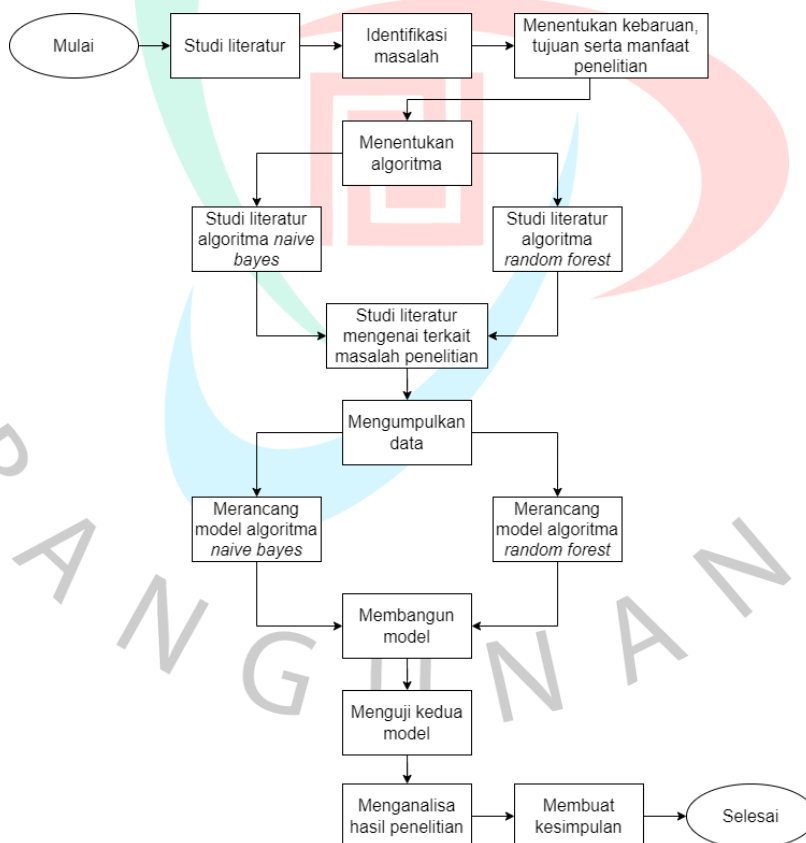


## BAB IV PERENCANAAN

Dalam penelitian terdapat langkah-langkah serta perencanaan yang dilakukan. Langkah dan perencanaan tersebut menjadi pedoman dalam melakukan penelitian sehingga terlaksana dengan sistematis dan sesuai dengan tujuan.

### 4.1. Langkah-langkah Penelitian

Langkah-langkah penelitian dibuat agar penelitian dapat dilakukan secara sistematis dan efisien. Langkah-langkah ini juga berguna sebagai gambaran serta pedoman untuk penulis. Adapun langkah-langkah tersebut dapat dilihat melalui diagram alir pada Gambar 9.



Gambar 4.1 Langkah-langkah penelitian

Berdasarkan diagram alir yang ditunjukkan pada Gambar 4.1, langkah-langkah penelitian dapat dijelaskan sebagai berikut:

1. Studi literatur merupakan kegiatan yang dilakukan untuk menemukan topik yang diteliti. Pada tahap ini dilakukan riset terhadap sejumlah jurnal, artikel, berita, maupun skripsi yang telah diterbitkan secara daring yang telah memiliki kredibilitas.
2. Identifikasi masalah merupakan langkah merumuskan masalah dari topik yang telah ditentukan. Pada tahapan ini juga ditentukan batasan-batasan penelitian sehingga penelitian dapat berjalan dengan efektif dan efisien.
3. Tahapan selanjutnya adalah menentukan kebaruan, tujuan serta manfaat penelitian. Tujuan penelitian dirumuskan agar penelitian ini dapat bermanfaat bagi masyarakat, ilmu pengetahuan dan peneliti. Selain itu pada tahap ini juga ditentukan kebaruan dari penelitian sebelumnya yang memiliki topik serupa.
4. Menentukan algoritma yang akan digunakan untuk menyelesaikan masalah yang telah dirumuskan.
5. Melaksanakan studi literatur terhadap kedua algoritma dengan melakukan riset terhadap penggunaan algoritma tersebut pada penelitian sebelumnya. Pada tahap ini akan diketahui cara kerja, proses, kinerja dan lain-lain dari masing-masing algoritma.
6. Studi literatur terhadap masalah terkait dilakukan untuk menambah pengetahuan penulis terkait masalah tersebut. Selain itu juga berguna untuk memberikan gambaran besar terkait kualitas air di DKI Jakarta.
7. Data penelitian diperoleh dari *website* DKI Jakarta yang berisikan dataset untuk nantinya digunakan sesuai dengan kebutuhan penelitian. Pada tahap ini data juga akan diproses terlebih dahulu agar data dapat menghasilkan informasi yang baik sesuai dengan tujuan penelitian.
8. Membuat rancangan dari masing-masing algoritma. Rancangan dibuat dan akan menghasilkan diagram alir yang berisikan gambaran langkah-langkah pembuatan model.

