BAB III

METODOLOGI PENELITIAN DAN PERANCANGAN SISTEM

3. 1 Metodologi Penelitian

Detail terkait dengan tahapan-tahapan yang dilakukan dalam metodologi penelitian beserta penjelasan dari setiap tahapan diberikan pada bagian berikut :

1. Perumusan Masalah

Kegiatan yang dilakukan pada tahapan ini adalah mengidentifikasi permasalahan yang terdapat pada media berita merahputih.com. Proses identifikasi dilakukan melalui tahap wawancara dengan Bapak Jourdy Irawan, selaku General Manager dari Departemen Human Resource (HR) PT. Merah Putih Media. Melalui permasalahan yang ada dilakukan proses perumusan masalah, pembatasan permasalahan yang akan dikaji dalam penelitian, serta tujuan dan manfaat dari dilakukannya penelitian.

2. Studi Literatur

Pada tahapan ini kegiatan yang dilakukan adalah meninjau teori-teori yang digunakan dalam penelitian. Teori-teori yang dimaksudkan terdiri dari teori terkait dengan algoritma Random Forest dengan menggunakan fitur ekstraksi TF-IDF dengan penghitungan performa model F1 Score.

3. Pengembangan Sistem

Pada tahapan ini kegiatan yang dilakukan adalah melakukan perancangan yang ditujukan untuk memperoleh gambaran mengenai alur kerja dari algoritma random forest, lalu dilanjutkan dengan implementasi program dari algoritma random forest.

4. Uji Coba Sistem

Pada tahapan ini akan dilakukan uji coba berdasarkan konten berita yang tersedia serta parameterisasi dari metode TF-IDF dan Random Forest. dalam tahap uji coba, performa dari model akan diukur dengan menggunakan F1-score sebagai metrik pengukuran performa model klasifikasi.

5. Evaluasi

Evaluasi dilakukan untuk mengetahui kelebihan dan kekurangan dari metode-metode yang digunakan dalam penelitian dan saran terkait dengan penelitian lanjutan yang dapat dilakukan.

6. Konsultasi Penelitian

Pada tahapan ini kegiatan yang dilakukan adalah penulisan laporan yang bertujuan untuk mendokumentasikan segala bentuk proses penelitian dan menyimpulkan hasil akhir dari penelitian yang dilakukan dalam pengerjaan penelitian. Penulisan laporan ini dilakukan berdasarkan proses bimbingan dan diskusi dengan dosen pembimbing.

3. 2 Pengumpulan Data

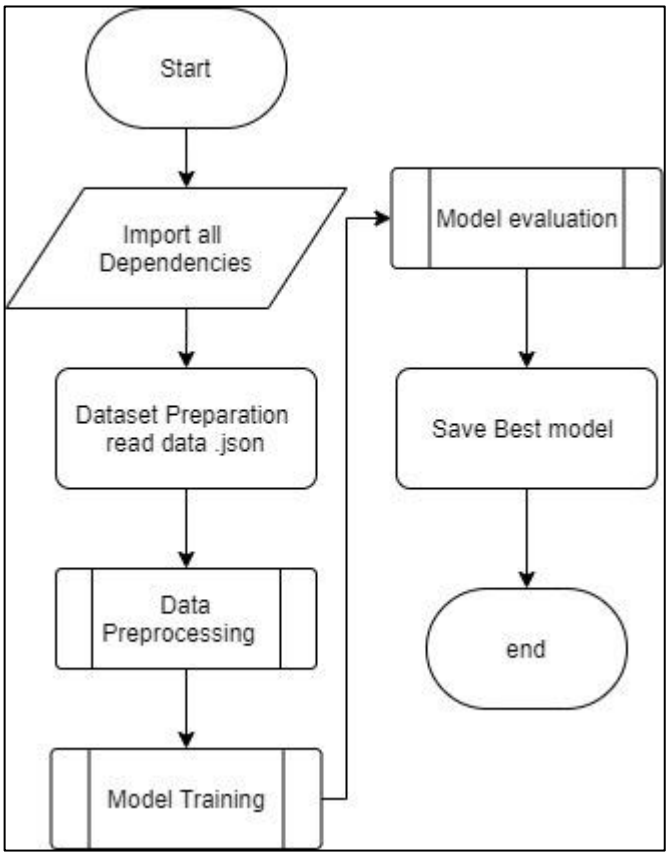
Data yang diperoleh pada penelitian ini adalah data dari hasil *web crawling* dari setiap artikel yang terdapat di portal berita merahputih.com. *Web crawling* yang dilakukan penelitian ini atas seizin dari pihak perusahaan terkait. Data dari hasil crawling disimpan dalam format file Javascript Object Notation (JSON) untuk mempermudah proses pengolahan data. Dataset dari hasil proses crawling hanya terdiri dari 3 subkategori berita yang terdapat pada kategori "Indonesiaku"

yaitu kuliner-articles, tradisi-articles, dan travel-articles data tersebut adalah subkategori dari konten berita Indonesiaku pada portal berita merahputih.com.

3. 3 Perancangan Aplikasi

Perancangan sistem terdiri dari *flowchart* dan perancangan antarmuka aplikasi *web.* Dalam penelitian ini *flowchart* akan terbagi menjadi 3 bagian, yaitu itu *flowchart* proses klasifikasi yang merupakan *flowchart* yang menggambarkan proses klasifikasi menggunakan algoritma *Random Forest Classifier*, *flowchart* algoritma *Random Forest Classifier*, dan *flowchart* aplikasi web untuk memprediksi berita pada portal berita merahputih.com.

