BAB 3

METODOLOGI PENELITIAN

3.1 Metodologi Penelitian

Penelitian "Implementasi Algoritma Random Forest untuk Klasifikasi Kategori Berita" terbagi menjadi beberapa proses tahapan sehingga dapat terlaksana dengan baik, yaitu sebagai berikut.

a. Identifikasi dan Perumusan Masalah

Permasalahan yang diangkat untuk diteliti adalah bagaimana membuat klasifikasi kategori berita dengan algoritma *Random Forest* berbasis *Word2Vec* dan nilai performanya berdasarkan perhitungan *accuracy* dan F1-*Score*.

b. Telaah Literatur

Pada tahap ini peneliti akan mencari informasi terkait penelitian yang dibutuhkan untuk menyelesaikan masalah yang dijabarkan pada rumusan masalah. Teori tentang berita yang menjadi objek klasifikasi dalam penelitian ini, *Random Forest* yang merupakan algoritma klasifikasi yang digunakan untuk mengklasifikasikan berita, *Word2Vec* yang merupakan algoritma yang digunakan untuk *word embedding* teks berita tersebut, *Text Preprocessing* yang merupakan metode untuk mengolah data, dan *Confusion Matrix* untuk menguji performa dari model klasifikasi yang telah dibuat.

c. Pengumpulan *Dataset*

Tahap dilakukannya *webcrawling* pada https://indeks.kompas.com/ untuk mendapatkan judul dan konten berita serta kategori berita untuk dijadikan data latih

dan data tes pada model yang akan dibuat. Berita yang dikumpulkan dibatasi hanya beberapa kategori, yaitu money, tekno, sains, otomotif, lifestyle, properti dan travel.

d. Pemrograman Sistem

Berdasarkan permasalahan yang disebutkan sebelumnya, dilakukan analisa untuk mengimplementasikan algoritma *Random Forest* dengan tepat dan sesuai. Sistem akan dimulai dengan melakukan pra-pemrosesan pada data yang telah di-*crawl*, pembuatan model, dan pengaplikasiannya dalam bentuk *website*.

e. Implementasi Algoritma

Algoritma *Random Forest* akan diimplementasikan untuk melakukan klasifikasi dari judul dan/atau konten berita yang dimasukan. Untuk mendapatkan nilai *hyperparameter* terbaik dilakukan *RandomSearchCV* pada model dan *dataset* yang telah diproses untuk mengurangi bias.

f. Evaluasi Hasil Implementasi Algoritma

Setelah dilakukan implementasi, algoritma harus dievaluasi untuk melihat performa dari model yang telah dibuat dalam mengklasifikasikan konten atau judul berita yang dimasukan. Baik tidaknya performa model dilihat dari nilai *accuracy* dan F1-*Score* yang dihasilkan saat dilakukan testing. Semakin mendekati nilai 1, maka performa model semakin baik.

g. Penyusunan Laporan

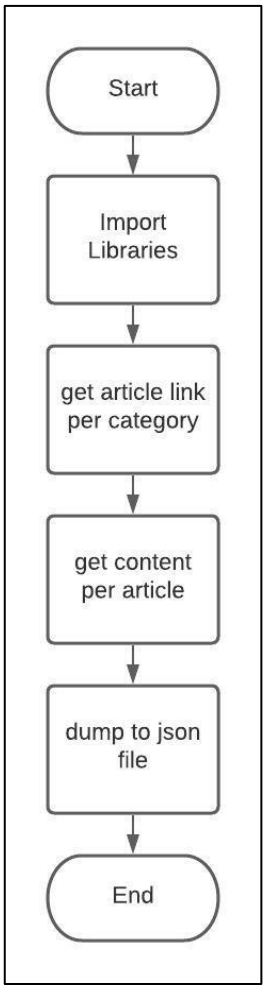
Tahap akhir dalam penelitian ini adalah penyusunan laporan yang meliputi hasil keseluruhan penelitian yang telah dilakukan sebagai dokumentasi. Laporan disusun dengan kaidah penyusunan dan penulisan laporan ilmiah yang sudah ditentukan, dimulai dari pendahuluan hingga kesimpulan dan saran.

