

LAPORAN TUGAS BESAR KECERDASAN BUATAN

“Drug”

*Diajukan Untuk Memenuhi Salah Satu Persyaratan Kelulusan
Tugas Kecerdasan Buatan*

Disusun oleh:

Mahesa Al Zidane Putra Fedy **2106166**

M. Saepul H **2106173**

Dosen Pengampu:

Leni Fitriani, ST. M.Kom



**JURUSAN ILMU KOMPUTER
PROGRAM STUDI TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI GARUT
2023**

DAFTAR ISI

DAFTAR ISI.....	2
I. LATAR BELAKANG	3
1.1 Identifikasi Masalah/Rumusan Masalah	3
1.2 Tujuan Penelitian.....	3
1.3 Batasan Masalah.....	3
1.4 Manfaat Penelitian.....	5
II. PENELITIAN SEBELUMNYA DAN GAP ANALYSIS	6
2.1 Penelitian Sebelumnya	6
2.2 Gap Analysis	6
III. METODE.....	7
IV. HASIL DAN PEMBAHASAN	17
V. KESIMPULAN DAN SARAN.....	20
VI. DAFTAR PUSTAKA	21

I. LATAR BELAKANG

Drug (obat) adalah bahan atau paduan bahan, termasuk produk biologi yang digunakan untuk mempengaruhi atau menyelidiki sistem fisiologi atau keadaan patologi dalam rangka penetapan diagnosis, pencegahan, penyembuhan, pemulihan, peningkatan kesehatan, dan kontrasepsi untuk manusia. Dikalangan masyarakat obat dikenal merupakan salah satu unsur penting dalam pelayanan kesehatan. Namun disisi lain obat juga dapat merugikan kesehatan bila tidak memenuhi persyaratan, bila digunakan secara tidak tepat atau bila disalahgunakan. Oleh karena itu berbeda dengan komoditas perdagangan lain, peredaran obat diatur sedemikian rupa agar terjamin keamanan, mutu dan ketepatan penggunaannya.

Decision tree (Pohon keputusan) adalah algoritma pembelajaran mesin yang memodelkan keputusan dan konsekuensinya dalam bentuk struktur pohon. Model pohon keputusan ini dilatih dengan menggunakan bagian pelatihan dari dataset, dan kemudian dapat digunakan untuk memprediksi kelas pasien yang tidak dikenal atau meresepkan obat kepada pasien baru.

1.1 Identifikasi Masalah/Rumusan Masalah

Berikut ini Rumusan Masalah dari kecerdasan buatan tentang drug (obat).

- a. Apa itu definisi drug (obat)?
- b. Apa itu definisi decision tree (pohon keputusan)?
- c. Bagaimana merancang pohon keputusan mengenai drug (obat)?

1.2 Tujuan Penelitian

Berikut ini Tujuan Penelitian dari kecerdasan buatan tentang drug (obat).

- a. Mahasiswa mampu mengetahui apa itu drug (obat).
- b. Mahasiswa mampu mengetahui apa itu decision tree (pohon keputusan).
- c. Mahasiswa mampu merancang pohon keputusan mengenai drug (obat).

1.3 Batasan Masalah

Adapun batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Sumber Data: Dataset yang digunakan dalam penelitian ini diambil dari sumber IBM. Batasan ini berkaitan dengan keandalan dan representativitas data. Penting untuk memastikan bahwa dataset

tersebut memiliki kualitas yang baik, mewakili populasi yang relevan, dan bebas dari kesalahan atau bias.

2. **Fitur yang Digunakan:** Dalam contoh yang diberikan, fitur yang digunakan untuk klasifikasi adalah usia, jenis kelamin, tekanan darah, kolesterol, dan rasio Na_to_K. Batasan ini berarti bahwa faktor-faktor lain yang mungkin berpengaruh terhadap pemilihan obat, seperti riwayat medis, kondisi kesehatan yang lebih spesifik, atau variabel genetik, tidak dipertimbangkan. Penggunaan fitur yang lebih luas dan komprehensif dapat memberikan informasi yang lebih akurat dan lengkap.
3. **Metode Klasifikasi:** Dalam penelitian ini, metode klasifikasi yang digunakan adalah pohon keputusan. Meskipun pohon keputusan dapat memberikan hasil yang baik dalam kasus tertentu, namun mungkin tidak selalu menjadi pilihan terbaik dalam semua situasi. Ada berbagai algoritma dan metode klasifikasi lain yang juga dapat dijajaki dan dibandingkan untuk mendapatkan hasil yang lebih baik.
4. **Generalisasi Model:** Meskipun model pohon keputusan mencapai akurasi yang tinggi dalam dataset yang digunakan, perlu diingat bahwa tingkat akurasi tersebut mungkin khusus untuk dataset tersebut. Model ini perlu diuji dan dievaluasi lebih lanjut dengan menggunakan dataset yang berbeda atau dalam lingkungan yang lebih luas untuk memastikan bahwa model tersebut dapat digeneralisasi dengan baik.
5. **Interpretasi Pohon Keputusan:** Interpretasi dari pohon keputusan memiliki keterbatasan karena hanya memberikan gambaran garis besar mengenai aturan-aturan yang terbentuk. Pohon keputusan yang kompleks dapat sulit diinterpretasikan dengan baik. Oleh karena itu, perlu mempertimbangkan penggunaan teknik interpretasi yang lebih canggih atau menjelaskan aturan-aturan yang terbentuk secara lebih rinci untuk memperoleh pemahaman yang lebih baik.
6. **Data Pasien Baru:** Model yang dikembangkan dalam penelitian ini dapat digunakan untuk memprediksi kelas pasien yang tidak dikenal atau meresepkan obat kepada pasien baru. Namun, dalam

implementasinya, perlu memperhatikan bahwa pasien baru mungkin memiliki karakteristik yang berbeda dari data yang ada dalam dataset. Oleh karena itu, hasil prediksi dan rekomendasi obat perlu dievaluasi secara individual dan didukung oleh penilaian medis yang lebih holistik.