3.3.1 Flowchart Klasifikasi

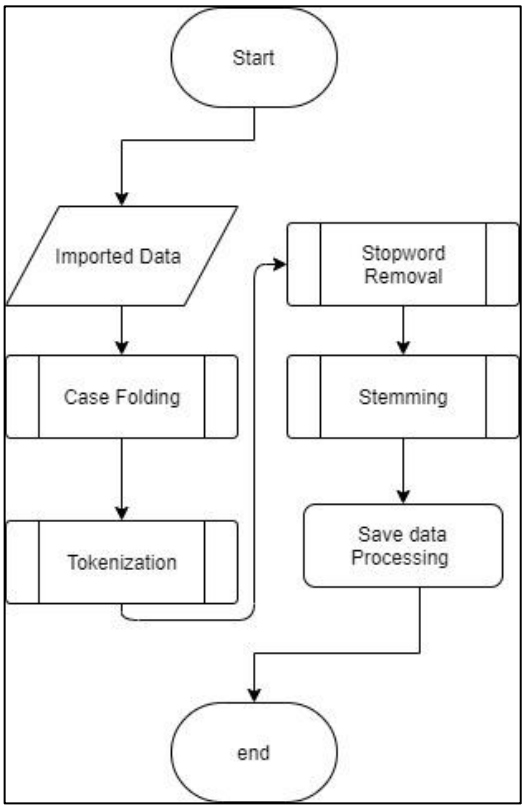


Gambar 3. 1 Flowchart klasifikasi umum

Flowchart yang ditunjukan oleh Gambar 3.1 merupakan *flowchart* dari proses klasifikasi menggunakan *Random Forest* (RFS) *Classifier* secara umum. Proses pertama dilakukan adalah meng*import dependencies* atau *library* yang diperlukan. Kemudian *dataset* yang akan digunakan di-load untuk digunakan pada tahap selanjutnya. Proses pelatihan model akan dilakukan sebanyak 3 kali dengan hanya memanfaatkan informasi konten dari setiap artikel yang ingin diklasifikasikan saja, judul konten dari setiap artikel saja, serta gabungan dari keduanya.

Dataset yang masuk akan dinormalisasi pada tahapan *text preprocessing.* Hal ini dilakukan untuk mengurangi kompleksitas dari proses klasifikasi dengan mengabaikan informasi yang dianggap kurang relevan untuk proses klasifikasi (menghilangkan kata berupa angka ataupun menghilangkan imbuhan pada setiap token). Selain itu, tahap *text pre-processing* diharapkan dapat membuat metode klasifikasi menjadi lebih efektif dan diharapkan performa yang lebih baik dan tepat sasaran. Dalam penelitian ini, tahap *preprocessing* yang dilakukan terdiri dari *Case folding, Tokenization, Stopwords Removal*, dan *Stemming.*

Urutan pelaksanaan tahapan dapat dilihat pada flowchart pada Gambar 3.2 dan penjelasan dari setiap tahapan yang dilakukan akan diberikan setelahnya.



Gambar 3. 2 Flowchart pre-processing

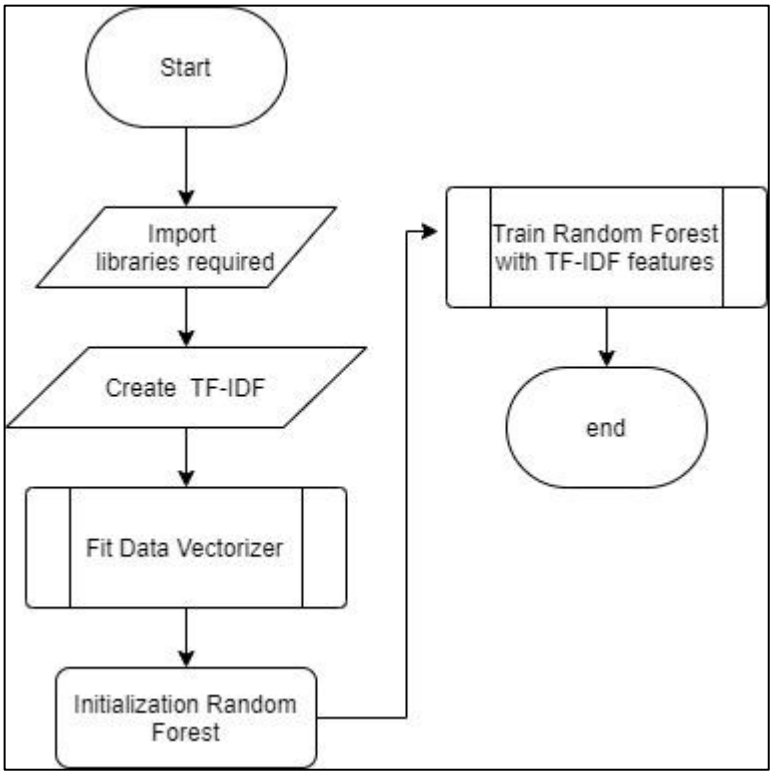
Pada proses *pre-processing*, pertama melakukan proses *case folding* untuk mengecilkan huruf yang berada dalam *dataset,* mengubah angka menjadi kata atau menghapus angka yang tidak dibutuhkan pada *dataset,* menghapus tanda baca pada teks, dan menghapus spasi, tab, dan *new line* pada teks. Proses *case folding* dilakukan untuk menormalisasikan teks pada *dataset* karena untuk mempermudah proses klasifikasi teks untuk membatasi hal-hal yang akan dan tidak digunakan dalam proses klasifikasi.

Tokenization dilakukan untuk memisahkan teks menjadi potongan bagian terkecil yang disebut dengan istilah token. *Stopwords removal* dilakukan untuk

menghilangkan kata umum yang biasanya muncul dalam jumlah besar dan tidak mengandung informasi yang cukup relevan dalam proses klasifikasi. *Stemming* dilakukan untuk merubah bentuk dasar untuk mengurangi kompleksitas dari proses klasifikasi..

