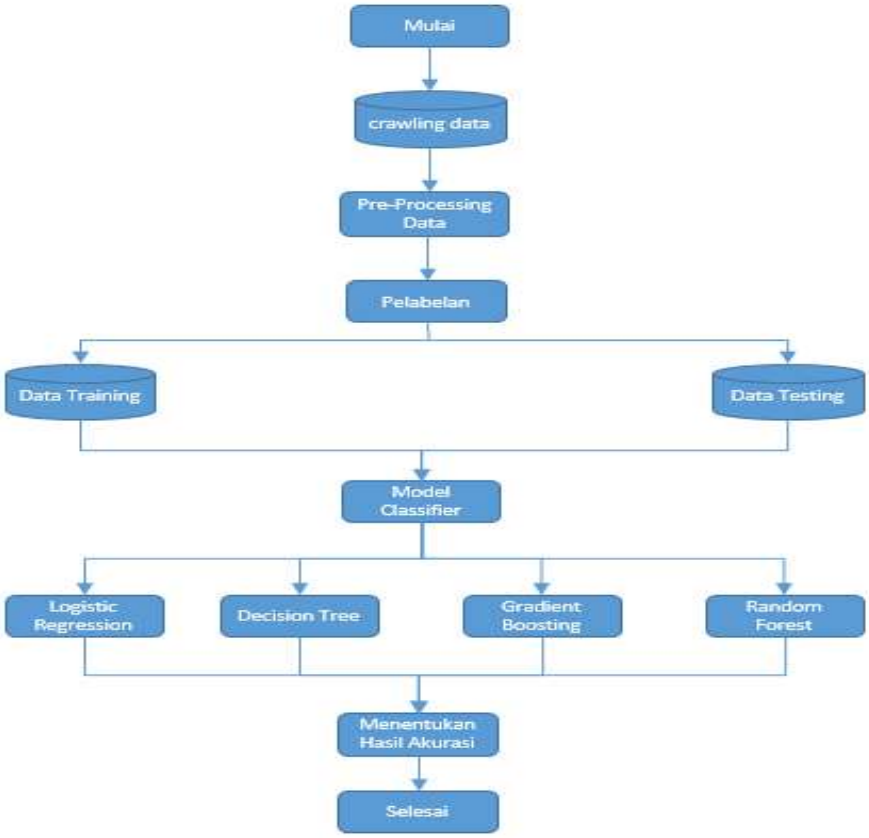
BAB III

METODE PENELITIAN

3.1 Desain Penelitian

Desain penelitian adalah rencana atau strategi yang digunakan oleh peneliti untuk menjawab pertanyaan penelitian dan mencapai tujuan penelitian. gambar 3.1 merupakan desain penelitian pada penelitian ini:



Gambar 3 1 Deain Penelitian
Sumber: (Ditendra et al. 2022)

Penjelasan dari langkah-langkah yang Anda berikan adalah sebagai berikut:

1. *Crawling* Data

Data yang diperlukan untuk penelitian diambil dari berbagai sumber menggunakan teknik crawling. Ini dapat mencakup pengambilan data dari situs web, media sosial, atau sumber lainnya.

2. *Preprocessing* Data

   Data yang telah diambil kemudian melalui tahap preprocessing, di mana dilakukan pembersihan data dari *noise*, *normalisasi* data, dan langkah-langkah lain untuk mempersiapkan data agar siap untuk digunakan dalam model *machine learning* (Widyaya and Budi 2021).

3. Pelabelan

   Data yang sudah bersih akan diberi label, yaitu kategori atau klasifikasi yang sesuai dengan tujuan penelitian. Label ini biasanya mencakup klasifikasi sebagai data *training* atau data testing.

4. Data *Training*

   Sebagian dari data yang telah diberi label digunakan sebagai data *training*. Ini adalah data yang digunakan oleh model *machine learning* untuk belajar dan menghasilkan pola dari fitur-fitur yang ada.

5. Data Testing

   Sisanya dari data yang telah diberi label digunakan sebagai data testing. Data ini tidak digunakan dalam proses pelatihan, tetapi digunakan untuk menguji kinerja model yang telah dilatih.