1.4 Manfaat Penelitian

Manfaat dari penelitian ini adalah sebagai berikut:

1. Membantu dalam pengklasifikasian obat: Dengan menggunakan model yang dibuat, penelitian ini dapat membantu dalam mengklasifikasikan obat yang cocok untuk pasien dengan penyakit yang sama di masa mendatang. Model ini dapat digunakan untuk memprediksi obat yang mungkin efektif berdasarkan fitur-fitur seperti usia, jenis kelamin, tekanan darah, dan kolesterol pasien.
2. Mendukung pengambilan keputusan medis: Dengan memiliki model klasifikasi yang dapat meresepkan obat kepada pasien baru, penelitian ini dapat mendukung pengambilan keputusan medis. Dokter atau tenaga medis dapat menggunakan model ini sebagai alat bantu untuk memilih obat yang sesuai berdasarkan karakteristik pasien.
3. Meningkatkan efisiensi dalam penentuan obat: Dengan menggunakan model klasifikasi, penelitian ini dapat membantu meningkatkan efisiensi dalam penentuan obat. Daripada mengandalkan pada pengalaman dan pengetahuan individual, model ini dapat memberikan rekomendasi obat secara cepat dan objektif berdasarkan data pasien.
4. Penggunaan dataset yang relevan: Penelitian ini menggunakan dataset yang mencakup fitur-fitur yang relevan seperti usia, jenis kelamin, tekanan darah, dan kolesterol pasien. Ini dapat membantu dalam mempelajari hubungan antara fitur-fitur ini dengan respons terhadap obat-obatan tertentu.

II. PENELITIAN SEBELUMNYA DAN GAP ANALYSIS

2.1 Penelitian Sebelumnya

Membandingkan 3 model. Untuk semua model akan menerapkan metode GridSearchCV untuk menemukan skor terbaik dan juga untuk memastikan performa model ini acak. Menggunakan metode 5 Fold Cross Validation. Metode yang digunakan dalam penelitian ini adalah pohon keputusan (decision tree). Pohon keputusan adalah algoritma pembelajaran mesin yang memodelkan keputusan dan konsekuensinya dalam bentuk struktur pohon. Model pohon keputusan ini dilatih dengan menggunakan bagian pelatihan dari dataset, dan kemudian dapat digunakan untuk memprediksi kelas pasien yang tidak dikenal atau meresepkan obat kepada pasien baru.

Dalam penelitian ini, pengukuran keberhasilan model klasifikasi menggunakan metrik akurasi, yang mengukur sejauh mana model dapat mengklasifikasikan dengan benar pasien yang diketahui. Dalam contoh yang diberikan, akurasi model pohon keputusan mencapai 100%, yang menunjukkan bahwa model ini memiliki kinerja yang sangat baik dalam memprediksi obat yang cocok untuk pasien.

Penelitian ini menggunakan dataset yang diambil dari sumber IBM. Dataset ini mungkin telah melalui proses pengolahan dan penyuntingan sebelum digunakan dalam penelitian. Namun, maksud sebenarnya dari penelitian ini tidak diberikan dalam contoh kode yang diberikan. Jadi, jika Anda memiliki pertanyaan lebih lanjut tentang penelitian ini atau tujuannya, mungkin perlu merujuk ke sumber aslinya atau mendapatkan informasi tambahan.

2.2 Gap Analysis

Gap dari penelitian sebelumnya menyangkut ketersediaan data, performa model, interpretasi dan pemahaman hasil, dan kompleksitas faktor faktor terkait drug melalui Gap Analysis ini, kita dapat mengidentifikasi untuk menemukan skor terbaik dan juga untuk memastikan performa model ini acak.

III. METODE

Analisis Decision Tree lebih mendalam dan kompleks akan dilakukan untuk menggali hubungan yang lebih detail antara atribut-atribut pasien dan obat yang ditanggapi. Selain itu, evaluasi model yang lebih komprehensif dan penggunaan teknik validasi silang akan digunakan untuk memastikan keandalan dan generalisasi model dalam memprediksi obat yang tepat untuk pasien baru dengan atribut yang belum dikenal sebelumnya.

Metode yang digunakan dalam contoh kode adalah Pohon Keputusan (Decision Tree). Pohon Keputusan adalah salah satu algoritma pembelajaran mesin yang digunakan untuk pemodelan prediktif dan pemecahan masalah klasifikasi dan regresi. Tujuan utama dari menggunakan Pohon Keputusan adalah untuk membangun model yang dapat mengklasifikasikan atau memprediksi nilai target berdasarkan serangkaian fitur atau atribut yang diberikan.