Setelah proses text *pre-processing* dilakukan, kemudian *dataset* akan masuk ke tahap data training dimana tahap ini akan dilakukan proses ekstraksi fitur dan pelatihan model Random Forest Classifier.

Tahapan data training dapat dilihat pada *flowchart* pada Gambar 3.3

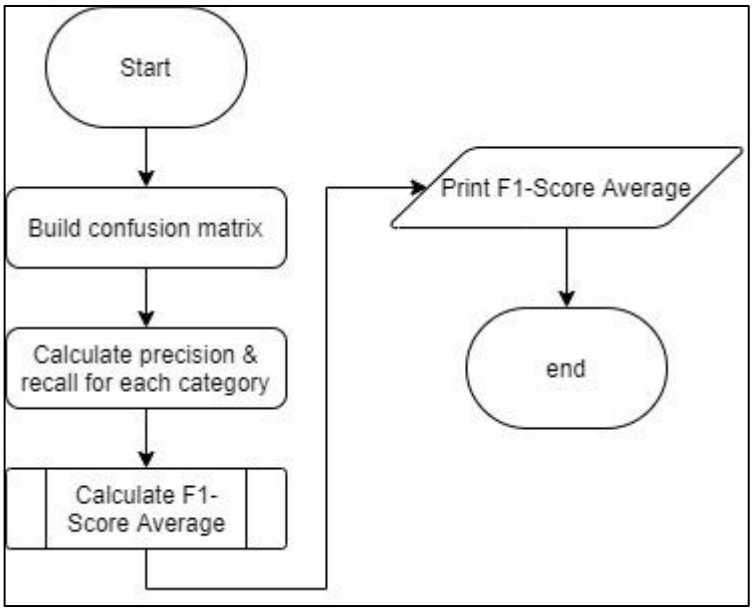


Gambar 3. 3 Flowchart data training

Proses training data dimulai dengan memasukan *library* yang dibutuhkan seperti Tfidfvectorizer, Random Forest Classifier, Pipeline, dan GridSearchCV. setelah itu mengaktifkan ekstraksi fitur TF-IDF setelah itu fit data vectorizer

untuk mempelajari kamus kosa kata dari semua token dalam dataset. Selanjutnya adalah proses instalasi algoritma Random Forest Classifier selanjutnya adalah proses training *dataset* dengan algoritma Random Forest Classifier menggunakan ekstraksi fitur TF-IDF setelah itu proses evaluasi menggunakan matrik perhitungan F1-score untuk menentukan bobot terbaik pada hasil training data. Setelah proses training data telah selesai maka data akan dievaluasi untuk mengetahui hasil dari proses penghitungan data training.

Tahapan data evaluation dapat dilihat pada *flowchart* pada gambar 3.4



Gambar 3. 4 Flowchart data evaluation

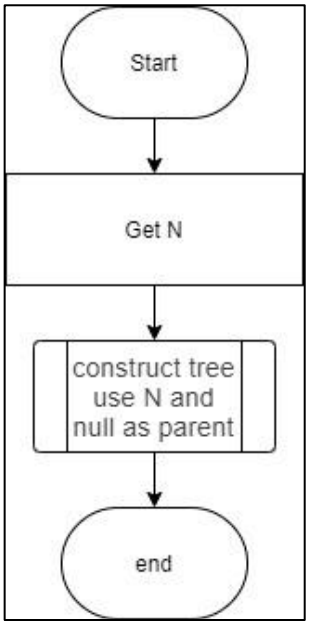
Dalam tahapan evaluasi data diawali dengan membuat confusion matrix untuk model evaluasi, setelah itu dilanjutkan dengan proses perhitungan precision dan recall pada tiap sub kategori berita yang ada. Proses selanjutnya adalah proses perhitungan rata-rata dari hasil perhitungan precision dan recall tiap sub kategori berita dengan menggunakan matrik perhitungan F1-Score, selanjutnya adalah mencetak hasil perhitungan F1-Score untuk melihat hasil rata-rata dari

perhitungan yang sudah dilakukan untuk menetukan hasil terbaik dari rata-rata tersebut.

3.3.2 Flowchart Algoritma

Algoritma Random Forest Classifier (RFC) bekerja berdasarkan mekanisme dataset bagging dan pelatihan sekumpulan Decision Tree, maka dari itu, untuk menjelaskan proses pelatihan yang terjadi dalam algoritma RFC terlebih dahulu akan dijelaskan proses pelatihan/ pembentukan sebuah Decision Tree.