3.2 Perancangan Sistem

Sistem yang dihasilkan dari penelitian ini berbasis *website* menggunakan *Anvil*. Sebelum diimplementasikan dalam bentuk *website*, dilakukan perancangan *flowchart* yang terdiri dari *flowchart webcrawling dataset*, *flowchart* untuk pembuatan model *Random Forest Classifier*, *flowchart website*, dan rancangan tampilan antarmuka *website*.

3.2.1 Flowchart Webcrawling Dataset

Tahap awal penelitian dilakukan pengumpulan data kategori dan konten artikel sebagai *dataset* dalam membuat model *Random Forest Classifier*. Pengumpulan data dilakukan dengan melakukan *webcrawling* pada portal berita Kompas. *Flowchart* dapat dilihat pada Gambar 3.1.



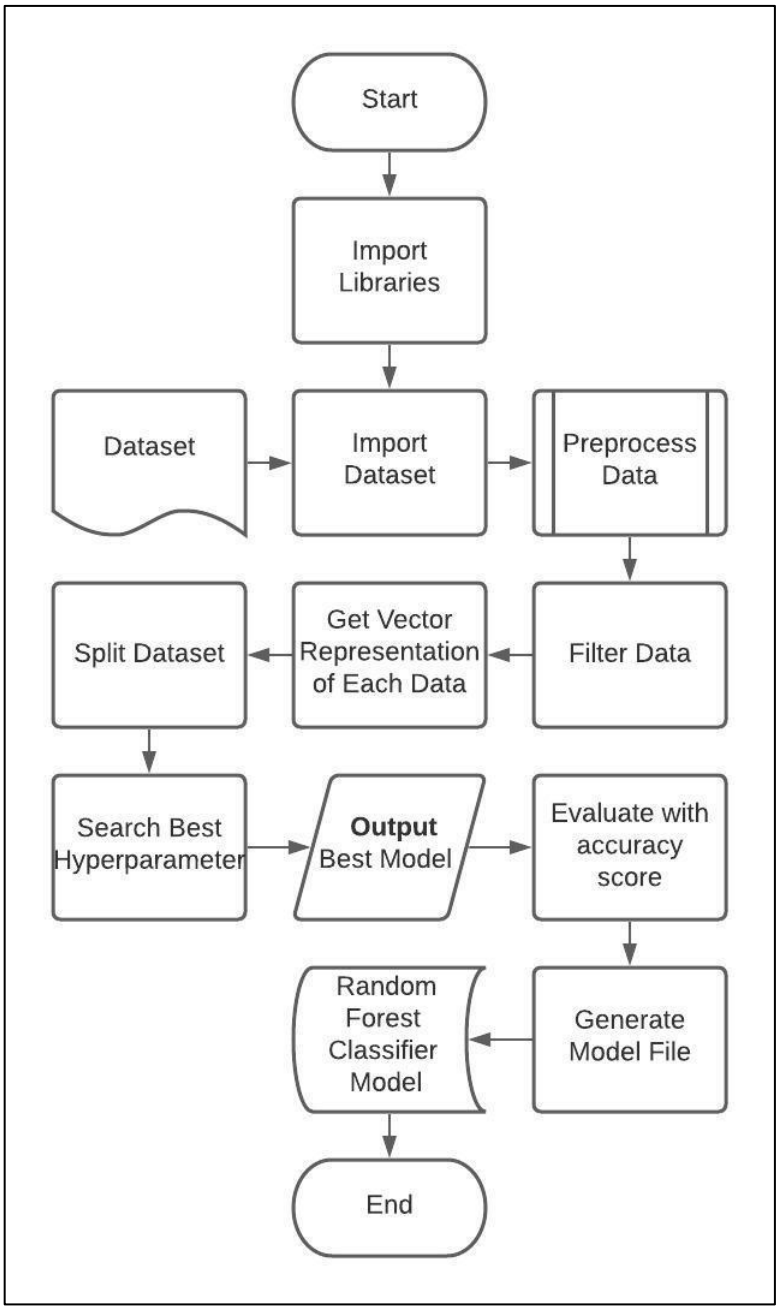
Gambar 3. 1 Flowchart *webcrawling dataset*