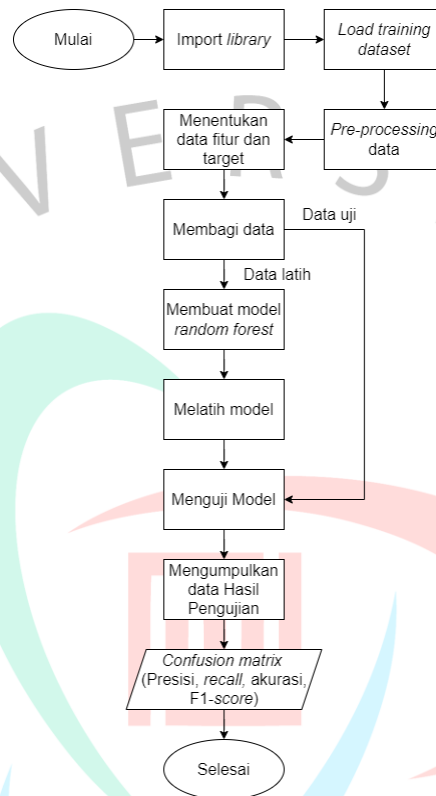
9. Membangun model dari masing-masing algoritma. Pada tahap ini terjadi proses *coding* untuk membangun model dari masing-masing algoritma.
10. Proses pengujian dilakukan menggunakan metode pengujian yang telah ditentukan. Pada tahap ini akan dikumpulkan hasil pengujian dari model yang telah dibangun.
11. Pada tahap ini akan dilakukan analisa terhadap hasil pengujian dari masing-masing algoritma. Analisa ini dilakukan untuk mengetahui algoritma mana yang paling efektif digunakan pada topik kualitas air.
12. Membuat kesimpulan berdasarkan penelitian yang telah dilakukan. Pada tahap ini akan dipaparkan hasil penelitian yang menjadi landasan dari kesimpulan. Pada kesimpulan disajikan hasil presisi, *recall*, akurasi, dan F1-*score* dari masing-masing algoritma dan disimpulkan algoritma yang paling efektif digunakan pada kasus kualitas air. Lalu pada akhir kesimpulan, peneliti akan memberikan saran terhadap penelitian selanjutnya.

#### **4.2. Rancangan Pengujian**

Rancangan pengujian memberikan gambaran jelas terkait model yang dibangun. Rancangan yang baik merupakan landasan dari pembuatan model yang baik. Rancangan diperlukan dalam proses pembuatan model agar penelitian ini dapat berjalan dengan sistematis dan memberikan hasil yang sesuai dengan tujuan penelitian ini.

#### 4.2.1. Diagram Alir *Random Forest*

Diagram alir dari algoritma *random forest* merupakan cara kerja pembuat model *random forest*.



