BAB III
METODOLOGI PENELITIAN

3.1 Metode Pengembangan *Life* Cycle

Metode *machine learning life cycle* adalah metode pembuatan model klasifikasi yang digunakan pada penelitian ini.

3.1.1 *Model Requirements*

Pada tahap *model requirements* dilakukan Analisa kebutuhan mulai dari pemilihan jenis data sampai dengan menentukan teknologi atau platform yang digunakan untuk pembuatan model seperti pada tabel 3.1 berikut:

Tabel 3.1 *Model Requirements*

No	Requirements	Hasil Analisa
1	Pemilihan Jenis Data	Hasil dari pemilihan jenis data dalam penelitian ini adalah data yang digunakan berupa teks atau tipe data *string* serta sumber atau source data yang digunakan diambil dari situs *website* yang telah di *publish* berdasarkan rentang waktu januari 2019 – agustus 2022.
2	Model Algoritma.	Model algoritma yang digunakan adalah *Gaussian Naïve Bayes*, *Bernoulli Naïve Bayes* dan *Multinomial Naïve Bayes*, yang nantinya dari ketiga algoritma tersebut hanya satu model algoritma yang digunakan berdasarkan hasil akurasi dari model algoritma dengan nilai akurasi tertinggi.
3	Pengukuran Kinerja Model	Untuk memastikan bahwa model bekerja dengan baik maka model *Machine Learning* harus mencakup

No	Requirements	Hasil Analisa
		matrik evaluasi kinerja, seperti akurasi, presisi, *Recall*, dan *F1-score* dll.
4	Teknologi atau platform	*Requirements* ini mencakup bahasa pemrograman serta perangkat lunak yang digunakan. Dimana bahasa pemrograman yang digunakan adalah *Python* serta *software* atau perangkat lunka lainnya yang digunakan yaitu *Anaconda Navigator*, *Jupyter Notebook* dan *Web Scraper*

3.1.2 *Data Collection*

Tahap selanjutnya yaitu proses pengumpulan *dataset* dari *website* atau *blog* yang telah di *publish* melalui *internet*, menurut (Roh dkk., 2021) terdapat 3 teknik pengumpulan data yang akan dilakukan pada penelitian ini yaitu:

1. *Data Discovery*

   Mengumpulkan *dataset* yang sudah ada sebelumnya melalui *platform online* seperti *repository*, *google dataset*, *search engine* dan lain-lain.

   Namun penggunaan *dataset* yang telah tersedia di *platform* atau *repository online* tidak dapat digunakan karena tidak memenuhi syarat dan sehingga tidak dapat digunakan untuk menunjang penelitian ini.
2. *Data Augmentation*

   Mengambil *dataset* tambahan untuk memperluas kumpulan data yang digunakan untuk menghindari *overfitting dataset* dan melakukan *testing* untuk mencari keakurasian *dataset*.

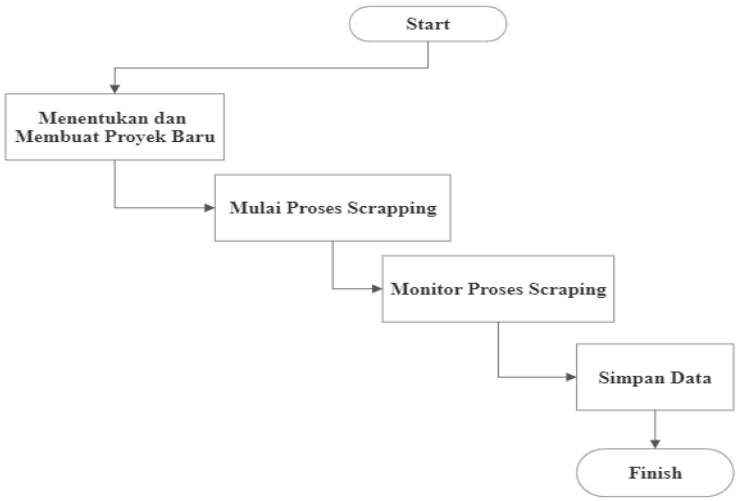
Membuat *dataset* tambahan dengan melakukan teknik *scraping* untuk mengambil data-data yang dibutuhkan dalam *website* dengan menggunakan *API website*.

