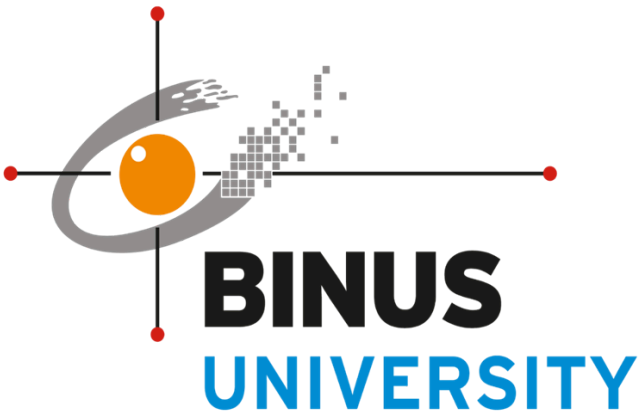
PREDIKSI & VISUALISASI TIPE PENYAKIT ANEMIA

DENGAN SPARK DECISION TREE ALGORITHM

BERDASARKAN SAMPEL DARAH



Big Data Processing - COMP6579001

Oleh:

2602090731 - Jesika Purnomo

2602111723 - Matheus Ariel Reinhart Sidharta

2602084016 - Misia Callista Abdipatra

2602128661 - Nicholas Cahyadi

2602091053 - Raffa Winters

PROGRAM STUDI COMPUTER SCIENCE

SEMESTER GENAP 2023/2024

DAFTAR ISI

BAB 1. LATAR BELAKANG

Tubuh kita memproduksi tiga jenis sel darah, yaitu sel darah putih untuk melawan infeksi, trombosit untuk proses pembekuan darah, dan sel darah merah untuk mengedarkan oksigen ke seluruh tubuh. Penyakit Anemia adalah kondisi dimana jumlah sel darah merah dalam tubuh kita kurang dari jumlah seharusnya. Kurangnya sel darah merah yang berperan untuk menghantarkan oksigen ke seluruh organ tubuh dapat menurunkan fungsi organ tubuh. Anemia umumnya terjadi karena beberapa kondisi antara lain produksi sel darah merah yang kurang, kehilangan banyak darah, hancurnya sel darah merah yang terlalu cepat. Penyakit anemia umumnya ditandai dengan gejala seperti lemas, mudah lelah atau mengantuk, pandangan berkunang-kunang, pusing kepala, dan pucat.

Institute For Health Metric And Evaluation (IHME) menyatakan bahwa kasus penyakit anemia terus meningkat. Sekitar seperempat populasi global diperkirakan menderita anemia, khususnya pada perempuan, ibu hamil, remaja putri, dan anak-anak dibawah usia 5 tahun. Pada tahun 2021, sekitar 1,92 miliar orang secara global menderita anemia, yang mana ini disebabkan oleh adanya peningkatan sebanyak 420 juta kasus anemia selama 3 dekade. Dari beberapa data yang disediakan oleh IHME, kita dapat lihat bahwa penyakit Anemia turut mengancam dan berdampak pada jumlah populasi dunia.

Jika tidak diobati, anemia bisa berdampak fatal seperti menurunnya kecerdasan dan kinerja otak, hal ini dikarenakan supply oksigen ke otak tidak cukup. Selain itu pada anak-anak, anemia akan menghambat tumbuh dan kembang, khususnya oleh anemia yang disebabkan zat besi, karena zat besi berperan untuk menunjang pertumbuhan sel anak. Pada wanita remaja dan dewasa, anemia bisa menyebabkan gangguan pada sistem reproduksi, sebab wanita yang mengidap anemia akan rentan mengalami komplikasi saat hamil dan bersalin. Kurangnya sel darah merah menyebabkan organ-organ tubuh tidak bekerja secara normal, salah satunya ialah jantung. Jika jantung tidak bekerja dengan optimal, jantung akan sulit untuk memenuh kebutuhan oksigen seluruh tubuh, yang jika berlangsung lama, jantung akan kehilangan kemampuan untuk berkontraksi dengan baik, dan terjadi gagal jantung.