Gambar 4.2 Diagram alir *random forest*

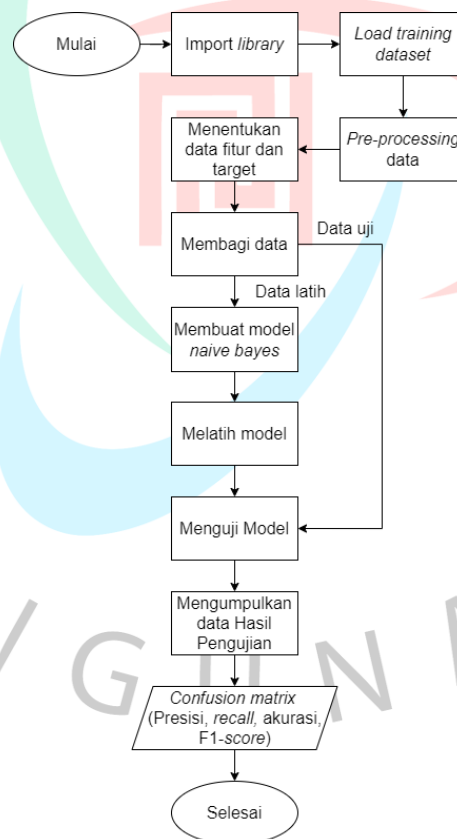
Berikut merupakan penjelasan dari diagram alir algoritma *random forest* di atas.

1. Langkah pertama yang dilakukan dalam membangun model *random forest* adalah melakukan *import library* yang dibutuhkan.
2. Selanjutnya memuat dataset yang dimiliki ke dalam python.
3. Dataset akan memasuki tahap *pre-processing* untuk menghilangkan nilai *null* dan dilakukan normalisasi.
4. Dataset yang telah siap akan dibagi dua menjadi data latih dan data training.
5. Selanjutnya akan dibangun model menggunakan algoritma *random forest*.

6. Model yang telah dibangun akan dilatih menggunakan data latih yang sebelumnya telah dibagi.
7. Lalu model dievaluasi dengan mengujinya menggunakan data uji dan membandingkan nilai hasil prediksinya dengan nilai aktual.
8. Data hasil evaluasi akan dikumpulkan pada tahapan ini.
9. Terakhir, hasil evaluasi tersebut akan ditampilkan performanya berdasarkan presisi, *recall*, akurasi, dan *F1-score*-nya.

#### 4.2.2. Diagram Alir *Naïve Bayes*

Diagram alir dari algoritma *naïve bayes* merupakan alur kerja pembuatan model *naïve bayes*.



Gambar 4.3 Diagram alur *naive bayes*

Berikut merupakan penjelasan dari diagram alir algoritma *naïve bayes* ditunjukkan melalui Gambar 4.3.

1. Untuk membangun model *naïve bayes* hal pertama yang dilakukan adalah meng-*import library* yang dibutuhkan.
2. Selanjutnya memuat dataset yang dimiliki ke dalam python.
3. Lalu pada dataset akan dilakukan *pre-processing* untuk menghilangkan nilai *null* serta dilakukan normalisasi.
4. Dataset yang telah siap dibagi menjadi dua bagian yakni, data latih dan data training.
5. Selanjutnya akan dibangun model menggunakan algoritma *naïve bayes*.
6. Model yang telah dibangun akan dilatih menggunakan data latih yang sebelumnya telah dibagi.
7. Lalu model dievaluasi dengan mengujinya menggunakan data uji dan membandingkan nilai hasil prediksinya dengan nilai aktual.
8. Data hasil evaluasi akan dikumpulkan pada tahapan ini.
9. Terakhir, hasil evaluasi tersebut akan ditampilkan performanya berdasarkan presisi, *recall*, akurasi, dan *F1-score*-nya.

#### **4.2.3. Perbedaan Model *Machine Learning***

Dalam membangun model *machine learning* untuk masing-masing model algoritma, terdapat perbedaan antara keduanya. Meskipun kedua model sama-sama dibangun menggunakan bahasa python, kode program serta parameter yang menjadi kriteria yang digunakan tidaklah sama.

Proses pembangunan model *machine learning* dilakukan ketika data sudah melalui tahapan *pre-processing* serta data telah dibagi menjadi data latih dan data uji. Berikut merupakan kode program yang digunakan dalam membangun model *machine learning* menggunakan algoritma *random forest* dapat dilihat melalui Gambar 4.4 dan 4.5.