3. *Data Generation*

   Setelah mengumpulkan *data discovery* dan *data augmentation* sangat besar kemungkinan akan terjadi perubahan *dataset* yang telah dikumpulkan, oleh karena itu pada tahap ini akan dilakukan *data generation* menggunakan teknik *SMOTE* (*Synthetic Minority Over-sampling*) untuk menghasilkan sampel data yang berkualitas dan menghindari kehilangan data secara acak.

Berdasarkan pemaparan dari ketiga poin yang disebutkan sebelumnya maka diambil keputusan bahwa teknik pengumpulan data yang digunakan adalah teknik *data augmentation* dan *data generation*, namun teknik *data generation* yaitu dengan menerapkan teknik *SMOTE* akan diterapkan pada tahap *feature engineering*.

Tahap awal dalam pengumpulan data dari konten *website* dilakukan dengan cara *scraping* data menggunakan *software web scraper*, gambar 3.1 berikut adalah proses yang dilakukan dalam mengambil data atau *scraping* data:



Gambar 3.1 Proses *Scraping*

Berdasarkan gambar 3.1 di atas proses yang dilakukan dalam pengambilan data dapat dipaparkan sebagai berikut:

1. Menentukan dan Membuat Proyek Baru

   Proses ini merupakan proses menentukan situs *web* yang ingin diambil datanya dan memastikan bahwa data tersebut dapat diakses secara publik.

   Sebelum membuat proyek baru terdapat beberapa hal yang dilakukan sebelum menentukan *website* yang akan di *scraping* sebagai berikut:

   a. *Keyword Search Engine*

      Untuk menentukan *website* yang akan diambil datanya dapat dilakukan dengan memasukan *keyword* yang spesifik pada search engine untuk mempermudah dalam menentukan *website* mana yang akan diambil datanya. Contoh dari *keyword* yang digunakan dapat dilihat pada tabel berikut:

Tabel 3.2 *Keyword Search Engine*

Keyword Pornografi	**Keyword Non-Pornografi**
Cerita Sex	Berita Terkini
Cerita Dewasa Hot	Portal Berita
Cerita Dewasa Panas	Website Pengetahuan
Cerita Sex Dewasa Bahasa Indonesia	Ilmu Pengetahuan
Cerita Dewasa	Definisi Bahasa
Cerita Sex Duda	Pengertian Machine Learning

Keyword Pornografi	Keyword Non-Pornografi
Cerita Sex Janda	Tutorial Machine Learning
Cerita Sex Remaja	Dunia Entertainment
Cerita Sex Gay	Dunia Olahraga
Cerita Sex Lesbian	Tutorial Belajar Pyhton

b. *Link Website*

Tujuan awal dari penelitian ini adalah dengan mengklasifikasikan konten *website* berdasarkan konten teks yang ada dalam *website* tersebut, namun tidak menutup kemungkinan bahwa berdasarkan *link* suatu *website* mungkin saja tidak terlihat memiliki unsur pornografi namun ketika membuka halaman *website* tersebut maka dapat ditemukan bahwa halaman *website* tersebut adalah *website* yang mengandung unsur pornografi didalamnya.

Contohnya adalah dimana pada saat peneliti ingin menentukan *website* yang akan di *scraping* ditemukan *link* suatu *website* yaitu http://139.99.33.206/. Tampak sekilas *link website* tersebut terlihat seperti *link website* biasa namun ketika membuka *link* tersebut maka konten yang terdapat dalam *website* tersebut berisi sekumpulan konten-konten pornografi, dan ketika kita mengambil semua data yang lebih detail dari *link* tersbut maka *link url website* yang sebelumnya adalah http://139.99.33.206/ akan menjadi http://139.99.33.206/cerita-sex-ngentot-dipanti-pijat/ http://139.99.33.206/cerita-selingkuh-payudara-vs/ http://139.99.33.206/cerita-dewasa-ngentot-dua-suster/

dan seterusnya tergantung dari jumalah data yang ada pada *website* tersebut. Setelah memperhatikan hal-hal di atas kemudian peneliti siap untuk membuat proyek baru dari situs *web* yang ingin diambil datanya serta menentukan data apa saja yang akan diambil dari *website* tersebut, dalam penelitian ini data yang diambil adalah *link*, isi konten (teks data) dan tanggal *publish website*.

