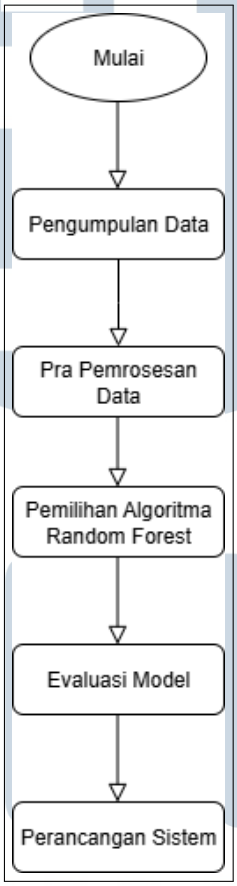
BAB 3
METODOLOGI PENELITIAN

Pada bagian ini dijabarkan langkah-langkah yang hendak dilakukan dalam menyusun dan mengerjakan penelitian. Adapun tahapan utama dalam penelitian ini mencakup pengumpulan data, pra-pemrosesan, perancangan model, implementasi algoritma, pengujian, evaluasi, dan penyusunan prototipe sistem berbasis web.



Gambar 3.1. Metodologi penelitian

3.1 Pengumpulan Data

Tahap pengumpulan data pada penelitian ini menggunakan pendekatan bioinformatika secara sistematis untuk memperoleh dataset skala besar yang relevan. Proses ini memanfaatkan database bioaktivitas publik ChEMBL melalui Application Programming Interface (API) yang disediakannya.

Target biologis yang dipilih adalah EGFR (ID: CHEMBL203) pada manusia, yang perannya krusial dalam kanker paru-paru. Menggunakan skrip Python dengan pustaka chembl webresource client, query diajukan ke database untuk menarik semua data aktivitas biologis yang diukur dengan standar IC50. Dari hasil query, data yang relevan seperti ID molekul, struktur kimia dalam format SMILES, nilai IC50 , HBA, HBD, Massa molekul (g/mol), logP diekstraksi untuk membentuk dataset mentah.

3.2 Pra-Pemrosesan Data

Tahapan pra-pemrosesan data dilakukan untuk memastikan bahwa data bersih, lengkap, konsisten, dan siap digunakan untuk pelatihan model machine learning. Proses ini sangat krusial untuk menghasilkan model yang akurat dan dapat digeneralisasi. Pada tahap pengumpulan data awal, berhasil dihimpun sebanyak 16711 senyawa dari berbagai literatur. Namun, dataset mentah ini masih mengandung beberapa inkonsistensi, seperti nilai yang hilang (missing values) dan data yang tidak lengkap, sehingga tidak dapat langsung digunakan. Langkah-langkah yang dilakukan mencakup:

1. Pemuatan dan Inspeksi Awal Data: Data mentah yang telah dikumpulkan memanfaatkan database bioaktivitas publik ChEMBL melalui API dan menggunakan pustaka Pandas di lingkungan Jupyter Notebook. Inspeksi awal dilakukan untuk memahami struktur data, tipe data setiap kolom, dan mengidentifikasi keberadaan nilai yang hilang (ditandai sebagai N/A atau kosong).

```
import pandas as pd
import numpy as np
from chembl_webresource_client.new_client import new_client
from rdkit import Chem
from rdkit.Chem import AllChem
from tqdm.notebook import tqdm

activity = new_client.activity

target_id = 'CHEMBL203'
print(f"Menggunakan target spesifik: Human EGFR (ID: {
    target_id})")

```

```
print("\nMengambil semua data aktivitas IC50 untuk target...")
activities_iterator = activity.filter(target_chembl_id=target_id, standard_type="IC50")

print("\nMemproses setiap data aktivitas...")
processed_data = []
for act in tqdm(activities_iterator, desc="Processing Activities"):
    smiles = act.get('canonical_smiles')
    value = act.get('standard_value')
    mol_id = act.get('molecule_chembl_id')

    if smiles and value:
        processed_data.append({
            'molecule_chembl_id': mol_id,
            'canonical_smiles': smiles,
            'standard_value': value
        })

df = pd.DataFrame(processed_data)

if df.empty:
    print("\nTidak ada data aktivitas valid yang ditemukan untuk target ini. Proses dihentikan.")
else:
    print(f"\nSelesai. Ditemukan {len(df)} data aktivitas yang lengkap.")
```

