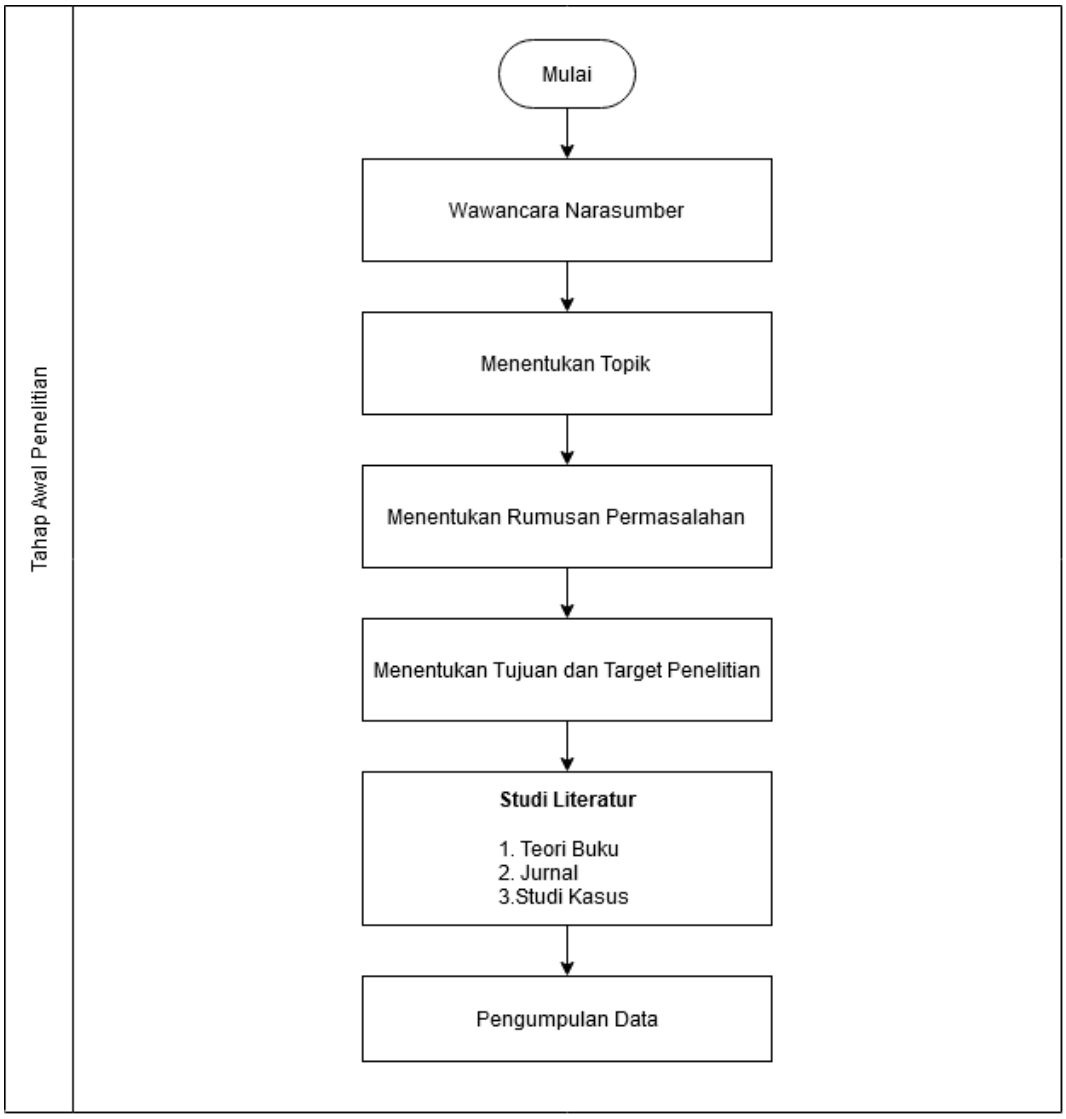
BAB 3

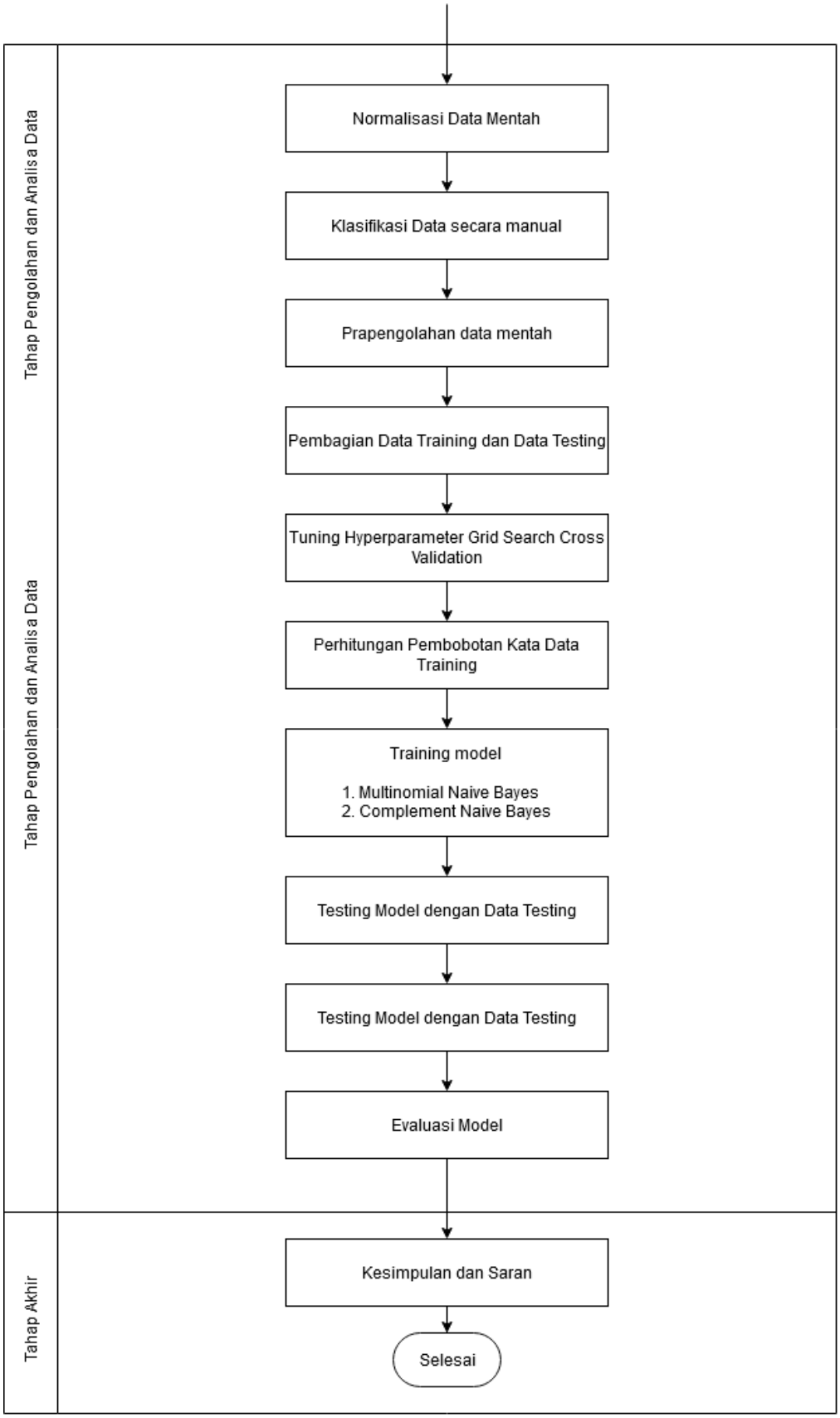
METODE PENELITIAN

Bab ini berisikan rancangan jalannya penelitian, yakni jalannya sistem, kebutuhan sistem, dan metode pengujian terhadap data yang telah didapatkan.

3.1 Diagram Penelitian

Tahapan proses yang akan dilakukan dalam penelitian ini digambarkan dalam diagram penelitian pada Gambar 3.1 sebagai berikut:





Gambar 3.1 Gambar Diagram Penelitian

3.2 Metodologi Penelitian

Metodologi penelitian yang digunakan dalam penelitian ini adalah sebagai berikut:

1. Wawancara

   Wawancara dilakukan guna untuk mengetahui pendapat praktisi psikolog mengenai penyakit mental khususnya depresi. Pendapat ini kemudian akan digunakan sebagai pedoman dalam menyelesaikan permasalahan yang ada, dalam hal ini mengklasifikasi *tweet* pada aplikasi Twitter ke dalam kategori terindikasi depresi. Wawancara dilakukan secara daring dengan praktisi psikolog yang bernama Patricia Astrid Nadia, S.Psi.

2. Studi Literatur

   Dalam tahap ini, dilakukan pembelajaran dan pendalaman teori-teori terkait pengambilan sampel melalui Twitter *API*, penyakit mental depresi, *text mining*, algoritma *Naïve Bayes*, *confusion matrix*, dan *Grid-Search Cross Validation*. Sumber informasi didapatkan melalui penelitian terlebih dahulu, buku, jurnal, referensi, *website* yang terkait dengan penelitian yang akan dilakukan.

3. Perancangan Sistem

   Secara garis besar, sistem dapat mengklasifikasikan *tweet* ke dalam kategori terindikasi depresi dan tidak terindikasi depresi. Untuk mencapai hasil ini, terdapat beberapa proses yang perlu dilakukan adalah sebagai berikut:

a. *Crawling* Twitter *Data*

Crawling Twitter *Data* dilakukan dengan cara mengkonsumsi *API* yang disediakan oleh Twitter. Proses *crawling twitter data* terdiri dari proses sebagai berikut:

i. Membuat autentikasi agar Twitter dapat mengenali *user* yang melakukan penarikan data.

ii. Membuat *class* yang akan digunakan sebagai pembungkus atau *container* dari fungsi-fungsi untuk menarik data.

iii. Membuat *Regular Expression* (RegEx) yang digunakan oleh sistem untuk meminimalisir terbentuknya data yang abstrak. Metode ini dilakukan guna untuk menghilangkan simbol, angka, *emoji*, dan *url*.

iv. Menjalankan fungsi penarikan data yang disertakan dengan kata kunci yang memiliki konotasi negatif atau gaya bahasa yang sering digunakan oleh penderita penyakit mental depresi (contoh: menyedihkan, kesepian, sengsara, sakit, diriku sendiri, tidak pernah, dll.) agar data yang diambil oleh sistem memiliki keterkaitan dengan kebutuhan data dari penelitian. Penarikan data dilakukan pada Pulau Jawa, Bali, dan Sumatera.

v. Memasukkan data ke dalam bentuk *Comma Separated Value* (CSV) guna untuk diproses lebih lanjut agar data menjadi lebih relevan.

Tabel 3.1 Contoh Data *Tweet Training*

Kelas	*Tweet*
Terindikasi Depresi	“aku tidak kehilangan seorang teman pun aku hanya baru sadar kalau tidak pernah memilikinya”
	“kesalahan pertamaku adalah mengenalmu kesalahan kedua aku adalah jatuh cinta padamu dan kesalahan aku berikutnya adalah aku tidak pernah bisa melupakanmu”
	“mungkin selama ini kau tidak pernah anggap aku sebagai kekasihmu”
	“kamu tidak tau pernah seperduli apa aku hingga aku menjadi secuek ini”
	“aku tidak pernah sekalipun menyesali diamku tetapi aku berkali-kali menyesali bicaraku”
Tidak Terindikasi Depresi	“cinta itu tidak pernah sendiri cinta selalu mendua karena cinta disini adalah aku dan kamu”
	“yang selalu sayang sama aku tidak pernah ngarepin balasan apapun”
	“buat aku yang pernah naksir teman sendiri”
	“aku memang pasrah karena jarak tapi percayalah aku tidak akan pernah menyerah”
	“sakit kepala tidak pernah datang tepat waktu”

b. *Text Preprocessing*

Text-preprocessing merupakan proses pengolahan data yang belum terstruktur atau abstrak menjadi data yang terstruktur). Proses yang dilakukan adalah sebagai berikut:

i. Normalisasi

Normalisasi dilakukan secara manual dengan mengubah data *tweet* (kata yang memiliki arti yang sama namun beda penulisan, typo, dll.) untuk menghindari abstraksi pada data yang digunakan. Pengubahan kata didasari oleh daftar kata baku yang terdapat pada Kamus Besar Bahasa Indonesia (KBBI).

