unggah Data *Tweet*