

## BAB 3

### METODOLOGI PENELITIAN

#### 3.1 Analisis Kebutuhan

Penelitian ini dilakukan dengan beberapa tahap sebagai berikut:

##### 1. Studi Literatur

Pada tahap pertama, dilakukan dengan melakukan identifikasi terhadap masalah. Di mana peneliti akan melakukan pendalaman terhadap teori-teori yang sudah ada, dengan membaca berbagai referensi yang memiliki tema atau topik yang sama dengan penelitian.

##### 2. Analisis Kebutuhan

Peneliti akan menganalisis kebutuhan yang diperlukan dalam proses penelitian. Di mulai dari kebutuhan *dataset*, *library* untuk membangun program dan *software* dan *hardware* yang dibutuhkan.

##### 3. Pengumpulan Data

Mencari dan mengumpulkan *dataset* yang sesuai dengan topik. *Dataset* yang dicari harus memiliki label sebagai berita asli dan berita *hoax* di dalamnya. *Dataset* didapatkan dari penelitian yang dilakukan oleh Faisal Rahutomo, Ingrid Yanuar dan Rosa Andrie Asmara yang berjudul “Online News Classification Using Multinomial Naive” pada tahun 2017. *Dataset* memiliki 600 data dengan 372 label tidak *hoax* dan 228 label *hoax*.

##### 4. Perancangan

Peneliti membangun program penelitiannya menggunakan program jupyter atau google colab untuk melakukan proses penelitiannya. Di mulai dari *text*

*processing*, perhitungan *TF-IDF*, membangun model sesuai dengan algoritma yang telah dipilih, lalu mengevaluasi model.

#### 5. Pengujian

Program yang sudah dibangun akan diuji menggunakan *dataset* yang telah dikumpulkan sebelumnya.

#### 6. Evaluasi

Mengevaluasi model yang telah dibuat, apakah sesuai dengan kebutuhan penelitian dan berjalan sesuai. Evaluasi dilakukan menggunakan *F1-Score* untuk dilihat performa *score* yang dihasilkan dari masing–masing model.

#### 7. Dokumentasi

Melakukan dokumentasi program yang telah dibuat. Di mulai dari dokumentasi *code*, hasil performa yang dihasilkan dari masing–masing model dan evaluasi yang dihasilkan tiap model hingga laporan.

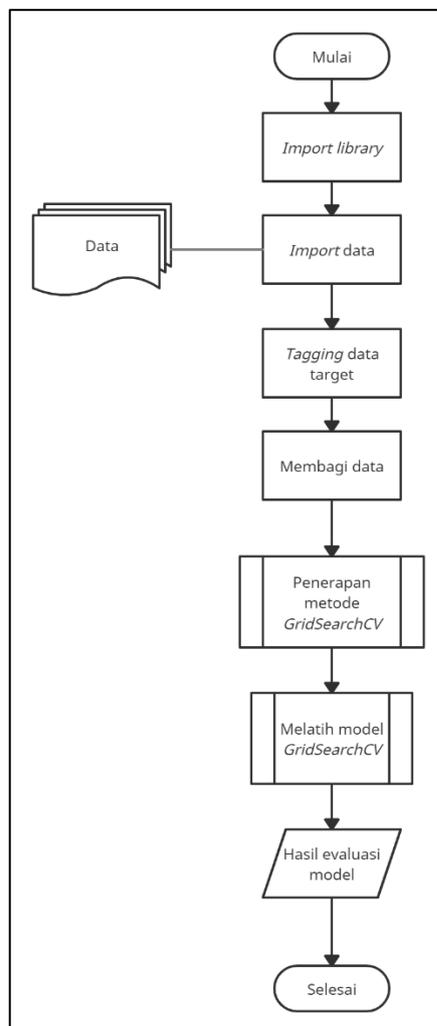
### **3.2 Perancangan Aplikasi**

Perancangan sistem dijelaskan dalam bentuk *flowchart* yang dibagi menjadi beberapa tahap yaitu *flowchart* utama, *flowchart* penerapan metode *GridSearchCV*, *flowchart* melatih model *GridSearchCV*, *flowchart text preprocessing* dan *flowchart TF-IDF*.