2. Mulai Proses *Scraping*

   Proses selanjutnya adalah memulai proses pengambilan data dari proyek yang dibuat, dalam proses ini dibutuhkan waktu yang cukup lama sesuai dengan jumlah data yang diambil dari halaman *website*.

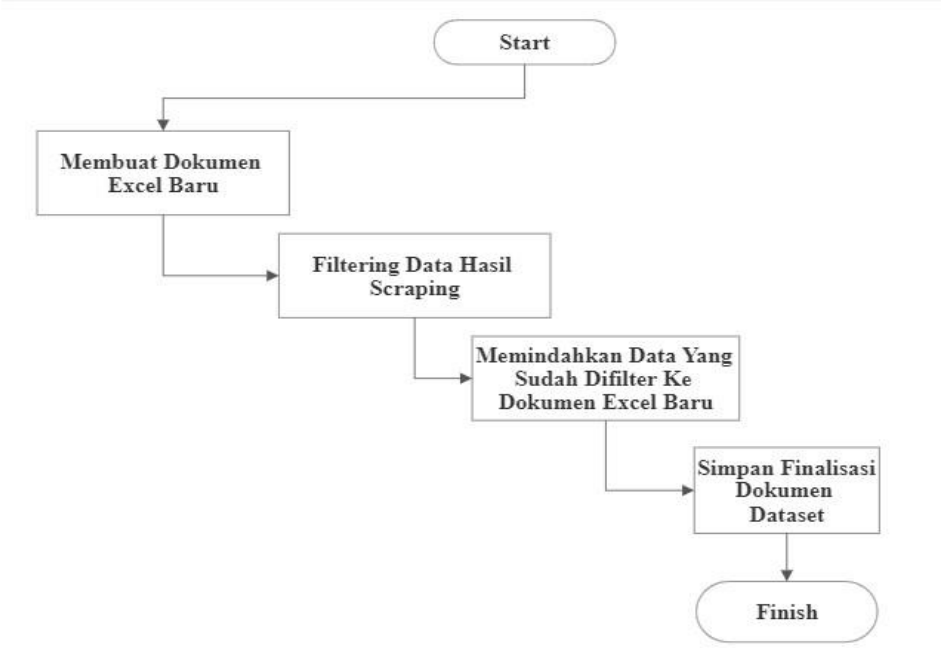
3. *Monitoring* Proses *Scraping*

   Proses *monitoring scraping* ini bertujuan untuk memastikan bahwa data yang diambil sesuai dengan yang diinginkan dan tanpa masalah.

4. Simpan Data

   Proses ini merupakan menyimpan hasil *scraping* data dalam bentuk tabel atau *file csv*.

Setelah proses *scraping* data selesai perlu dilakukannya pengolahan kembali data hasil dari *scraping* sebagai proses finalisasi *dataset* yang siap digunakan untuk pembuatan model *Machine Learning*, proses finalisasi *datatset* yang dilakukan dapat dilihat pada gambar 3.2 berikut:



Gambar 3.2 Finalisasi

Berdasarkan gambar 3.2 di atas proses finalisasi data dapat dijelaskan sebagai berikut:

1. Membuat Dokumen *Excel* Baru

   Proses pertama yang dilakukan dalam tahap finalisasi *dataset* ini adalah dengan membuat *file* dokumen *excel* baru yang bertujuan sebagai *file dataset* yang final yang nantinya digunakan untuk pelatihan model *Machine Learning*.

2. Filtering Data Hasil *Scraping*

   Proses *filtering* ini merupakan proses memilih *dataset* agar sesuai dengan syarat penggunaan *dataset* sebagaimana telah dijelaskan pada batasan masalah yaitu tanggal atau tahun *publish website* yang digunakan adalah *website-website* atau *blog* yang di *publish* dari tahun 2019-2022 dan hal kedua yang dilakukan adalah memfilter data yang tidak berhasil di ambil seperti konten *website* (teks) kosong atau hanya berisi 1 kata serta tanggal atau tahun yang tidak berhasil diambil pada proses *scraping*, tujuan dari proses *filtering*