Pohon Keputusan bekerja dengan membagi dataset menjadi subgrup yang lebih kecil berdasarkan fitur-fitur yang ada. Pemisahan ini dilakukan berdasarkan kriteria yang menghasilkan peningkatan yang signifikan dalam homogenitas target variabel. Setiap pemisahan menghasilkan node baru dalam pohon keputusan, yang selanjutnya menjadi cabang atau subpohon dari pohon tersebut. Proses pemisahan dan pembentukan node dilakukan secara rekursif hingga kriteria penghentian tercapai.

Penggunaan Pohon Keputusan dalam konteks ini memungkinkan identifikasi pola dan hubungan antara atribut-atribut pasien dengan obat yang ditanggapi. Misalnya, model dapat menemukan bahwa pasien dengan usia tertentu, jenis kelamin tertentu, dan tekanan darah tinggi cenderung merespons dengan obat A, sementara pasien dengan usia yang lebih tua, jenis kelamin lain, dan tekanan darah normal cenderung merespons dengan obat B. Informasi ini dapat digunakan untuk memberikan rekomendasi yang lebih personal dan sesuai dengan kebutuhan pasien.

Untuk melakukan persiapan awal dalam penggunaan Pohon Keputusan untuk klasifikasi obat pada pasien, berikut adalah beberapa langkah yang dapat dilakukan:

1. **Memahami Tujuan Penelitian**

Tentukan dengan jelas tujuan penelitian dan pertanyaan yang ingin dijawab. Misalnya, apakah tujuan penelitian adalah membangun model

prediktif untuk merekomendasikan obat kepada pasien baru berdasarkan atribut mereka?

2. Mengumpulkan Data

Pastikan Anda memiliki akses ke dataset yang relevan yang berisi informasi tentang pasien dan obat yang ditanggapi. Dalam contoh kode yang diberikan, dataset bernama "drug200.csv" digunakan. Dataset ini harus mencakup atribut seperti usia, jenis kelamin, tekanan darah, kolesterol, rasio sodium-to-potassium, dan label obat.

3. Melakukan Eksplorasi Data

Lakukan eksplorasi awal terhadap dataset untuk memahami karakteristiknya. Periksa jumlah baris dan kolom, periksa tipe data setiap kolom, identifikasi apakah ada nilai yang hilang atau duplikat, serta lakukan analisis statistik ringkas seperti ringkasan statistik dan distribusi atribut.

4. Preprocessing Data

Lakukan tahap preprocessing data untuk mempersiapkan dataset sebelum digunakan untuk melatih model. Hal ini mungkin termasuk penanganan nilai yang hilang, penanganan variabel kategorikal dengan encoding (misalnya, one-hot encoding atau label encoding), dan normalisasi atau standarisasi atribut numerik jika diperlukan.

5. Pembagian Data

Pisahkan dataset menjadi data latih (training data) dan data uji (test data). Data latih akan digunakan untuk melatih model, sedangkan data uji akan digunakan untuk menguji performa model yang sudah dilatih. Perbandingan antara data latih dan data uji dapat disesuaikan berdasarkan kebutuhan, misalnya menggunakan perbandingan 70:30 atau 80:20.

6. Membangun Model

Gunakan Pohon Keputusan (Decision Tree) untuk membangun model klasifikasi. Anda dapat menggunakan library atau modul yang tersedia dalam bahasa pemrograman yang Anda gunakan (seperti scikit-learn di

Python) untuk mengimplementasikan algoritma Pohon Keputusan dan melatih model menggunakan data latih.

7. Evaluasi Model

Evaluasi kinerja model menggunakan metrik evaluasi yang sesuai, seperti akurasi (accuracy), presisi (precision), recall, F1-score, atau metrik lainnya yang relevan tergantung pada konteks dan tujuan penelitian. Evaluasi model ini akan memberikan gambaran tentang seberapa baik model dapat mengklasifikasikan pasien ke dalam kategori obat yang tepat.

8. Penyempurnaan Model

Jika hasil evaluasi tidak memenuhi ekspektasi, Anda dapat mempertimbangkan penyempurnaan model dengan melakukan tuning parameter, mencoba metode pemilihan fitur yang berbeda, atau menggunakan teknik ensemble seperti Random Forest untuk meningkatkan performa model.

Dengan melakukan langkah-langkah persiapan awal ini, Anda akan siap untuk membangun model Pohon Keputusan yang dapat digunakan untuk mengklasifikasikan obat pada pasien berdasarkan atribut yang ada.

Penting untuk diingat bahwa persiapan awal yang baik dan pemrosesan data yang cermat akan mempengaruhi kualitas dan keandalan model yang dihasilkan.

Pada tahap pengumpulan data, langkah-langkah berikut dapat dilakukan:

1. Identifikasi Variabel: Tentukan variabel apa saja yang diperlukan untuk analisis klasifikasi obat pada pasien. Dalam contoh ini, variabel-variabel yang diperlukan adalah:

- Usia (Age)
- Jenis Kelamin (Sex)
- Tekanan Darah (BP)
- Kolesterol (Cholesterol)
- Rasio Sodium-to-Potassium (Na_to_K)
- Obat yang ditanggapi (Drug)