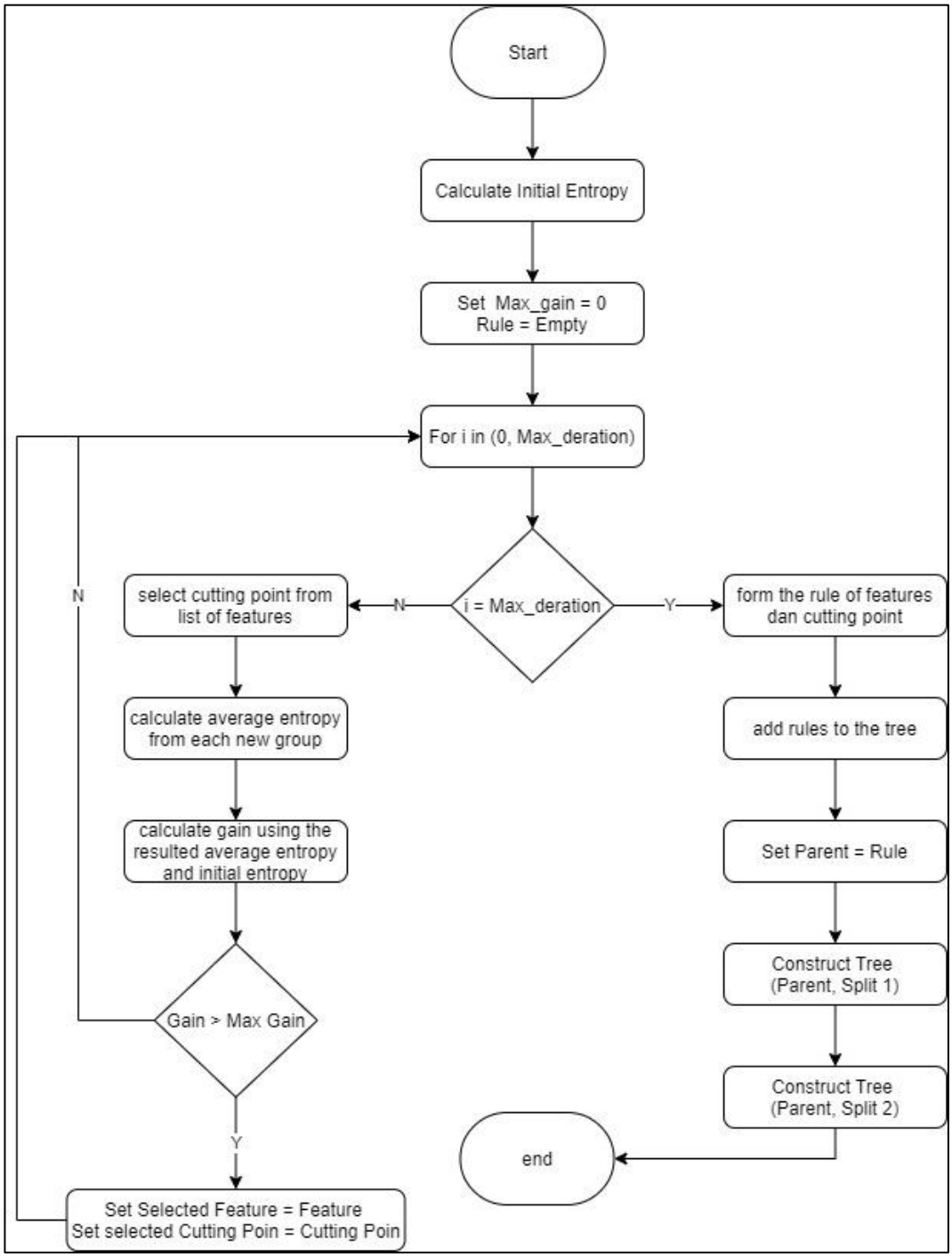
Tahapan proses Decision tree dapat dilihat pada *flowchart* Gambar 3.5



Gambar 3. 5 Flowchart Decision Tree

Diawali dengan get N yaitu proses mengambil dataset pada sub-kategori berita yang akan dijadikan Decision Tree setelah itu proses pelatihan decision tree mula-mula dilakukan berdasarkan sebuah data input N dan null sebagai parent node.

Proses pembuatan *tree* atau pohon dapat dilihat pada *flowchart tree* pada Gambar 3.6 .



Gambar 3. 6 Flowchart Construction Tree

Pada proses pembuatan *tree* atau pohon diawali dengan menghitung *entropy* awal setelah itu menentukan *max-gain* = 0 dan men set *rule* = 0 setelah di set

