BAB III
METODE PENELITIAN

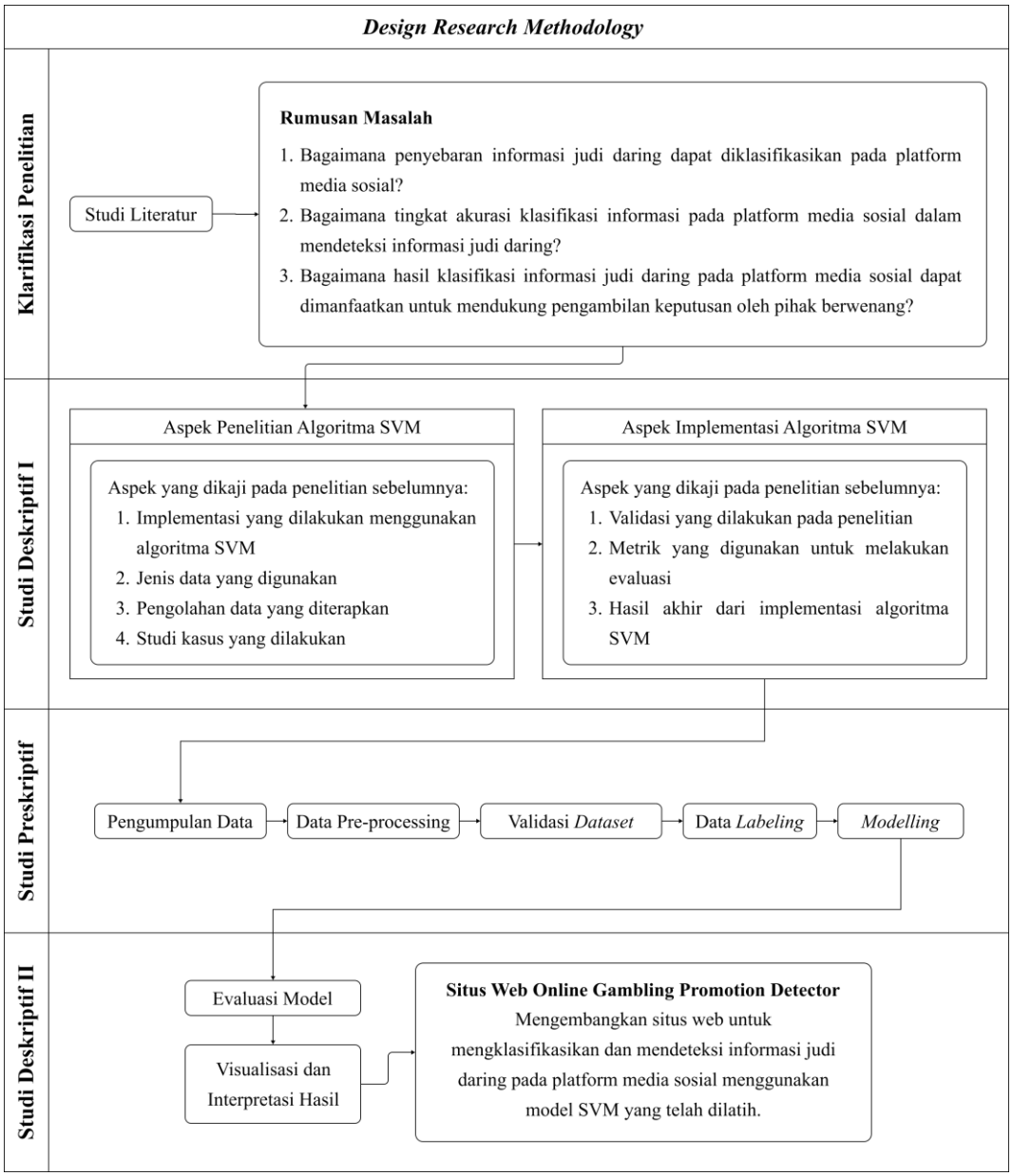
3.1 Desain Penelitian

Desain penelitian yang digunakan dalam penelitian ini adalah *Design Research Methodology* (DRM). DRM adalah sebuah kerangka metodologi yang bertujuan untuk merencanakan dan mengimplementasikan penelitian desain dengan lebih efektif dan efisien. Menurut Blessing dkk. (1998) metodologi DRM terdiri dari empat tahap, yaitu:

1. Klarifikasi Penelitian (*Research Clarification*), tahap ini bertujuan untuk menjelaskan pemahaman peneliti tentang penelitian yang dilakukan dan keseluruhan tujuan yang ingin dicapai. Tahap ini melibatkan kajian literatur, formulasi masalah, tujuan, dan kriteria keberhasilan penelitian.
2. Studi Deskriptif I (*Descriptive Study I*), tahap ini bertujuan untuk mendapatkan pemahaman yang lebih dalam tentang situasi yang ada dengan mengidentifikasi dan mengklarifikasi faktor-faktor yang mempengaruhi kriteria keberhasilan. Tahap ini melibatkan pengumpulan dan analisis data empiris, baik dari studi lapangan maupun laboratorium.
3. Studi Preskriptif (*Prescriptive Study*), tahap ini bertujuan untuk mengolah hasil dari studi deskriptif I dan mengembangkan desain pendukung (*design support*) yang dapat berupa metode, pedoman, alat, atau prototipe yang bertujuan untuk meningkatkan, menghilangkan, atau mengurangi pengaruh faktor-faktor penting yang telah diidentifikasi.
4. Studi Deskriptif II (*Descriptive Study II*), tahap ini bertujuan untuk menguji dan mengevaluasi desain pendukung yang telah dikembangkan dengan mengaplikasikannya pada situasi yang sesuai dan mengamati dampaknya terhadap kriteria keberhasilan. Tahap ini melibatkan validasi, verifikasi, dan perbaikan desain pendukung.

Dengan memahami keempat tahap metodologi DRM yang telah diuraikan, menjadi lebih jelas bagaimana kerangka kerja ini membantu dalam merencanakan dan mengimplementasikan penelitian desain, serta memastikan bahwa hasil

penelitian dapat diuji dan dievaluasi secara efektif. Gambar 3.1 adalah kerangka metodologi DRM yang diadopsi dalam penelitian ini.



Gambar 3.1 Desain Penelitian

3.1.1 Klarifikasi Penelitian

Pada tahap awal penelitian ini, dilakukan klarifikasi penelitian sebagai langkah untuk mengidentifikasi inti masalah yang berkaitan dengan penyebaran informasi judi daring di platform media sosial. Studi literatur yang komprehensif dijalankan dengan merujuk pada berbagai sumber, termasuk jurnal, skripsi yang telah ada, dan materi dari internet, untuk meningkatkan pemahaman serta

memperkuat konsep solusi yang akan diterapkan. Proses ini merupakan fondasi yang krusial untuk merampungkan dan mengembangkan formulasi masalah. Literatur yang diteliti mencakup topik seperti judi daring, media sosial, klasifikasi, *text mining*, *web crawling*, *machine learning*, dan algoritma Support Vector Machine (SVM). Selain itu, metode evaluasi seperti *Confusion Matrix* dan K-Fold Cross Validation juga dipelajari. Tahapan klarifikasi penelitian ini menghasilkan pemahaman awal mengenai dinamika penyebaran informasi judi daring. Selanjutnya, dilakukan penelaahan literatur yang relevan dan formulasi dari identifikasi masalah. Tahap ini fokus pada pemecahan permasalahan dengan penerapan algoritma SVM untuk mengklasifikasikan informasi yang berkaitan dengan judi daring.