Penyakit anemia dapat disebabkan oleh beberapa faktor yang berbeda, sehingga anemia dapat

dibagi menjadi beberapa macam, antara lain:

- Normocytic Hypochromic Anemia

  Anemia Hipokromik Normositik adalah jenis anemia dimana sel darah merah memiliki ukuran yang normal (normositik), namun memiliki warna yang lebih pucat dari seharusnya. Pemucatan ini disebabkan oleh pengurangan hemoglobin sel darah merah yang jumlahnya tidak proporsional dengan volume sel.

- Normocytic Normochromic Anemia

  Anemia Normokromik Normositik adalah jenis anemia dimana sel darah merah memiliki ukuran dan warna yang normal (normositik dan normokromik). Anemia jenis ini umumnya disebabkan oleh penyakit lain, namun dapat juga dipicu oleh kelainan primer darah.

- Iron Deficiency Anemia

  Anemia defisiensi zat besi adalah anemia yang disebabkan oleh kurangnya zat besi dalam tubuh, yang akan mengakibatkan penurunan jumlah sel darah merah sehat. Zat besi adalah mineral yang berperan penting untuk menghasilkan sel darah merah (hemoglobin), yang berfungsi untuk mengangkut oksigen ke seluruh bagian tubuh. Kurangnya produksi hemoglobin membuat tidak tercukupinya asupan oksigen dalam darah berkurang sehingga tubuh tidak mendapat oksigen yang cukup.

- Microcytic Anemia

  Anemia Mikrositik adalah anemia yang terjadi ketika ukuran sel darah merah lebih kecil dari biasanya, dikarenakan tidak memiliki cukup hemoglobin.

- Leukimia

  Leukimia adalah salah satu jenis penyakit kanker yang ditandai dengan kurangnya sel darah merah (hemoglobin), karena produksi sel darah yang tidak normal, mengakibatkan ketidakseimbangan komposisi darah. Karena ketidakseimbangan komposisi darah ini, terkadang Leukimia dianggap sebagai Anemia.

- Thrombocytopenia

  Trombositopenia adalah penyakit dimana jumlah trombosit dalam darah yang terlalu rendah, atau terjadi penurunan jumlah platelet darah hingga dibawah batas minimalnya. Normalnya jumlah platelet berkisar antara 150.000 - 450.000 / mikroliter. Kekurangan jumlah platelet (trombosit) dalam darah dapat terjadi karena penurunan produksi platelet pada sumsum tulang, atau proses hancurnya platelet yang lebih cepat dibandingkan dengan proses produksinya.

- Macrocytic Anemia

  Anemia Makrositik adalah kelainan darah yang terjadi ketika sumsum tulang memproduksi sel darah merah yang berukuran lebih besar dari seharusnya, sehingga menyebabkannya tidak dapat membawa cukup oksigen ke seluruh tubuh.

Dataset yang kami gunakan bernama “Anemia Types Classification” yang dipublikasikan di Kaggle oleh Ehab Aboelnaga. Dataset tersebut berisi 1281 baris data dan terdapat 15 kolom atribut. Dataset tersebut berasal dari beberapa data CBC (Complete Blood Count) yang dikumpulkan untuk dapat memprediksi tipe anemia. Tujuan utama dari penggunaan dataset ini adalah untuk dapat mengembangkan model yang dapat memprediksi tipe anemia berdasarkan dengan data CBC pasien. Rincian atribut yang ada di dalam dataset ini antara lain:

- HGB (Hemoglobin): Jumlah hemoglobin dalam darah, berperan penting untuk membawa oksigen
- PLT (Platelet): Jumlah trombosit dalam darah, berperan penting dalam pembekuan darah
- WBC (White Blood Cell): Jumlah sel darah putih, untuk sistem kekebalan tubuh
- RBC (Red Blood Cell): Jumlah sel darah merah
- MCV (Mean Corpuscular Volume): Rata - rata volume pada sel darah merah
- MCH (Mean Corpuscular Hemoglobin): Rata - rata jumlah hemoglobin per sel darah merah
- MCHC (Mean Corpuscular Hemoglobin Concentration): Rata - rata konsentrasi hemoglobin di dalam sel darah merah
- PDW (Mean Platelet Volume): Pengukuran variasi di dalam distribusi ukuran trombosit dalam darah