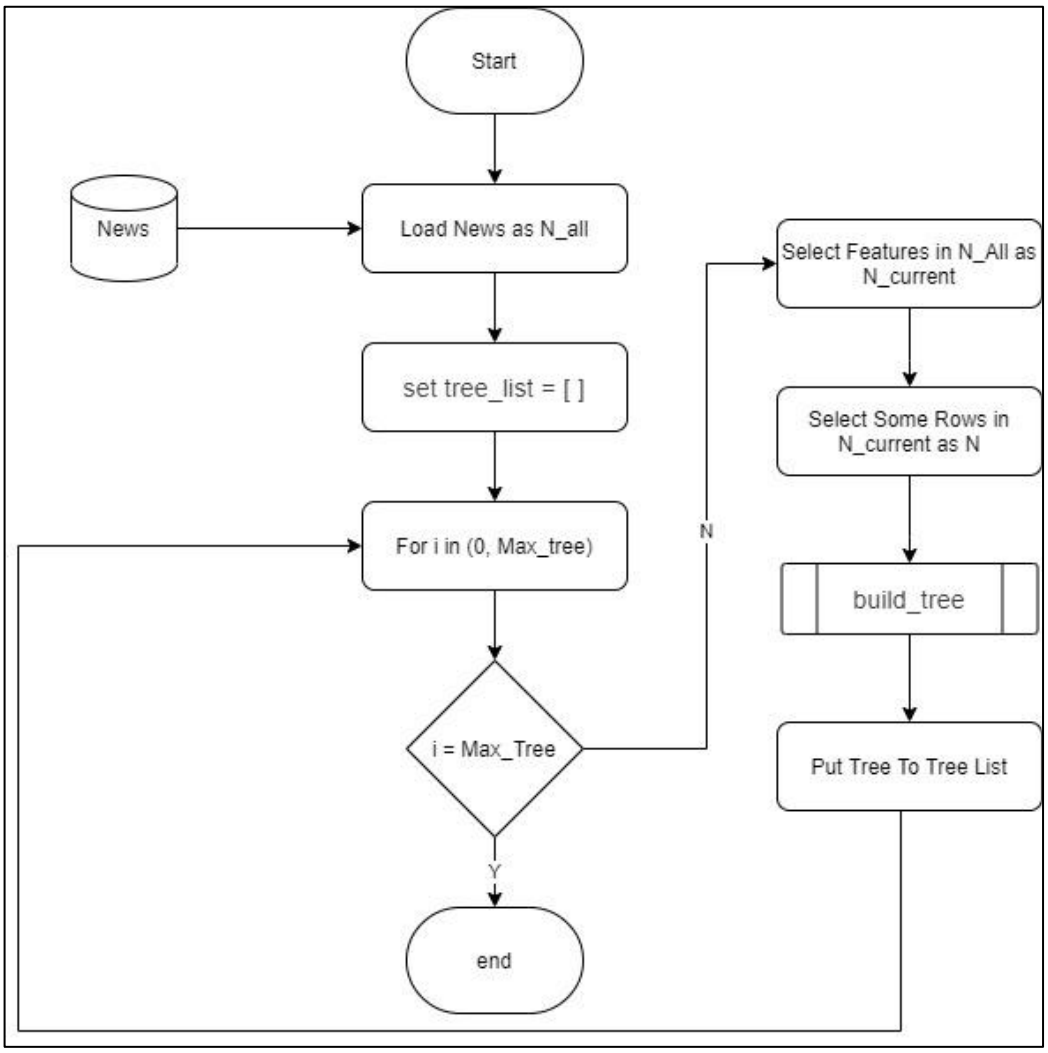
melakukan proses pengulangan *For* i *in* (0, *max_ deration*) pengulangan ini untuk pengecekan pada proses i = *max_deration* ketika kondisi *yes* makan berlanjut pada proses pembentukan *rule* dari fitur dan *cutting point* lalu berlanjut pada proses menambahkan *rule* ke dalam *tree* atau pohon setelah itu berlanjut pada proses menentukan *parent* = *rule* setelah itu proses berlanjut pada proses *construct tree* atau membuat pohon dengan menentukan (*parent, split 1*) proses ini berlanjut untuk *construct tree* atau membuat pohon dengan menentukan (*parent, split 2*) prosesi ini berjalan ketika kondisi pengecekan hasil pengulangan *yes.*

Ketika kondisi pengecekan hasil pengulangan *no* maka akan menjalankan proses pemilihan fitur *cutting poin* lalu berlanjut pada proses penghitungan rata-rata atau *calculate average entropy from each new group* setelah proses perhitungan rata-rata berlanjut pada proses *calculate gain using the resulted average entropy and initial entropy* atau perhitungan *gain* dari hasil *calculate average entropy from each new group* setelah itu proses penghitungan di cek apakan *gain* > *max gain* ketikan hasil pengecekan dengan kondis *yes* maka akan masuk proses *set selected feature* = *feature* dan *set selected cutting poin* = *Cutting poin* ketika proses ini sudah dilakukan maka akan balik ke prose pengulangan lalu di cek kembali apakah kondisi i = *max_entropy* jika *yes* maka akan melakukan proses pada kondis *yes* pada i = *max_entropy,* ketika kondisi pengecekan mendapatkan kondisi *no* maka akan melakukan proses kondisi *no* pada tahap pengecekan i = *max_entropy*.

Ketikan hasil pengecekan *gain* > *max gain* dengan kondisi *no* maka akan kembali ke proses pengulangan pertama lalu mengulangi proses tahapan kondis

no pada pengecekan i = *max_entropy* ketika proses semua sudah dijalankan proses telah selesai.

Ketika proses Decision tree sudah diproses dan mendapatkan hasil maka proses Algoritma Random Forest Classifier untuk klasifikasi sub-kategori berita dapat dilakukan, berikut ini adalah *flowchart* Algoritma Random Forest Classifier dapat dilihat pada Gambar 3.7.



Gambar 3. 7 Flowchart Algoritma Random Forest Classifier

Proses algoritma Random Forest Classifier diawali dengan mengambil data pada *dataset* news dengan inisialisasi *N_all* setelah itu menentukan pada *tree*