Tabel 3.2 Kata Baku

Kata Baku			
Bentuk non standar 1	**…**	**Bentuk non standar 8**	**Bentuk standar (baku)**
Knp	Napa	-	kenapa
Bpk	Bp	Bokap	bapak
gue	Gua	w	Aku
elo	Ente	Lo	Kamu

Berikut merupakan contoh normalisasi kata pada *tweet* yang belum mengalami tahapan *preprocessing data* lainnya:

Tabel 3.3 Contoh Normalisasi Pada Data *Tweet* Awal

Kata tidak baku		**Kata Baku**
Pak		Bapak
Pengen		Mau
Bikin		Buat
Utk		Untuk
dgn	→	Dengan
Mending		Agak
Yuk		Ayo
aja		Saja

ii. *Case Folding*

Case folding dilakukan untuk mengubah semua huruf menjadi huruf kecil dalam dokumen, menghilangkan *whitespace*, dilakukan dengan menggunakan fungsi yang disediakan Python yaitu *lower, punctuation,* dan *split*.

iii. *Tokenizing*

Tokenizing dilakukan untuk memisahkan tiap kata dalam dokumen, dilakukan dengan menggunakan fungsi pada *library NLTK*, yaitu *word_tokenize*.

iv. *Filtering*

Filtering dilakukan untuk menghilangkan kata-kata yang tidak penting, dilakukan dengan menggunakan *stopwords* yang telah disediakan oleh *library NLTK*.

v. *Stemming*

Stemming merupakan pengubahan kata-kata dalam dokumen menjadi bentuk dasarnya, dilakukan dengan menggunakan fungsi yang disediakan oleh *library* Sastrawi, yaitu stem.

c. *Term Frequency – Inverse Document Frequency* (TF-IDF)

TF-IDF merupakan tahapan untuk melakukan pembobotan setiap kata dalam kalimat yang akan digunakan sebagai acuan dalam training, dilakukan dengan menggunakan fungsi yang disediakan oleh *library scikit-learn*, yaitu *TfidfVectorizer*.

d. *Training*

Training merupakan pembelajaran sistem dengan menggunakan pembobotan nilai yang dihasilkan TF-IDF dan N-Gram.

e. *Testing*

Testing merupakan proses menguji hasil pelatihan.

f. Klasifikasi Data

Klasifikasi Data merupakan tahapan untuk mengkategorikan setiap *tweet* ke dalam kategori terindikasi depresi dan tidak terindikasi depresi.

Setelah mendapatkan informasi yang dibutuhkan untuk melakukan penelitian, perancangan sistem dapat dibangun dengan membuat gambaran umum sistem, pembuatan *flowchart*, dan implementasi algoritma *Naïve Bayes* terhadap informasi atau data yang dimiliki.

4. Implementasi Sistem

Tahap ini merupakan tahapan implementasi dari tahapan yang telah dibangun pada tahap-tahap sebelumnya, dilakukan dengan menggunakan bahasa pemrograman Python.

5. Pengujian dan Evaluasi

Setelah sistem selesai dibangun, dilakukan implementasi pengujian terhadap implementasi algoritma *Multinomial Naïve Bayes* dan *Complement Naïve Bayes* untuk mengklasifikasikan *tweet*. Data yang digunakan untuk proses pengujian merupakan data yang telah dilabel berdasarkan panduan dari praktisi psikolog terkait melakukan diagnosa indikasi penyakit mental depresi. Pengujian dilakukan dengan melakukan validasi dengan mencari *parameter* terbaik, *f1_micro*, serta *confusion*

matrix dari algoritma *Multinomial Naïve Bayes* dan *Complement Naïve Bayes* dalam melakukan klasifikasi *tweet* yang belum terstruktur.

6. Dokumentasi

   Tahap ini merupakan tahapan dalam membuat laporan sebagai bentuk dokumentasi dari sistem yang telah dibangun, dokumentasi dilakukan secara bertahap serta penelitian mendalam mengenai implementasi algoritma *Multinomial Naïve Bayes* dan *Complement Naïve Bayes* dalam melakukan klasifikasi *tweet* yang belum terstruktur.

3.3 Teknik Pengumpulan Data

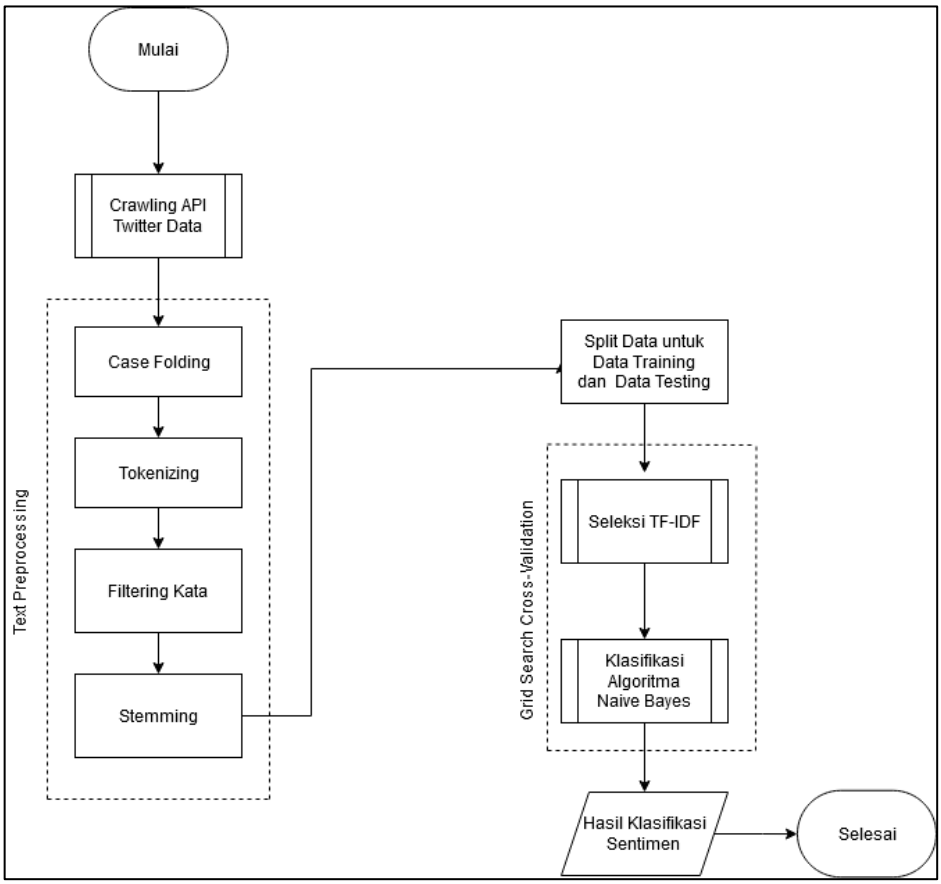
Pengumpulan data untuk mengetahui permasalahan yang ada dan kebutuhan publik akan sistem yang dapat mengklasifikasikan *tweet* yang memiliki indikasi penyakit mental depresi dilakukan dengan cara wawancara secara daring dengan praktisi psikolog, Patricia Astrid Nadia, S.Psi.

Data yang dibutuhkan untuk tahap *training* dan *validation* merupakan data *tweet* yang sudah diberi label menurut pedoman atau arahan dari praktisi psikolog. Untuk data *tweet* yang sesuai dengan pedoman yang diberikan akan dikategorikan sebagai data *tweet* dengan label “terindikasi depresi”. Untuk tweet yang tidak termasuk ke dalam pedoman yang diberikan, akan dikategorikan sebagai data *tweet* dengan label “tidak terindikasi depresi”.

Komposisi jumlah data *training* dan *testing* masing-masing kategori memiliki peranan yang penting karena bisa memengaruhi hasil perhitungan akurasi, *f-score*, *confusion matrix*. Pembagian data akan dibahas lebih lanjut pada pengujian di Bab 4.

3.4 Gambaran Umum Sistem

Sistem yang dibangun merupakan sistem analisa untuk menghitung akurasi, *f-score*, *confusion matrix* terhadap data yang telah diberikan label dengan mengimplementasikan algoritma *Multinomial Naïve Bayes* dan *Complement Naïve Bayes*. Sistem ini berfungsi untuk mengklasifikasikan data *tweet* ke dalam kategori "terindikasi depresi" dan "tidak terindikasi depresi". Kategori "terindikasi depresi" digunakan untuk data *tweet* yang termasuk ke dalam pedoman dari praktisi psikolog, sedangkan "tidak terindikasi depresi" digunakan untuk data *tweet* yang tidak termasuk ke dalam pedoman dari praktisi psikolog. Model sistem analisa digambarkan pada Gambar 3.2.



Gambar 3.2 Gambaran Umum Sistem Klasifikasi *Tweet*

Sistem diawali dengan mendaftarkan diri kepada pihak Twitter agar mendapatkan Twitter *API* yang akan digunakan pada tahap *Crawling* Twitter *Data* dapat dilihat pada Gambar 3.1. Twitter *API* kemudian akan digunakan oleh sistem sebagai autentikasi dalam melakukan penarikan data, setelah sistem dapat mengenali autentikasi yang digunakan, maka sistem dapat melakukan penarikan data dengan menggunakan fungsi yang sudah dibangun, fungsi ini berguna untuk menarik data dari daerah dan radius (satuan Kilometer) dan fungsi untuk mengurangi abstraksi data (menggunakan *Regular Expression*). Setelah data berhasil ditarik oleh sistem, sistem akan memasukkan data-data hasil penarikan tersebut ke dalam suatu dokumen sesuai dengan nama daerahnya.

Secara garis besar, setelah melakukan penarikan data atau *Crawling Data* Twitter, akan dilakukan tahap *text preprocessing* yang terjadi pada tahap *text preprocessing* adalah mengubah kata tidak baku menjadi kata baku (normalisasi), mengubah setiap huruf menjadi huruf kecil, menghilangkan *whitespace*, dan tanda baca (*case folding*) dengan menggunakan fungsi yang disediakan Python, yaitu *lower*, pemisahan setiap kata dalam kalimat (*tokenizing*) dengan menggunakan *library NLTK*, menghilangkan kata tidak penting (*filtering*) dengan menggunakan *library NLTK* dan Bahasa Indonesia sebagai acuan dari sistem dalam melakukan *filtering*, dan proses terakhir yang dilakukan oleh sistem adalah mengubah kata menjadi bentuk dasar (*stemming*) dengan menggunakan fungsi yang disediakan *library* Sastrawi pada StemmerFactory, yaitu stem (Nugroho, 2019; Robani, 2018).

Setelah sistem melakukan proses *text preprocessing*, maka akan dilanjutkan proses pembobotan kata dengan menggunakan *Term Frequency – Inverse Document Frequency* (TF-IDF). Pada saat yang bersamaan, akan dijalankan juga

Grid Search Cross Validation sebagai alat bantu sistem dalam menentukan *parameter* terbaik dalam melakukan pembobotan kata terhadap data yang dimiliki agar hasil yang diberikan oleh sistem merupakan hasil yang paling optimal.