3.1.2 Studi Deskriptif I

Selanjutnya, tahap Studi Deskriptif I dilakukan untuk meningkatkan pemahaman mengenai masalah penyebaran informasi judi daring di platform media sosial serta menginvestigasi fenomena tersebut. Tahap ini memfokuskan pada pengembangan solusi berdasarkan hasil studi literatur dan formulasi masalah yang telah dikerjakan sebelumnya. Capaian yang dituju pada tahap ini adalah proses awal pengukuran dan penentuan pembaharuan dari solusi penelitian. Di tahap ini, peneliti melakukan analisis seputar penelitian terdahulu yang berkaitan dengan implementasi algoritma SVM, dengan tujuan memperdalam pemahaman tentang masalah yang diteliti serta analisisnya. Ini meliputi mengkaji jenis data yang digunakan, pengolahan data yang diterapkan, dan evaluasi implementasi SVM. Selanjutnya, dilakukan analisis beberapa kajian mengenai metrik yang digunakan untuk melakukan evaluasi dan melihat bagaimana hasil akhir dari implementasi algoritma SVM.

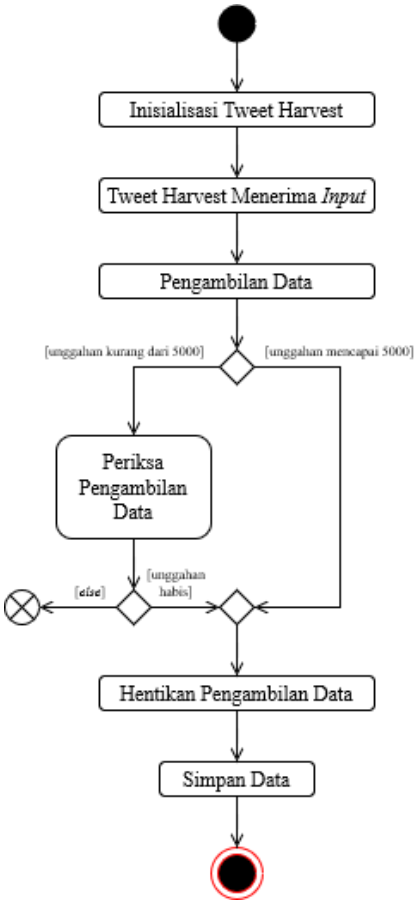
3.1.3 Studi Preskriptif

Studi Preskriptif dalam penelitian ini terbagi menjadi beberapa tahapan yang terstruktur. Mulai dari pengumpulan data, data *pre-processing*, validasi *dataset*, data *labeling*, data *splitting*, ekstraksi fitur, hingga *modelling*.

3.1.3.1 Pengumpulan Data

Pengumpulan data dalam penelitian ini menggunakan Google Colab dan *library* Tweet Harvest untuk melakukan *web crawling* di platform X. Tweet Harvest

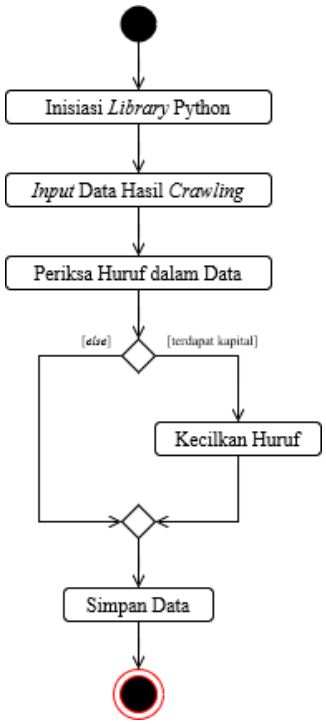
memerlukan beberapa informasi, termasuk nama untuk *file* hasil *crawling*, kata kunci berupa *search parameter* untuk *crawling*, jumlah unggahan yang akan dikumpulkan, serta X *auth token* yang didapatkan dari akun peneliti. Peneliti menamai *file* berdasarkan kata kunci yang diikuti oleh tahun dan bulan, kemudian menggunakan *search parameter* yaitu kata kunci "judi", bahasa, dan rentang waktu pengambilan data. Agar *crawling* dapat dilakukan bertahap, maka rentang waktu pengambilan data adalah per bulan, dengan target pengumpulannya adalah 5000 unggahan. Proses ini dimulai dengan Tweet Harvest yang membuka browser Chromium, navigasi ke halaman pencarian X menggunakan akun peneliti, memasukkan kata kunci, *search parameter* yang telah ditentukan, dan selanjutnya melakukan pengambilan data. Unggahan yang berhasil dikumpulkan disimpan dalam kolom full_text dengan format CSV dalam direktori di Google Colab. Tahap pengumpulan data digambarkan seperti pada Gambar 3.2.



Gambar 3.2 Diagram Aktivitas Pengumpulan Data

3.1.3.2 Data *Pre-processing*

Data *pre-processing* dilakukan menggunakan bantuan Google Colab dan beberapa *library* Python. Proses diawali dengan *case folding*, di mana *file* hasil pengumpulan data dibaca, dan semua huruf pada kolom full_text dikonversi menjadi huruf kecil. Hasilnya disimpan dalam kolom baru bernama case_folded dan disimpan dalam *file* CSV. Proses *case folding* dalam tahap data *pre-processing* digambarkan seperti pada Gambar 3.3.



Gambar 3.3 Diagram Aktivitas *Case Folding*

Langkah berikutnya adalah proses *cleaning*, di mana teks yang telah melalui proses *case folding* dibersihkan dari elemen-elemen yang tidak relevan atau mengganggu, seperti URL, *tag* pengguna, *hashtag*, *tag* HTML, tanda baca, karakter non-ASCII, angka, dan lainnya, seperti yang dijelaskan pada Tabel 3.1.

Tabel 3.1

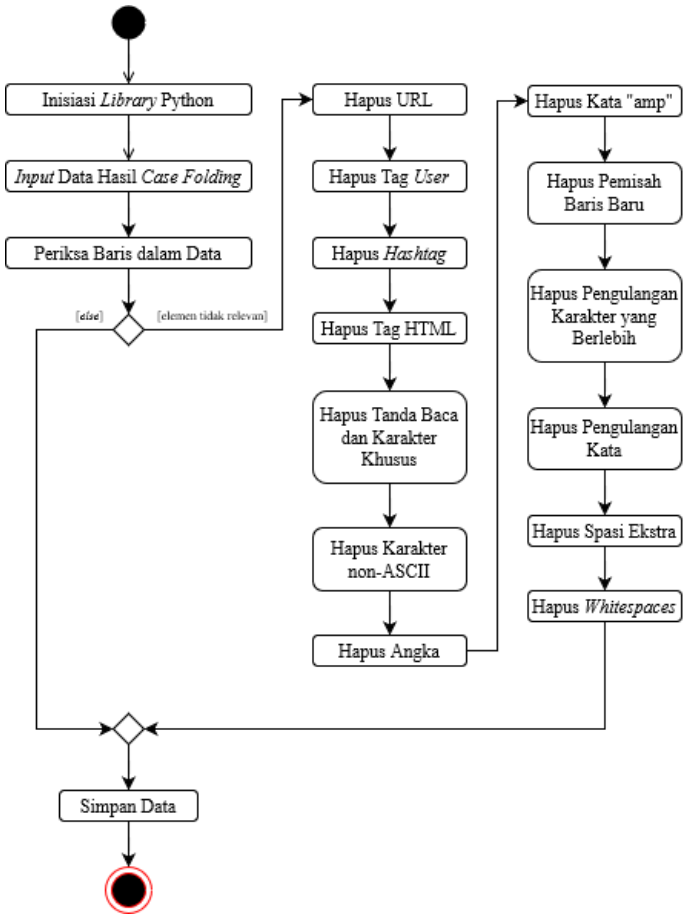
Tahapan *Cleaning*

Tahapan *Cleaning*	Penjelasan
Menghapus URL	URL sering kali tidak memberikan informasi yang relevan untuk analisis teks dan dapat mengganggu proses analisis.