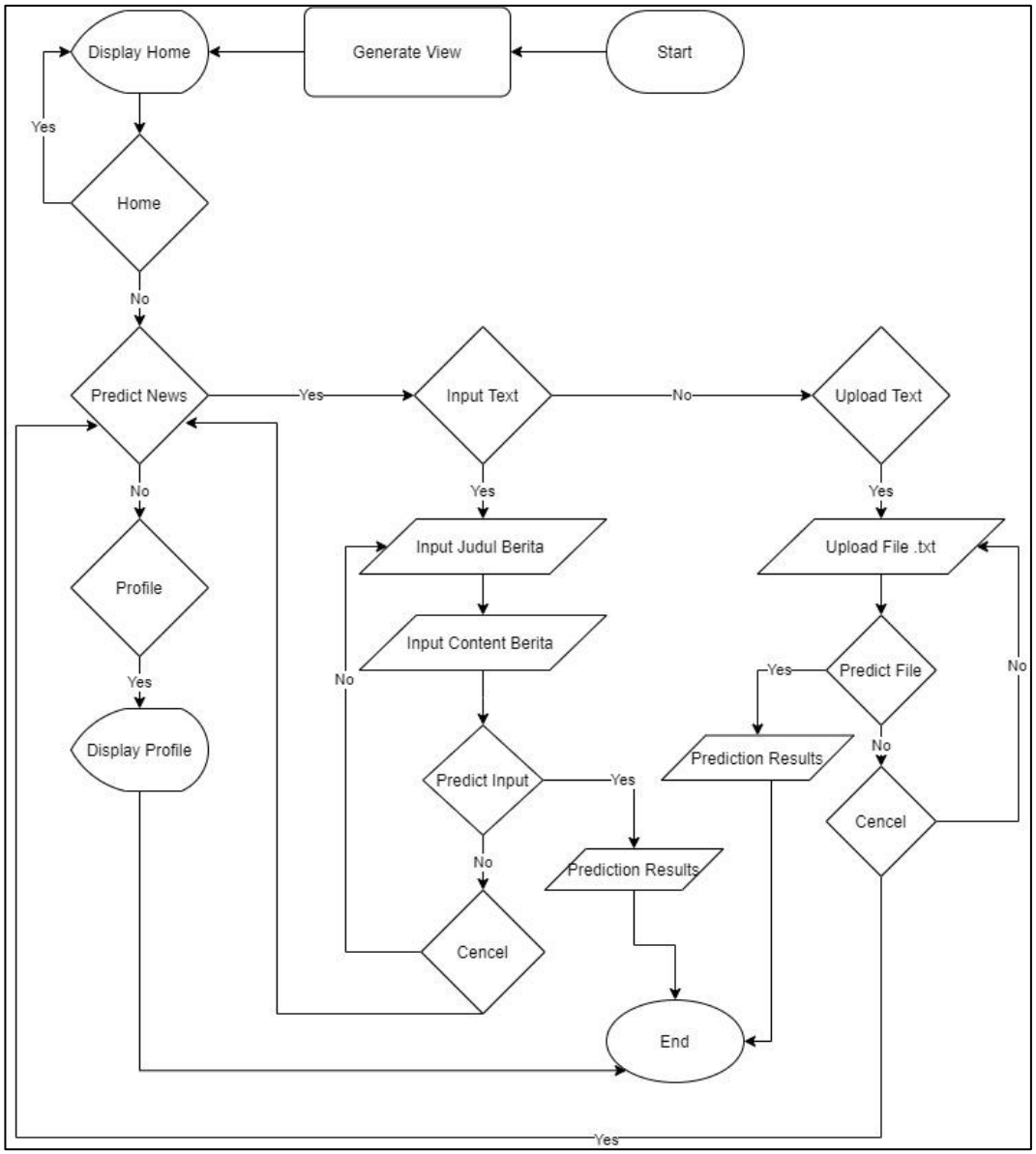
dengan menggunakan *array* kosong setelah itu masuk proses pengulangan *for* i *in* (0, *max_tree*) setelah itu proses berlanjut ke proses pengecekan apakah i = *Max_Tree* ketika proses mendapat kondisi *yes* maka proses akan selesai.

Ketika kondisi yang didapatkan dari hasil pengecekan adalah *no* maka proses akan berlanjut untuk menjalani proses pemilihan fitur pada *N_All* untuk dijadikan *N_current* ketika proses ini sudah dijalankan maka berlanjut ke proses pemilihan baris dalam *N_current* sebagai data atau *N* setelah itu akan menjalankan proses *build_tree* setelah proses ini kelar maka pohon atau *tree* akan dimasukan kedalam daftar pohon atau *tree,* ketika data sudah dimasukan kedalam daftar pohon maka akan kembali dalam proses pengulangan *for* i *in* (0, *max_tree*) setelah itu akan dilakukan pengecekan pada proses i = *Max_Tree* ketika ketika mendapat kondis *yes* maka proses akan berakhir ketika proses mendapat kondis *no* maka akan kembali menjalankan proses *no* pada i = *Max_Tree.*

3.3.3 Flowchart Web Aplikasi

Flowchart web aplikasi ini adalah *flowchart* dari web aplikasi klasifikasi Algoritma Random Forest Classifier pada sub-kategori berita pada portal berita merahputih.com.

Berikut ini adalah *flowchart* web aplikasi Algoritma Random Forest pada sub-kategori berita dapat dilihat pada Gambar 3.8



Gambar 3. 8 Flowchart web aplikasi

Pada proses ini diawali dengan rendering tampilan web setelah itu *user* melihat tampilan home dalam web aplikasi Algoritma Random Forest pada sub-kategori berita terdapat 3 pilihan menu yaitu home, *Predict news,* dan *profile.* Ketika *user* memilih menu home maka *user* akan kembali ke halaman home atau halaman awal pada web aplikasi Algoritma Random Forest pada sub-kategori berita, ketika *user* tidak memilih halam home maka *user* dapat memilih menu lainnya seperti menu *Predict news* atau *profile.*

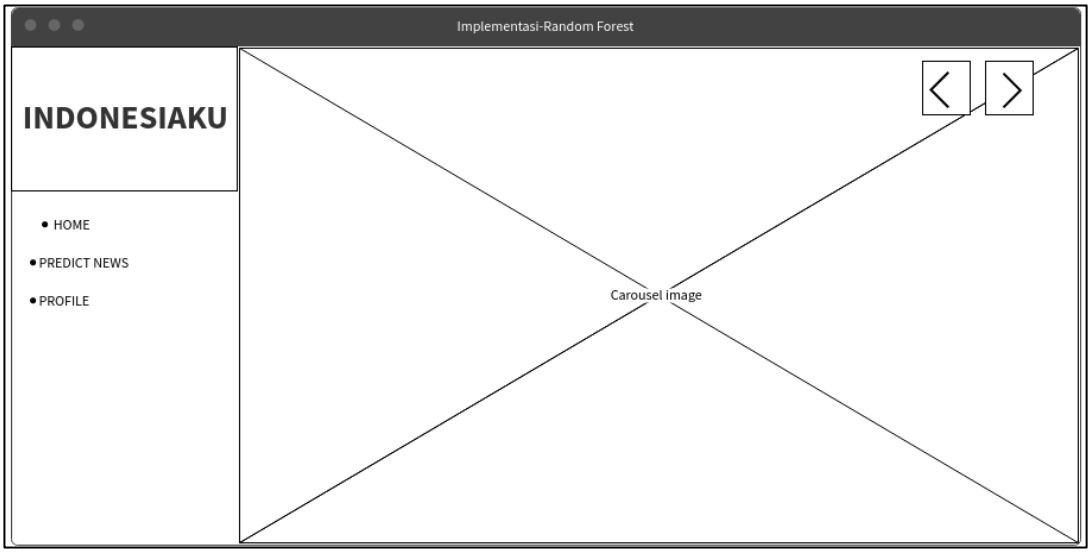
Pada saat *user* memilih menu *predict news* maka *user* dapat memilih jenis *predict news* yaitu *input text* dan *upload text.* Ketika *user* memilih *input text* maka *user* dapat memasukan judul berita yang akan diprediksi ketika sudah memasukan judul user dapat memasukan isi dari *content* berita ketika sudah *user* dapat memilih *button predict news* atau *cancel*, ketika *user* memilih *button predict news* maka *user* akan melihat hasil prediksi dari berita yang dimasukan, ketika user memilih *button cancel* maka *user* akan kembali ke halaman *predict news.*