Tahap selanjutnya adalah proses *training*, di mana sistem akan dilatih menggunakan hasil pembobotan kata yang dilakukan pada tahap sebelumnya, proses pelatihan ini dilakukan untuk mencari probabilitas kemunculan kata dan peluangnya pada suatu data *tweet.* Hasil *training* ini kemudian akan digunakan pada tahap selanjutnya sebagai acuan sistem dalam melakukan klasifikasi data.

Setelah hasil *training* sistem telah didapatkan, maka proses *testing* terhadap data yang belum terstruktur akan dijalankan oleh sistem, sistem akan menggunakan perhitungan *Naïve Bayes* dan hasil *training* dalam menentukan klasifikasi dari data *tweet*. Proses pengklasifikasian data ini akan dilakukan dengan mencari perhitungan tertinggi dari setiap kategori yang ada, pada penelitian ini terdapat dua kategori, yaitu terindikasi depresi dan tidak terindikasi depresi.

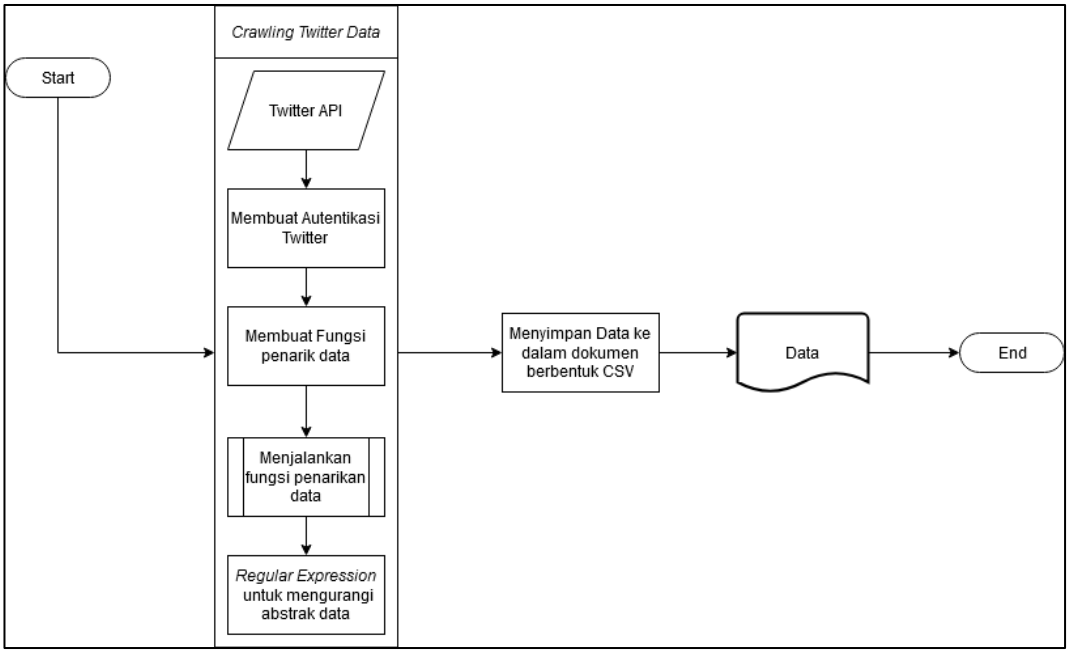
3.5 Perancangan Sistem

Sistem yang dibangun merupakan sistem analisa dengan menggunakan bahasa pemrograman Python. Sistem dirancang untuk melakukan penarikan data melalui Twitter *API*, melakukan *text-preprocessing* terhadap data mentah yang diambil oleh sistem, melakukan *training* terhadap data yang telah dibersihkan atau dilakukan *text-preprocessing* dan menjadikan hasil *training* sebagai acuan sistem dalam melakukan *testing* terhadap data yang belum dilakukan klasifikasi. Perancangan sistem ini teridiri dari *flowchart* untuk menjelaskan alur sistem dalam menganalisa dan mengklasifikasikan data.

3.5.1 Flowchart Sistem Analisa

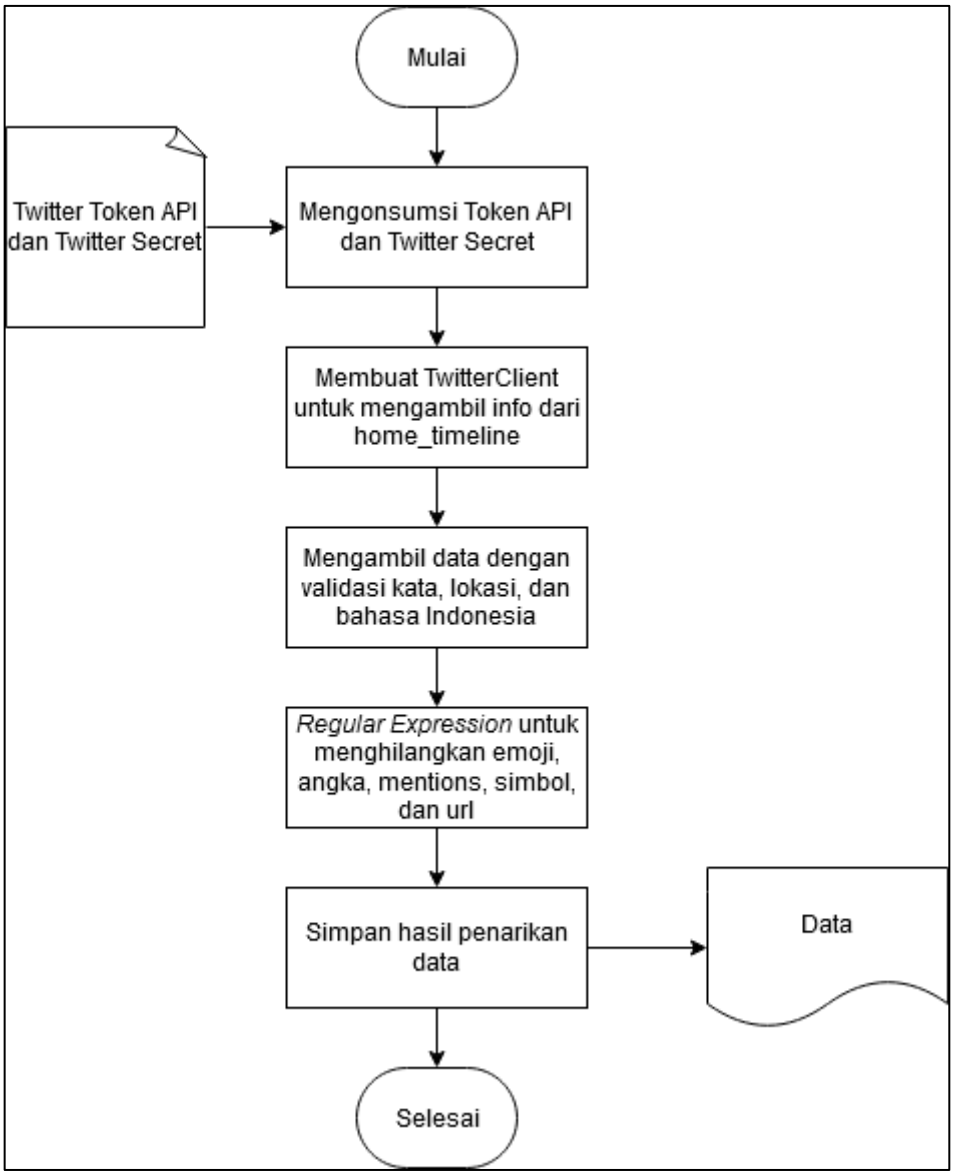
Gambar 3.2 di atas merupakan alur sistem dalam melakukan penarikan data secara keseluruhan, *training* dan *testing* dalam melakukan klasifikasi *tweet* dilakukan dengan menggunakan algoritma *Multinomial Naïve Bayes* dan *Complement Naïve Bayes*.

Sistem diawali dengan melakukan penarikan data melalui aplikasi Twitter, proses ini merupakan salah satu proses inti sistem, karena data yang dibutuhkan dalam penelitian merupakan *tweet* yang dibuat oleh pengguna aplikasi Twitter. Proses penarikan data dapat dilihat pada Gambar 3.3.



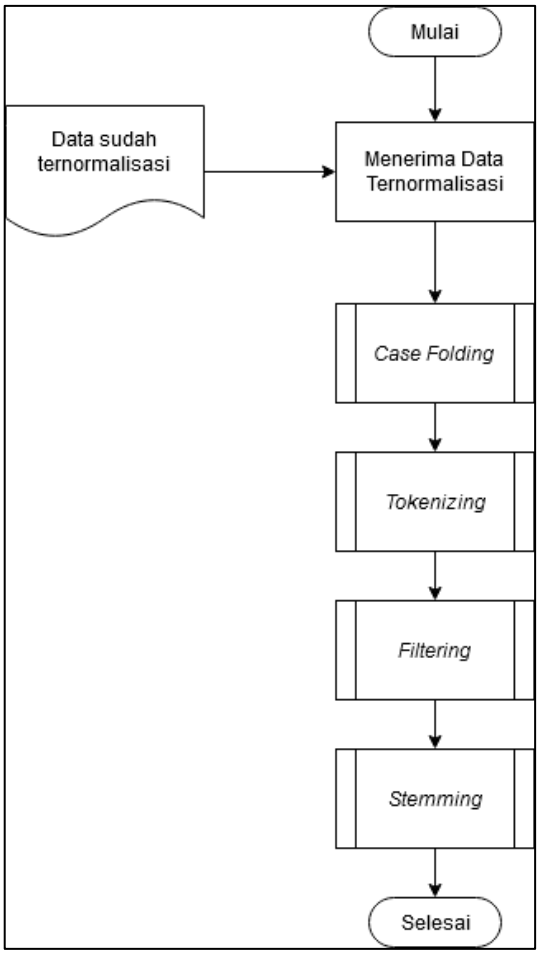
Gambar 3.3 *Crawling* Twitter *Data*

Setelah Twitter melakukan autentikasi terhadap *API* dan *Token* yang digunakan, sistem akan menjalankan fungsi untuk melakukan penarikan data. Fungsi penarikan data dapat dilihat pada Gambar 3.4.



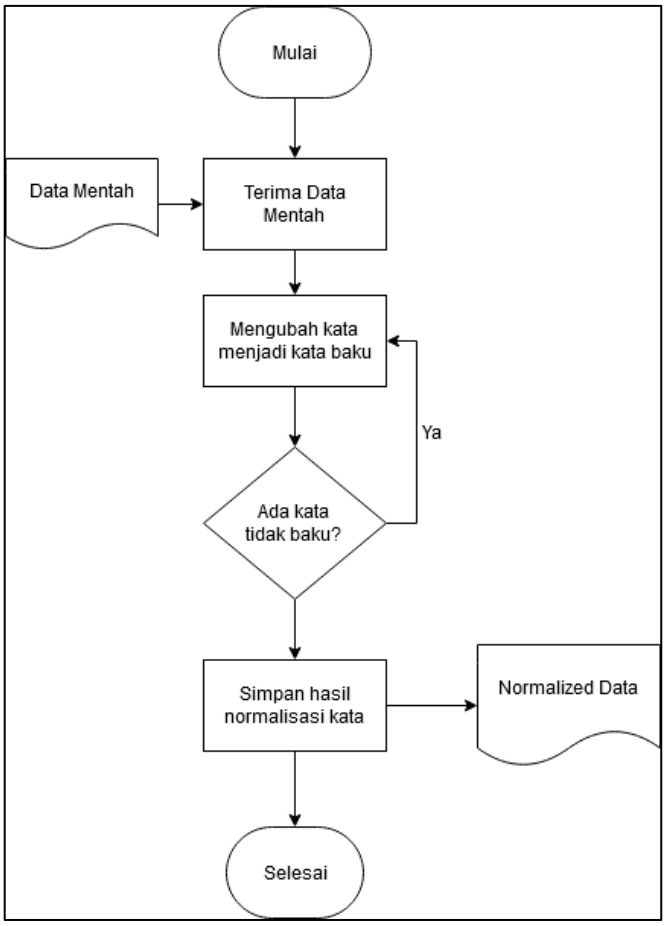
Gambar 3.4 Fungsi Penarikan Data

Pada modul *text-preprocessing*, terdapat empat modul besar yang memiliki peranan penting dalam melakukan transformasi data mentah menjadi data bersih. Modul-modul dalam *text-preprocessing* antara lain: *case folding*, *tokenizing*, *filtering*, dan *stemming*. Proses *text-preprocessing* dapat dilihat pada Gambar 3.5.