Menghapus *tag user*	*Tag user* seperti @username biasanya tidak relevan untuk analisis sentimen atau topik dan hanya menambah *noise* pada data.
Menghapus *hashtag*	Meskipun *hashtag* dapat memberikan konteks, tetapi sering kali mereka mengandung gabungan kata yang sulit dipisahkan dan diinterpretasikan oleh model.
Menghapus *tag* HTML	*Tag* HTML tidak relevan untuk analisis teks dan harus dihilangkan agar tidak mengganggu proses analisis.
Menghapus tanda baca atau karakter khusus	Tanda baca dan karakter khusus sering tidak diperlukan dalam analisis teks dan dapat dihilangkan untuk mengurangi kompleksitas data.
Menghapus karakter non-ASCII atau karakter *unicode*	Karakter non-ASCII atau *unicode* bisa jadi tidak terbaca oleh beberapa sistem analisis dan lebih baik dihilangkan.
Menghapus angka	Angka sering tidak memberikan kontribusi pada analisis sentimen atau topik dan dapat dihilangkan untuk fokus pada teks.
Menghapus “amp”	Kata “amp” sering muncul sebagai entitas HTML untuk “dan” dan tidak memiliki arti dalam analisis teks.
Menghapus baris baru	Pemisah baris baru dapat mengganggu analisis teks jika tidak diubah menjadi spasi.
Menghapus karakter tunggal	Karakter tunggal sering tidak memiliki arti dalam analisis teks dan dapat dihilangkan.
Mengganti semua urutan karakter yang berulang lebih dari dua kali dalam baris menjadi dua pengulangan karakter	Pengulangan karakter yang berlebihan tidak menambah nilai analitis dan dapat dihilangkan untuk konsistensi.

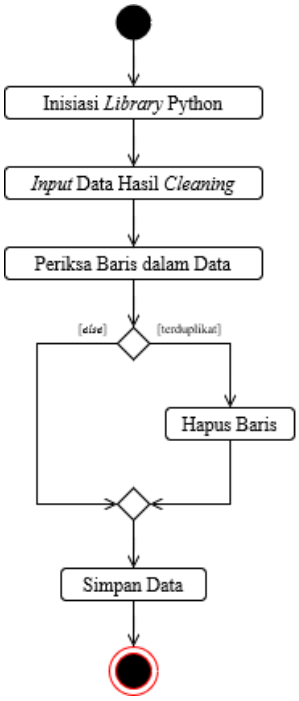
Menghapus kata-kata yang berulang	Pengulangan kata dapat dianggap sebagai *noise* dan mengganggu analisis frekuensi kata.
Menghapus spasi ekstra	Spasi ekstra tidak diperlukan dan dapat mengganggu proses *tokenization*.
Menghapus *whitespace*	*Whitespace* yang tidak perlu di awal dan akhir teks harus dihilangkan untuk kebersihan data.

Proses *cleaning* ini menghasilkan teks yang lebih bersih yang disimpan dalam kolom baru bernama cleaned dan disimpan dalam *file* CSV. Proses *cleaning* dalam tahap data *pre-processing* digambarkan seperti pada Gambar 3.4.



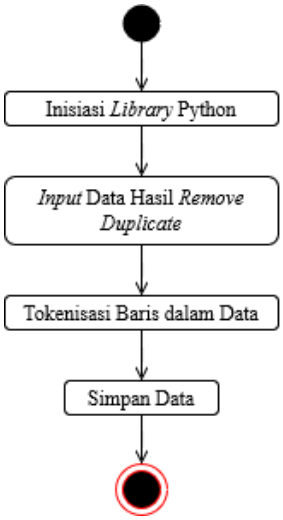
Gambar 3.4 Diagram Aktivitas *Cleaning*

Selanjutnya dilakukan proses *remove duplicate* untuk menghapus duplikat unggahan yang mungkin ada dalam *dataset*, memastikan bahwa data yang digunakan adalah unik dan tidak mengandung duplikasi. Proses *remove duplicate* dalam tahap data *pre-processing* digambarkan seperti pada Gambar 3.5.



Gambar 3.5 Diagram Aktivitas *Remove Duplicate*

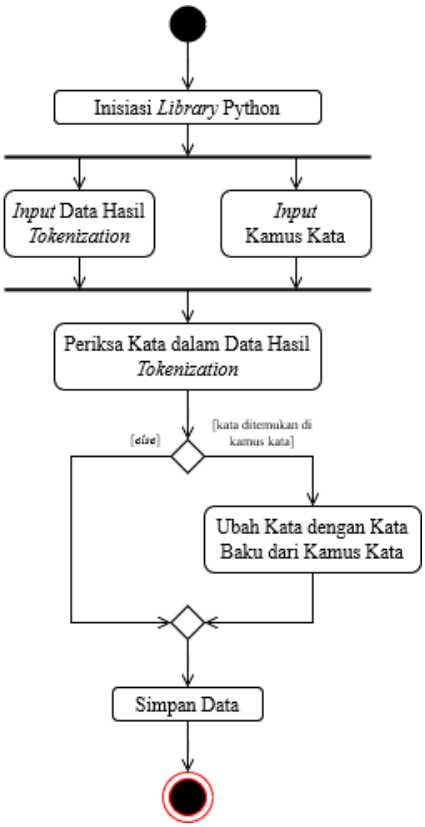
Langkah selanjutnya adalah *tokenization*, di mana teks yang telah dibersihkan dan dihapus duplikatnya dibagi menjadi *token-token* individual menggunakan NLTK RegexpTokenizer. Ini memungkinkan untuk mengidentifikasi setiap kata secara terpisah. Hasil *tokenization* disimpan dalam kolom baru bernama tokenized dan disimpan dalam *file* CSV. Proses *tokenization* dalam tahap data *pre-processing* digambarkan seperti pada Gambar 3.6.