Pada saat *user* memilih *upload text* maka *user* dapat memasukan file yang berisikan berita yang akan diprediksi, dalam file yang akan *upload* harus berisi judul dan isi dari berita yang akan diprediksi dalam *upload news* dapat memasukan file lebih dari satu untuk diprediksi dan juga file yang di *upload* harus berbentuk *.txt,* setelah memasukan file berita *user* dapat memilih *button predict news* atau *cancel,* pada saat *user* memilih *button predict news* maka *user* dapat melihat hasil prediksi dalam bentuk tabel yang berisikan judul dan hasil prediksi dari berita yang sudah di *upload,* ketika user memilih *button cancel* maka *user* akan kembali ke halaman *predict news.* Pada saat user memilih menu *profile* dalam menu ini *user* dapat melihat *profile* dari penulis.

3. 4 Rancangan Tampilan Antarmuka

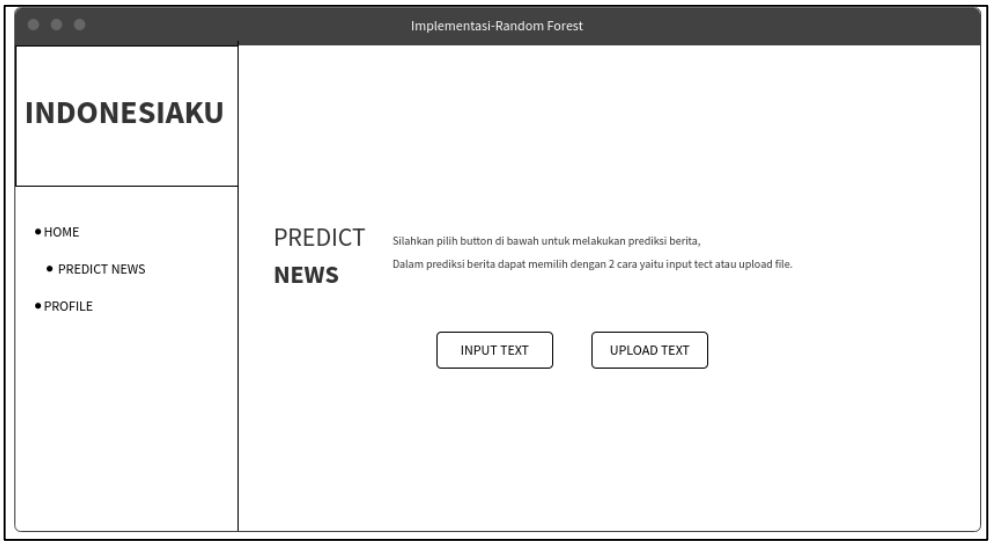
Tampilan antarmuka aplikasi *web* dibuat secara *single page web application* yang dibagi beberapa bagian atau *section,* yaitu bagian Home, bagian *Predict News* dalam bagian ini terdapat sub-bagian lagi yaitu bagian *Predict News* menggunakan *input* berita dan bagian *Predict News* menggunakan *upload* berita, dan bagian *profile.*

Gambar 3.9 merupakan rancangan antarmuka untuk bagian *Home*, pada bagian ini terdapat *image* dan poster yang berisi tentang penjelasan singkat mengenai aplikasi web dan penjelasan kategori yang menjadi acuan dalam prediksi berita, *image* dan poster pada halam ini menggunakan *carousel* untuk transisi gambar dan poster.



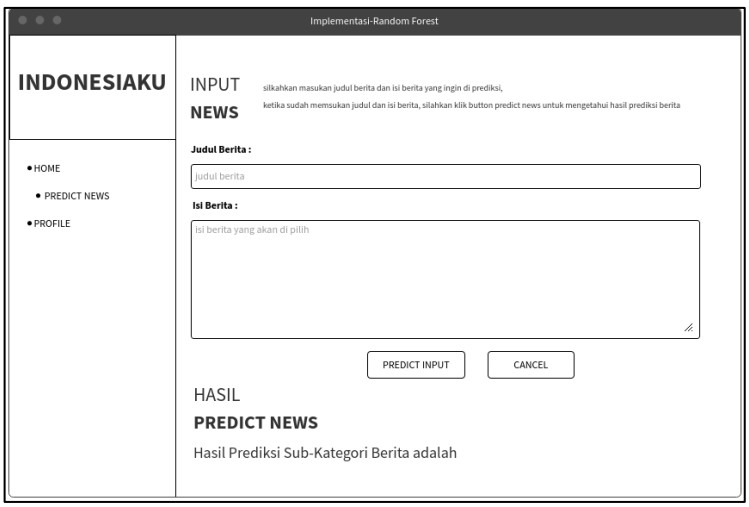
Gambar 3. 9 Rancangan antarmuka bagian home