6. Model *Classifier*

   Model *classifier* adalah kerangka kerja atau struktur dasar yang digunakan untuk mengklasifikasikan data ke dalam kategori yang telah ditentukan.

7. *Logistic Regression*

   Salah satu metode *machine learning* yang digunakan untuk masalah klasifikasi, khususnya ketika variabel *dependen* adalah *biner* (Abubakar et al. 2023).

8. *Decision Tree*

   Pemanfaatan pohon keputusan terutama terletak pada kemampuannya untuk menyederhanakan proses pengambilan keputusan, mengurai langkah-langkahnya dari yang rinci menjadi lebih sederhana. Hal ini memungkinkan pengambil keputusan untuk lebih mudah menginterpretasikan solusi dari permasalahan yang dihadapi. Pohon keputusan juga digunakan untuk melakukan eksplorasi data dan mengungkap hubungan yang mungkin tersembunyi (Arnomo 2021).

9. *Gradient Boosting*

   Metode yang menggabungkan beberapa model lemah (*weak learners*) menjadi satu model yang kuat untuk meningkatkan performa (Sri Diantika et al. 2023).

10. *Random Forest*

    Metode yang menggunakan banyak pohon keputusan untuk meningkatkan akurasi dan mengurangi *overfitting* (Ramadhan et al. 2022).

11. Menentukan Hasil Akurasi

    Setelah model-model di atas dilatih dan diuji dengan data testing, hasil akurasi dievaluasi dengan menggunakan metrik yang sesuai, seperti akurasi, presisi, *recall*, dan *f1-score* (Ditendra et al. 2022).

3.2 Object Penelitian

Detik.com portal berita online terkemuka di Indonesia yang menyediakan berita terkini dari berbagai kategori, seperti politik, ekonomi, olahraga, dan hiburan. *turnbackhoax*.id situs web yang bertujuan mengidentifikasi dan membunkum berita *hoax* di Indonesia. Fokusnya adalah memberikan informasi untuk membantu masyarakat membedakan antara berita yang valid dan *hoax*, serta meningkatkan literasi digital.

1. Berita dari Detik.com (https://*news*.detik.com/):

   Objek Penelitian: Berita online yang dipublikasikan di situs web Detik.com.

   Ciri-ciri Berita:

   Kategori Berita: Politik, Ekonomi, Olahraga, Hiburan, dll.

   Struktur Kalimat: Panjang kalimat, kompleksitas kalimat.

   Tanggal Publikasi: Waktu publikasi berita.

2. Berita dari TurnBackHoax.id (https://turnbackhoax.id):

   Objek Penelitian: Informasi yang diberikan oleh TurnBackHoax.id terkait berita *hoax*.

   Ciri-ciri Berita:

   Kategori Berita *Hoax*: Jenis berita *hoax* yang sering ditemui.

   Polanya: Apakah terdapat pola tertentu dalam cara berita *hoax* dibuat atau disebarluaskan.

   Sumber *Hoax*: Identifikasi sumber berita yang sering menjadi *hoax*.

3.3 Populasi dan Sampel

Dalam sub bab ini penulis akan menjelaskan mengenai keterangan yang dimiliki oleh populasi dan sampel yang dijadikan bahan dalam pelaksanaan kegiatan penelitian sebagai berikut.

3.3.1 Populasi

Berdasarkan uraian tersebut maka, populasi dalam penelitian ini adalah seluruh *dataset* berita dari kumpulan data valid maupun hoax sebanyak 4312 baris 5 column.