Gambar 3.6 Diagram Aktivitas *Tokenization*

Setelah *tokenisasi*, dilakukan *slang normalization* di mana kata-kata slang dalam *dataset* diperbaiki menjadi bentuk standar. Proses ini memanfaatkan kamus

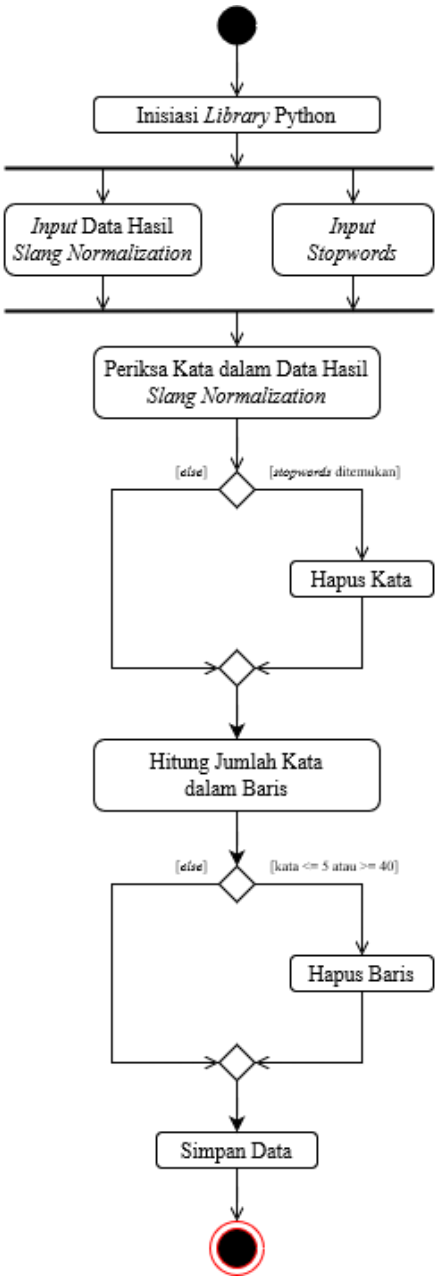
kata yang mencakup *slang word* yang umumnya digunakan di Indonesia, yaitu Colloquial Indonesian Lexicon (Aliyah Salsabila dkk., 2018), serta penambahan *slang word* manual yang terkait dengan perjudian daring. Jika dalam baris terdapat kata yang terdapat juga pada kamus kata, maka kata tersebut diganti dengan kata baku yang terdapat di kamus kata. Hasil *slang normalization* disimpan dalam kolom baru bernama normalized dan disimpan dalam file CSV. Proses *slang normalization* dalam tahap data *pre-processing* digambarkan seperti pada Gambar 3.7.



Gambar 3.7 Diagram Aktivitas *Slang Normalization*

Kemudian, dilakukan *filtering* dengan menghapus *stopwords* dan menghapus baris yang jumlah katanya kurang dari atau sama dengan 5, serta lebih dari atau sama dengan 40. *Stopwords* yang digunakan didasarkan pada *corpus stopwords* berbahasa Indonesia yang diperoleh dari NLTK. Prosesnya adalah, data hasil *slang normalization* dilakukan perhitungan jumlah kata. Hasil perhitungan tersebut disimpan di samping kolom yang terkait, kemudian dilakukan pengecekan apakah jumlah kata kurang dari atau sama dengan 5 atau lebih dari atau sama dengan 40. Jika angka tersebut tidak sesuai, maka baris akan dihapus. Setelah baris berhasil dihapus, kolom hasil perhitungan tersebut juga dihapus. Selanjutnya, jika

terdapat kata yang juga terdapat pada *corpus stopwords*, kata tersebut akan dihapus. Hasil dari proses *filtering* disimpan dalam kolom baru yang dinamakan stopwords_removed dan selanjutnya disimpan dalam *file* CSV. Proses *filtering* dalam tahap data *pre-processing* digambarkan seperti pada Gambar 3.8.



Gambar 3.8 Diagram Aktivitas *Filtering*

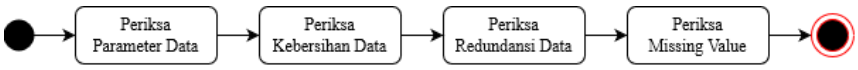
Terakhir, dilakukan *stemming* menggunakan *library* PySastrawi untuk mereduksi kata-kata ke bentuk dasarnya. Proses ini membantu dalam konsistensi dan reduksi dimensi kata-kata dalam *dataset*. Hasil dari *stemming* disimpan dalam

file CSV. Proses *stemming* dalam tahap data *pre-processing* digambarkan seperti pada Gambar 3.9.

Gambar 3.9 Diagram Aktivitas *Stemming*

3.1.3.3 Validasi *Dataset*

Validasi *dataset* dilakukan untuk memastikan bahwa *dataset* yang akan digunakan dalam pembuatan model memiliki kualitas tinggi, relevan, dan representatif. Validasi *dataset* dimulai dengan pemeriksaan parameter data, apakah sesuai dengan kriteria yang ditentukan atau tidak. Kemudian dilanjutkan dengan pemeriksaan kebersihan data, yang bertujuan untuk mengetahui apakah setelah proses *cleaning* pada tahap data *pre-processing* masih terdapat data yang tidak bersih. Selanjutnya dilakukan pemeriksaan redundansi data, yaitu untuk mengetahui apakah masih terdapat data yang terduplikasi. Terakhir, dilakukan pemeriksaan *missing value*, di mana data diperiksa untuk memastikan apakah ada nilai yang hilang. Gambar 3.10 merupakan tahapan dari validasi *dataset* yang dilakukan dalam penelitian ini.



Gambar 3.10 Tahapan Validasi *Dataset*

3.1.3.4 Data *Labeling*

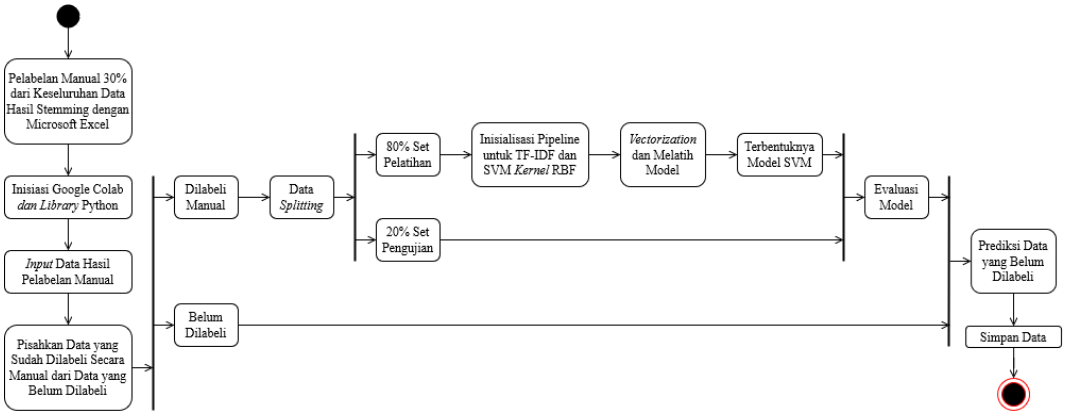
Setelah melewati proses validasi *dataset*, langkah selanjutnya adalah pemberian label secara semi-manual menggunakan Microsoft Excel dan algoritma *machine learning*. Proses pelabelan data menggunakan Microsoft Excel dilakukan secara manual dengan merujuk pada kriteria label yang ditentukan pada Tabel 3.2 di bawah ini.