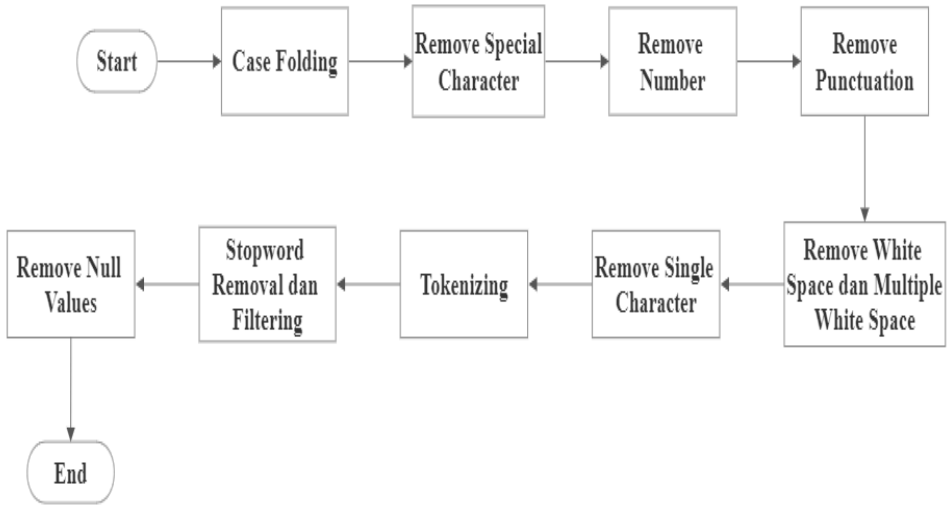
ini adalah untuk memastikan kualitas dan integritas dari *dataset* yang akan digunakan.

3. Memindahkan Data Yang Sudah *Difilter* Ke Dokumen *Excel* Baru
   Setelah proses *filtering* data dilakukan maka data-data yang sudah difilter tersebut kemudian diproses dengan memindahkan data-data tersebut ke *file* dokumen *excel* baru.
4. Simpan Finalisasi Dokumen *Dataset*
   Setelah semua data sudah berhasil terkumpul atau disatukan di dalam *file excel* baru kemudian *file* tersebut disimpan dan diberi nama sebagai finalisasi *dataset*, hal ini menandakan bahwa proses pengumpulan data atau *data collection* telah selesai dilakukan dan data-data tersebut telah siap digunakan sebagai *dataset* untuk pembuatan model *Machine Learning*.

3.1.3 *Data Cleaning*

Pada tahap ini dilakukan pembersihan data yang bertujuan untuk memastikan kualitas minimum dari sampel data seperti akurasi, kelengkapan, konsistensi, keunikan, dan integritas.

Pada tahap *data cleaning* ini terbagi menjadi beberapa proses untuk *cleaning* data, proses-proses tersebut dapat dilihat pada gambar berikut:



Gambar 3.3 *Data Cleaning*

Berikut adalah penjelasan dari gambar 3.3 di atas:

1. *Case Folding*

   Case folding adalah proses perubahan huruf kapital menjadi huruf kecil, jadi semua teks yang mengandung huruf kapital akan diubah menjadi huruf kecil. Tujuan dari *case folding* ini adalah agar kata-kata yang sama tidak terdeteksi berbeda hanya karena perbedaan terdapat huruf kapital.

2. *Remove Special Character*

   Remove special character (atau karakter khusus) merupakan teknik pengolahan teks yang digunakan untuk menghapus karakter-karakter tertentu pada suatu teks, seperti karakter non-alfanumerik, simbol, tanda baca, atau karakter *special* lainnya. Tujuannya adalah untuk membersihkan teks dari karakter-karakter yang tidak diperlukan yang dapat mengganggu analisis teks selanjutnya.

3. *Remove Number*

   Remove number merupakan proses menghapus angka yang ada dalam *dataset* yang berfungsi untuk berfokus pada konten teks dan mengurangi dimensi data.

4. *Remove Punctuation*

   Remove punctuation merupakan proses menghapus tanda baca yang terdapat dalam *dataset* seperti titik (.), koma (,) dll.

5. *Remove White Space dan Multiple White Space*

   Remove white space merupakan proses menghapus spasi ekstra dalam *string* teks, seperti spasi di awal atau akhir *string*, atau banyak spasi di antara kata, sedangkan *remove multiple white space* merupakan proses mengganti beberapa spasi berturut-turut dalam *string* teks dengan satu spasi. Ini sering dilakukan untuk membuat teks lebih mudah dibaca dan untuk membakukan pemformatan.