Kode 3.1: Load dataset

Dalam kolom-kolom dari tabel tersebut terdapat kolom *molecular fingerprint*. Lalu, setelah mengambil dataset untuk *molecular fingerprint* diambil juga dataset untuk fitur HBA, HBD, Massa Molekul dan LogP. Kemudian, fitur-fitur tersebut digabungkan.

2. Menggabungkan Fitur: Setelah di load dataset, semua fitur digabungkan dan berhasil dibuat dataset gabungan.

```
        print("Menggabungkan semua fitur...")

target_dan_smiles = final_df[['canonical_smiles', 'Potensi Antikanker']]

```

```
df_gabungan = pd.concat([target_dan_smiles, deskriptor_awal,
    df_fingerprints], axis=1)

df_gabungan.dropna(inplace=True)
df_gabungan['Potensi Antikanker'] = df_gabungan['Potensi
    Antikanker'].astype(int)


print(f"Dataset gabungan berhasil dibuat dengan {len(
    df_gabungan)} baris dan {df_gabungan.shape[1]} kolom.")
```

Kode 3.2: Menggabungkan fitur

3. Penanganan Missing Values: Setelah inspeksi awal, dari total 16711 senyawa, teridentifikasi bahwa 6637 di antaranya memiliki data yang tidak lengkap pada kolom fitur yang krusial. Karena kelengkapan fitur-fitur ini esensial untuk model klasifikasi, maka diputuskan untuk mengeliminasi baris data yang tidak lengkap tersebut. Setelah proses eliminasi tersebut, diperoleh dataset akhir yang bersih dan siap digunakan sebanyak 10074 senyawa, di mana setiap senyawa memiliki data yang lengkap untuk fitur-fitur yang dibutuhkan.

```
        print("\nMemulai pembersihan data...")
    df['standard_value'] = pd.to_numeric(df['standard_value'
    ])
    df_clean = df.sort_values('standard_value', ascending=
    True).drop_duplicates('molecule_chembl_id', keep='first')
    print(f"Setelah pembersihan, tersisa {len(df_clean)}
    senyawa unik.")
```

Kode 3.3: Pembersihan data

4. Ekstraksi dan Pemilihan Fitur: Fitur-fitur yang akan digunakan dalam model klasifikasi adalah HBD, HBA, massa molekul, logP dan *molecular fingerprint*. Fitur-fitur ini sudah tersedia di dataset yang dikumpulkan.

5. Balancing dataset menggunakan Random Undersampling: Dataset mentah yang sudah didapat adalah dataset yang tidak balance sehingga digunakan metode *Random Undersampling* untuk membuat datasetnya balance dengan cara eliminasi kelas mayoritas.

```
df_majority = df_full[df_full['Potensi Antikanker'] == 1]
```

```
df_minority = df_full[df_full['Potensi Antikanker'] == 0]

if len(df_minority) > len(df_majority):
    df_majority, df_minority = df_minority, df_majority

print(f"\nUkuran Awal - Kelas Mayoritas (Label {df_majority['
    Potensi Antikanker'].iloc[0]}): {len(df_majority)}")
print(f"Ukuran Awal - Kelas Minoritas (Label {df_minority['
    Potensi Antikanker'].iloc[0]}): {len(df_minority)}")

df_majority_downsampled = df_majority.sample(n=len(
    df_minority), random_state=42)

df_balanced = pd.concat([df_majority_downsampled, df_minority
    ])

df_balanced = df_balanced.sample(frac=1, random_state=42).
    reset_index(drop=True)

balanced_file_path = 'dataset_undersampled_balanced.csv'
df_balanced.to_csv(balanced_file_path, index=False)

print(f"\nDataset yang sudah seimbang berhasil disimpan
    sebagai '{balanced_file_path}'")
print("\nDistribusi kelas setelah penyeimbangan (
    Undersampling):")
print(df_balanced['Potensi Antikanker'].value_counts())
print(f"\nTotal data yang siap digunakan sekarang: {len(
    df_balanced)}")
print(f"Jumlah kolom fitur: {len(df_balanced.columns)}")
print("\nContoh 5 baris pertama dari dataset yang seimbang:")
display(df_balanced.head())
```