- PCT (Platelet Distribution Width): Tes untuk membantu dalam diagnosis penderita sepsis akibat dari infeksi bakteri atau apabila pasien mempunyai risiko tinggi terkena sepsis
- LYMp (Persentase Limfosit): Nilai yang menunjukan limfosit dalam jumlah total sel darah putih
- NEUTp (Persentase Neutrofil): Nilai yang menunjukan jumlah neutrofil per liter dalam darah
- LYMn (Jumlah Limfosit): Nilai yang menunjukan jumlah limfosit per liter dalam darah
- NEUTn (Jumlah Neutrofil): Nilai yang menunjukkan jumlah neutrofil per liter dalam darah
- HCT (Hipertensi): Nilai yang mengukur tekanan darah
- Diagnosis: Tipe anemia berdasarkan dengan parameter CBC

Dengan menganalisis dataset ini, kita dapat menentukan tipe anemia dan dapat mengembangkan model yang akurat dalam memprediksi dan mengklasifikasikan tipe penyakit anemia.

BAB 2. METODOLOGI

Untuk membangun model prediksi tipe-tipe anemia ini, kami melakukan beberapa tahap terhadap dataset blood sample yang telah kami pilih, agar didapat wawasan-wawasan yang dapat digunakan untuk model yang kami bangun.

Link Code: https://colab.research.google.com/finaltaskbdp

Tahap - tahap dalam pengolahan dataset antara lain sebagai berikut:

1. **Data Preparation**

```
%pip install pyspark
%pip install pandas
```

```
Requirement already satisfied: pyspark in /usr/local/lib/python3.10/dist-packages (3.5.1)
Requirement already satisfied: py4j==0.10.9.7 in /usr/local/lib/python3.10/dist-packages (from pyspark) (0.10.9.7)
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (2.0.3)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas) (2023.4)
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-packages (from pandas) (2024.1)
Requirement already satisfied: numpy>=1.21.0 in /usr/local/lib/python3.10/dist-packages (from pandas) (1.25.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)
```

```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
from pyspark.sql import SparkSession
from pyspark.ml.feature import VectorAssembler, StandardScaler
from pyspark.ml.classification import DecisionTreeClassifier
from pyspark.ml.evaluation import MulticlassClassificationEvaluator

spark = SparkSession.builder.getOrCreate()

# Read data
anemiaDataset = spark.read.option("inferSchema", "true").csv("diagnosed_cbc_data_v4.csv", header=True)

# Select kolom column yang relefan
anemiaDataset = anemiaDataset.select("WBC", "LYMp", "NEUTp", "LYMn", "NEUTn", "RBC", "HGB", "HCT", "MCV", "MCH", "MCHC", "PLT", "PDW", "PCT", "Diagnosis")

anemiaDataset.show()
```

```
+----+----+-----+----+-----+----+----+-----+----+----+----+-----+----+----+--------------------+
| WBC|LYMp|NEUTp|LYMn|NEUTn| RBC| HGB|  HCT| MCV| MCH|MCHC|  PLT| PDW| PCT|           Diagnosis|
+----+----+-----+----+-----+----+----+-----+----+----+----+-----+----+----+--------------------+
|10.0|43.2| 50.1| 4.3|  5.0|2.77| 7.3| 24.2|87.7|26.3|30.1|189.0|12.5|0.17|Normocytic hypoch...|
|10.0|42.4| 52.3| 4.2|  5.3|2.84| 7.3| 25.0|88.2|25.7|20.2|180.0|12.5|0.16|Normocytic hypoch...|
| 7.2|30.7| 60.7| 2.2|  4.4|3.97| 9.0| 30.5|77.0|22.6|29.5|148.0|14.3|0.14|Iron deficiency a...|
| 6.0|30.2| 63.5| 1.8|  3.8|4.22| 3.8| 32.8|77.9|23.2|29.8|143.0|11.3|0.12|Iron deficiency a...|
| 4.2|39.1| 53.7| 1.6|  2.3|3.93| 0.4|316.0|80.6|23.9|29.7|236.0|12.8|0.22|Normocytic hypoch...|
| 6.6|27.3| 65.4| 1.8|  4.3|3.96| 8.8| 29.7|75.2|22.2|79.6|207.0|11.5|0.18|Other microcytic ...|
|16.7|19.1| 68.2| 3.2| 11.4|5.15|14.2| 44.8|87.1|27.5|31.6|151.0|12.8|0.14|            Leukemia|
| 9.3|27.4| 64.0| 2.6|  5.9|4.39|12.0| 37.9|86.4|27.3|31.6|194.0|15.9|0.19|Normocytic hypoch...|
| 5.2|19.7| 72.4| 1.0|  3.8|4.85|13.2| 41.0|84.7|27.2|32.1|181.0|10.0|0.15|             Healthy|
|10.5|12.4| 79.0| 1.3|  8.3|4.57|12.4| 38.9|85.3|27.1|31.8|164.0|11.3|0.14|Normocytic hypoch...|
| 5.6|35.6| 56.9| 2.0|  3.2|4.31|11.0| 36.3|84.3|25.5|30.3|171.0|13.1|0.15|Normocytic hypoch...|
| 8.6|41.4| 50.3| 3.6|  4.3|4.28|11.0| 35.0|81.9|25.7|31.4| 88.0|17.2|0.09|Normocytic hypoch...|
| 5.1|31.3| 60.6| 1.6|  3.1|5.63|14.6| 45.6|81.0|25.9|32.0|185.0|11.3|0.16|             Healthy|
| 5.2|23.0| 68.6| 1.2|  3.6|4.62|13.3| 40.9|88.7|28.7|32.5|146.0|14.1|0.13|    Thrombocytopenia|
|12.3|20.7| 71.2| 2.6|  8.7|5.78|16.5| 50.2|86.9|28.5|32.8|190.0|14.1|0.18|            Leukemia|
|10.4| 9.5| 85.1| 1.0|  8.8| 5.3|14.5| 45.2|85.3|27.3|32.0|211.0|15.6|0.21|            Leukemia|
| 4.5|43.4| 51.2| 2.0|  2.3|5.35|13.5| 44.0|82.4|25.7|31.3|202.0|12.3|0.18|             Healthy|
| 3.1|48.5| 45.5| 1.5|  1.4|5.05| 9.6| 34.6|68.6|18.8|27.1|263.0|13.6|0.24|Iron deficiency a...|
| 4.7|21.5| 68.0| 1.0|  3.2| 5.5|14.7| 44.7|81.4|26.7|32.8|139.0|14.9|0.14|    Thrombocytopenia|
|13.8|18.9| 72.5| 2.6| 10.0|4.54| 9.6| 32.4|71.4|21.1|29.6|280.0|16.1|0.29|Iron deficiency a...|
+----+----+-----+----+-----+----+----+-----+----+----+----+-----+----+----+--------------------+
only showing top 20 rows
```