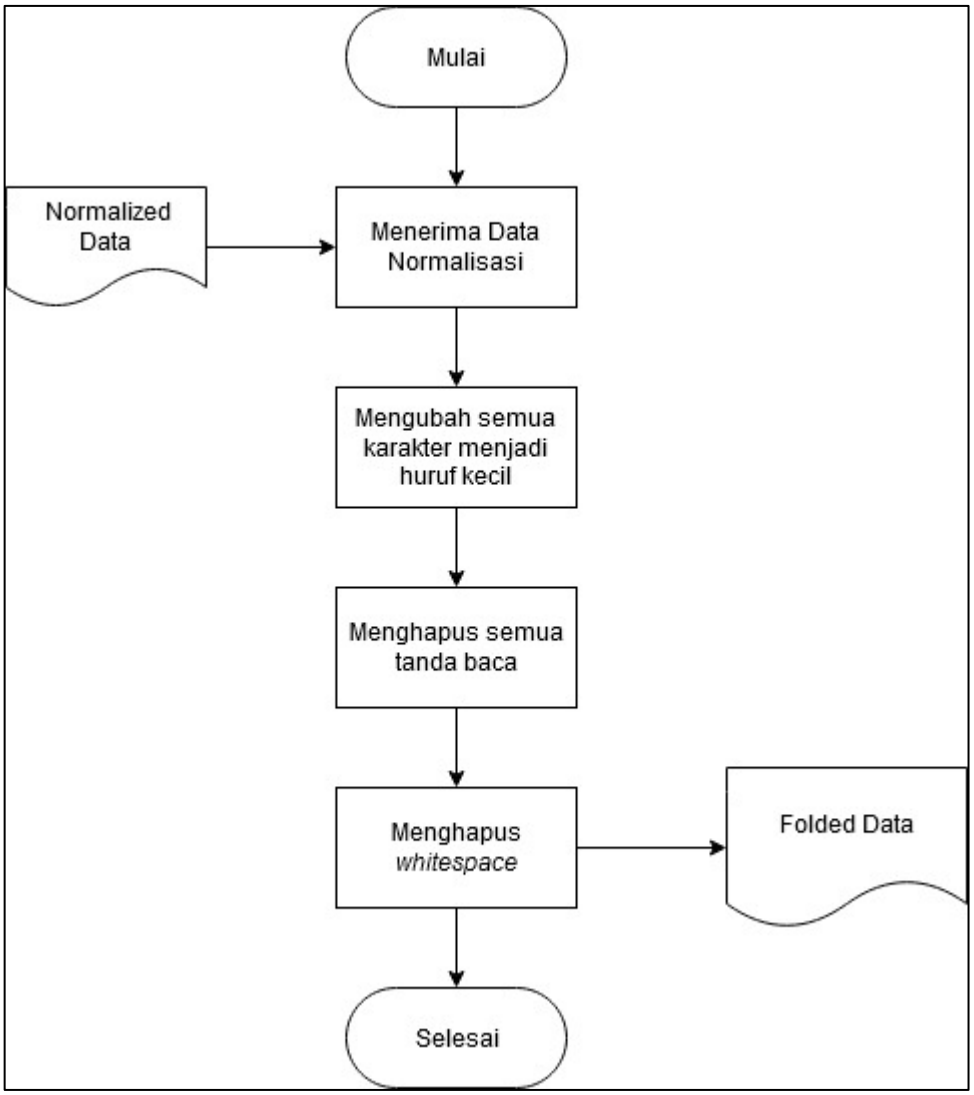
Gambar 3.5 *Flowchart* Modul *Text Preprocessing*

Pada tahap normalisasi, dilakukan pengubahan kata secara manual yang memiliki makna yang sama tetapi memiliki perbedaan penulisan, dilakukan dengan cara manual, yaitu mengubah kata pada dokumen yang telah dilakukan *Regular Expression*. Proses normalisasi data dilihat pada Gambar 3.6.



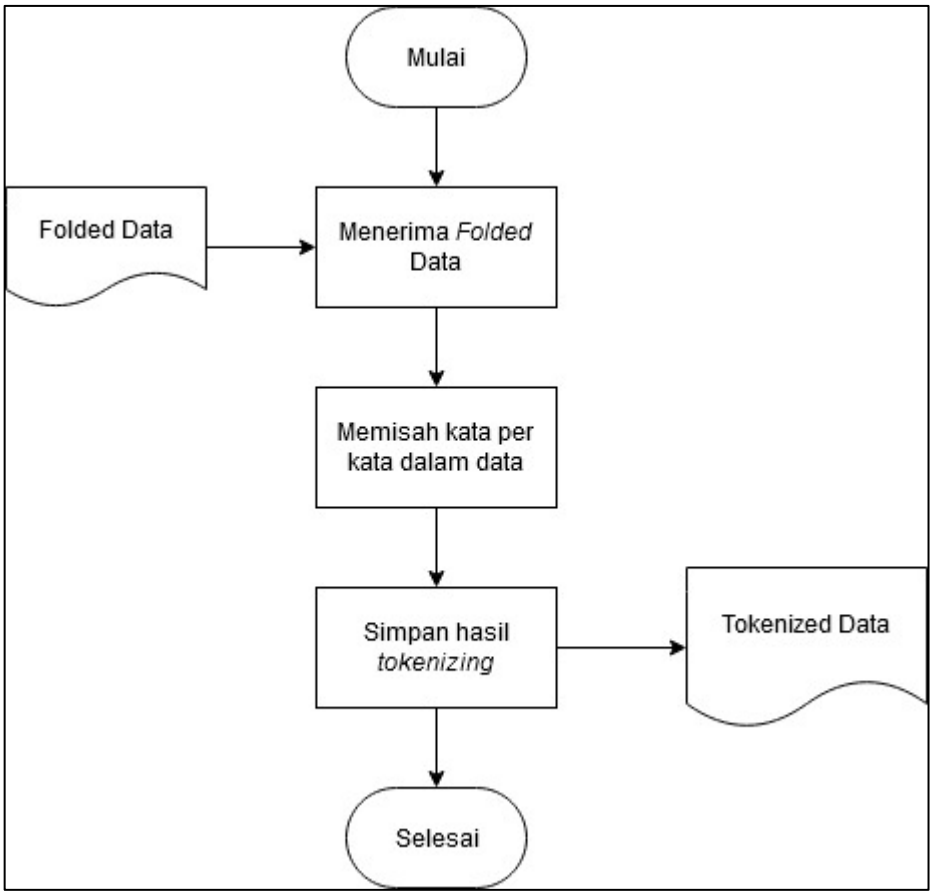
Gambar 3.6 *Flowchart* Proses Normalisasi

Pada modul *case folding* dilakukan proses transformasi data yang telah melewati tahap *Regular Expression* dalam menghilangkan beberapa abstraksi data (contoh: emoji, angka, dll.). Proses *case folding* dilakukan dengan menggunakan fungsi *lower*, *punctuation*, dan *split*. Proses *case folding* dapat dilihat pada Gambar 3.7.



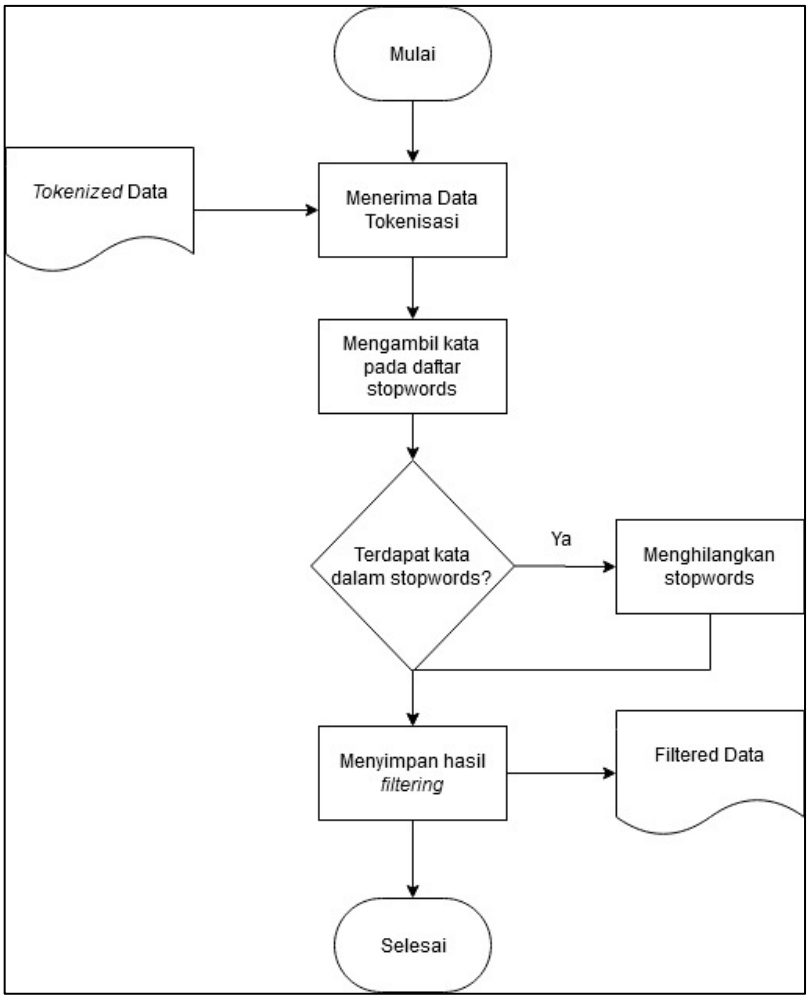
Gambar 3.7 *Flowchart* Modul *Case Folding*

Selanjutnya, dilakukan model *tokenizing*, dilakukan pemisahan kata per kata dalam dokumen agar lebih mudah untuk diubah pada modul *filtering* dan *stemming*. Proses *tokenizing* dapat dilihat pada Gambar 3.8.