6. *Remove Single Character*

   Remove single character adalah sebuah proses untuk menghapus karakter tunggal atau karakter yang hanya muncul sekali dalam sebuah teks atau *string*. Proses ini digunakan untuk membersihkan atau menyederhanakan teks, terutama jika karakter tunggal tersebut tidak memberikan informasi yang signifikan atau dapat mempengaruhi analisis teks yang dilakukan.

7. *Tokenizing*

   Tokenizing merupakan proses memisahkan atau memecahkan yang awalnya berupa kalimat menjadi kata-kata atau memutus urutan *string* menjadi potongan-potongan kata seperti kata-kata berdasarkan tiap kata yang menyusunnya. *Tokenizing* digunakan untuk memperoleh potongan kata menjadi suatu entitas dan memiliki nilai dalam penyusunan matriks dokumen pada proses berikutnya.

8. *Stopword Removal dan Filtering*

   Proses *stopword removal* dan *filtering* dilakukan karena sebagian besar di dalam data teks terdapat kata-kata umum yang tidak memiliki makna dan tentunya ini akan mempengaruhi keakuratan hasil analisis dan biasanya kata-kata ini muncul dalam frekuensi yang cukup banyak. Contoh *stopword* dalam kata Bahasa Inggris misalnya adalah *are, is, i, am, was, were, they, you, the,* dan lain sebagainya. Dan untuk *filtering* merupakan proses manual menambahkan *stoplist* kata sebagai *stoplist* tambahan sebagai kata yang tidak memiliki arti.

9. *Stemming*

   Proses *stemming* dilakukan untuk menghilangkan *suffix* dan *prefix* pada suatu teks. Di mana proses ini mengambil kata dasar dari teks tersebut, contohnya amazing menjadi amaze.

10. *Remove Null Values*

    Menghapus *null value* merupakan proses menghilangkan data yang tidak memiliki nilai atau *null* dalam *dataset*.

3.1.4 *Data Labeling*

Pada tahap ini dilakukan labeling pada *dataset* yang telah dikumpulkan apakah termasuk dalam kategori pornografi atau non-pornografi dengan menggunakan daftar kata-kata pornografi dan non-pornografi yang telah dimodifikasi dan disesuaikan dengan tema dari *dataset*. Dalam tahap *data labeling* ini terdapat beberapa proses yang dilakukan seperti pada gambar 3.4 berikut:

Gambar 3.4 Proses *Data Labeling*

Dari gambar 3.4 di atas dapat dijelaskan bahwa:

1. Kamus Kata

   Proses membuat kamus kata pornografi dan non-pornografi adalah sebuah proses yang melibatkan identifikasi dan klasifikasi kata-kata berdasarkan sifat atau sentiment yang terkait dengan kata tersebut.

2. Hitung Total Kata

   Proses menghitung jumlah kata pada sebuah dokumen adalah proses yang melibatkan identifikasi dan pengurutan kata-kata yang ada dalam sebuah teks. Dalam konteks penelitian ini, menghitung

jumlah kata pada dokumen digunakan untuk menganalisis frekuensi kata yang ada dalam setiap dokumen.

3. Jumlah Kata Pornografi dan Non-Pornografi

   Setelah menghitung total kata dalam dokumen proses selanjutnya adalah menghitung frekuensi kemunculan kata dalam dokumen berdasarkan kamus kata pornografi dan non-pornografi.

4. Persentase Pornografi dan Non-Pornografi

   Persentase kata pornografi dan non-pornografi adalah suatu ukuran yang menunjukkan perbandingan jumlah kata pornografi dan jumlah kata non-pornografi dalam sebuah dokumen. Ini merupakan indeks yang membantu untuk mengukur kadar konten pornografi dalam suatu dokumen tekstual.

   Persentase ini dapat dihitung dengan cara membagi jumlah kata pornografi dalam dokumen dengan jumlah total kata dalam dokumen. Hasil dari perhitungan ini dapat dinyatakan dalam bentuk persentase, misalnya “50%” kata dalam dokumen adalah pornografi.