Pada tahap ini kami menyiapkan library - library yang digunakan, antara lain:

- Matplotlib: untuk membuat plot yang digunakan untuk visualisasi data.

- SparkSession dari library pyspark.sql: untuk membaca dataset menggunakan Spark.
- VectorAssembler dari library pyspark.ml.feature: untuk membuat fitur (data) menjadi vektor.
- StandardScaler dari library pyspark.ml.feature: untuk melakukan normalisasi pada data.
- DecisionTreeClassifier dari library pyspark.ml.classification: untuk membangun model yang menggunakan algoritma Decision Tree.
- MulticlassClassificationEvaluator dari library pyspark.ml.evaluation: untuk mengevaluasi tingkat akurasi dari model yang dibangun.
- Seaborn: untuk membuat visualisasi data heatmap.
- Pandas: untuk menganalisis data serta membangun sebuah machine learning model.

2. Data Preprocessing

Pada bagian data preprocessing kami mengubah setiap fitur yang ada pada dataset menjadi bahasa yang mudah untuk dipahami, seperti WBC menjadi whiteBloodCell dan lainnya.

```
anemiaDataset = anemiaDataset.withColumnRenamed("WBC", "whiteBloodCell")
anemiaDataset = anemiaDataset.withColumnRenamed("LYMp", "presentaseLimfosit")
anemiaDataset = anemiaDataset.withColumnRenamed("NEUTp", "presentaseNeutrofil")
anemiaDataset = anemiaDataset.withColumnRenamed("LYMn", "jumlahLimfosit")
anemiaDataset = anemiaDataset.withColumnRenamed("NEUTn", "jumlahNeutrofil")
anemiaDataset = anemiaDataset.withColumnRenamed("RBC", "redBloodCell")
anemiaDataset = anemiaDataset.withColumnRenamed("HGB", "hemoglobin")
anemiaDataset = anemiaDataset.withColumnRenamed("HCT", "hipertensi")
anemiaDataset = anemiaDataset.withColumnRenamed("MCV", "meanCorpuscularVolume")
anemiaDataset = anemiaDataset.withColumnRenamed("MCH", "meanCorpuscularHemoglobin")
anemiaDataset = anemiaDataset.withColumnRenamed("MCHC", "meanCorpuscularHemoglobinConcentration")
anemiaDataset = anemiaDataset.withColumnRenamed("PLT", "platelet")
anemiaDataset = anemiaDataset.withColumnRenamed("PDW", "plateletDistributionWidth")
anemiaDataset = anemiaDataset.withColumnRenamed("PCT", "meanPlateletVolume")
```

```
anemiaDataset.show(10)
```