Kode 3.4: Balancing dataset

6. Pembagian Data Latih dan Data Uji: Dataset yang telah bersih dan *balance* kemudian dibagi menjadi data latih (training data) dan data uji (testing data). Proporsi pembagian yang digunakan adalah 80 % untuk data latih dan 20 % untuk data uji. Pembagian ini dilakukan dengan strategi stratified sampling untuk memastikan bahwa distribusi kelas (potensial vs. tidak potensial) terjaga proporsinya di kedua subset data.

```
X = balanced_df.drop(columns=['Potensi Antikanker', 'canonical_smiles'])
y = balanced_df['Potensi Antikanker']

X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                    test_size=0.2,

    random_state=42,
                                                    stratify=y)

print("Proses pembagian data selesai.")
print("==================================")
print(f"Jumlah total data latih (X_train): {X_train.shape[0]}")
print(f"Jumlah total data uji (X_test): {X_test.shape[0]}")
print("\nDistribusi kelas pada Data Latih (y_train):")
print(y_train.value_counts())
print("\nDistribusi kelas pada Data Uji (y_test):")
print(y_test.value_counts())
```

Kode 3.5: Data latih dan data uji

```
Proses pembagian data selesai.
==================================
Jumlah total data latih (X_train): 2257
Jumlah total data uji (X_test): 565

Distribusi kelas pada Data Latih (y_train):
1    1129
0    1128
Name: Potensi Antikanker, dtype: int64

Distribusi kelas pada Data Uji (y_test):
0    283
1    282
Name: Potensi Antikanker, dtype: int64
```

Gambar 3.2. Pembagian data latih dan data uji

3.3 Pemilihan Algoritma Random Forest dan Hyperparameter tuning

Model klasifikasi dibangun menggunakan algoritma Random Forest yang diimplementasikan melalui pustaka scikit-learn dalam lingkungan Jupyter Notebook dengan bahasa pemrograman Python.

1. Pemilihan dan Inisialisasi Model: Random Forest dipilih karena kemampuannya dalam menangani data dengan dimensi tinggi, mengurangi overfitting, dan kinerjanya yang stabil. Model diinisialisasi dengan parameter tertentu yang telah dioptimasi atau dipilih berdasarkan praktik umum:

```
param_dist = {
    'n_estimators': [100, 200, 300, 500],
    'max_depth': [10, 20, 30, 40, None],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4],
    'max_features': ['sqrt', 'log2']
}

rf = RandomForestClassifier(random_state=42, class_weight='
    balanced')
```

Kode 3.6: Pemilihan model

2. Hyperparameter tuning: Tuning dilakukan menggunakan RandomizedSearchCV (Metode Pencarian Cerdas), metode ini tidak mencoba semuanya dan hanya akan mencoba sejumlah kombinasi acak dari parameter yang kita tentukan.

```
random_search = RandomizedSearchCV(estimator=rf,
                                   param_distributions=
    param_dist,
                                   n_iter=50,
                                   cv=5,
                                   verbose=1,
                                   random_state=42,
                                   n_jobs=-1,
                                   scoring='f1_macro')

print("Memulai proses hyperparameter tuning...")

random_search.fit(X_train, y_train)
print("Tuning selesai.")
```

```
14
15
16 print(f"\nParameter terbaik yang ditemukan: {random_search.
       best_params_}")
17
18 best_model = random_search.best_estimator_
```

Kode 3.7: Hyperparameter tuning

3. Prediksi: Setelah model dilatih, kemampuannya untuk memprediksi kelas baru diuji menggunakan data uji (X-test).

```
1 y_pred = best_model.predict(X_test)
```