#### **3.2.1 Flowchart Utama**

Pada Gambar 3.1 menunjukkan gambar *flowchart* dari sistem yang akan dibuat. Langkah pertama adalah dengan melakukan *import library* yang dibutuhkan di dalam program. Setelah itu *import dataset* yang telah dikumpulkan ke dalam program. Lalu dari data yang telah diimpor, akan dilakukan proses *tagging*. Hal ini

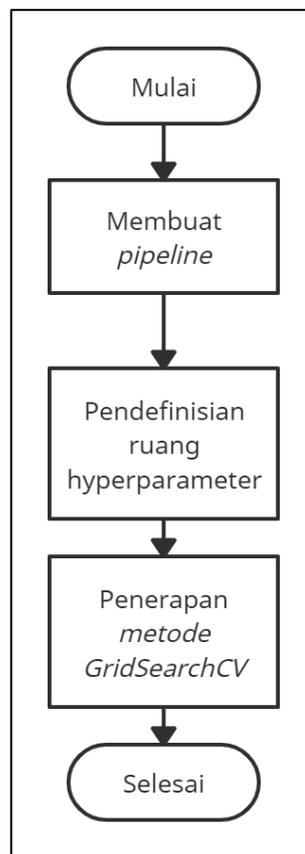
dilakukan untuk mengubah data target yang memiliki bentuk teks menjadi angka, dengan mengubah data target dengan teks 'Valid' menjadi '1' dan Hoax menjadi '0'. Setelah itu data akan dibagi menjadi dua bagian yaitu X sebagai fitur data dan y sebagai target. Lalu menerapkan metode *GridSearchCV* yang akan menjadi proses pembuatan model dan *hyperparameter*. Setelah itu akan dilakukan *fit* model yang akan dibagi menjadi beberapa proses yaitu *text preprocessing*, *TF-IDF* dan *fit* model dan evaluasi model. Lalu akan memunculkan *output F1-Score* dari model dan parameternya.



Gambar 3.1 Flowchart Utama

### 3.2.2 Flowchart Penerapan Metode GridSearchCV

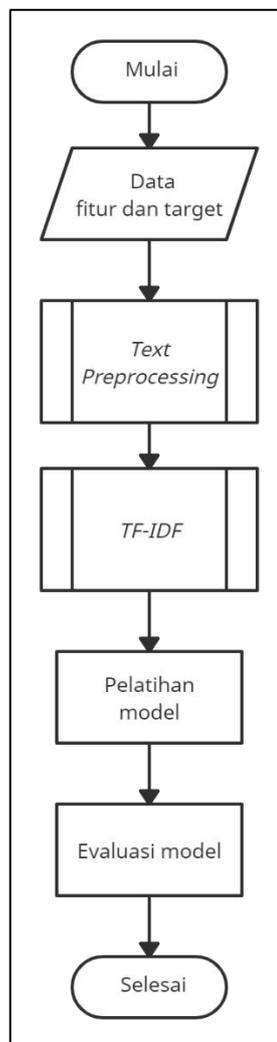
Pada Gambar 3.2 dijelaskan proses penerapan metode *GridSearchCV*. Proses ini berawal dari pembuatan *pipeline* dengan bantuan *library* dari *pipeline*. *Pipeline* ini berguna untuk mengumpulkan model yang berisikan *TF-IDF*, *classifier* dan *recursive feature elimination*. Lalu mendefinisikan ruang *hyperparameter* dari masing-masing model dari *pipeline*. Lalu menerapkan metode *GridSearchCV* dengan parameter yang sesuai dengan masing-masing *classifier* dengan bantuan *library* dari *GridSearchCV*.