Tabel 3.2
Kriteria Label

Label Positif	Promosi perjudian, baik secara eksplisit maupun implisit
	Penyebaran pengalaman perjudian yang menguntungkan
	Dukungan terhadap perjudian melalui opini yang positif
	Keinginan atau niat untuk berpartisipasi dalam perjudian
	Merekomendasikan atau memberikan informasi mengenai situs perjudian
Label Negatif	Kebalikan dari label positif
	Penolakan terhadap perjudian
	Ajakan untuk menghindari perjudian
	Opini negatif mengenai perjudian
	Penyampaian informasi yang objektif dan akurat tanpa mengarahkan pembaca ke suatu kesimpulan tertentu
	Ungkapan kebingungan atau pertanyaan

Target pelabelan manual adalah lebih dari 30% keseluruhan data. Data yang belum memiliki label akan diberi label menggunakan algoritma *machine learning*, khususnya algoritma SVM dengan *kernel* RBF, dengan target minimum akurasi sebesar 95%. Proses pelabelan dengan *machine learning* dimulai dengan

memisahkan data yang sudah diberi label secara manual dengan data yang belum diberi label. Selanjutnya, *modelling* SVM dilakukan dengan memanfaatkan data yang sudah diberi label, yang hasilnya akan memberikan klasifikasi pada data yang belum diberi label. Hasil dari kedua proses tersebut kemudian digabung menjadi satu *file* CSV. Tahap data *labeling* digambarkan pada Gambar 3.11.



Gambar 3.11 Diagram Aktivitas Data *Labeling*

3.1.3.5 *Modelling*

Proses ini dimulai dengan membagi seluruh *dataset* menjadi dua bagian yaitu set pelatihan dan set pengujian. Pertama, 80% dari data digunakan sebagai set pelatihan, sementara 20% sisanya digunakan sebagai set pengujian. Pembagian dilakukan secara acak, tetapi konsistensi tetap dijaga dengan menetapkan *random state* 42, sehingga pembagian data akan selalu sama setiap kali skrip dijalankan.

Setelah pembagian, set pelatihan akan digunakan untuk melatih model, sedangkan set pengujian akan digunakan untuk menguji kinerja model yang telah dilatih. Selanjutnya dilakukan ekstraksi fitur dengan menghitung nilai TF-IDF untuk setiap kata dalam setiap dokumen teks. Misalnya, terdapat dokumen yang berisi promosi judi daring dalam Tabel 3.3.

Tabel 3.3

Contoh Dokumen untuk Ekstraksi Fitur

Kalimat 1	Promo besar judi daring, daftar sekarang dan menangkan hadiah!
Kalimat 2	Judi daring terbaik dengan bonus menarik, segera daftar!
Kalimat 3	Ayo judi daring sekarang, nikmati promo dan bonus besar!

Pertama, dilakukan penghitungan *Term Frequency* (TF) dengan mencari tahu frekuensi setiap kata dalam dokumen dan jumlah kata dari setiap kalimat. Selanjutnya, frekuensi setiap kata dibagi dengan jumlah kata dalam kalimat tersebut. Penghitungan ini digambarkan pada Tabel 3.4, sebagai berikut:

Tabel 3.4

Term Frequency

Term	Kalimat 1	Kalimat 2	Kalimat 3
promo	1/9	0/8	1/9
besar	1/9	0/8	1/9
judi	1/9	1/8	1/9
daring	1/9	1/8	1/9
daftar	1/9	1/8	0/9
sekarang	1/9	0/8	1/9
dan	1/9	0/8	1/9
menangkan	1/9	0/8	0/9
hadiah	1/9	0/8	0/9
terbaik	0/9	1/8	0/9
dengan	0/9	1/8	0/9
bonus	0/9	1/8	1/9
menarik	0/9	1/8	0/9
segera	0/9	1/8	0/9
ayo	0/9	0/8	1/9
nikmati	0/9	0/8	1/9

Contohnya, pada Tabel 3.4 diketahui bahwa pada kalimat 1 dengan term "promo", TF-nya adalah 1/9 di mana 1 merupakan frekuensi kata "promo" yang muncul pada kalimat 1, sedangkan 9 merupakan jumlah kata pada kalimat 1.

Selanjutnya, dilakukan penghitungan *Inverse Document Frequency* (IDF) dengan logaritma basis 10. Penghitungan ini digambarkan pada Tabel 3.5, sebagai berikut:

Tabel 3.5

Inverse Document Frequency

Term	$\boldsymbol{df}$	$\boldsymbol{IDF(t) = \log\left(\frac{N}{1 + df(t)}\right)}$
promo	2	$log\left(\frac{3}{3}\right) = 0$
besar	2	$log\left(\frac{3}{3}\right) = 0$
judi	3	$log\left(\frac{3}{4}\right) = -0.1249$
daring	3	$log\left(\frac{3}{4}\right) = -0.1249$
daftar	2	$log\left(\frac{3}{3}\right) = 0$
sekarang	2	$log\left(\frac{3}{3}\right) = 0$
dan	2	$log\left(\frac{3}{3}\right) = 0$
menangkan	1	$log\left(\frac{3}{2}\right) = 0.1761$
hadiah	1	$log\left(\frac{3}{2}\right) = 0.1761$
terbaik	1	$log\left(\frac{3}{2}\right) = 0.1761$
dengan	1	$log\left(\frac{3}{2}\right) = 0.1761$
bonus	2	$log\left(\frac{3}{3}\right) = 0$
menarik	1	$log\left(\frac{3}{2}\right) = 0.1761$
segera	1	$log\left(\frac{3}{2}\right) = 0.1761$
ayo	1	$log\left(\frac{3}{2}\right) = 0.1761$
nikmati	1	$log\left(\frac{3}{2}\right) = 0.1761$

Pada Tabel 3.5, N merupakan banyaknya kalimat pada dokumen, df merupakan frekuensi term yang muncul dari keseluruhan dokumen, sedangkan nilai 1 ditambahkan untuk menghindari pembagian dengan nol.

Setelah nilai TF dan IDF diperoleh, keduanya digabungkan untuk menghasilkan nilai TF-IDF untuk setiap term dalam keseluruhan dokumen. Misalnya, untuk kata "promo" dalam kalimat pertama, nilai TF-IDF dihitung sebagai TF×IDF, yang mencerminkan seberapa pentingnya kata "promo" dalam kalimat pertama relatif terhadap keseluruhan dokumen. Berikut ini adalah gambaran penghitungan TF-IDF dalam Tabel 3.6:

Tabel 3.6

Penghitungan TF-IDF

Term	Kalimat 1	Kalimat 2	Kalimat 3
promo	1/9 x 0 = 0	0/8 x 0 = 0	1/9 x 0 = 0
besar	1/9 x 0 = 0	0/8 x 0 = 0	1/9 x 0 = 0
judi	1/9 x -0.1249 = -0.0139	1/8 x -0.1249 = -0.0156	1/9 x -0.1249 = -0.0139
daring	1/9 x -0.1249 = -0.0139	1/8 x -0.1249 = -0.0156	1/9 x -0.1249 = -0.0139
daftar	1/9 x 0 = 0	1/8 x 0 = 0	0/9 x 0 = 0
sekarang	1/9 x 0 = 0	0/8 x 0 = 0	1/9 x 0 = 0
dan	1/9 x 0 = 0	0/8 x 0 = 0	1/9 x 0 = 0
menangkan	1/9 x 0.1761 = 0.0196	0/8 x 0.1761 = 0	0/9 x 0.1761 =
hadiah	1/9 x 0.1761 = 0.0196	0/8 x 0.1761 = 0	0/9 x 0.1761 =
terbaik	0/9 x 0.1761 = 0	1/8 x 0.1761 = 0.0220	0/9 x 0.1761 =
dengan	0/9 x 0.1761 = 0	1/8 x 0.1761 = 0.0220	0/9 x 0.1761 =
bonus	0/9 x 0 = 0	1/8 x 0 = 0	1/9 x 0 = 0

menarik	0/9 x 0.1761 = 0	1/8 x 0.1761 = 0.0220	0/9 x 0.1761 =
segera	0/9 x 0.1761 = 0	1/8 x 0.1761 =0.0220	0/9 x 0.1761 =
ayo	0/9 x 0.1761 = 0	0/8 x 0.1761 = 0	1/9 x 0.1761 =
nikmati	0/9 x 0.1761 = 0	0/8 x 0.1761 = 0	1/9 x 0.1761 =