5. *Labeling Data*

   Setelah mnghitung Persentase pornografi dan non-pornografi pada dokumen, proses selanjutnya yang dilakukan adalah pelabelan data apabila hasil perhitungan dari Persentase kata pornografi lebih besar dari Persentase kata non-pornografi maka dokumen tersebut diberi label pornografi (YA) dan apabila Persentase kata non-pornografi lebih besar dari Persentase kata pornografi maka dokumen akan diberi label non-pornografi (TIDAK).

   Setelah ditentukanya label dalam keseluruhan dokumen. Proses selanjutnya adalah mengubah label pada dokumen yang tadinya berupa kata atau tipe data *string* akan diubah menjadi tipe data *numeric* atau *integer* yaitu 0 dan 1 dimana 0 merupakan nilai untuk label non-pornografi dan 1 untuk label pornografi.

3.1.5 *Feature Engineering*

Proses *Feature Engineering* dilakukan dengan memecah data menjadi data *training* dan *testing*. Untuk data *training* berjumlah 70% dari keseluruhan data sebagai data *training* dan 30% dari keseluruhan dari *data testing*. Pada tahap *feature engineering* juga menerapkan dua metode *engineering* yaitu *TFIDF* dan *SMOTE*, kedua teknik tersebut dapat dijelaskan sebagai berikut:

1. *Term Frequency – Inverse Document Frequency* (*TFIDF*)
   TFIDF merupakan proses pengukuran statistik yang mengevaluasi seberapa relevan suatu kata dengan dokumen dalam kumpulan dokumen. Hal ini dilakukan dengan mengalikan dua matrik yaitu berapa kali sebuah kata muncul dalam dokumen, dan frekuensi *inverse* dokumen dari kata tersebut diseluruh kumpulan dokumen. Namun berbeda halnya apabila menghitung *TFIDF* pada *python* atau dalam proses membuat model klasifikasi sehingga rumus untuk menghitung *TFIDF* pada *python* dapat dilihat pada persamaan-persamaan berikut:

   $$TF = ft, j \quad (3.1)$$

   Dimana TF adalah frekuensi (f) dari (t) di dalam dokumen (j).

   $$BobotTF(term, j) = \frac{"term" \, frek \, dalam \, dokumen \, j}{total \, kata \, dalam \, dokumen \, j} \quad (3.2)$$

   Dimana "$term$" $frek \, dalam \, dokumen \, j$ adalah jumlah kemunculan "$term$" $dalam \, dokumen \, j$ dan "$total \, kata \, dalam \, dokumen \, j$" adalah jumlah total kata dalam dokumen j.

   Persamaan yang digunakan untuk menghitung *IDF* adalah sebagai berikut:

$$IDF(term) = LOG\frac{\text{total dokumen}+1}{dok\ yg\ mengandung\ term + 1} + 1 \quad 3.3$$

Dimana $total\ dokumen\ +\ 1$ adalah jumlah total dokumen dalam *dataset* ditambah satu dan $dok\ yg\ mengandung\ term + 1$ adalah jumlah total dokumen yang mengandung term dalam *dataset* ditambah satu.

Untuk menghitung bobot *TFIDF* dapat dilakukan menggunakan persamaan berikut:

$$TFIDF(term) = TF(ft, j) * IDF(term) \quad 3.4$$

Rumus TF-IDF pada dasarnya digunakan untuk menentukan bobot kata dalam sebuah dokumen dalam kumpulan dokumen *(corpus)* yang lebih besar. Tujuannya adalah untuk memberikan nilai yang lebih tinggi pada kata-kata yang lebih penting atau lebih jarang muncul dalam dokumen tersebut.

Dalam implementasi pada *python*, menambahkan nilai +1 pada rumus *IDF* bertujuan untuk menghindari kemungkinan terjadinya pembagian dengan nilai 0 jika ada kata yang tidak muncul dalam dokumen tersebut. Dalam rumus *IDF*, bobot sebuah kata didefinisikan sebagai jumlah kemunculan *term* tersebut dalam dokumen tersebut dibagi dengan total jumlah dokumen yang mengandung *term* tersebut. Jika ada *term* yang tidak muncul sama sekali dalam dokumen tersebut, maka pembagian dengan nilai 0 akan menghasilkan error.