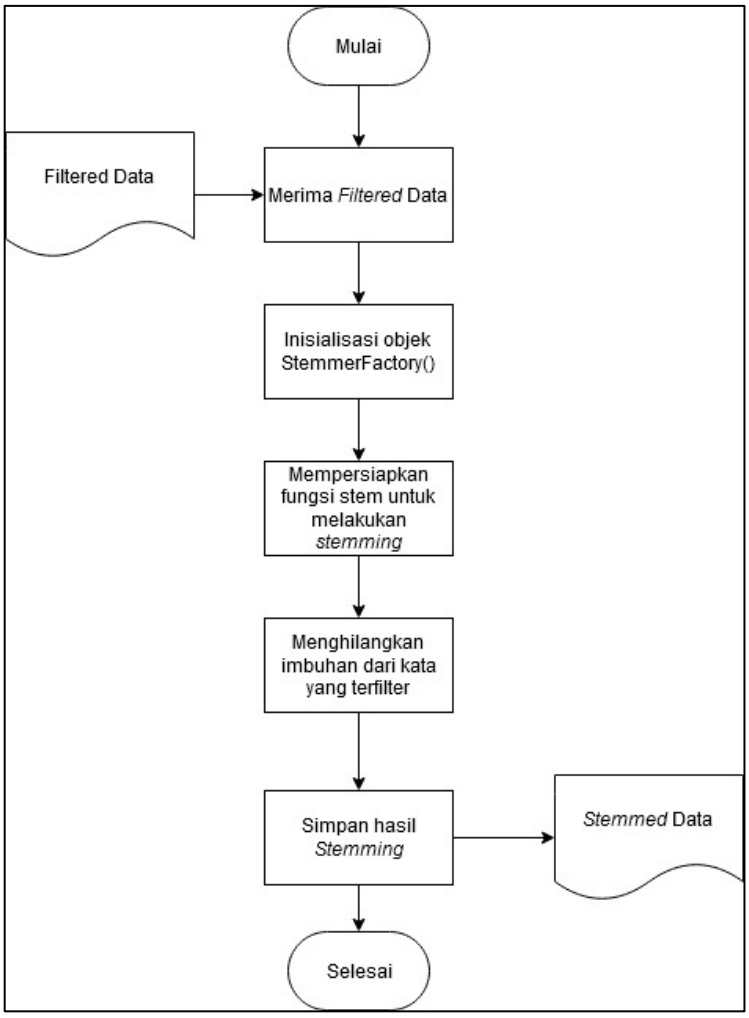
Gambar 3.8 *Flowchart* Modul *Tokenizing*

Setelah proses *tokenizing* berhasil dilewati, *text preprocessing* akan dilanjutkan dengan modul *filtering*, proses *filtering* dilakukan dengan menggunakan *stopword* yang telah disediakan oleh *library* NLTK. Proses ini dilakukan guna menghapus kata-kata yang tidak penting untuk mengurangi variasi kata. Proses dapat dilihat pada Gambar 3.9.



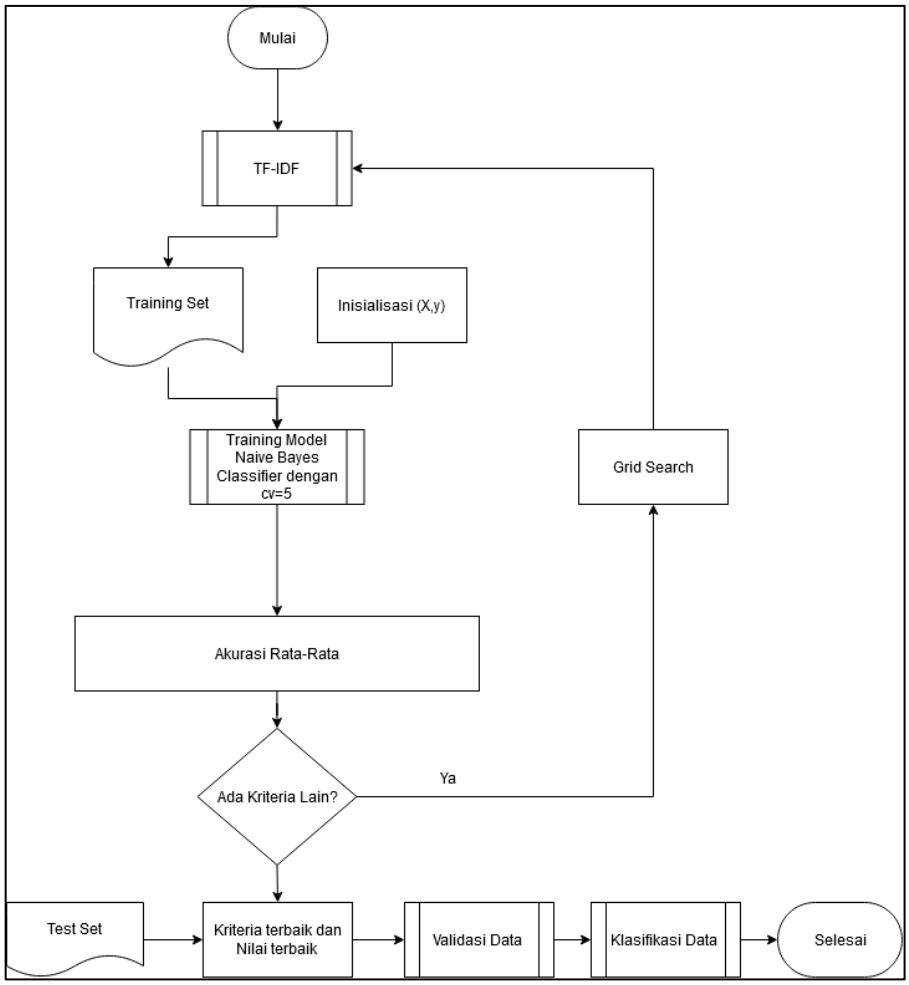
Gambar 3.9 *Flowchart* Modul *Filtering*

Langkah terakhir pada modul *text preprocessing* merupakan proses *stemming*. *Stemming* merupakan proses mengubah token data ke bentuk paling dasar dari kata tersebut. Proses ini menggunakan *library* Sastrawi. Diawali dengan melakukan inisialisasi terhadap objek StemmerFactory(). Sama seperti tahap *filtering*, sistem akan melakukan pengecekkan terhadap token, apabila terdapat kata yang sama dalam objek StemmerFactory(), imbuhan dari kata tersebut akan dihapus, agar menjadi bentuk dasar dari kata itu sendiri. Proses *Stemming* dapat dilihat pada Gambar 3.10.



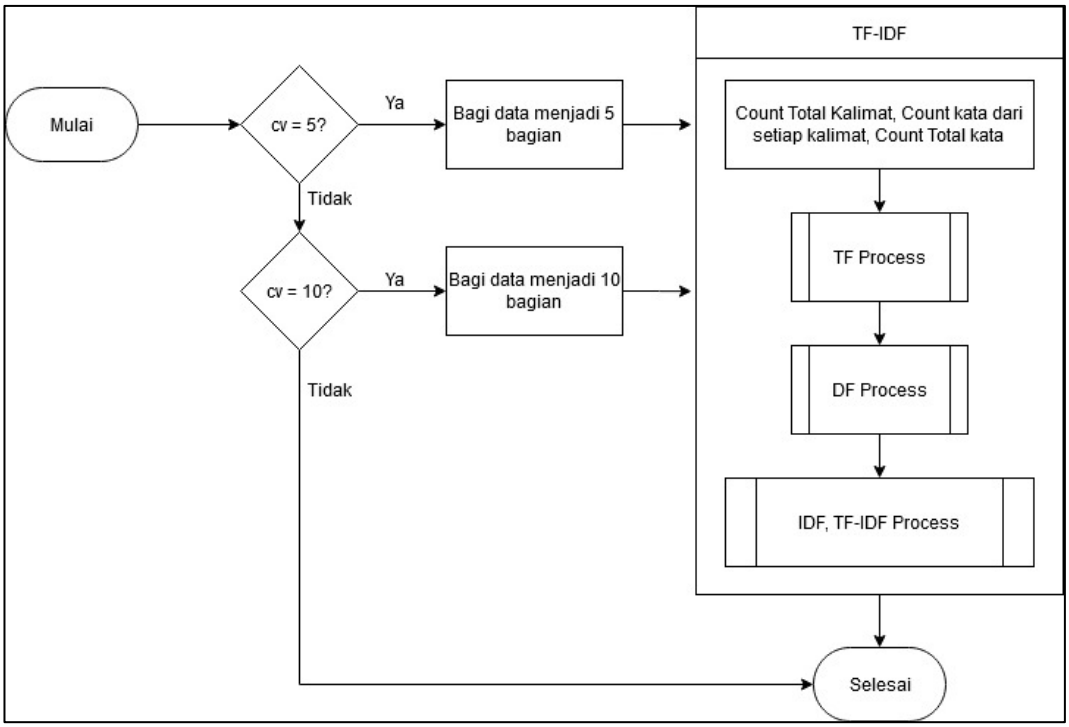
Gambar 3.10 *Flowchart* Modul *Stemming*

Setelah melakukan proses *text-preprocessing*, sistem akan menjalankan *Training* dengan menjalankan *Grid-Search Cross-Valdiation*. Hal ini dijalankan guna mencari *parameter* terbaik dari *Term Frequency – Inverse Document Frequency* (TF-IDF) yang dilanjutkan dengan klasifikasi dengan mengimplementasikan algoritma *Multinomial Naïve Bayes* dan *Complement Naïve Bayes*. Pemberian nilai Cross Validation (CV), didasari oleh jumlah data yang dimiliki, teknik yang dilakukan CV adalah data training (CV – 1) dan data testing adalah 1 dan melakukan penilaian secara rata-rata sebanyak n. Proses *Grid Search Cross Validation* dapat dilihat pada Gambar 3.11.



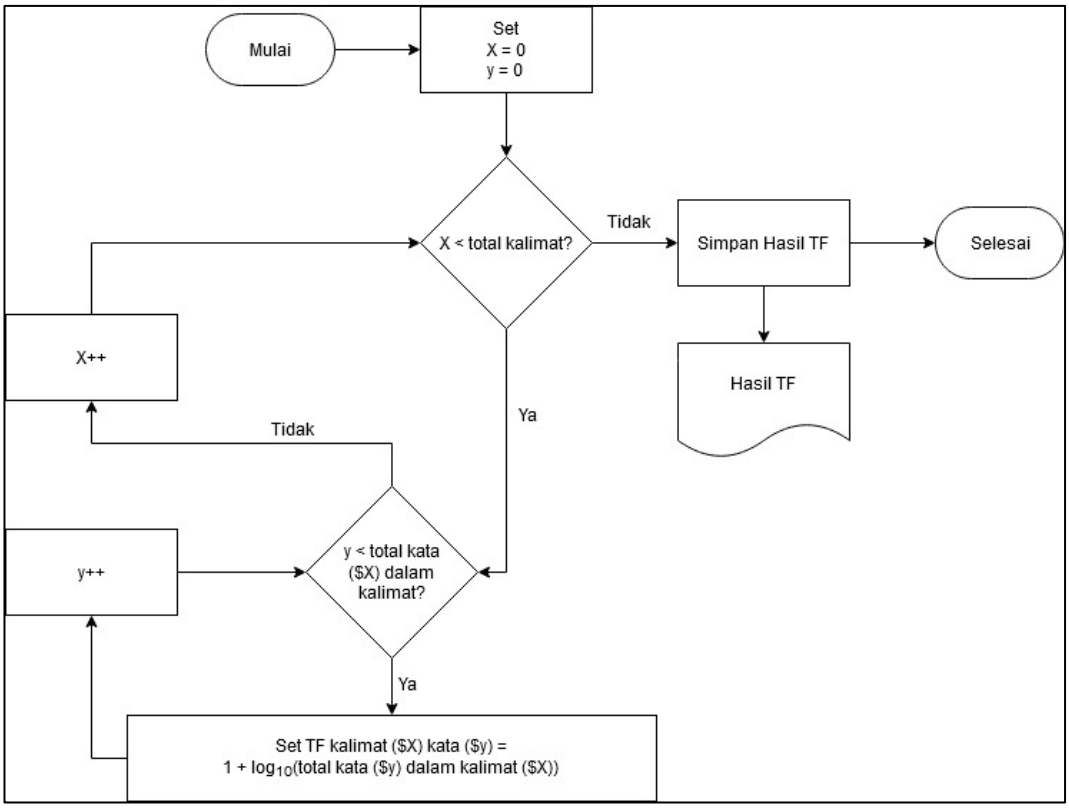
Gambar 3.11 *Flowchart* Modul *Grid-Search Cross Validation*

Setelah proses *text preprocessing* selesai dilakukan, hasil dari tahap tersebut akan dilanjutkan ke tahap TF-IDF yaitu proses pembobotan setiap kata, pembobotan kata ini akan dilakukan dengan menggunakan bantuan *cross validation* pada *library scikit-learn* untuk mencari parameter terbaik dalam melakukan pembobotan kata, TF-IDF. Proses TF-IDF dapat dilihat pada Gambar 3.12.