Nilai TF-IDF mencerminkan seberapa pentingnya sebuah kata dalam sebuah kalimat tertentu, dengan mempertimbangkan frekuensinya dalam kalimat tersebut dan kemunculannya di seluruh dokumen. Untuk memudahkan proses penghitungan TF-IDF, dapat dilakukan dengan bantuan *library* seperti Scikit-Learn. *Library* ini menyediakan berbagai alat yang memudahkan penghitungan TF-IDF secara otomatis dan efisien.

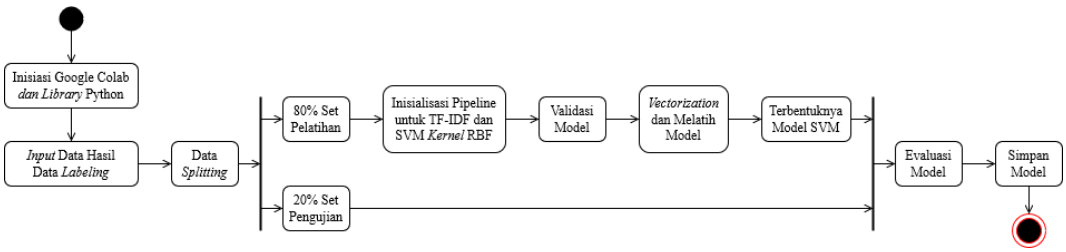
Sebuah *pipeline* diinisiasi dengan menggabungkan dua komponen utama, yaitu TF-IDF dan SVM. *Pipeline* dibuat menggunakan fungsi dari *library* Scikit-Learn. Komponen pertama dalam *pipeline* adalah TfidfVectorizer, yang digunakan untuk melakukan ekstraksi fitur dari teks mentah. Pada komponen tersebut, jumlah fitur atau kata unik yang dipertimbangkan adalah 1000 kata yang paling sering muncul. Ini membantu mengurangi dimensi data dan menghilangkan kata-kata yang kurang relevan, sehingga meningkatkan efisiensi komputasi dan kinerja model. Komponen kedua dalam *pipeline* adalah SVC (Support Vector Classifier), yang merupakan implementasi dari algoritma SVM untuk klasifikasi. SVC diinisiasi dengan parameter *kernel* RBF. *Kernel* tersebut dipilih karena kemampuannya dalam mengklasifikasikan data dengan tingkat akurasi yang relatif tinggi dibandingkan dengan *kernel* lain.

Pipeline ini memungkinkan peneliti untuk mengaplikasikan transformasi yang sama pada set pelatihan dan set pengujian, serta menyederhanakan alur kerja dengan menggabungkan beberapa langkah menjadi satu objek yang mudah digunakan. *Pipeline* juga membantu menghindari kebocoran data, karena fitur yang diekstraksi dari data pengujian akan konsisten dengan fitur yang diekstraksi dari data pelatihan, tanpa perlu melakukan fit ulang pada data pengujian.

Pipeline ini kemudian akan digunakan untuk melatih model pada data pelatihan dan untuk membuat prediksi pada data pengujian, memastikan bahwa seluruh proses dari ekstraksi fitur hingga klasifikasi dilakukan dengan cara yang efisien dan terintegrasi.

Selanjutnya adalah melakukan validasi pada model yang telah diinisiasi dalam *pipeline*. Validasi dilakukan dengan menggunakan teknik K-Fold Cross Validation. Karena *dataset* yang dikumpulkan cukup besar, digunakan nilai K = 10 pada tahap K-Fold Cross Validation, yang mana nilai tersebut dapat digunakan dengan baik karena setiap *fold* akan memiliki cukup banyak data. Teknik ini membagi set pelatihan menjadi 10 *subset* yang sama besar. Pembagian data dilakukan secara acak dengan menetapkan *random state* 42 untuk menghasilkan *fold* yang lebih representatif dan memastikan pengacakan dilakukan dengan cara yang sama setiap kali skrip dijalankan. Pada setiap iterasi, satu *subset* digunakan sebagai set pengujian, sementara 9 *subset* lainnya digunakan sebagai set pelatihan. Proses ini diulang sebanyak 10 kali, sehingga setiap *subset* digunakan sebagai set pengujian sekali. Hasil dari K-Fold Cross Validation menunjukkan skor akurasi untuk setiap iterasi, rata-rata akurasi keseluruhan, *standard deviation*, dan *variance*.

Setelah validasi, model SVM dilatih pada seluruh set pelatihan yang telah disiapkan. Selama proses pelatihan, model SVM mempelajari pola-pola yang ada dalam set pelatihan dengan mencari garis pemisah yang optimal antara kelas data yang berbeda. Model SVM memperhitungkan sejumlah faktor, termasuk margin terbesar antara kelas data yang berbeda dan penalti terhadap data yang berada di sisi yang salah dari garis pemisah. Model yang telah dilatih ini akan digunakan untuk memprediksi label pada set pengujian yang telah dipisahkan sebelumnya. Tahap *modelling* digambarkan pada Gambar 3.12.



Gambar 3.12 Diagram Aktivitas *Modelling*

3.1.4 Studi Deskriptif II

Tahap Studi Deskriptif II dalam penelitian ini meliputi evaluasi, interpretasi, visualisasi hasil, serta *deployment* model, diikuti dengan penarikan kesimpulan dan pemberian saran.

3.1.4.1 Evaluasi

Proses ini melibatkan penerapan model SVM yang telah dilatih untuk mengklasifikasikan kelas dari set pengujian. Hasil prediksi kemudian dibandingkan dengan label sebenarnya dari set pengujian untuk mengevaluasi performa model.

Pertama, hasil prediksi dari model SVM pada set pengujian dikumpulkan bersama dengan label sebenarnya dari data tersebut. Setiap prediksi model akan diperiksa untuk menentukan apakah benar atau salah berdasarkan label sebenarnya dari data.

Selanjutnya, *Confusion Matrix* dibuat berdasarkan hasil prediksi dan label sebenarnya. Proses pembuatan *Confusion Matrix* ini melibatkan pengelompokan hasil prediksi menjadi empat kategori: *True Positives* (TP), *False Positives* (FP), *True Negatives* (TN), dan *False Negatives* (FN).

Setelah *Confusion Matrix* dibuat, langkah berikutnya adalah menghitung metrik evaluasi performa model, seperti akurasi, *precision*, *recall*, dan F1-*score*.

Tahapan dalam Studi Preskriptif dan Studi Deskriptif II, yaitu pengumpulan data, data *pre-processing*, data *labeling*, *modelling*, dan evaluasi model telah selesai, selanjutnya adalah menyimpan model dengan format joblib dan pickle, untuk keperluan klasifikasi pada situs web deteksi promosi judi daring.