whiteBloodCell	presentaseLimfosit	presentaseNeutrofil	jumlahLimfosit	jumlahNeutrofil	redBloodCell	hemoglobin	hipertensi	meanCorpuscularVolume	meanCorpuscularHemoglobin	meanCorpuscularHe
10.0	43.2	50.1	4.3	5.0	2.77	7.3	24.2	87.7	26.3	
10.0	42.4	52.3	4.2	5.3	2.84	7.3	25.0	88.2	25.7	
7.2	30.7	60.7	2.2	4.4	3.97	9.0	30.5	77.0	22.6	
6.0	30.2	63.5	1.8	3.8	4.22	3.8	32.8	77.9	23.2	
4.2	39.1	53.7	1.6	2.3	3.93	0.4	316.0	80.6	23.9	
6.6	27.3	65.4	1.8	4.3	3.96	8.8	29.7	75.2	22.2	
16.7	19.1	68.2	3.2	11.4	5.15	14.2	44.8	87.1	27.5	
9.3	27.4	64.0	2.6	5.9	4.39	12.0	37.9	86.4	27.3	
5.2	19.7	72.4	1.0	3.8	4.85	13.2	41.0	84.7	27.2	
10.5	12.4	79.0	1.3	8.3	4.57	12.4	38.9	85.3	27.1	

only showing top 10 rows

```
# Menghilangkan null data
anemiaDataset = anemiaDataset.na.drop()

# Mengganti Column Diagnosis menjadi Integer
from pyspark.sql.functions import when
anemiaDataset = anemiaDataset.withColumn("Diagnosis", when(anemiaDataset["Diagnosis"] == "Healthy", 0)
                                                      .when(anemiaDataset["Diagnosis"] == "Normocytic hypochromic anemia", 1)
                                                      .when(anemiaDataset["Diagnosis"] == "Iron deficiency anemia", 2)
                                                      .when(anemiaDataset["Diagnosis"] == "Other microcytic anemia", 3)
                                                      .when(anemiaDataset["Diagnosis"] == "Leukemia", 4)
                                                      .when(anemiaDataset["Diagnosis"] == "Thrombocytopenia", 5)
                                                      .when(anemiaDataset["Diagnosis"] == "Normocytic normochromic anemia", 6)
                                                      .when(anemiaDataset["Diagnosis"] == "Leukemia with thrombocytopenia", 7)
                                                      .otherwise(8))
cols = list(anemiaDataset.columns)
cols.remove("Diagnosis")
assembler = VectorAssembler(inputCols=cols, outputCol="features")
anemiaDataset = assembler.transform(anemiaDataset)

# Split data (70% for training, 30% for testing)
(trainingData, testingData) = anemiaDataset.randomSplit([0.7, 0.3])

trainingData.show(5)
testingData.show(5)
```