Pada Gambar 3.10 terdapat rancangan antarmuka untuk bagian *predict news*, pada bagian ini *user* dapat memilih sub-bagian pada *predict news* yaitu *input text* atau *upload text.*



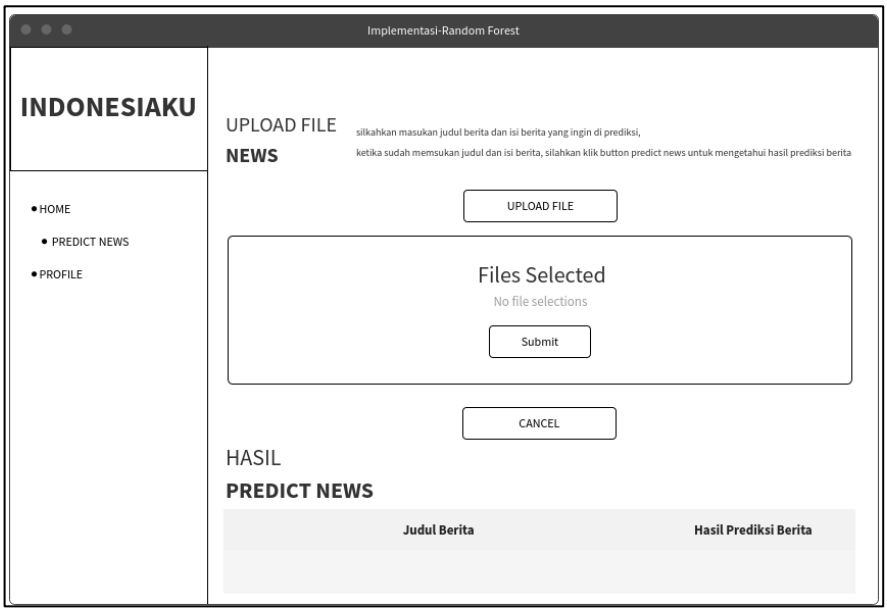
Gambar 3. 10 Rancangan antarmuka bagian predict news

Pada gambar 3.11 rancangan antarmuka untuk sub-bagian *input text* pada *predict news* dalam rancangan antarmuka ini user dapat menulis berita yang ingin diprediksi, bagian yang harus diisi oleh *user* adalah judul berita yang ingin di prediksi dan isi dari berita yang ingin diprediksi, selanjutnya user dapat menekan *button predict input* untuk melakukan prediksi pada saat itu *user* dapat melihat hasil prediksi dan juga user dapat menekan *button cancel* untuk kembali kehalaman *predict news.*



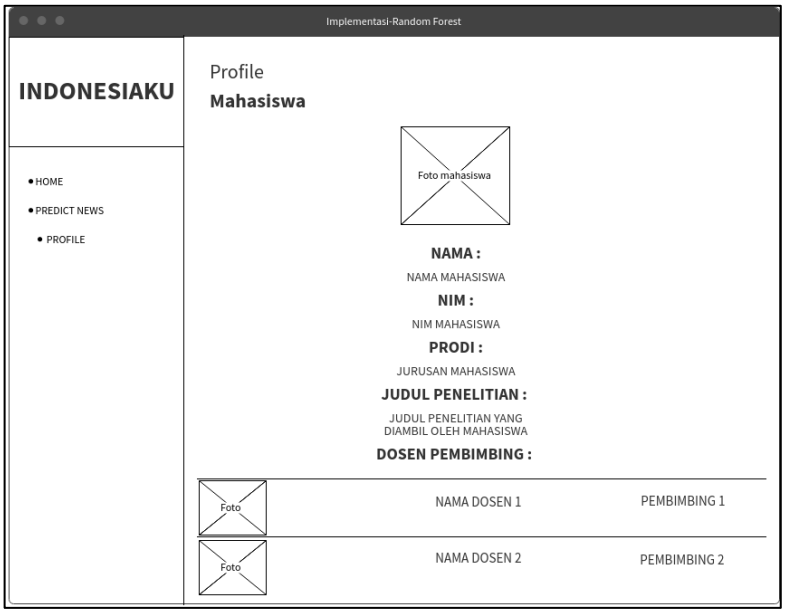
Gambar 3. 11 Rancangan antarmuka sub-bagian input text

Pada Gambar 3.12 rancangan antarmuka untuk sub-bagian *upload file* pada *predict news* dalam rancangan antarmuka ini *user* dapat meng-*upload file* berita yang ingin diprediksi dalam sub-bagian ini *user* dapat meng-*upload file* lebih dari satu, *file* yang dapat di*upload* oleh *user* adalah *file* yang berekstensi .txt. Setelah *file* di*upload user* dapat menekan *button* s*ubmit* untuk memprediksi *file* berita yang sudah di*upload,* hasil prediksi dari *file* berita sudah di *upload* dapat dilihat pada tabel hasil prediksi. *User* dapat menekan *button cancel* untuk kembali ke halaman *predict news.*



Gambar 3. 12 Rancangan antarmuka sub-bagian upload file

Pada Gambar 3.13 rancangan antarmuka untuk *profile* pada bagian ini *user* dapat melihat *profile* mahasiswa, pada bagian ini berisikan informasi tentang mahasiswa, penelitian yang diambil oleh mahasiswa, dan dapat melihat informasi tentang dosen pembimbing mahasiswa.



Gambar 3. 13 Rancangan antarmuka profile