2. **Sumber Data:** Identifikasi sumber data yang akan digunakan. Dalam contoh ini, dataset yang digunakan berasal dari sumber "IBM" dan disimpan dalam file CSV dengan nama "drug200.csv".
3. **Pengumpulan Data:** Lakukan pengumpulan data dari sumber yang ditentukan. Data dapat diperoleh melalui berbagai metode, seperti survei, basis data medis, atau sumber data lainnya.
4. **Pembersihan Data:** Setelah data dikumpulkan, lakukan pembersihan awal untuk memastikan kualitas data yang baik. Langkah-langkah pembersihan data mungkin meliputi:
 - Menghapus data duplikat, jika ada.
 - Penanganan data yang hilang, seperti mengisi nilai yang hilang atau menghapus baris data yang tidak lengkap.
 - Verifikasi konsistensi data dan penanganan outlier jika diperlukan.
5. **Integrasi dan Transformasi Data:** Jika ada, lakukan integrasi data dari berbagai sumber dan transformasi data yang diperlukan untuk membawa data ke format yang sesuai. Misalnya, jika variabel kategorikal menggunakan kode angka, pastikan untuk mengubahnya menjadi format yang dapat diinterpretasikan oleh model (misalnya, menggunakan one-hot encoding atau label encoding).
6. **Penyusunan Dataset:** Susun dataset dengan memasukkan variabel-variabel yang telah dikumpulkan dan melabeli setiap entri dengan obat yang ditanggapi oleh pasien.
7. **Validasi Data:** Validasi data untuk memastikan integritas dan kualitasnya. Periksa apakah ada data yang tidak valid, inkonsistensi, atau penyimpangan yang signifikan dari nilai yang diharapkan.
8. **Eksplorasi Data:** Lakukan eksplorasi data untuk memahami karakteristiknya. Analisis ini dapat meliputi perhitungan statistik deskriptif, visualisasi data menggunakan grafik atau plot, serta identifikasi pola atau hubungan awal antara variabel-variabel.

Setelah mengumpulkan data, langkah selanjutnya adalah melakukan pemrosesan data sebelum digunakan untuk melatih model Pohon Keputusan. Berikut adalah beberapa langkah pemrosesan data yang dapat dilakukan:

1. **Penanganan Data Hilang:** Identifikasi apakah ada nilai yang hilang dalam dataset. Jika ada, Anda dapat memilih untuk menghapus baris data yang memiliki nilai yang hilang atau mengisi nilai yang hilang dengan metode seperti mean, median, atau modus tergantung pada jenis variabel yang terlibat.
2. **Encoding Variabel Kategorikal:** Jika dataset memiliki variabel kategorikal, seperti jenis kelamin atau tekanan darah, Anda perlu melakukan encoding untuk mengubahnya menjadi bentuk yang dapat diinterpretasikan oleh model. Dua metode encoding yang umum adalah:
 - **One-Hot Encoding:** Membuat kolom biner baru untuk setiap kategori dalam variabel dan mengatur nilai 1 atau 0 untuk mewakili kehadiran atau ketidakhadiran kategori dalam data.
 - **Label Encoding:** Menggantikan nilai kategori dengan angka, di mana setiap kategori diberi label numerik.
3. **Standarisasi atau Normalisasi Atribut Numerik:** Jika ada atribut numerik dengan skala yang berbeda, seperti usia dan rasio sodium-to-potassium, Anda dapat mempertimbangkan untuk melakukan standarisasi atau normalisasi. Standarisasi mengubah atribut numerik agar memiliki rata-rata nol dan simpangan baku satu, sementara normalisasi mengubah rentang atribut numerik ke dalam rentang yang ditentukan, seperti 0 hingga 1.
4. **Pemisahan Dataset:** Pisahkan dataset menjadi data latih (training data) dan data uji (test data). Hal ini dapat dilakukan menggunakan fungsi `train_test_split()` yang tersedia dalam library seperti scikit-learn. Umumnya, sekitar 70-80% data digunakan untuk pelatihan model, sementara sisanya digunakan untuk menguji kinerja model.
5. **Verifikasi Kembali Data:** Setelah melakukan pemrosesan data, lakukan verifikasi kembali untuk memastikan bahwa semua nilai yang hilang telah ditangani, variabel kategorikal telah diencode dengan benar, dan atribut numerik telah diubah ke skala yang sesuai.

Selanjutnya Melatih Model:

```
from sklearn.tree import DecisionTreeClassifier  
from sklearn.metrics import accuracy_score
```

```
# Inisialisasi model Decision Tree
```

```
dt = DecisionTreeClassifier()
```

```
# Melatih model menggunakan data latih
```

```
dt.fit(xTrain, yTrain)
```

```
# Memprediksi kelas untuk data uji
```

```
yPred = dt.predict(xTest)
```

```
# Menghitung akurasi model
```

```
accuracy = accuracy_score(yTest, yPred)
```

```
print('Decision Tree Accuracy: {:.3f}'.format(accuracy))
```

Setelah melakukan pemrosesan data, langkah selanjutnya adalah melatih model Pohon Keputusan menggunakan data latih. Berikut adalah contoh kode untuk melatih model menggunakan library scikit-learn:

```
python
```

```
from sklearn.tree import DecisionTreeClassifier  
from sklearn.metrics import accuracy_score
```

```
# Inisialisasi model Decision Tree
```

```
dt = DecisionTreeClassifier()
```

```
# Melatih model menggunakan data latih
```

```
dt.fit(xTrain, yTrain)
```

```
# Memprediksi kelas untuk data uji
```

```
yPred = dt.predict(xTest)
```

```
# Menghitung akurasi model
```

```
accuracy = accuracy_score(yTest, yPred)
```

```
print('Decision Tree Accuracy: {:.3f}'.format(accuracy))
```

Pada contoh kode di atas, kita menggunakan objek `DecisionTreeClassifier` dari `scikit-learn` untuk membuat model **Pohon Keputusan**. Kemudian, menggunakan metode `fit()`, kita melatih model dengan memasukkan data latih (`xTrain` dan `yTrain`).