- Menghilangkan nilai null pada data. Nilai null merupakan data yang tidak tersedia di dalam dataset
- Melakukan konversi data pada kolom Diagnosis menjadi numerik
- Melakukan split data menjadi dua subset yaitu menjadi 70% training data dan 30% testing data

```
+---+------+------+-------+-------+----+----+-------+----+----+----+-----+-----------+-------+---------+--------------------+
|WBC|  LYMp| NEUTp|   LYMn|  NEUTn| RBC| HGB|    HCT| MCV| MCH|MCHC|  PLT|        PDW|    PCT|Diagnosis|            features|
+---+------+------+-------+-------+----+----+-------+----+----+----+-----+-----------+-------+---------+--------------------+
|0.8|  45.3|  48.4|    1.7|    1.9|3.99| 7.3|   28.1|70.5|18.2|25.9|146.0|       12.8|   0.15|        2|[0.8,45.3,48.4,1....|
|2.0|25.845|77.511|1.88076|5.14094|3.18| 9.7|46.1526|94.0|30.5|32.4| 84.0|14.31251157|0.26028|        6|[2.0,25.845,77.51...|
|2.0|  30.7|  64.4|    0.6|    1.3|4.62| 9.3|   31.8|68.9|20.1|29.2|134.0|       13.3|   0.13|        2|[2.0,30.7,64.4,0....|
|2.0|  33.8|  61.4|    0.7|    1.2|4.51| 9.3|   31.0|68.8|20.6|30.0|135.0|       13.3|   0.13|        2|[2.0,33.8,61.4,0....|
|2.4|  24.8|  68.0|    0.6|    1.6|4.66|13.8|   42.0|90.2|29.2|32.8|112.0|       18.5|   0.12|        5|[2.4,24.8,68.0,0....|
+---+------+------+-------+-------+----+----+-------+----+----+----+-----+-----------+-------+---------+--------------------+
only showing top 5 rows

+---+------+------+-------+-------+----+----+-------+----+----+----+-----+-----------+-------+---------+--------------------+
|WBC|  LYMp| NEUTp|   LYMn|  NEUTn| RBC| HGB|    HCT| MCV| MCH|MCHC|  PLT|        PDW|    PCT|Diagnosis|            features|
+---+------+------+-------+-------+----+----+-------+----+----+----+-----+-----------+-------+---------+--------------------+
|2.4|25.845|77.511|1.88076|5.14094|4.52|14.7|46.1526|92.4|30.9|33.9|149.0|14.31251157|0.26028|        5|[2.4,25.845,77.51...|
|2.6|  21.4|  71.4|    0.6|    1.8|4.77| 9.0|   30.8|64.6|18.8|29.2|111.0|       13.1|   0.12|        2|[2.6,21.4,71.4,0....|
|2.6|25.845|77.511|1.88076|5.14094|5.84|14.9|46.1526|75.0|25.5|25.5|160.0|14.31251157|0.26028|        0|[2.6,25.845,77.51...|
|2.7|  43.4|   1.2|    0.2|   4.77|13.1|41.0|    2.0|86.4|27.4|31.7|169.0|       14.3|   0.17|        1|[2.7,43.4,1.2,0.2...|
|2.7|  43.4|  49.5|    1.2|    1.3|4.77|13.1|   41.2|74.1|27.4|31.7|169.0|       14.3|   0.17|        0|[2.7,43.4,49.5,1....|
+---+------+------+-------+-------+----+----+-------+----+----+----+-----+-----------+-------+---------+--------------------+
only showing top 5 rows
```

Berikut merupakan data training dan testing yang diurutkan serta melakukan standard scaler pada data training dan testing menggunakan StandardScaler sehingga fitur pada data training dan juga testing mempunyai skala yang sama.

```
# Standard Scaler
scaler = StandardScaler(inputCol="features", outputCol="scaledFeatures")
scaler_model = scaler.fit(trainingData)
trainingData = scaler_model.transform(trainingData)
testingData = scaler_model.transform(testingData)
```