3.3.2 Sampel

Berdasarkan uraian tersebut maka, membuat sampel berita dapat mengambil sejumlah contoh yang mewakili variasi dalam data untuk keperluan analisis atau pelatihan model *machine learning*. Pemilihan sampel yang baik sangat penting untuk memastikan representativitas terhadap populasi dan hasil yang dapat diandalkan.

```
import random

# Data berita sebagai populasi
populasi_berita = [
    {"judul": "Vaksin COVID-19 Menyebabkan Kerusakan DNA Manusia", "deskripsi": "Berita ini menyebarkan klaim palsu tentang vaksin COVID-19.", "tanggal": "2022-01-15", "is_fake": 1},
    {"judul": "Penemuan Baru Dapat Menghentikan Penuaan", "deskripsi": "Sebuah penelitian menemukan terobosan dalam penghentian proses penuaan.", "tanggal": "2022-01-16", "is_fake": 0},
    # Tambahkan berita lain sesuai kebutuhan
]

# Menggunakan random sampling untuk membuat sampel
jumlah_sampel = 2
sampel_berita = random.sample(populasi_berita, jumlah_sampel)

# Menampilkan data sampel
for berita in sampel_berita:
    print("Judul:", berita["judul"])
    print("Deskripsi:", berita["deskripsi"])
    print("Tanggal:", berita["tanggal"])
    print("Hoaks?", "Ya" if berita["is_fake"] else "Tidak")
    print("--------------------")
```

```
Judul: Penemuan Baru Dapat Menghentikan Penuaan
Deskripsi: Sebuah penelitian menemukan terobosan dalam penghentian proses penuaan.
Tanggal: 2022-01-16
Hoaks? Tidak
--------------------
Judul: Vaksin COVID-19 Menyebabkan Kerusakan DNA Manusia
Deskripsi: Berita ini menyebarkan klaim palsu tentang vaksin COVID-19.
Tanggal: 2022-01-15
Hoaks? Ya
--------------------
```

Gambar 3 2 Sampel

3.4 Teknik Pengumpulan Data

Pengumpulan data adalah tahap awal yang penting dalam penelitian ini, di mana berbagai berita *hoax* dan *non-hoax* dikumpulkan dari sumber-sumber yang dapat diandalkan. Data yang berkualitas dan representatif akan menjadi dasar dalam pengembangan dan evaluasi model deteksi berita *hoax*.

1. Mengumpulkan dataset berita yang mencakup berita *hoax* dan non-*hoax* untuk melatih dan menguji model deteksi.
2. Berita online mengambil berita dari sumber-sumber berita online yang diverifikasi keasliannya, seperti situs berita resmi dan terpercaya.
3. Penyusunan *dataset*
4. Mengorganisir berita *hoax* dan non-*hoax* dalam format yang sesuai, seperti *csv*.
5. Setiap data berisi teks berita dan label yang menandakan apakah berita tersebut *hoax* atau non-*hoax*.

3.5 *Ekstrasi* Fitur

Ekstrasi fitur adalah bahwa proses ini mengubah teks berita menjadi representasi angka yang dapat dimengerti oleh algoritma *machine learning*. Teknik ekstraksi fitur, seperti *tf-idf*, memberikan bobot pada kata-kata dalam teks berdasarkan pentingnya kata tersebut dalam dokumen dan seluruh dataset (Purniawan, Sasmita, and Pratama 2022). Representasi numerik ini memungkinkan model *machine learning* untuk memahami pola dan hubungan antara kata-kata

dalam teks, yang pada gilirannya membantu dalam pengenalan pola yang mendukung klasifikasi berita *hoax* dan *non-hoax*.

Dengan menggunakan ekstraksi fitur, dapat mengubah teks berita menjadi bentuk yang dapat diolah oleh algoritma, memungkinkan model untuk belajar dari data pelatihan dan membedakan ciri-ciri antara berita *hoax* dan *non-hoax*. Hal ini menjadi langkah penting dalam membangun model deteksi berita *hoax* yang efektif dan akurat.

Implementasi *TF-IDF* :

Dokumen 1: "Saya suka belajar pemrograman."
Dokumen 2: "Pemrograman sangat berguna untuk masa depan."

Sekarang, menghitung bobot *tf-idf* untuk kata "pemrograman" dalam kedua dokumen.