Gambar 3.2 Flowchart Penerapan Metode GridSearchCV

### 3.2.3 Flowchart Melatih Model GridSearchCV

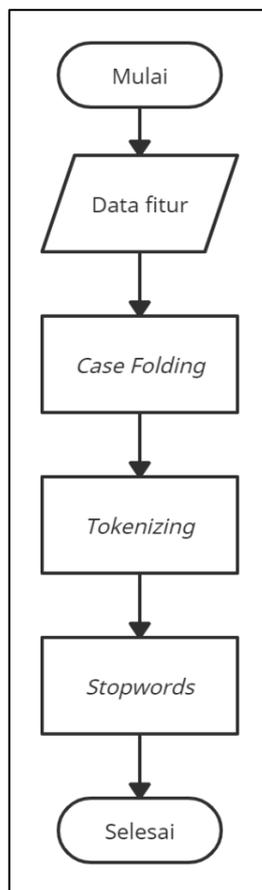
Pada Gambar 3.3 dijelaskan proses pelatihan model. Proses ini dimulai dengan memasukkan data X sebagai data fitur dan y sebagai target. Lalu dilanjutkan dengan memproses data tersebut melalui proses *text preprocessing*, untuk mengubah data yang masih berbentuk teks panjang menjadi *token* atau *term*. Setelah itu data tersebut akan melalui proses *TF-IDF* untuk mengubah data yang berbentuk teks menjadi angka dengan bantuan *library* dari *TfidfVectorizer*. Lalu data tersebut sudah bisa melalui proses pelatihan model.



Gambar 3.3 Flowchart Melatih Model GridSearchCV

### 3.2.4 Flowchart Text Preprocessing

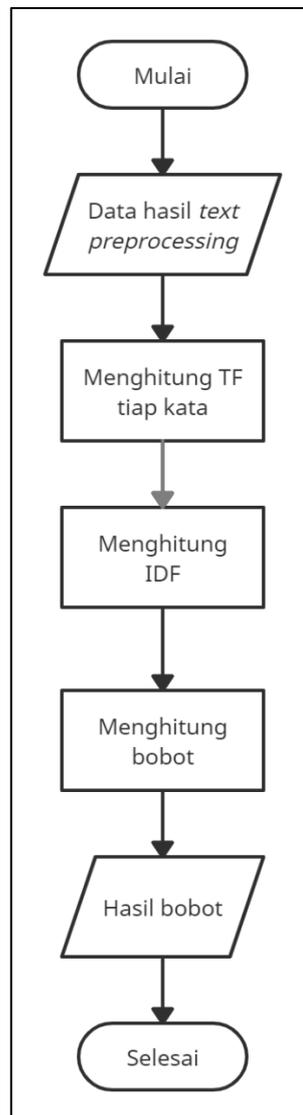
Pada Gambar 3.4 dijelaskan proses *text preprocessing*. Proses dimulai dari mengambil data. Semua teks pada data tersebut akan diubah menjadi huruf kecil. Setelah itu dilakukan proses *tokenizing* yang merupakan proses untuk memisahkan atau memecah teks menjadi terpisah-pisah yang akan disebut *token* atau *term*. Lalu *token* tersebut akan melalui proses *stopwords removing* dengan bantuan *library* dari *stopwords* untuk membuang kata yang tidak memiliki arti penting di dalam proses *text preprocessing*.



Gambar 3.4 Flowchart Text Preprocessing

### 3.2.5 Flowchart TF-IDF

Pada Gambar 3.5 dijelaskan proses *TF-IDF* untuk mengubah *token* yang didapatkan dari proses *text preprocessing* menjadi nilai yang dapat digunakan oleh model. Lalu proses selanjutnya dilakukan dengan mencari nilai TF, DF dan IDF. Lalu menghitung *weight* setiap *term* dari nilai-nilai yang sudah dihitung sebelumnya. Lalu menghasilkan *weight* pada tiap *term*.



Gambar 3.5 Flowchart TF-IDF