3. Data Visualization

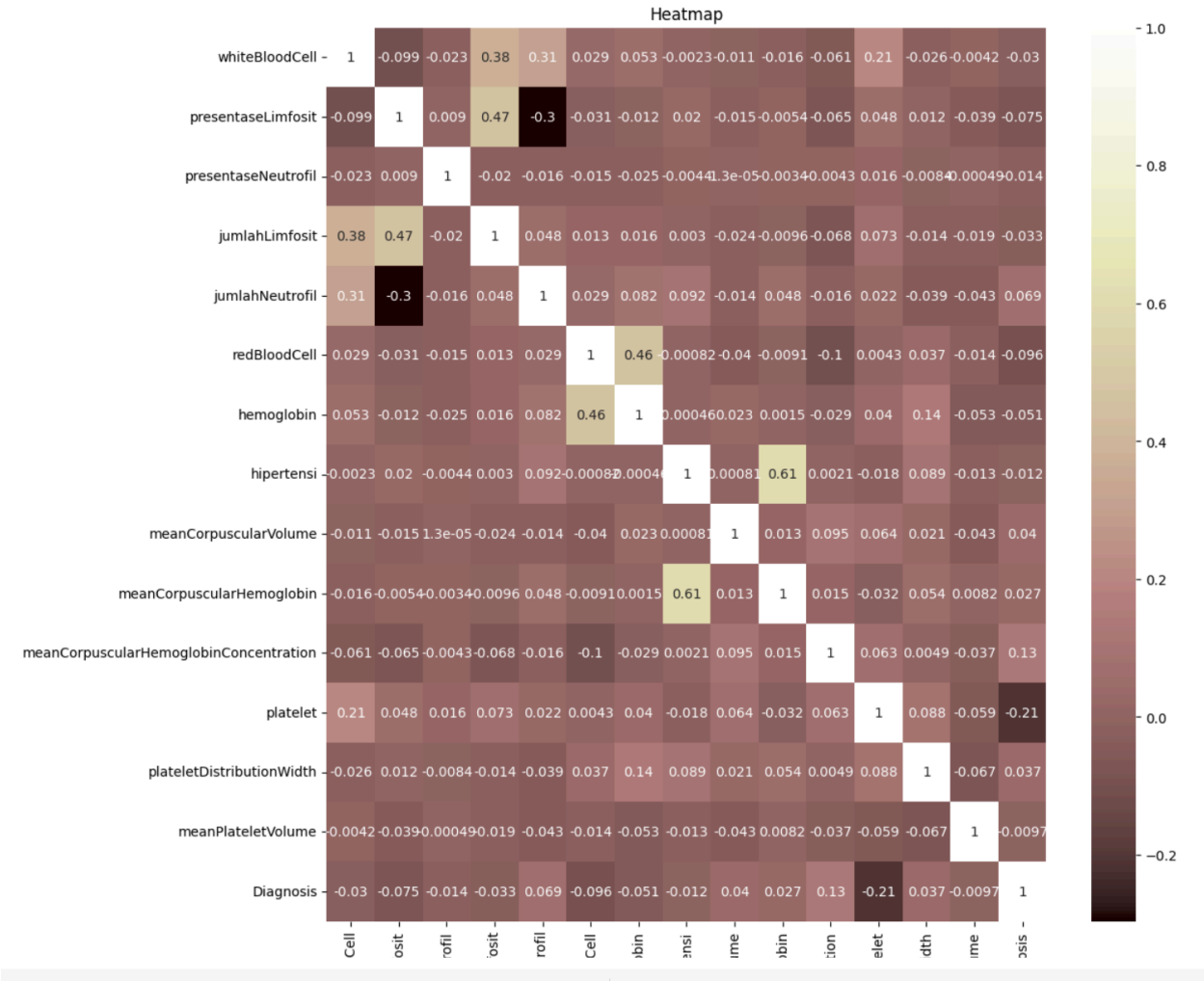
- Memilih fitur terkait dengan darah seperti whiteBloodCell, persentaseLimfosit dan lainnya
- Mengubah format data menjadi Pandas DataFrame agar mempermudah melakukan analisis
- Menghitung korelasi antar fitur yang ada
- Melakukan perubahan kode pada diagnosis menjadi tipe jenis anemia
- Melakukan perhitungan jumlah data pada setiap jenis tipe anemia
- Menggunakan plot seperti diagram pie untuk menampilkan visualisasi persentase dari setiap jenis anemia
- Menggunakan heatmap untuk mengetahui hubungan atau relasi antar fitur yang berada di dalam data
- Menggunakan barplot untuk mengetahui fitur mana yang memiliki peran penting dalam memprediksi anemia

```
[ ]  anemiaDataset = anemiaDataset.select("whiteBloodCell", "presentaseLimfosit", "presentaseNeutrofil", "jumlahLimfosit",
                                          "jumlahNeutrofil", "redBloodCell", "hemoglobin", "hipertensi", "meanCorpuscularVolume",
                                          "meanCorpuscularHemoglobin", "meanCorpuscularHemoglobinConcentration", "platelet",
                                          "plateletDistributionWidth", "meanPlateletVolume", "Diagnosis")
     anemiaDataset_pd = anemiaDataset.toPandas()

     correlation_matrix = anemiaDataset_pd.corr()
```