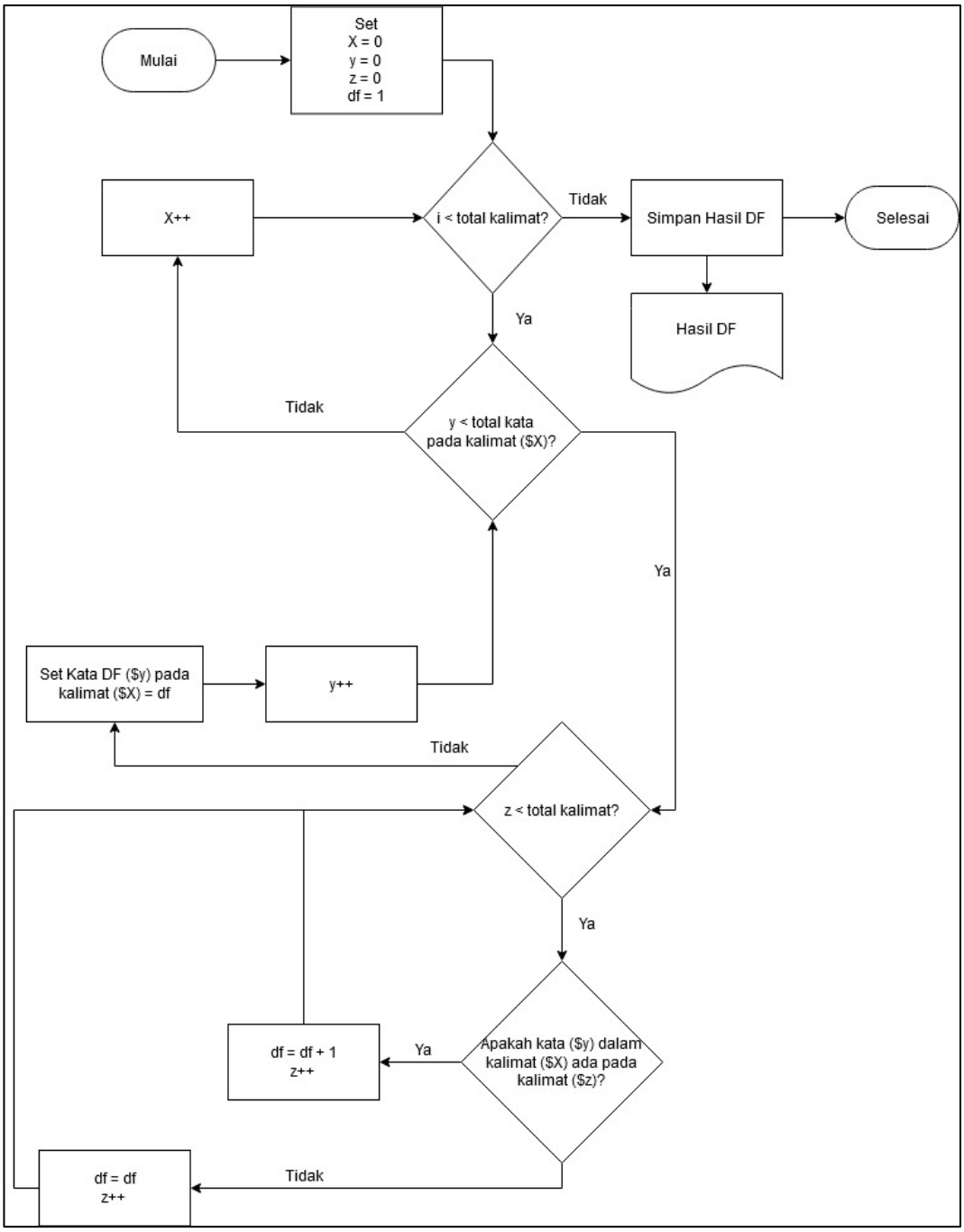
Gambar 3.12 *Flowchart TF-IDF*

Pada *flowchart* TF-IDF, terdapat modul proses *Term Frequency* (TF), modul ini dilakukan untuk mencari jumlah kemunculan sebuah kata di setiap kalimat yang ada didalam suatu dokumen. Jumlah kemunculan kata ini kemudian akan dianggap sebagai nilai TF dari kata tersebut.proses ini akan diulang sampai setiap kata dalam setiap kalimat memiliki nilai TF. Proses *Term Frequency* dapat dilihat pada Gambar 3.13.



Gambar 3.13 *Flowchart* Modul *Term Frequency*

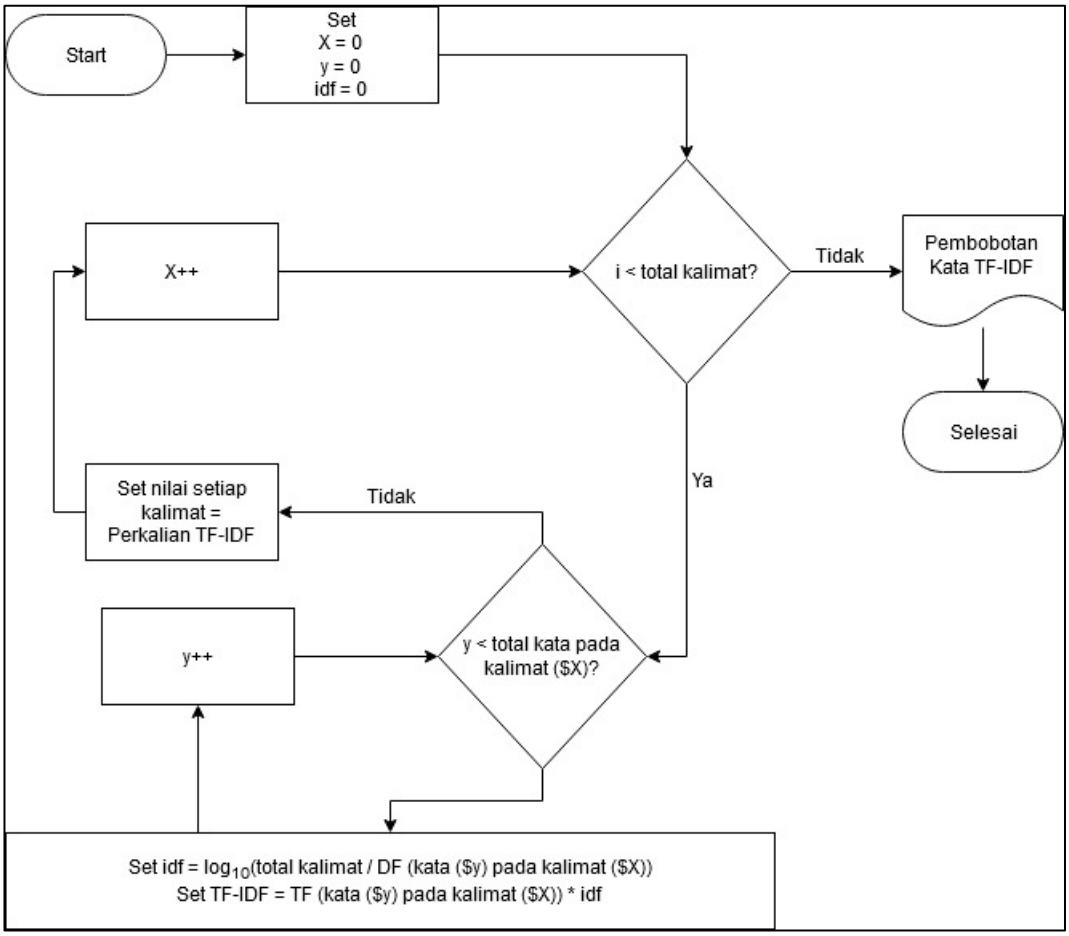
Setelah melakukan proses pembobotan pada proses *Term Frequency*, selanjutnya sistem akan menjalankan *Document Frequency* (DF) yang bertujuan untuk mencari jumlah kemunculan kata di setiap kalimat yang ada pada teks asli. Nilai DF akan ditentukan dari jumlah kemunculan kata pada kalimat-kalimat yang ada. Seperti pada proses TF, proses DF akan dilakukan sampai setiap kata memiliki nilai DF. Proses pembobotan DF dapat dilihat pada Gambar 3.14.



Gambar 3.14 *Flowchart* Proses *Document Frequency*

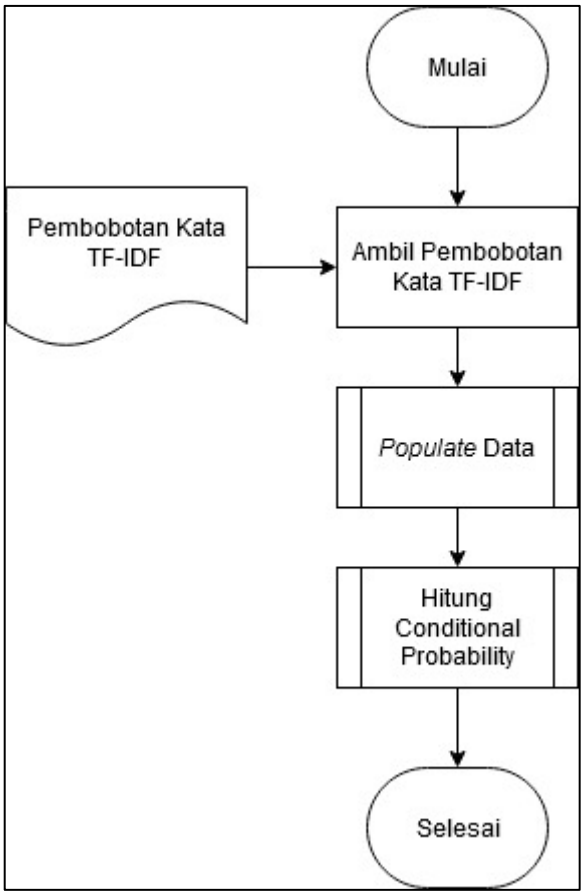
Setelah selesai melakukan pembobotan pada tahap *Document Frequency*, sistem dapat melanjutkan ke dalam proses *Inverse Document Frequency* (IDF) dan *Term Frequency-Inverse Document* Frequency (TF-IDF) untuk setiap kata. Langkah pertama dari proses ini adalah menghitung nilai IDF, ketika langkah

tersebut selesai dikerjakan, maka TF-IDF dapat dihasilkan dengan cara mengkalikan nilai antara TF dan IDF. Nilai yang dihasilkan pada proses TF-IDF akan digunakan sebagai acuan dalam melakukan klasifikasi dengan menggunakan implementasi algoritma *Naïve Bayes*. Proses dapat dilihat pada Gambar 3.15.



Gambar 3.15 *Flowchart* Proses *TF-IDF*

Ketika proses TF-IDF selesai dilakukan, sistem akan melanjutkan ke dalam tahap *training*. Pada pross ini sistem akan mempelajari pola dari data dalam melakukan klasifikasi, proses *training* ini akan menggunakan nilai pembobotan dari proses TF-IDF untuk dijadikan acuan dalam melakukan klasifikasi. Proses *training* dapat dilihat pada Gambar 3.16.