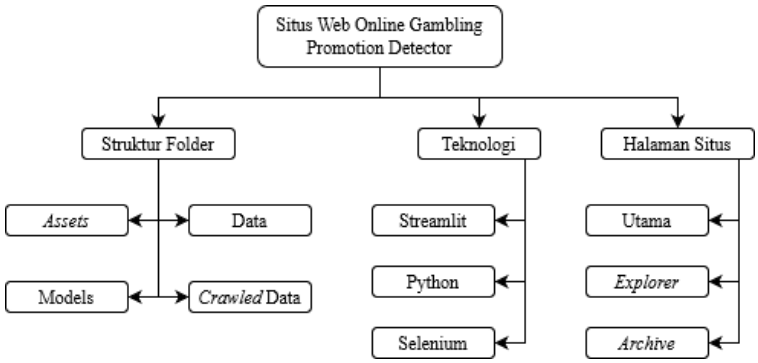
3.1.4.2 Visualisasi dan Interpretasi

Visualisasi dan interpretasi hasil dilakukan dengan bantuan Google Colab dan beberapa *library* Python. Tahapan ini dimulai dengan membuat visualisasi untuk memudahkan pemahaman data yang diklasifikasikan. Pertama, proporsi label positif dan negatif divisualisasikan menggunakan *pie chart*. Interpretasi dari *pie chart* ini menunjukkan keseimbangan atau ketidakseimbangan antara jumlah unggahan promosi judi daring (label positif) dan unggahan non-promosi (label negatif), salah satunya nantinya akan diketahui bagaimana kata kunci "judi" kebanyakan berisi unggahan promosi atau bukan. Kedua, dilakukan analisis kata-kata yang paling sering muncul dalam *dataset* dengan membuat *word cloud* dari

keseluruhan teks, yang menggambarkan frekuensi kata-kata dengan ukuran yang lebih besar menunjukkan frekuensi yang lebih tinggi. Interpretasi dari *word cloud* ini memberikan wawasan tentang kata-kata kunci yang dominan dalam unggahan, baik yang berisi promosi maupun yang tidak. Dari sini juga dapat diketahui *bigram* yang sering digunakan dalam unggahan, yang dapat menjadi wawasan untuk pihak berwenang. Ketiga, dibuat *bar chart* untuk menampilkan frekuensi kata dalam label positif dan negatif, memungkinkan peneliti melihat perbedaan dalam penggunaan kata antar label. Interpretasi dari *bar chart* ini membantu mengidentifikasi kata-kata yang paling sering digunakan dalam unggahan promosi dibandingkan dengan unggahan non-promosi. Terakhir, histogram digunakan untuk mengetahui apakah panjang unggahan berpengaruh terhadap labelnya.

3.1.4.3 Situs Web Online Gambling Promotion Detector

Pengembangan situs web deteksi promosi judi daring melibatkan beberapa langkah. Pertama, menentukan teknologi yang akan digunakan. Teknologi yang tepat untuk pengembangan situs web ini adalah bahasa pemrograman Python, *framework* Streamlit untuk bagian *front-end*, dan Selenium untuk melakukan data *crawling* di media sosial Facebook, Instagram, dan X. Selanjutnya, struktur situs web dibuat dan aset-asetnya dipersiapkan, termasuk folder *assets*, data, *models*, dan folder untuk menyimpan data hasil *crawling*. Gambar 3.13 merupakan rancangan pengembangan situs web.

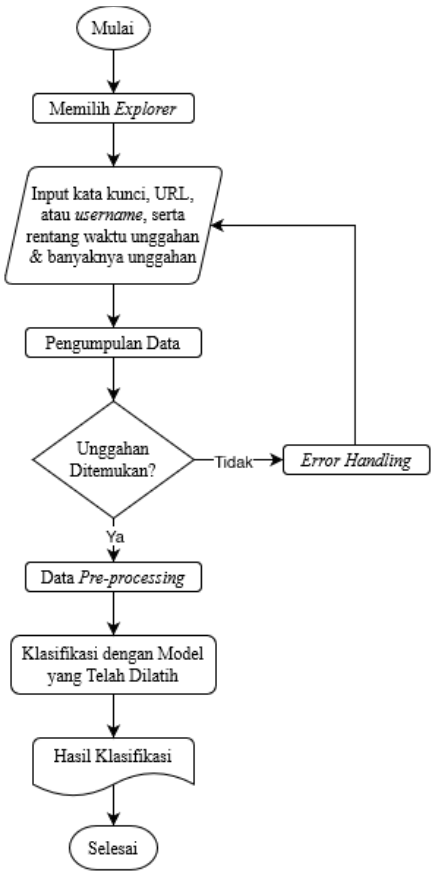


Gambar 3.13 Rancangan Situs Web

Tahap berikutnya adalah pengembangan situs web yang dimulai dengan beberapa langkah. Pertama, pembuatan halaman utama untuk menampilkan visualisasi dan interpretasi data. Kedua, pembuatan halaman *explorer* yang menyediakan berbagai pilihan untuk mengumpulkan dan mengklasifikasikan

konten dari media sosial seperti Facebook (unggahan pribadi, halaman, dan grup), Instagram (unggahan pribadi, unggahan dengan *hashtag*, dan komentar), dan X (unggahan pribadi dan pencarian unggahan). Selain itu, terdapat pilihan *plain text* di mana pengguna dapat memasukkan teks, *file* CSV, atau Excel secara manual.

Pengguna dapat memilih salah satu opsi *explorer* yang akan digunakan. Setelah memilih dan mengisi beberapa *field* yang tersedia, pengguna dapat menekan tombol untuk *crawling* dan klasifikasi. Sistem akan secara otomatis melakukan *crawl*, *pre-processing* hasil *crawl*, dan kemudian melakukan klasifikasi menggunakan model SVM. Hasil klasifikasi ditampilkan dalam bentuk tabel yang menyertakan visualisasi data yang dikumpulkan. Ketiga, dibuat halaman *archive* yang berisi daftar data yang telah dikumpulkan dari setiap media sosial atau *plain text*, yang mana data tersebut dapat dihapus atau diunduh. Selain itu, terdapat bagian *about* yang berisi informasi tentang sistem dan kontak. Penggunaan situs web Online Gambling Promotion Detector ditampilkan pada Gambar 3.14.



Gambar 3.14 *Flowchart* Online Gambling Promotion Detector

Setelah situs web berhasil dikembangkan, dilakukan pengujian untuk menilai kemampuannya dalam mengklasifikasikan informasi judi daring. Pengujian ini memanfaatkan fungsi pengumpulan data untuk mengumpulkan sampel serta alat klasifikasi, yang kemudian akan digunakan untuk menghitung nilai akurasi, *precision*, *recall*, dan F1-*score*.

3.2 Populasi dan Sampel

Populasi penelitian ini terdiri dari unggahan di platform media sosial X yang mencakup kata kunci "judi". Kriteria ini didesain untuk memperoleh konten yang berkaitan dengan perjudian daring, seperti promosi situs judi, pengalaman bermain, ajakan berjudi, atau transaksi terkait. Mengingat "judi" adalah kata kunci berbahasa Indonesia, sampel yang diambil akan terbatas pada unggahan yang memiliki atribut berbahasa Indonesia di platform X. Selain itu, unggahan yang diambil adalah unggahan yang dibuat dalam rentang waktu antara Januari 2019 hingga Juni 2024, yang merupakan tahun di mana meningkatnya kasus perjudian daring.