Term Frequency (TF):

1. Dokumen 1:

   Jumlah kata dalam dokumen: 5

   Jumlah kemunculan kata "pemrograman": 1

   TF("pemrograman", Dokumen 1) = 1/5

2. Dokumen 2:

Jumlah kata dalam dokumen: 7

Jumlah kemunculan kata "pemrograman": 1

TF("pemrograman", Dokumen 2) = 1/7

Inverse Document Frequency (IDF):

Total dokumen dalam koleksi (N) = 2

Jumlah dokumen yang mengandung "pemrograman" (n_t) = 2

IDF("pemrograman") = log(2 / (2 + 1)) ≈ 0.176

***TF-IDF Weight*:**

1. Dokumen 1:

- *TF-IDF*("pemrograman", Dokumen 1) = TF("pemrograman", Dokumen 1) * IDF("pemrograman") ≈ (1/5) * 0.176

2. Dokumen 2:

- *TF-IDF*("pemrograman", Dokumen 2) = TF("pemrograman", Dokumen 2) * IDF("pemrograman") ≈ (1/7) * 0.176

3.6 Model Klasifikasi

Model klasifikasi memainkan peran kunci dalam mengidentifikasi dan membedakan berbagai jenis berita. Dalam pengembangan model ini, menjelajahi beberapa algoritma klasifikasi yang digunakan untuk melatih model agar dapat memberikan prediksi akurat terkait keaslian suatu berita. Berikut adalah gambaran

singkat tentang penggunaan empat algoritma klasifikasi yang berbeda: *Logistic Regression, Decision Tree, Gradient Boosting, dan Random Forest* (Baiq Nurul Azmi, Arief Hermawan, and Donny Avianto 2023).

1. *Logistic Regression*

   Dalam konteks analisis berita, *Logistic Regression* dapat diterapkan untuk memprediksi probabilitas keaslian suatu berita berdasarkan fitur-fitur seperti frekuensi kata kunci, panjang artikel, dan sentimen umum.

2. *Decision Tree*

   Decision Tree dapat digunakan untuk membuat aturan keputusan yang jelas dalam mengklasifikasikan berita. Misalnya, jika suatu berita memiliki judul sensasional dan deskripsi yang tidak konsisten, mungkin lebih cenderung diklasifikasikan sebagai *hoax*.

3. *Gradient Boosting*

   Dengan menggunakan *Gradient Boosting*, model dapat memperbaiki kesalahan prediksi model sebelumnya. Sebagai contoh, jika model sebelumnya kesulitan mengklasifikasikan berita dengan judul yang ambigu, *Gradient Boosting* dapat memberikan penekanan lebih pada fitur-fitur yang relevan untuk meningkatkan akurasi.

4. *Random Forest*

   Random Forest dapat digunakan untuk memitigasi *overfitting* dan meningkatkan generalisasi model. Sebagai contoh, jika ada banyak variasi

dalam *dataset*, *Random Forest* dapat membantu meningkatkan ketepatan prediksi dengan mempertimbangkan banyak pohon keputusan yang berbeda.

Logistic Regression:

P(Y=1)=1/(1+e-(b0+b1 X1+b2 X2+⋯+bn Xn))

di mana:

- P(Y = 1) adalah probabilitas bahwa *output* Y adalah 1.
- e adalah basis logaritma natural.
- b_0,b_1,b_2,…b_n adalah parameter model *(koefisien)*.
- x_(1,) x_2 〖,…,x〗_n adalah nilai-nilai fitur input.

Nilai-nilai fitur x_(1,) x_2 〖,…,x〗_ndikombinasikan dengan parameter model b_0,b_1,b_2,…b_n melalui fungsi logistik untuk menghasilkan probabilitas kelas positif (1). Probabilitas tersebut kemudian digunakan untuk membuat keputusan klasifikasi. Jika P(Y = 1) lebih besar dari atau sama dengan 0.5, data diklasifikasikan sebagai kelas 1; sebaliknya, jika P(Y = 1) kurang dari 0.5, data diklasifikasikan sebagai kelas 0.

3.7 Pelatihan dan Evaluasi

Mengembangkan dan menguji model klasifikasi yang mampu memprediksi apakah suatu berita merupakan *hoax* atau bukan melibatkan serangkaian langkah yang mencakup data pelatihan dan evaluasi kinerja model. Berikut adalah langkah-

langkah tersebut, dengan penyesuaian untuk mencakup algoritma klasifikasi yang berbeda seperti *Logistic Regression*, Decision *Tree*, *Gradient Boosting*, dan *Random Forest*.