Kode 3.8: Model memprediksi

4. Laporan klasifikasi dan Confusion matrix: dijalankan kode untuk *print* laporan klasifikasi dan confusion matrix

```
1 print("\nLaporan Klasifikasi:")
2 print(classification_report(y_test, y_pred))
3
4 print("\nConfusion Matrix:")
5 cm = confusion_matrix(y_test, y_pred)
6 disp = ConfusionMatrixDisplay(confusion_matrix=cm)
7 disp.plot(cmap=plt.cm.Blues)
8 plt.show()
```

Kode 3.9: Laporan klasifikasi dan confusion matrix

3.4 Evaluasi Model

Untuk mengevaluasi performa dan robustisitas model, digunakan metode 5-Fold Cross-Validation. Dalam metode ini, dataset dibagi menjadi lima bagian. Proses pelatihan dan pengujian diulang sebanyak lima kali, di mana setiap bagian secara bergantian digunakan sebagai data uji. Skor akurasi dari kelima iterasi tersebut kemudian dirata-ratakan untuk menghasilkan skor evaluasi akhir yang lebih stabil.

Selain itu, untuk analisis yang lebih mendalam, dilakukan juga satu kali pembagian data latih-uji (80:20) untuk menghasilkan metrik-metrik evaluasi seperti Laporan Klasifikasi (Precision, Recall, F1-Score), Confusion Matrix, dan Kurva ROC (AUC).

```
Menjalankan 5-Fold Cross-Validation untuk beberapa metrik...
Proses Cross-Validation selesai.

==================================
HASIL CROSS-VALIDATION
==================================
Akurasi Rata-rata      : 0.8646
Presisi Rata-rata (Macro) : 0.8671
Recall Rata-rata (Macro)  : 0.8646
F1-Score Rata-rata (Macro): 0.8644
```

Gambar 3.3. Cross validation

3.5 Perancangan Sistem

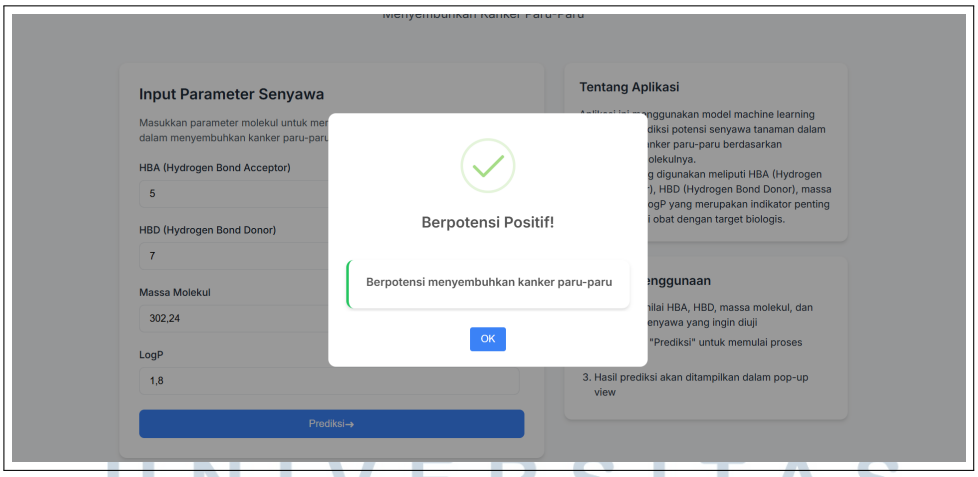
Setelah model klasifikasi Random Forest berhasil dibangun dan dievaluasi, sebuah prototipe sistem klasifikasi sederhana dikembangkan menggunakan HTML, CSS, JavaScript. Sistem ini dirancang untuk memberikan antarmuka yang ramah pengguna, memungkinkan pengguna untuk memasukkan parameter kimiawi suatu senyawa dan menerima prediksi apakah senyawa tersebut berpotensi sebagai senyawa antikanker paru-paru