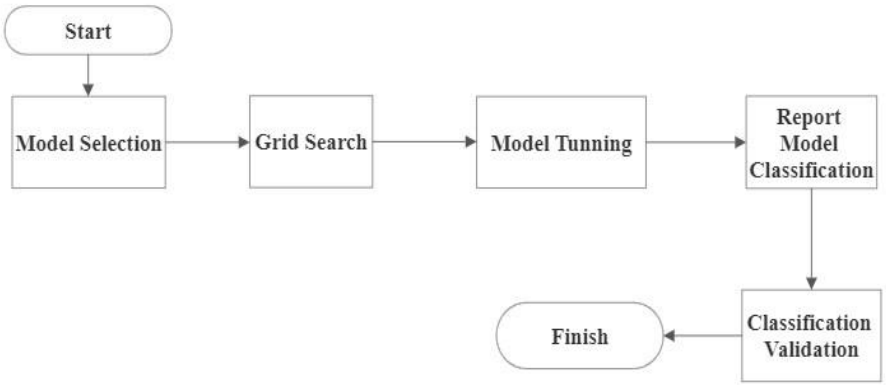
Dengan menambahkan nilai +1 pada rumus *IDF*, dapat memastikan bahwa pembagian tidak akan menghasilkan nilai 0, sehingga nilai bobot dapat dihitung dengan benar.

2. *Synthetic Minority Over-Sampling* (*SMOTE*)

   Teknik *Synthetic Minority Over-Sampling* (*SMOTE*) adalah teknik statistik untuk meningkatkan jumlah data dalam himpunan *dataset* dengan cara yang seimbang, penerapan teknik *SMOTE* ini bertujuan untuk menghindari terjadinya *overfitting* dan *underfitting* pada model *Machine Learning*.

3.1.6 *Model Training*

Model training merupakan proses pelatihan model *Machine Learning* menggunakan algoritma *Naïve Bayes*. Dalam tahap ini terdapat beberapa proses yang dilakukan proses-proses tersebut dapat dilihat pada gambar 3.5 berikut:



Gambar 3.5 Proses *Model Training*

Proses-proses yang ada pada gambar 3.5 dapat diberi penjelasan sebagai berikut:

1. *Model Selection*

   Proses *model selection* ini berfungsi untuk menentukan model yang akan digunakan untuk menjadi model *Machine Learning*, terdapat tiga model yang digunakan untuk model *Machine Learning* dalam penelitian ini model tersebut adalah *Gaussian Naïve Bayes, Bernoulli Naïve Bayes* dan *Multinomial Naïve Bayes.*

2. *Grid Search*

   Proses *grid search* merupakan proses yang dilakukan untuk mencari parameter terbaik dari model *Machine Learning* yang disebutkan pada proses *model selection*.

3. *Model Tuning*

   Proses *model tuning* merupakan proses mengimplementasikan *hyperparameter* pada model *machine learning* yang telah dipilih berdasarkan pemilihan model pada *model selection* dan pencarian parameter terbaik pada proses *grid search.*

4. *Report Model Classification*

   Report model classification merupakan proses menampilkan hasil dari model klasifikasi secara otomatis ke dalam kategori atau kelas yang telah ditentukan. Tujuan menampilkan hasil model klasifikasi adalah untuk mengkategorikan hasil klasifikasi secara akurat dan konsisten ke dalam kelas yang benar, *Report* atau hasil klasifikasi adalah sebagai berikut:

   a. *Overfitting* dan *Underfitting* Model

      Pengecekan *overfitting* dan *underfitting* dimana *overfitting* terjadi ketika model dilatih terlalu baik pada data pelatihan dan berkinerja buruk pada data yang tidak terlihat atau *data testing*. Ini karena model telah mempelajari *noise* dalam data pelatihan dan menjadi terlalu terspesialisasi untuknya. *Underfitting* terjadi ketika model tidak mampu menangkap pola dasar dalam data pelatihan dan berkinerja buruk baik pada data pelatihan maupun data yang tidak terlihat. Ini karena modelnya terlalu sederhana dan belum mempelajari hubungan yang diperlukan dalam data.

   b. *Null Accuracy*

      Null accuracy merupakan nilai akurasi yang didapatkan berdasarkan frekuensi kelas prediksi yang paling sering muncul pada label kelas.

c. *Confusion Matrix*

Confusion matrix merupakan teknik yang memberikan kesimpulan dari hasil klasifikasi dengan menampilkan dan membandingkan nilai aktual atau nilai sebenarnya dengan nilai hasil prediksi model yang dapat digunakan untuk menghasilkan matrik evaluasi seperti *Accuracy* (akurasi), *Precision*, *Recall*, dan *F1-Score*. Berikut adalah penjelasannya:

- *Accuracy*

  Nilai akurasi didapatkan dari jumlah data bernilai positif yang diprediksi positif dan data bernilai negatif yang diprediksi negatif dibagi dengan jumlah seluruh data di dalam *dataset*.