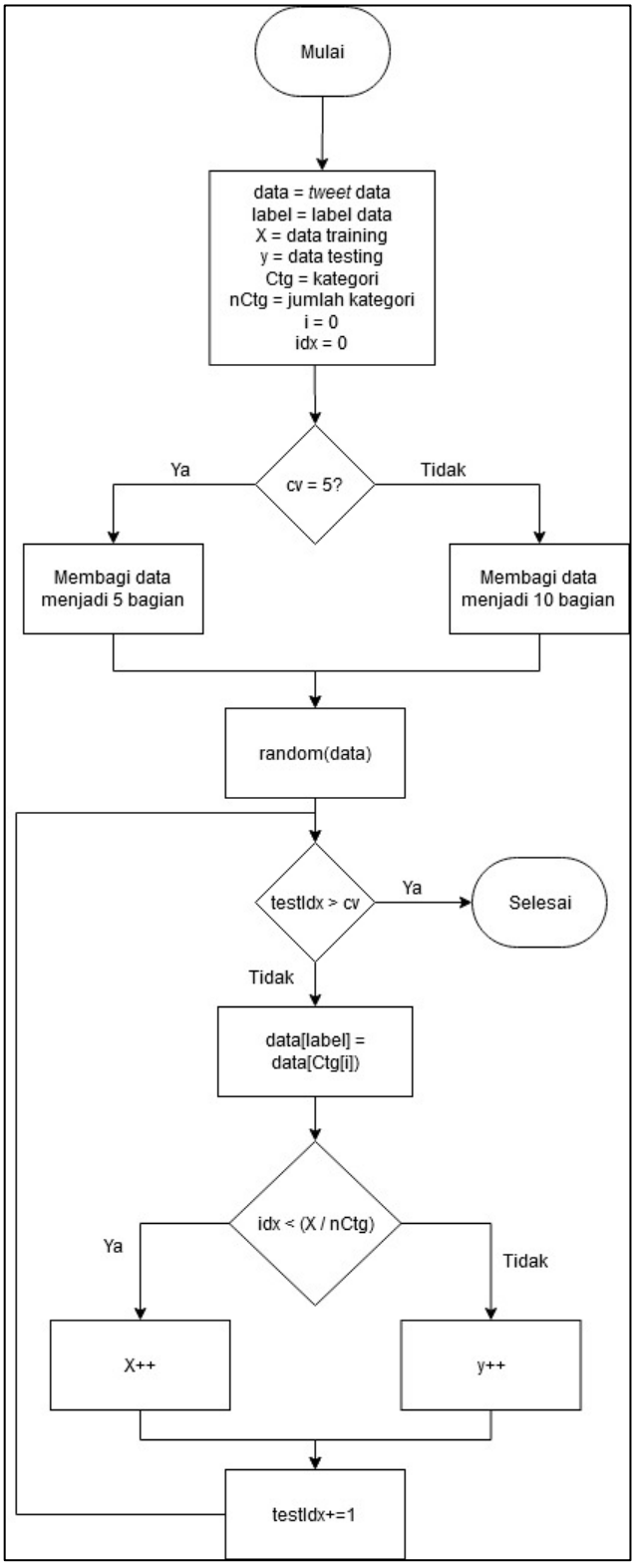


Gambar 3.16 *Flowchart* Modul *Training*

Pada proses training, terdapat dua modul besar yang dijalankan oleh sistem, yaitu *populate* data dan menghitung *conditional probability*. *Populate* data merupakan modul untuk membagi data *training* dan *testing*, sementara perhitungan *conditional probability* merupakan modul yang menghitung probabilitas kemunculan kata dalam suatu kelas. Nilai *conditional probability* akan digunakan pada tahap *testing*.

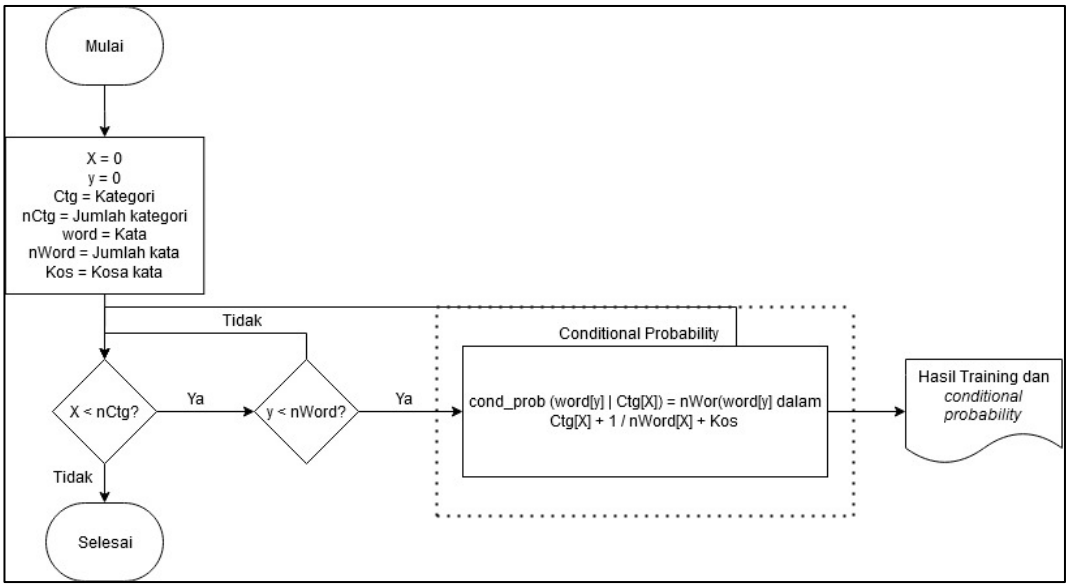
Pada modul *populate* data, sistem akan melakukan pengacakan data agar mendapatkan hasil pembelajaran atau *training* yang terbaik. Pembagian data *training* dan *testing* pada penelitian ini dilakukan dengan menggunakan *library Grid-Search Cross Validation* dengan nilai cv=5, cv merupakan proses pembagian

data oleh sistem menjadi sejumlah n (pada penelitian ini, n bernilai 5). Proses *populate* data dapat dilihat pada Gambar 3.17.



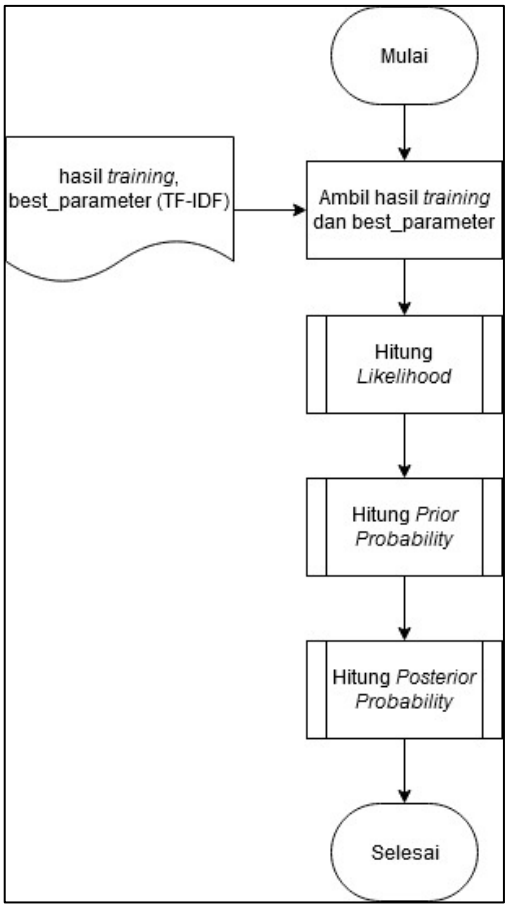
Gambar 3.17 *Flowchart* Modul *Populate* Data

Setelah *populate* data sudah selesai dijalankan pada proses *training*, maka sistem akan menjalankan perhitungan *conditional probability* yang merupakan perhitungan peluan kemunculan sebuah kata pada dokumen *training* tertentu. Pada perhitungan *conditional probability*, akan digunakan *Laplace's law of smoothing*, yaitu tahapan untuk mencegah kemunculan *conditional probability* yang bernilai 0 (*out of vocabulary*). Proses *conditional probability* dapat dilihat pada Gambar 3.18.



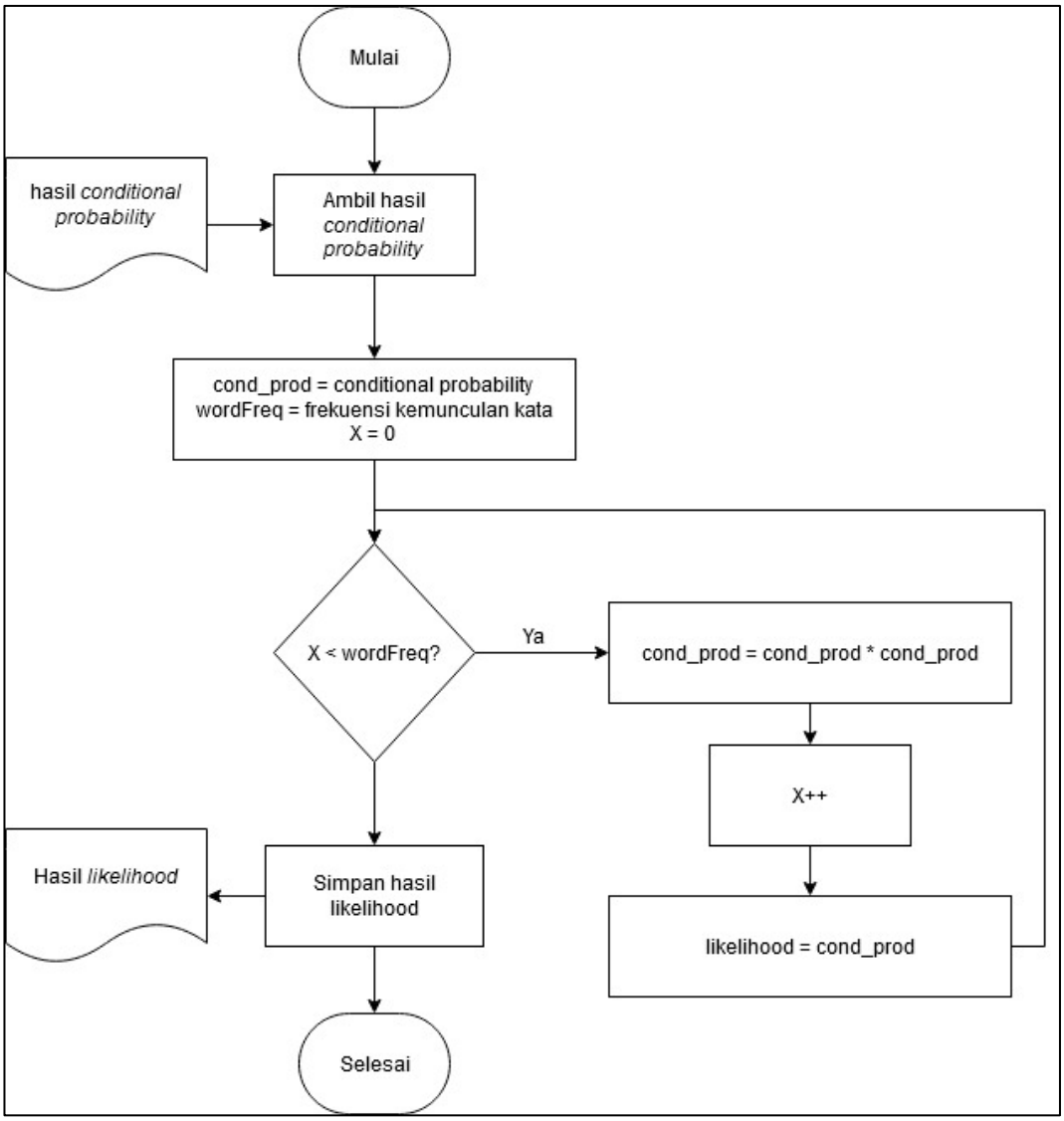
Gambar 3.18 *Flowchart* Modul *Conditional Probability*

Ketika proses *training* sudah selesai dilakukan, maka sistem akan melanjutkannya dengan melakukan proses *testing*. Proses ini menggunakan hasil *training* berupa *conditional probability* dalam melakukan klasifikasi data, tujuan utama dari proses *testing* adalah melakukan pengecekkan apakah model pada proses *training* sudah optimal. Proses *testing* memiliki 3 modul penting, yaitu menghitung *likelihood*, *prior probability*, dan *posterior probability* dengan menggunakan parameter terbaik dari TF-IDF. Proses dapat dilihat pada Gambar 3.19.