Hal yang dilakukan pertama adalah *import libraries* dimana *library* utama yang digunakan untuk melakukan *crawling* adalah BeautifulSoup dari bs4 dan *library* pendukung yaitu request yang berguna untuk membangun koneksi dengan halaman web dan menerima respon berupa teks html.

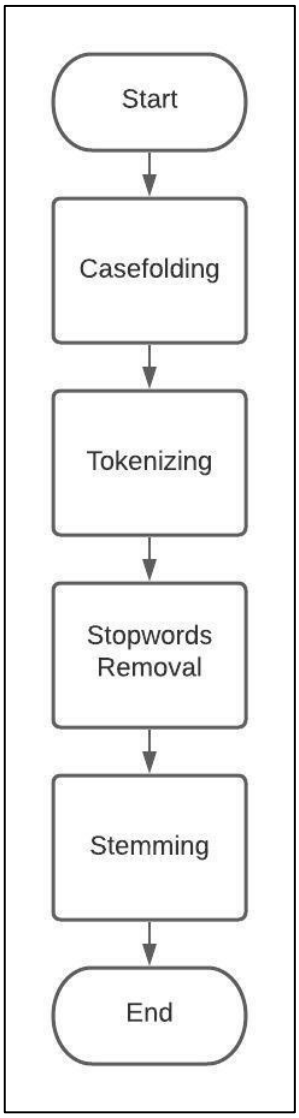
Webcrawling disini dapat dibagi menjadi dua fase. Fase pertama adalah mengambil semua *link* artikel pada suatu kategori. *Link* ini kemudian akan digunakan untuk mencari konten dari artikel yang dirujuknya. Fase kedua adalah mengambil judul dan konten berita dari teks html yang diterima setelah koneksi terbentuk. Kemudian semua data dimasukan ke dalam satu *file JSON* untuk digunakan pada pembuatan model.

3.2.2 Flowchart Pembuatan File Model Random Forest Classifier

Untuk dapat mengaplikasikan model yang dibuat dengan cepat, model tersebut harus diekspor menjadi sebuah *file* yang dapat dengan mudah dimuat sehingga mempercepat proses prediksi kategori. *File* model akan diekspor menjadi *file* berekstensi *joblib* menggunakan modul Python bernama *joblib*.



Gambar 3. 2 Flowchart pembuatan model *Random Forest Classifier*



Gambar 3. 3 Flowchart pra-pemrosesan data

Pada Gambar 3.2, hal pertama yang dilakukan adalah meng-*import library* yang diperlukan untuk pembuatan model. Selanjutnya *dataset* juga di-*import* untuk kemudian diproses menjadi data yang lebih terstruktur dan rapih. Pada Gambar 3.3 terdapat *flowchart* tahap pra-pemrosesan data yang terbagi menjadi empat proses, yaitu *Casefolding*, *Tokenizing*, *Stopwords Removal* dan *Stemming*.

Casefolding adalah tahap untuk menyamakan *case* semua karakter dimana pada umumnya diubah menjadi *lowercase*. Kemudian teks yang telah dijadikan *lowercase* dipisah setiap katanya melalui proses *tokenizing.* Selanjutnya dari kata-

kata tersebut dilakukan *stopwords removal* yaitu penghilangan kata-kata yang dapat diabaikan atau tidak sesuai konteks. Terakhir adalah tahap *stemming* yaitu merubah kata-kata yang berhimbuhan menjadi kata dasarnya.