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import f1_score, precision_score, recall_score, accuracy_score
```

Gambar 4.4 *Import library* yang dibutuhkan

```
clf=RandomForestClassifier(criterion = "gini", n_estimators=100)
clf.fit(X_train,y_train)
y_pred=clf.predict(X_test)
```

Gambar 4.5 Model *machine learning random forest*

Model *machine learning random forest* dibangun menggunakan library yang tersedia dari python yaitu sklearn. Dengan library tersebut, peneliti dapat menentukan kriteria pembagian pohon. Pada penelitian kali ini penelitian menggunakan gini indeks sebagai kriterianya. Selanjutnya peneliti juga menentukan jumlah pohon dalam model *random forest* tersebut.

Selanjutnya dalam membangun model *machine learning* dengan algoritma *naïve bayes*, peneliti perlu melakukan *pre-processing* terhadap data serta membagi data menjadi data latih dan data uji. Berikut merupakan kode program yang digunakan dalam membangun model *machine learning* menggunakan algoritma *naïve bayes* dapat dilihat melalui Gambar 4.6 dan 4.7.

```
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import f1_score, precision_score, recall_score, accuracy_score
```

Gambar 4.6 *Import library* yang dibutuhkan

```
model = GaussianNB(var_smoothing=1e-17)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
```

Gambar 4.7 Model *machine learning naive bayes*

Model *machine learning naïve bayes* dibangun menggunakan library yang tersedia dari python yaitu sklearn. Dengan library tersebut, peneliti menentukan jenis *classifier* yang digunakan. Pada penelitian kali ini penelitian menggunakan

Gaussian sebagai *classifier*-nya. Selanjutnya peneliti juga nilai pada *var\_smoothing* agar nilai 0 tidak langsung menghasilkan probabilitas nol.

Maka dari itu, peneliti menjabarkan perbedaan dari masing-masing algoritma yang ditunjukkan melalui Tabel 4.1 berikut.

Tabel 4.1 Perbedaan model *machine learning random forest* dan *naive bayes*

| Random Forest                                                                         | Naïve Bayes                                                                          |
|---------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> <li>Menentukan kriteria pembagian pohon</li> </ul> | <ul style="list-style-type: none"> <li>Menentukan jenis <i>classifier</i></li> </ul> |
| <ul style="list-style-type: none"> <li>Menentukan jumlah pohon</li> </ul>             | <ul style="list-style-type: none"> <li>Menentukan <i>var_smoothing</i></li> </ul>    |

#### 4.2.4. Data

Dataset yang diolah merupakan data yang berasal dari Jakarta Open data yang telah disaring sesuai dengan batasan yang telah ditetapkan. Batasan tersebut berguna untuk menjaga alur penelitian agar tetap pada inti permasalahan. Proses penyaringan data dilakukan dengan memilih fitur yang akan digunakan untuk proses klasifikasi. Data tersebut memiliki ketentuan diantaranya sebagai berikut.

1. Data yang digunakan merupakan data dari daerah DKI Jakarta.
2. Data berasal kurang dari lima tahun dihitung dari tahun ini.
3. Kategori yang klasifikasi menggunakan standar Metoda Indeks Pencemaran yakni, memenuhi baku mutu, tercemar ringan, tercemar sedang, dan tercemar berat.
4. Fitur yang menjadi *input* dalam pengujian adalah kekeruhan, air raksa, besi (Fe), fluorida, kesadahan, krom heksavalen, mangan (Mn), nitrat, nitrit, seng (Zn), sulfat, detergen, KMnO4, total coliform, dan E.coli.
5. Fitur yang menjadi *output* dalam pengujian adalah kategori.
6. Hasil akhir adalah akurasi, presisi, recall, F1-score dari pemodelan kualitas air.



Pada proses penyaringan data, data pada fitur kategori akan diubah menjadi integer sesuai dengan batasan yang tertera diatas. Perubahan tersebut dapat dilihat pada Tabel 4.2.

Tabel 4.2 Kode data

| No | Kolom    | Keterangan         | Kode Integer |
|----|----------|--------------------|--------------|
| 1  | Kategori | Memenuhi Baku Mutu | 0            |
| 2  | Kategori | Tercemar Ringan    | 1            |
| 3  | Kategori | Tercemar Sedang    | 2            |
| 4  | Kategori | Tercemar Berat     | 3            |

Proses perubahan data ini terjadi pada tahap *pre-precossing*. Pada tahap tersebut data akan diubah dari non-numerik menjadi numerik. Perubahan tersebut dilakukan agar mua data yang masuk ke dalam dataset berjenis numerik.