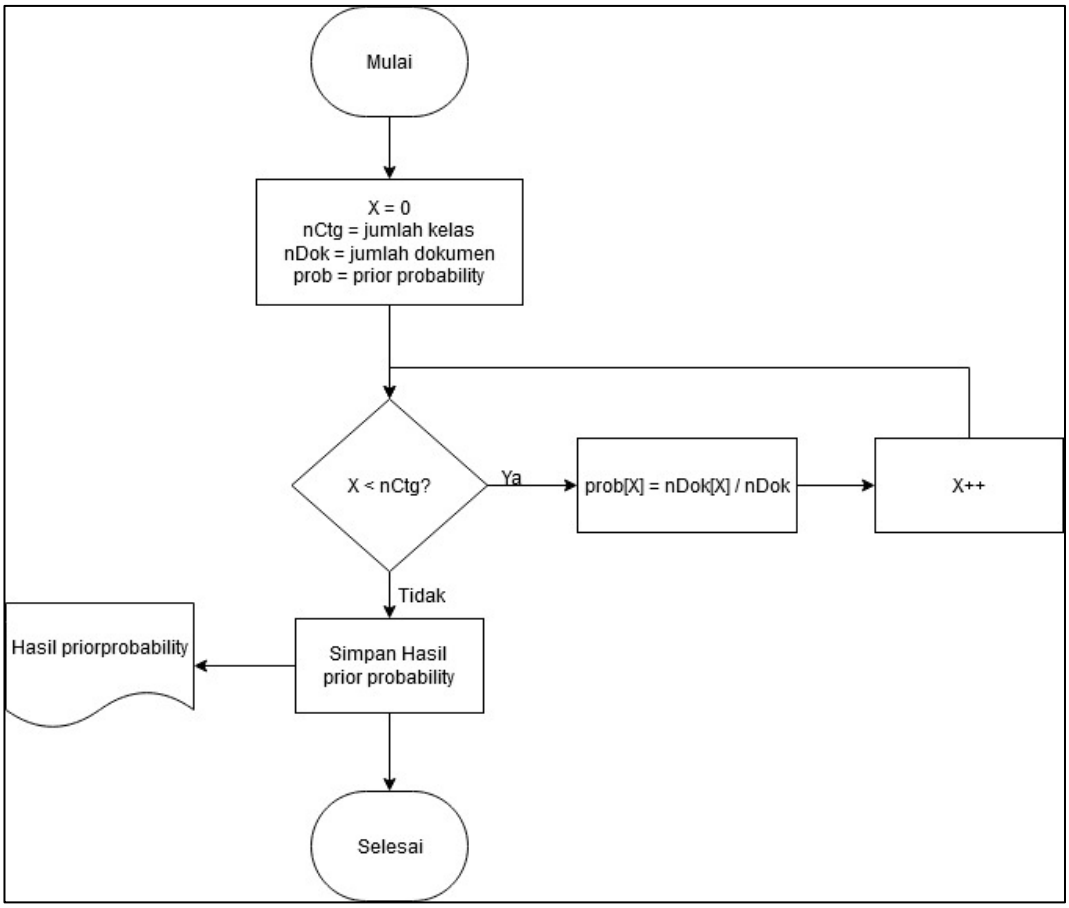
Gambar 3.19 *Flowchart* Modul *Testing*

Modul *likelihood* merupakan modul untuk mencari peluang kemunculan kata dalam dokumen tertentu, nilai dari *likelihood* berasal dari perkalian kemunculan kata (*conditional probability*) dan dirinya sendiri sebanyak n. Proses perhitugnan *likelihood* dapat dilihat pada Gambar 3.20.



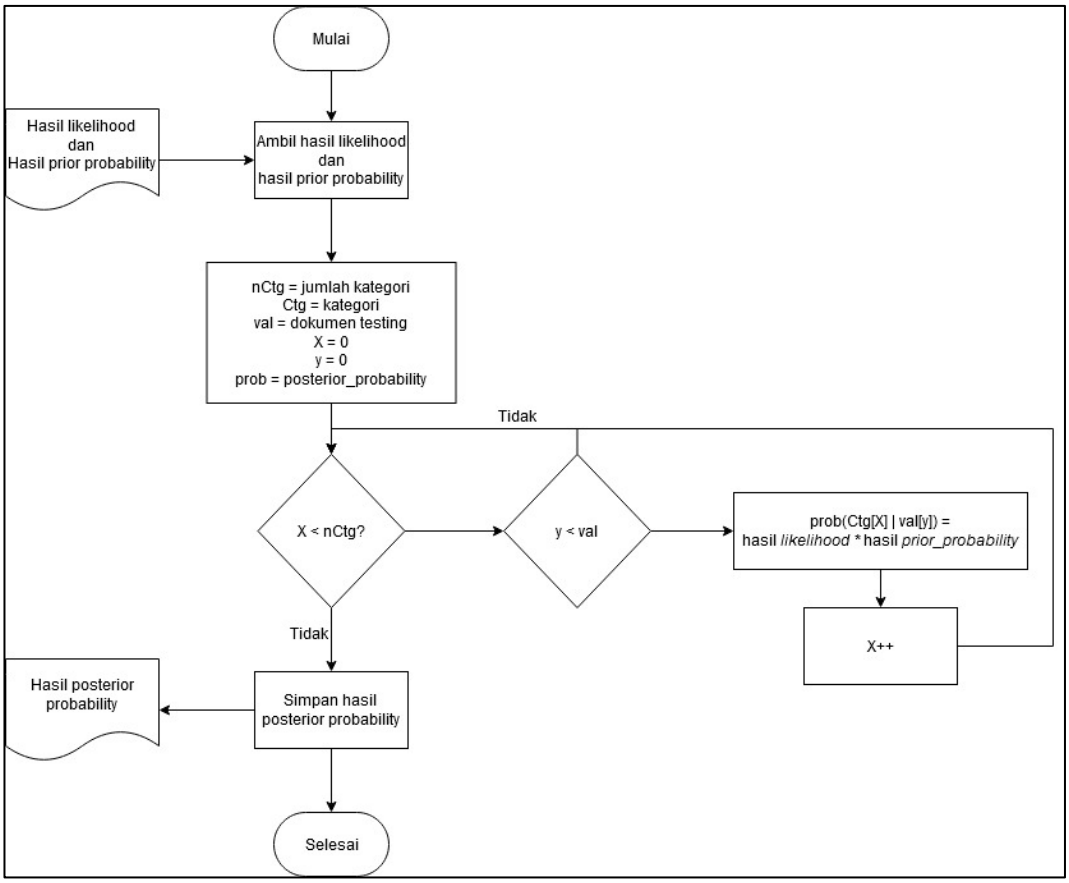
Gambar 3.20 *Flowchart* Modul *Likelihood*

Setelah proses perhitungan *likelihood* sudah selesai dijalankan, maka sistem akan melanjutkan dengan melakukan perhitungan terhadap *prior probability*. *Prior probability* merupakan proses perhitungan peluang kemunculan suatu dokumen dalam dokumen *training*. Pada penelitian ini, *prior probability* dilakukan untuk mencari peluang kemunculan *tweet* dengan indikasi depresi atau tidak teindikasi depresi pada data *training*. Proses modul *prior probability* dapat dilihat pada Gambar 3.21.



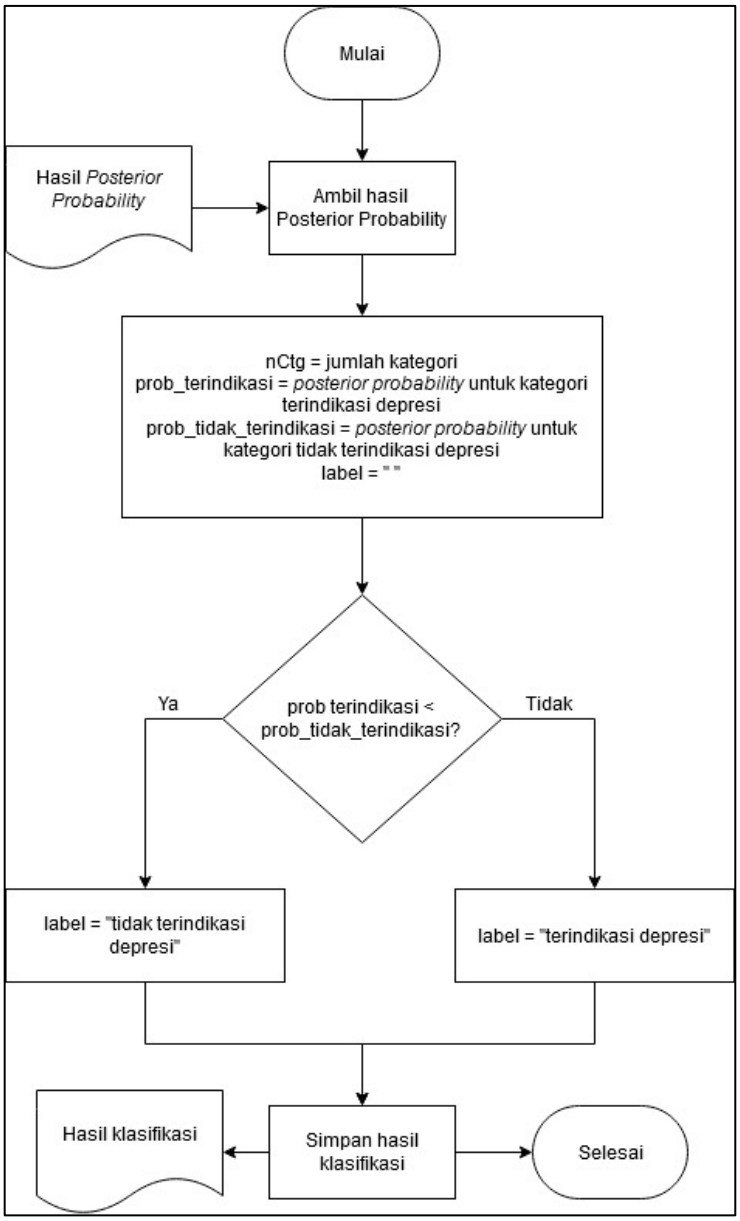
Gambar 3.21 *Flowchart* Modul *Prior Probability*

Tahap terakhir dari proses *testing* adalah perhitungan *posterior probability* yang dilakukan untuk menghitung peluang suatu *tweet* untuk masuk ke dalam kategori terindikasi depresi dan tidak terindikasi depresi. Proses perhitungan *Posterior Probability* dapat dilihat pada Gambar 3.22.



Gambar 3.22 *Flowchart* Modul *Posterior Probability*

Ketika proses *testing* selesai, maka sistem akan menjalankan tahap terakhir yaitu melakukan klasifikasi berdasarkan nilai *posterior probability* dari setiap kategori untuk tiap *tweet*. Pengkategorian akan didasari oleh nilai *posterior probability* yang lebih besar. Proses modul klasifikasi dapat dilihat pada Gambar 3.23.



Gambar 3.23 *Flowchart* Modul Klasifikasi Data *Tweet*

Setelah proses klasifikasi selesai, sistem akan mengeluarkan hasil berupa hasil prediksi terhadap data *test* dan menampilkan nilai *f1_micro* dan parameter terbaik yang dihasilkan dengan menggunakan *Grid Search Cross Validation.*