1. Arsitektur Sistem: Sistem prototipe ini mengadopsi arsitektur sederhana berbasis client-server lokal, di mana antarmuka pengguna dibangun dengan HTML, CSS, JavaScript dan model Random Forest yang telah dilatih berfungsi sebagai backend untuk melakukan prediksi.
    - User Interface (HTML, CSS, JavaScript): Menyediakan formulir input untuk parameter senyawa dan menampilkan hasil prediksi.
    - Model Backend (Python/Random Forest): Memuat model Random Forest yang telah dilatih dan memproses input dari UI untuk menghasilkan prediksi.
    - Data Model: Model Random Forest yang telah diserialisasi (disimpan) dalam format .joblib.
2. Desain Antarmuka Pengguna (UI): Desain *User Interface* dibangun menggunakan HTML, CSS, JavaScript yang memungkinkan pembuatan aplikasi web interaktif dengan kode Python murni. UI terdiri dari:
    - Judul aplikasi yang jelas.
    - Kolom input numerik

- Tombol ”Prediksi” untuk memicu proses klasifikasi.
- Area output yang menampilkan hasil prediksi (misalnya, ”Senyawa ini Berpotensi sebagai Antikanker” atau ”Senyawa ini Tidak Berpotensi sebagai Antikanker”)

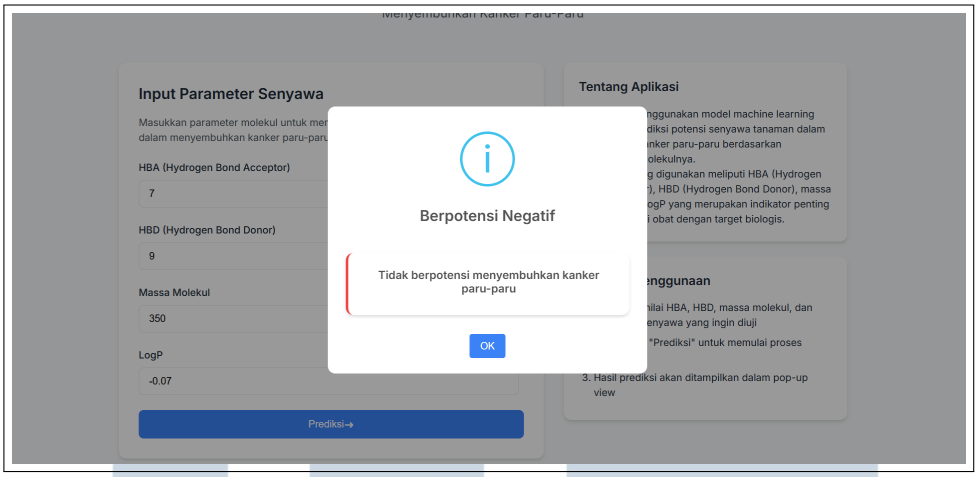
3. Fungsionalitas Sistem:

- Pengguna memasukkan nilai-nilai numerik dan format SMILES pada kolom input yang telah disediakan.
- Setelah semua input terisi, pengguna menekan tombol ”Prediksi”.
- Sistem akan mengambil nilai input, melakukan pra-pemrosesan (standarisasi) yang sama seperti pada data pelatihan, dan kemudian memasukkannya ke dalam model Random Forest yang telah dimuat.
- Model akan menghasilkan prediksi (0 atau 1) dan probabilitas.
- Hasil prediksi akan ditampilkan kepada pengguna di antarmuka web, menunjukkan apakah senyawa tersebut ”Berpotensi” atau ”Tidak Berpotensi” sebagai senyawa antikanker paru-paru.



Gambar 3.4. Ilustrasi hasil prediksi senyawa yang berpotensi

Gambar diatas merupakan ilustrasi jika senyawa adalah senyawa yang berpotensi menyembuhkan penyakit kanker paru-paru.



Gambar 3.5. Ilustrasi hasil prediksi senyawa yang tidak berpotensi

Gambar diatas merupakan ilustrasi jika senyawa adalah senyawa yang tidak berpotensi menyembuhkan penyakit kanker paru-paru.