Pelatihan Model Klasifikasi:

1. Data Pelatihan:
    a) *Dataset* terdiri dari berita *hoax* dan non-*hoax* dengan label yang menandakan keasliannya.
2. Pembagian Data:
    a) *Dataset* dibagi menjadi dua subset: data pelatihan *(train set)* dan data validasi *(validation set)*.
    b) Data pelatihan digunakan untuk melatih model, dan data validasi digunakan untuk menguji performa model secara objektif.
3. Proses Pelatihan:
    a) Model klasifikasi, seperti *Logistic Regression, Decision Tree, Gradient Boosting, atau Random Forest*. Diterapkan pada data pelatihan.
    b) Model mempelajari pola hubungan antara fitur-fitur *(tf-idf)* dari teks berita dan label (*hoax* atau *non-hoax*) pada data pelatihan.
    c) Secara iteratif, model mengidentifikasi pola-pola yang membedakan berita *hoax* dan *non-hoax* dan menyesuaikan parameter internal untuk meningkatkan akurasi.
4. Penyetelan Parameter:
    a) Penyetelan parameter algoritma klasifikasi dilakukan untuk

mendapatkan hasil optimal.

b) Misalnya, dalam *Logistic Regression*, parameter seperti regulasi dapat disesuaikan untuk meningkatkan performa model.

Evaluasi Model Klasifikasi:

1. Data Validasi:
    a) Data validasi adalah dataset terpisah yang tidak digunakan selama pelatihan.
2. Prediksi dan Evaluasi:
    a) Model melakukan prediksi pada data validasi dan menghasilkan label prediksi untuk setiap berita.
    b) Prediksi dibandingkan dengan label sebenarnya dalam data validasi untuk mengevaluasi kinerja model.
3. Metrik Evaluasi:
    a) Metrik evaluasi seperti akurasi, presisi, *recall*, dan *f1-score* digunakan untuk mengukur performa model.
    b) Metrik ini memberikan pandangan komprehensif tentang kemampuan model dalam mengklasifikasikan berita *hoax* dan *non-hoax*.

Metrik performa model klasifikasi:

1. *True Positive (TP)*: Jumlah observasi positif yang benar diprediksi oleh model.
2. *True Negative (TN)*: Jumlah observasi negatif yang benar diprediksi oleh model.

3. *False Positive (FP)*: Jumlah observasi negatif yang salah diprediksi sebagai positif oleh model.
4. *False Negative (FN)*: Jumlah observasi positif yang salah diprediksi sebagai negatif oleh model.
5. Dengan menggunakan TP, TN, FP, dan FN, kita dapat menghitung beberapa metrik evaluasi sebagai berikut:

- Akurasi *(Accuracy)*: Mengukur sejauh mana model dapat memprediksi secara benar, dinyatakan sebagai rasio dari total prediksi yang benar terhadap total observasi (Baiq Nurul Azmi et al. 2023).

$$\text{Akurasi} = \frac{\text{TP+TN}}{\text{TP + TN + FP + FN}}$$

- Presisi *(Precision)*: Mengukur sejauh mana prediksi positif model adalah benar, dinyatakan sebagai rasio dari TP terhadap total prediksi positif.

$$\text{Presisi} = \frac{\text{TP}}{\text{TP + FP}}$$

- *Recall (Sensitivitas atau True Positive Rate)*: Mengukur sejauh mana model dapat mendeteksi observasi positif, dinyatakan sebagai rasio dari TP terhadap total observasi positif yang seharusnya terdeteksi.

$$\textit{Recall} = \frac{\text{TP}}{\text{TP + FN}}$$

- *F1-Score*: Merupakan harmonic mean dari presisi dan *recall*. Berguna ketika terdapat *trade-off* antara presisi dan *recall*.

$$\textit{F1-Score} = \frac{\text{2x (Presisi x } \textit{Recall}}{\text{Presisi + } \textit{Recall}}$$