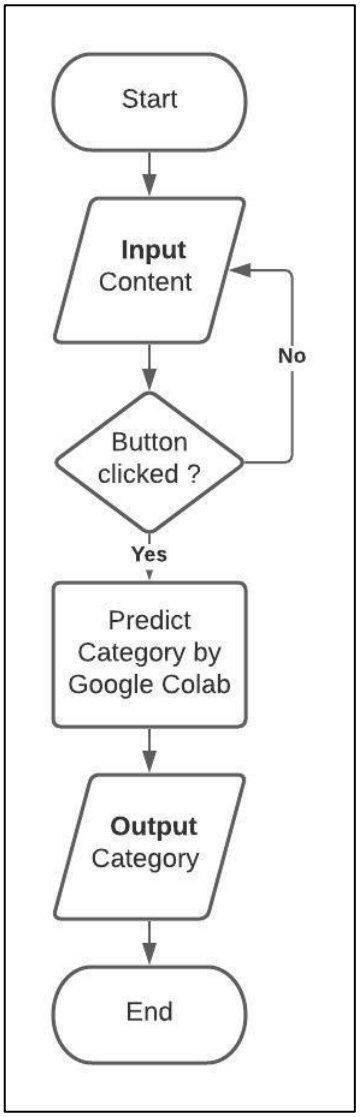
Setelah setiap konten berita selesai diproses dilanjutkan tahap penyaringan data dengan dua kriteria, yaitu konten memiliki lebih dari satu kata dan kata tersebut memiliki representasi vektor. Setiap kata tersebut kemudian dicari representasi vektornya menggunakan *pretrained Word2Vec model* dengan ukuran dimensi 200 dan dilakukan *split* menjadi data latih dan data tes. Sebelum melakukan *training* model, dilakukan pencarian *hyperparameter* terbaik dengan *Cross-Validation* menggunakan *RandomizedSearchCV*. *RandomizedSearchCV* adalah fungsi yang berguna untuk mencari *hyperparameter* terbaik dari rangkaian parameter yang diberikan secara acak. Fungsi ini digunakan karena jumlah *dataset* pada penelitian ini cukup besar dan keterbatasan *session* pada Google Colab yang membuat proses *training* harus dilakukan dalam waktu yang cukup singkat (sekitar 12 jam). Nilai K-*fold* yang digunakan sebesar tiga (3) dengan nilai iterasi sebanyak sepuluh (10) kali.

Setelah ditemukan *hyperparameter* terbaik dilakukan evaluasi model yang dibentuk dengan *hyperparameter* tersebut dengan menggunakan F1-*Score* dan *accuracy*. Akhirnya model pun di-*generate* menjadi *file* model berekstensi *joblib* menggunakan *library* joblib.

3.2.3 Flowchart Website

Model *Random Forest Classifier* diaplikasikan menjadi *website* menggunakan Anvil dan Google Colab. Anvil akan membuat tampilan *frontend*

bagi *user* untuk dapat memberikan *input* berupa konten berita pada *text area* yang tersedia sehingga saat *button* ditekan akan muncul kategori dari konten berita yang dimasukan. Model *Random Forest Classifier* dimuat pada Google Colab yang terhubung langsung dengan *frontend* Anvil tersebut yang bertugas untuk mengklasifikasikan konten yang diberikan lalu mengirimkan kategori dari konten tersebut berupa *string*.