- *Precision*

  Precision adalah peluang kasus yang diprediksi positif yang pada kenyataannya termasuk kasus kategori positif.

- *Recall*

  Recall adalah peluang kasus dengan kategori positif yang dengan tepat diprediksi positif.

- *F1-Score*

  Nilai *F1-Score* didapatkan dari hasil *Precision* dan *Recall* antara kategori hasil prediksi dengan kategori sebenarnya.

5. *Classification Validation*

Pada proses ini akan dilakukan pengecekan atau validasi dari model klasifikasi yang telah dibuat menggunakan dua teknik atau cara sebagai berikut:

a. Kurva *ROC*

Kurva *ROC* (*Receiver Operating Characteristic*) untuk mengevaluasi ambang batas yang berbeda untuk masalah pembelajaran mesin klasifikasi.

b. *Cross Validation*

Cross Validation adalah teknik untuk mengevaluasi kinerja model *Machine Learning* dengan membagi data yang tersedia menjadi *set* pelatihan dan validasi, melatih model pada *set* pelatihan dan mengevaluasi kinerjanya pada *set* validasi. Proses diulang berkali-kali dengan partisi data yang berbeda teknik *cross validation* yang digunakan adalah *K-Fold Cross-Validation* dimana data dibagi menjadi *k fold* yang masing-masing digunakan satu kali untuk validasi dan *k-1* kali untuk *training*.

3.1.7 *Model Evaluation*

Model evaluation Pada tahap ini dilakukan analisa untuk mendapatkan dan mengkonfirmasi apakah model *Machine Learning* yang telah dibuat sudah sesuai dengan tujuan ataukah belum dengan cara membandingkan tingkat akurasi data dengan pendapat ahli bahasa Indonesia.

Sebagai contoh: Jika teks 1 hasil dari algoritma adalah pornografi, maka akan dibandingkan dengan pendapat ahli bahasa Indonesia.

3.1.8 Operation

Operation pada *machine learning* tidak terlepas dari tahap *environment*, *deployment*, dan *monitoring*/*updating*. Berikut penjelasan singkat mengenai hal-hal di atas:

1. *Environment* (Lingkungan)

   Environment pada *machine learning* mengacu pada infrastruktur dan *tools* yang digunakan untuk mengeksekusi model *machine learning*. Ini mencakup *software*, *hardware*, dan konfigurasi untuk

menjalankan model. Hal ini sangat penting untuk memastikan bahwa model *machine learning* berjalan dengan baik tanpa *error.*

2. *Deployment* (Penerapan)

   Penerapan pada *machine learning* berkaitan dengan melakukan pemindahan model dari lingkungan pengembangan ke lingkungan produksi. Di sinilah model dikemas menjadi sebuah *software* atau layanan yang bisa digunakan secara langsung oleh pengguna. *Deployment* pada *machine learning* termasuk memilih lingkungan yang tepat, mengumpulkan data, dan menetapkan waktu pembaruan. Hal ini sangat penting memastikan model dapat digunakan dengan baik dan benar, menjaga keamanan, serta meningkatkan performa dan efisiensi.

3. *Monitoring*/*Updating* (Pemantauan/Pembaruan)

   Pemantauan dan Pembaruan pada *machine learning* berkaitan dengan memantau kinerja model *machine learning* yang sudah diterapkan dan memperbaiki model ketika diperlukan. Dalam *machine learning*, perubahan data yang mendasar dapat mempengaruhi tingkat akurasi model dan oleh karena itu *monitoring* dilakukan secara terus-menerus. Jika model menunjukan kinerja yang menurun atau model yang mengalami kesalahan ketika memprediksi data masa depan, model harus segera diperbaharui dengan meng-*updatenya*.