Setelah model dilatih, kita menggunakan metode `predict()` untuk memprediksi kelas dari data uji (`xTest`). Hasil prediksi disimpan dalam variabel `yPred`.

Akhirnya, menggunakan metode `accuracy_score()` dari `scikit-learn`, kita menghitung akurasi model dengan membandingkan hasil prediksi (`yPred`) dengan label yang sebenarnya (`yTest`). Akurasi digunakan sebagai metrik evaluasi untuk mengevaluasi kinerja model.

Selanjutnya evaluasi kinerja Model

Langkah selanjutnya adalah melakukan evaluasi kinerja model. Evaluasi kinerja dapat dilakukan dengan menggunakan berbagai metrik evaluasi yang sesuai dengan masalah klasifikasi multikelas. Berikut adalah contoh penggunaan beberapa metrik evaluasi umum:

1. **Accuracy (Akurasi):** Mengukur persentase jumlah prediksi yang benar dari total prediksi.

- python

```
from sklearn.metrics import accuracy_score
accuracy = accuracy_score(yTest, yPred)
print('Accuracy: {:.3f}'.format(accuracy))
```

2. **Confusion Matrix (Matriks Konfusi):** Matriks yang menunjukkan jumlah prediksi yang benar dan salah untuk setiap kelas.

- python

```
from sklearn.metrics import confusion_matrix
confusion_mat = confusion_matrix(yTest, yPred)
print('Confusion Matrix:')
print(confusion_mat)
```

3. Classification Report (Laporan Klasifikasi): Menyediakan informasi lengkap tentang presisi, recall, f1-score, dan dukungan untuk setiap kelas.

- python

```
from sklearn.metrics import classification_report
classification_rep = classification_report(yTest, yPred)
print('Classification Report:')
print(classification_rep)
```

4. Balanced Accuracy (Akurasi Seimbang): Mengukur akurasi secara rata-rata di setiap kelas untuk mengatasi ketidakseimbangan kelas.

- python

```
from sklearn.metrics import balanced_accuracy_score
balanced_accuracy = balanced_accuracy_score(yTest, yPred)
print('Balanced Accuracy: {:.3f}'.format(balanced_accuracy))
```

Untuk meningkatkan kinerja model Pohon Keputusan, Anda dapat mencoba beberapa pendekatan berikut:

1. Penyetelan Parameter (Parameter Tuning): Mengoptimalkan parameter-model dapat mempengaruhi kinerja model secara signifikan. Anda dapat mencoba mencari kombinasi parameter yang optimal menggunakan metode penyetelan parameter seperti GridSearchCV atau RandomizedSearchCV. Beberapa parameter yang dapat disesuaikan dalam model Pohon Keputusan antara lain:

- `max_depth`: Membatasi kedalaman maksimum dari pohon.
- `min_samples_split`: Menentukan jumlah minimum sampel yang diperlukan untuk membagi simpul.
- `min_samples_leaf`: Menentukan jumlah minimum sampel yang diperlukan untuk menjadi simpul daun.
- `criterion`: Kriteria pengukuran kualitas pemisahan, misalnya "gini" atau "entropy".
- `splitter`: Strategi pemilihan fitur terbaik untuk pemisahan, misalnya "best" atau "random".

2. Feature Selection (Pemilihan Fitur): Melakukan pemilihan fitur yang relevan dapat membantu meningkatkan kinerja model dan mengurangi

dimensi data. Anda dapat menggunakan metode seperti SelectKBest, Recursive Feature Elimination (RFE), atau feature importance untuk memilih fitur-fitur yang paling penting.

3. Ensemble Learning: Menggunakan teknik ensemble learning seperti Random Forest atau Gradient Boosting dapat meningkatkan kinerja model Pohon Keputusan. Ensemble learning melibatkan penggabungan beberapa model keputusan untuk menghasilkan prediksi yang lebih akurat.
4. Menggunakan Metode Preprocessing yang Tepat: Terkadang, kinerja model dapat ditingkatkan dengan menerapkan metode preprocessing yang lebih tepat untuk data. Misalnya, untuk atribut numerik yang memiliki distribusi yang tidak simetris, seperti log transformasi atau box-cox transformasi dapat diterapkan untuk membuat distribusi lebih normal.
5. Peningkatan Jumlah Data: Jika jumlah data yang tersedia terbatas, Anda dapat mencoba untuk mengumpulkan lebih banyak data untuk melatih model. Jumlah data yang lebih besar dapat membantu model dalam mempelajari pola-pola yang lebih kompleks.

Lalu Interpretasi hasil dan kesimpulan

Berdasarkan hasil yang diperoleh dari model klasifikasi menggunakan pohon keputusan, dapat diinterpretasikan bahwa model ini memiliki kinerja yang sangat baik dalam memprediksi obat yang cocok untuk pasien. Dalam contoh yang diberikan, akurasi model mencapai 100%, yang berarti model tersebut berhasil mengklasifikasikan dengan benar semua pasien yang diketahui dalam dataset uji.