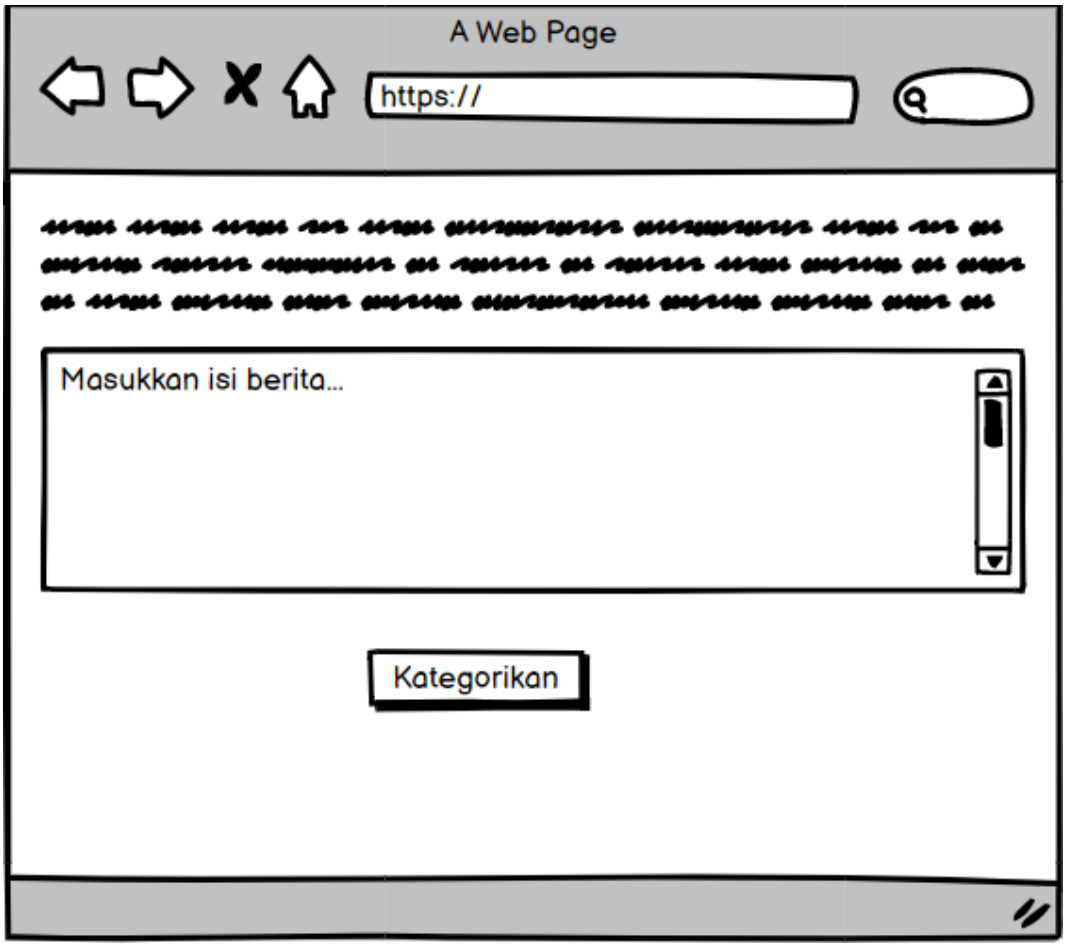


Gambar 3. 4 Flowchart *website* menggunakan Anvil

Pada Gambar 3.4, mula-mula *user* melakukan *input* konten berita pada *text area* yang disediakan. Konten tersebut dikirimkan dalam bentuk *string* ke Google

Colab yang sudah terhubung dengan Anvil dan siap melakukan prediksi kategori dari konten yang diterima. Konten yang diterima melalui tahap pra-pemrosesan terlebih dahulu sebelum akhirnya dijadikan dalam bentuk representasi vektor yang kemudian dicari kategorinya menggunakan model *Random Forest Classifier* yang telah dimuat. Akhirnya hasil kategori tersebut dikirim kembali ke *website* untuk ditampilkan kepada *user*.