3.3 Instrumen Penelitian

Instrumen penelitian yang digunakan dalam penelitian ini meliputi kombinasi dari perangkat lunak, perangkat keras, dan instrumen lainnya yang dirancang untuk mencapai tujuan penelitian yang telah ditetapkan.

Perangkat lunak yang digunakan untuk mendukung penelitian ini adalah sebagai berikut:

- Google Colab
- Microsoft Excel
- Microsoft Windows 11 Pro (64-bit)
- Microsoft Visual Studio Code
- Microsoft Edge
- Python
- Streamlit
- Selenium

Berikut adalah daftar perangkat keras yang berperan dalam mendukung penelitian ini:

- Intel® Core™ i5-9300H CPU @ 2.40GHz
- RAM 16 GB DDR4-2666

- Samsung SSD 980 500 GB

Selain instrumen perangkat lunak dan perangkat keras, terdapat instrumen pendukung lain yang turut berkontribusi dalam penelitian ini.

Library Python digunakan untuk meningkatkan efektivitas dan efisiensi dalam seluruh proses, mulai dari pengumpulan data hingga *deployment*. Tabel 3.7 merupakan daftar *library* Python yang digunakan dalam penelitian ini.

Tabel 3.7

Daftar *Library* Python yang Digunakan

Nama *Library*	Versi	Kegunaan
Scikit-Learn	1.5.0	Digunakan untuk pembuatan model *machine learning.*
Pandas	2.2.2	Digunakan untuk manipulasi dan analisis data.
Matplotlib	3.9.0	Digunakan untuk visualisasi data.
Natural Language Toolkit	3.8.1	Digunakan untuk pemrosesan bahasa alami.
Tweet Harvest	2.4.1	Digunakan pada tahap pengumpulan data untuk mengumpulkan unggahan di media sosial X.
NumPy	1.26.4	Digunakan untuk operasi numerik.
Beautiful Soup	4.12.3	Digunakan pada situs web untuk penguraian data dari HTML dan XML.
PySastrawi	1.2.0	Digunakan untuk *stemming* bahasa Indonesia.
Wordcloud	1.9.3	Digunakan untuk membuat visualisasi awan kata.
Streamlit	1.36.0	Digunakan pada tahap *deployment* untuk mengembangkan tampilan situs web agar lebih menarik.
Selenium	4.22.0	Digunakan pada situs web sebagai alat untuk *crawling* media sosial.

Pengumpulan data dilakukan menggunakan Google Colab dan *library* Tweet Harvest dengan metode *web crawling* pada platform X, yang memanfaatkan parameter yang disediakan oleh X sebagaimana tercantum dalam Tabel 3.8.

Tabel 3.8

Parameter Pencarian di X

Parameter	Contoh Penggunaan Parameter	Hasil
since	judi since:2019-01-31	Menampilkan unggahan yang mengandung kata "judi" dan diunggah sejak 1 Januari 2019
until	judi since:2023-12-31	Menampilkan unggahan yang mengandung kata "judi" dan diunggah hingga 31 Desember 2023
lang	judi lang:id	Menampilkan unggahan yang mengandung kata "judi" dan berbahasa Indonesia

Penggunaan parameter seperti yang telah dijelaskan pada Tabel 3.8 dapat digabung dalam sekali pencarian. Contohnya yaitu, "judi lang:id since:2019-01-01 until:2023-12-31", yang berarti menampilkan unggahan yang mengandung kata kunci "judi", berbahasa Indonesia, dan diunggah sejak 1 Januari 2019 hingga 31 Desember 2023.

Hasil pengumpulan data dari X disimpan dengan format CSV dengan berbagai atribut unggahan sebagaimana yang tercantum dalam Tabel 3.9.

Tabel 3.9

Atribut Unggahan di X

Atribut	Penjelasan
created_at	Menunjukkan waktu dan tanggal ketika unggahan dibuat
id_str	ID unik dari unggahan

full_text	Teks penuh dari unggahan
quote_count	Jumlah berapa kali unggahan di-*quote* oleh pengguna lain
reply_count	Jumlah balasan yang diterima unggahan
retweet_count	Jumlah berapa kali unggahan di unggah ulang oleh pengguna lain
favorite_count	Jumlah berapa kali unggahan difavoritkan oleh pengguna lain
lang	Kode bahasa dari teks unggahan
user_id_str	ID pengguna yang membuat unggahan
conversation_id_str	ID percakapan yang di mana unggahan termasuk di dalamnya
username	Nama unik pengguna yang membuat unggahan
tweet_url	URL dari unggahan

Meskipun proses pengumpulan data menghasilkan berbagai atribut dari unggahan, hanya atribut full_text yang digunakan dalam penelitian ini karena ia mengandung unggahan lengkap yang bermanfaat untuk *text mining*.

3.4 Analisis Data

Analisis data dilakukan dengan mempertimbangkan beberapa metrik evaluasi penting yang diperoleh dari *Confusion Matrix* dan *classification report*. *Confusion Matrix* terdiri dari empat komponen utama: *True Positives* (TP), *True Negatives* (TN), *False Positives* (FP), dan *False Negatives* (FN). Masing-masing komponen ini memberikan informasi penting tentang seberapa baik model dalam memprediksi kelas-kelas yang berbeda.

Dari nilai-nilai dalam *Confusion Matrix*, beberapa metrik evaluasi dapat dihitung untuk menilai kinerja model:

- Akurasi mengukur seberapa sering model membuat prediksi yang benar, baik untuk kelas positif maupun negatif. Akurasi dihitung dengan rumus:

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \quad (10)$$

- *Precision* memberikan indikasi seberapa baik model dalam mengidentifikasi positif atau negatif tanpa menghasilkan banyak kesalahan positif atau negatif.

$$Precision^{Positif} = \frac{TP}{TP + FP} \quad (11)$$

$$Precision^{Negatif} = \frac{TN}{TN + FN} \quad (12)$$

- *Recall* memberikan gambaran tentang seberapa baik model dalam mendeteksi kasus positif atau negatif yang sebenarnya.

$$Recall^{Positif} = \frac{TP}{TP + FN} \quad (13)$$

$$Recall^{Negatif} = \frac{TN}{TN + FP} \quad (14)$$

- F1-*score* memberikan ukuran keseimbangan dari *precision* dan *recall*.

$$F1\ score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (15)$$

Analisis mendalam terhadap metrik-metrik ini membantu menilai keefektifan model dalam mendeteksi konten promosi perjudian daring. Sebagai contoh, nilai *precision* yang tinggi menunjukkan bahwa model jarang menghasilkan prediksi positif yang salah, sedangkan nilai *recall* yang tinggi menunjukkan bahwa model mampu mengidentifikasi sebagian besar kasus positif yang ada. F1-*score*, sebagai gabungan dari *precision* dan *recall*, memberikan gambaran yang komprehensif mengenai keseimbangan antara keduanya.

Dengan menganalisis hasil *Confusion Matrix* dan metrik evaluasi seperti akurasi, *precision*, *recall*, dan F1-*score*, peneliti dapat memperoleh pemahaman yang jelas tentang kinerja model dan mengidentifikasi area-area untuk perbaikan lebih lanjut. Hasil analisis ini akan digunakan untuk menyempurnakan model dan meningkatkan akurasi deteksi konten promosi perjudian daring.