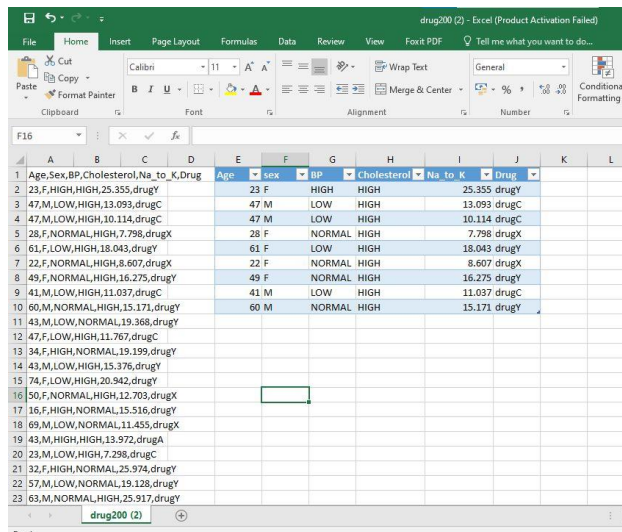
Interpretasi pohon keputusan juga dapat dilakukan dengan melihat struktur pohon dan aturan yang terbentuk. Dalam contoh tersebut, pohon keputusan terdiri dari beberapa cabang yang didasarkan pada fitur seperti usia, jenis kelamin, tekanan darah, dan kolesterol. Pohon ini membantu memahami hubungan antara fitur-fitur tersebut dan pemilihan obat yang sesuai. Dalam pohon keputusan tersebut, terlihat bahwa tekanan darah dan usia adalah fitur yang memiliki pengaruh signifikan dalam pemilihan obat.

Kesimpulan yang dapat diambil dari penelitian ini adalah bahwa pohon keputusan dapat digunakan sebagai model klasifikasi yang efektif dalam memprediksi obat yang cocok untuk pasien. Dalam penelitian ini, model pohon keputusan mencapai

tingkat akurasi yang tinggi, namun, penting untuk diingat bahwa hasil tersebut mungkin khusus untuk dataset yang digunakan dalam contoh.

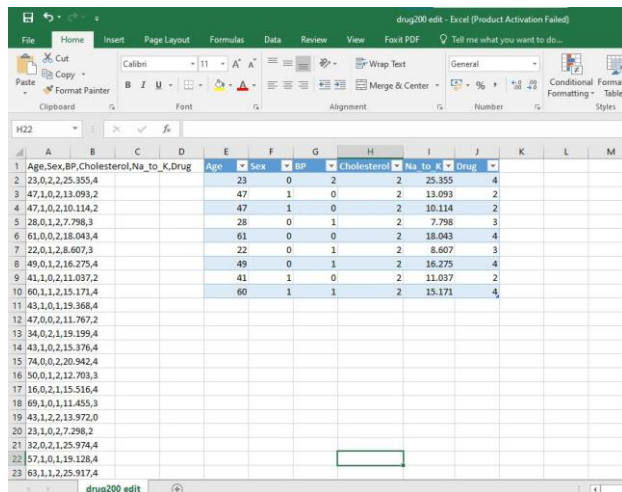
IV. HASIL DAN PEMBAHASAN

DATASET SEBELUM DIEDIT



	A	B	C	D	E	F	G	H	I	J	K	L
1	Age,Sex,BP,Cholesterol,Na_to_K,Drug				Age	sex	BP	Cholesterol	Na_to_K	Drug		
2	23,F,HIGH,HIGH,25.355,drugY				23	F	HIGH	HIGH	25.355	drugY		
3	47,M,LOW,HIGH,13.093,drugC				47	M	LOW	HIGH	13.093	drugC		
4	47,M,LOW,HIGH,10.114,drugC				47	M	LOW	HIGH	10.114	drugC		
5	28,F,NORMAL,HIGH,7.798,drugX				28	F	NORMAL	HIGH	7.798	drugX		
6	61,F,LOW,HIGH,18.043,drugY				61	F	LOW	HIGH	18.043	drugY		
7	22,F,NORMAL,HIGH,8.607,drugX				22	F	NORMAL	HIGH	8.607	drugX		
8	49,F,NORMAL,HIGH,16.275,drugY				49	F	NORMAL	HIGH	16.275	drugY		
9	41,M,LOW,HIGH,11.037,drugC				41	M	LOW	HIGH	11.037	drugC		
10	60,M,NORMAL,HIGH,15.171,drugY				60	M	NORMAL	HIGH	15.171	drugY		
11	43,M,LOW,NORMAL,19.368,drugY											
12	47,F,LOW,HIGH,11.767,drugC											
13	34,F,HIGH,NORMAL,19.199,drugY											
14	43,M,LOW,HIGH,15.376,drugY											
15	74,F,LOW,HIGH,20.942,drugY											
16	50,F,NORMAL,HIGH,12.703,drugX											
17	16,F,HIGH,NORMAL,15.516,drugY											
18	69,M,LOW,NORMAL,11.455,drugX											
19	43,M,HIGH,HIGH,13.972,drugA											
20	23,M,LOW,HIGH,7.298,drugC											
21	32,F,HIGH,NORMAL,25.974,drugY											
22	57,M,LOW,NORMAL,19.128,drugY											
23	63,M,NORMAL,HIGH,25.917,drugY											