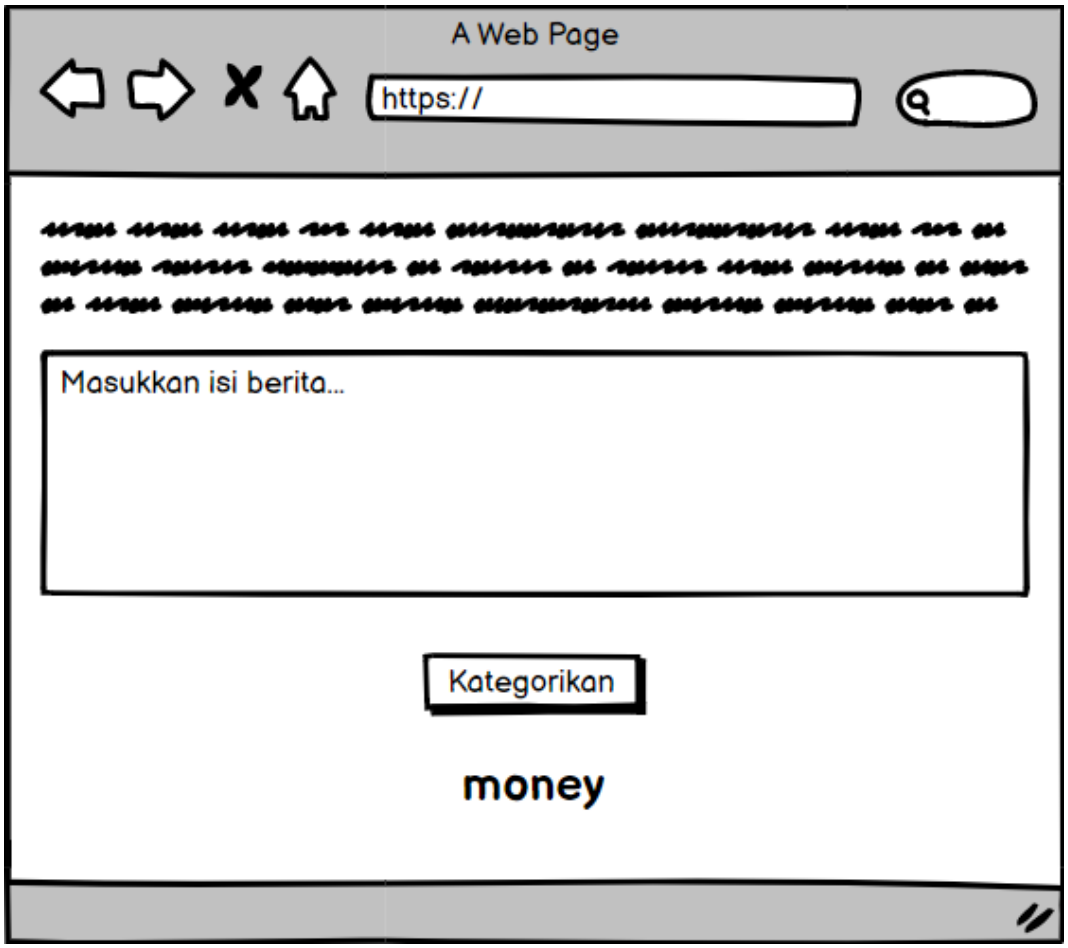
3.2.4 Rancangan Tampilan Antarmuka Website



Gambar 3. 5 Rancangan tampilan awal

Pada Gambar 3.5 terdapat teks paragraf di atas *text area* yang merupakan petunjuk awal penggunaan aplikasi agar *user* dapat lebih mudah menggunakan *website* tersebut. Setelah *user* melakukan *input* kemudian menekan tombol

“Kategorikan”, proses klasifikasi akan dilaksanakan dimana kemudian membawa *user* pada tampilan di Gambar 3.6.



Gambar 3. 6 Rancangan tampilan hasil kategori

Input yang diterima langsung mengalami proses klasifikasi yang dilakukan oleh model *Random Forest Classifier* yang telah dimuat pada Google Colab. Setelah proses klasifikasi selesai, hasilnya langsung dikirimkan ke Anvil yang lansgung menampilkan kategori konten berita tersebut seperti Gambar 3.6. Hasil kategori akan ditampilkan di bawah tombol sesuai dengan hasil klasifikasi yang dikirim dari model *Random Forest Classifier*.