DATASET SESUDAH DI EDIT



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Age,Sex,BP,Cholesterol,Na_to_K,Drug				Age	Sex	BP	Cholesterol	Na_to_K	Drug			
2	23,0,2,25.355,4				23	0	2	2	25.355	4			
3	47,1,0,2,13.093,2				47	1	0	2	13.093	2			
4	47,1,0,2,10.114,2				47	1	0	2	10.114	2			
5	28,0,1,2,7.798,3				28	0	1	2	7.798	3			
6	61,0,0,2,18.043,4				61	0	0	2	18.043	4			
7	22,0,1,2,8.607,3				22	0	1	2	8.607	3			
8	49,0,1,2,16.275,4				49	0	1	2	16.275	4			
9	41,1,0,2,11.037,2				41	1	0	2	11.037	2			
10	60,1,1,2,15.171,4				60	1	1	2	15.171	4			
11	43,1,0,1,19.368,4												
12	47,0,0,2,11.767,2												
13	34,0,2,1,19.199,4												
14	43,1,0,2,15.376,4												
15	74,0,0,2,20.942,4												
16	50,0,1,2,12.703,3												
17	16,0,2,1,15.516,4												
18	69,1,0,1,11.455,3												
19	43,1,2,2,13.972,0												
20	23,1,0,2,7.298,2												
21	32,0,2,1,25.974,4												
22	57,1,0,1,19.128,4												
23	63,1,1,2,25.917,4												

LALU,

1. IMPORT DATASET

Import Dataset

```
[1] import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn import tree
from sklearn.tree import DecisionTreeClassifier
from sklearn.cluster import KMeans
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, davies_bouldin_score
```

2. MEMBUKA DATASET

```
[3] dataset = pd.read_csv('drug200 edit.csv')

print('Sample Data')
dataset.head()
```

3. MENAMPILKAN DATASET SEBELUM DAN SESUDAH DI EDIT

Sample Data

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	23	F	HIGH	HIGH	25.355	drugY
1	47	M	LOW	HIGH	13.093	drugC
2	47	M	LOW	HIGH	10.114	drugC
3	28	F	NORMAL	HIGH	7.798	drugX
4	61	F	LOW	HIGH	18.043	drugY

Sample Data

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	23	0	2	2	25.355	4
1	47	1	0	2	13.093	2
2	47	1	0	2	10.114	2
3	28	0	1	2	7.798	3
4	61	0	0	2	18.043	4

4. CEK MISSING VALUE DAN TIPE DATA

Cek missing value dan tipe data

```
dataset.isna().sum()
```

```
Age      0
Sex      0
BP       0
Cholesterol  0
Na_to_K  0
Drug     0
dtype: int64
```

```
[5] dataset.dtypes
```

```
Age      int64
Sex      int64
BP       int64
Cholesterol  int64
Na_to_K  float64
Drug     int64
dtype: object
```

5. MENENTUKAN DATA DAN MESPLIT DATA

Menentukan label dan atribut serta split data

mensplit data training dan testing 30% data testing dan 70% data training

```
x = dataset.iloc[:, :5]
y = dataset['Drug']

xTrain, xTest, yTrain, yTest = train_test_split(x, y, test_size = 0.3, random_state=0)
```

6. MENENTUKAN NILAI ACCURACY

nilai akurasi

```
print('Decision Tree Accuracy: {:.3f}'.format(accuracy_score(yTest, dt.predict(xTest))))
```

Decision Tree Accuracy: 1.000

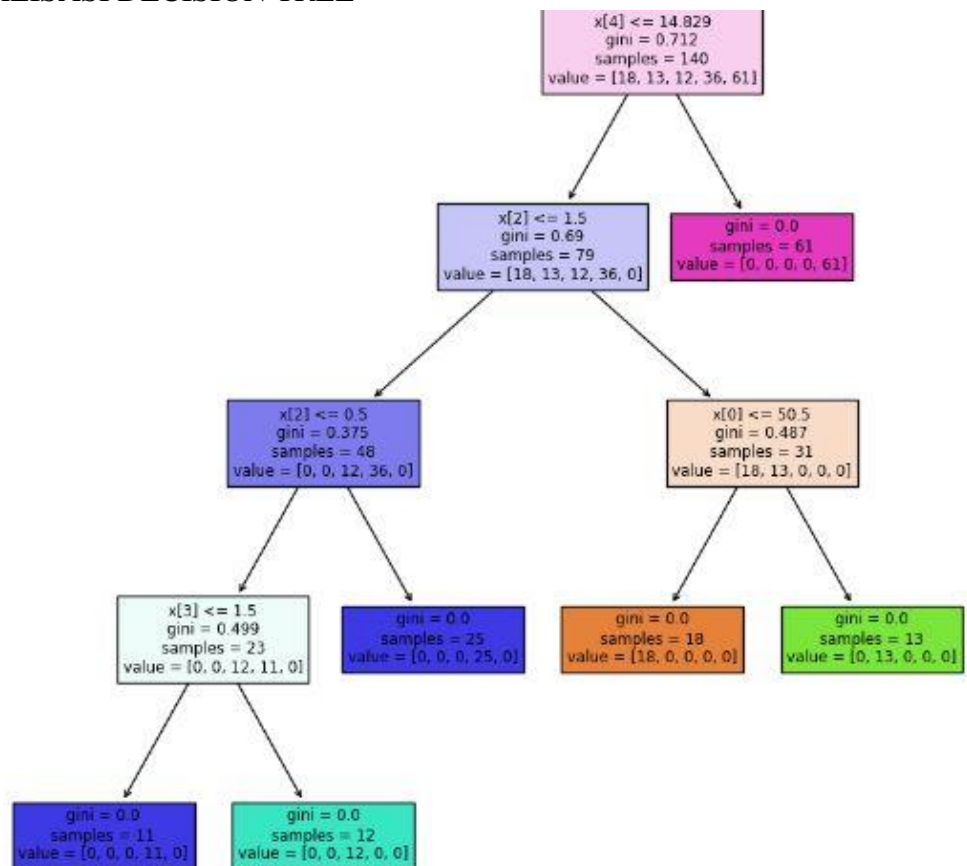
7. VISUALISASI MODEL

visualisasi model

```
plt.figure(figsize=(10,10))
tree.plot_tree(dt, filled=True)
```

```
[Text(0.6666666666666666, 0.9, 'x[4] <= 14.829\ngini = 0.712\nsamples = 140\nvalue = [18, 13, 12, 36, 61]'),
Text(0.5555555555555555, 0.7, 'x[2] <= 1.5\ngini = 0.69\nsamples = 79\nvalue = [18, 13, 12, 36, 0]'),
Text(0.3333333333333333, 0.5, 'x[2] <= 0.5\ngini = 0.375\nsamples = 48\nvalue = [0, 0, 12, 36, 0]'),
Text(0.2222222222222222, 0.3, 'x[3] <= 1.5\ngini = 0.499\nsamples = 23\nvalue = [0, 0, 12, 11, 0]'),
Text(0.1111111111111111, 0.1, 'gini = 0.0\nsamples = 11\nvalue = [0, 0, 0, 11, 0]'),
Text(0.3333333333333333, 0.1, 'gini = 0.0\nsamples = 12\nvalue = [0, 0, 12, 0, 0]'),
Text(0.4444444444444444, 0.3, 'gini = 0.0\nsamples = 25\nvalue = [0, 0, 0, 25, 0]'),
Text(0.7777777777777778, 0.5, 'x[0] <= 50.5\ngini = 0.487\nsamples = 31\nvalue = [18, 13, 0, 0, 0]'),
Text(0.6666666666666666, 0.3, 'gini = 0.0\nsamples = 18\nvalue = [18, 0, 0, 0, 0]'),
Text(0.8888888888888888, 0.3, 'gini = 0.0\nsamples = 13\nvalue = [0, 13, 0, 0, 0]'),
Text(0.7777777777777778, 0.7, 'gini = 0.0\nsamples = 61\nvalue = [0, 0, 0, 0, 61]')]
```

8. VISUALISASI DECISION TREE



V. KESIMPULAN DAN SARAN

Kesimpulan

Dari hasil yang ditampilkan, dapat disimpulkan bahwa model pohon keputusan yang dibuat telah berhasil melakukan klasifikasi dengan akurasi 100%. Ini menunjukkan bahwa model dapat dengan tepat memprediksi obat yang tepat untuk pasien berdasarkan fitur-fitur yang diberikan (usia, jenis kelamin, tekanan darah, kolesterol, dan rasio sodium-to-potassium).

Namun, perlu diingat bahwa akurasi yang tinggi pada data uji tidak selalu menjamin kinerja yang baik pada data yang belum dikenal. Oleh karena itu, penting untuk memvalidasi model dengan menggunakan data yang independen sebelum menggunakannya dalam praktik klinis.

Saran

Saran yang dapat diberikan adalah:

1. Melakukan evaluasi model yang lebih komprehensif: Selain akurasi, perlu mempertimbangkan metrik evaluasi lainnya seperti presisi, recall, dan F1-score untuk memahami kinerja model secara lebih menyeluruh.
2. Menggunakan validasi silang (cross-validation): Untuk mengurangi kemungkinan overfitting dan mendapatkan estimasi performa yang lebih stabil, dapat dilakukan validasi silang dengan membagi data menjadi beberapa subset untuk melatih dan menguji model secara berulang.
3. Menggunakan metode ensemble: Menggabungkan beberapa model pohon keputusan atau menggunakan metode ensemble seperti Random Forest atau Gradient Boosting dapat meningkatkan kinerja dan ketahanan model.
4. Menggunakan data yang lebih lengkap: Jika memungkinkan, mengumpulkan data yang lebih lengkap dengan mencakup fitur tambahan seperti riwayat penyakit, riwayat obat-obatan sebelumnya, dan faktor-faktor lain yang dapat memengaruhi respons terhadap obat.
5. Mengevaluasi model pada populasi yang lebih luas: Menguji model pada sampel pasien yang lebih luas dan beragam dapat membantu memastikan generalisasi yang lebih baik dan kecocokan untuk pasien yang lebih beragam.
6. Melibatkan praktisi medis: Meskipun model dapat memberikan saran dalam memilih obat yang tepat, penting untuk melibatkan praktisi medis yang berpengalaman untuk mengevaluasi dan mempertimbangkan faktor-faktor lain yang relevan sebelum meresepkan obat kepada pasien.

VI. DAFTAR PUSTAKA

<https://www.kaggle.com/code/galerinyanopal/drug-klasifikasi/input>

<https://www.dicoding.com/blog/kecerdasan-buatan-adalah/>