a. Heatmap Relasi Antar Fitur

```
[ ]  plt.figure(figsize=(12, 12))
     sns.heatmap(correlation_matrix, annot=True, cmap='pink')
     plt.title("Heatmap")
     plt.show()
```



b. Jenis Distribusi Penyakit Anemia

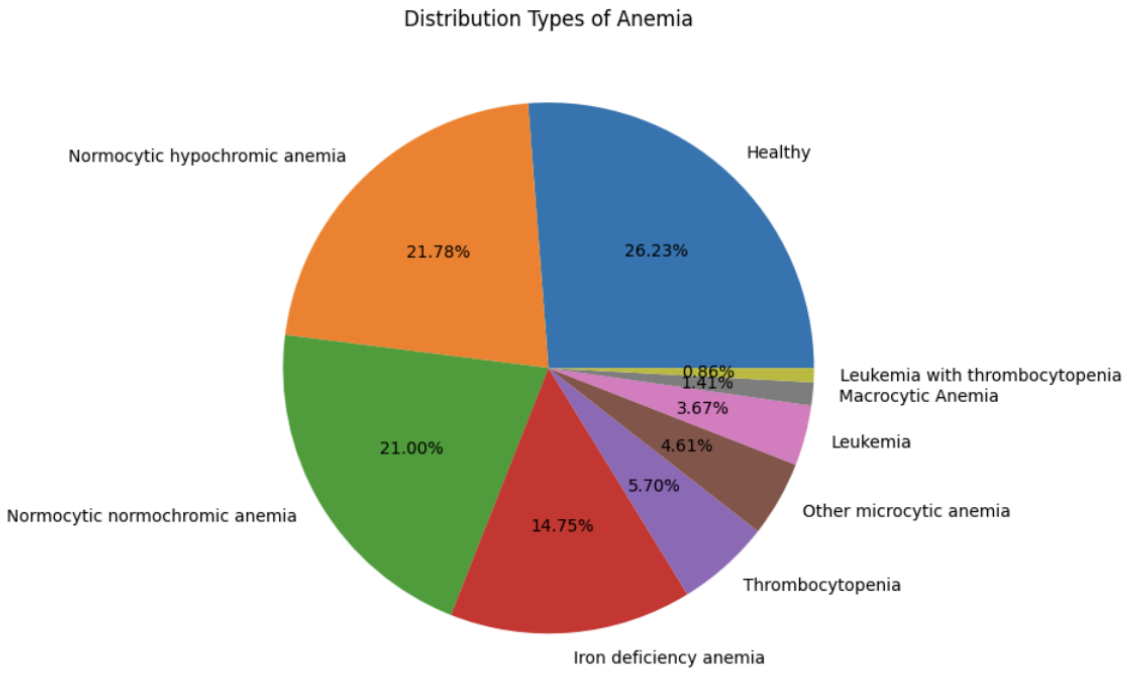
Kami menggunakan plot pie untuk memvisualisasikan distribusi dari setiap jenis penyakit anemia.

```
[ ] anemia_type_mapping = {
        0: 'Healthy',
        1: 'Normocytic hypochromic anemia',
        2: 'Iron deficiency anemia',
        3: 'Other microcytic anemia',
        4: 'Leukemia',
        5: 'Thrombocytopenia',
        6: 'Normocytic normochromic anemia',
        7: 'Leukemia with thrombocytopenia',
        8: 'Macrocytic Anemia',
    }

    anemiaDataset_pd['Diagnosis_mapped'] = anemiaDataset_pd['Diagnosis'].replace(anemia_type_mapping)
    anemia_counts = anemiaDataset_pd['Diagnosis_mapped'].value_counts()
    print(anemia_counts)

    plt.figure(figsize=(10, 6))
    plt.pie(anemia_counts.values, labels=anemia_counts.index, autopct="%.2f%%")
    plt.title('Distribution Types of Anemia')
    plt.tight_layout()
    plt.show()
```

```
Diagnosis_mapped
Healthy                          336
Normocytic hypochromic anemia    279
Normocytic normochromic anemia   269
Iron deficiency anemia           189
Thrombocytopenia                  73
Other microcytic anemia           59
Leukemia                          47
Macrocytic Anemia                 18
Leukemia with thrombocytopenia    11
Name: count, dtype: int64
```



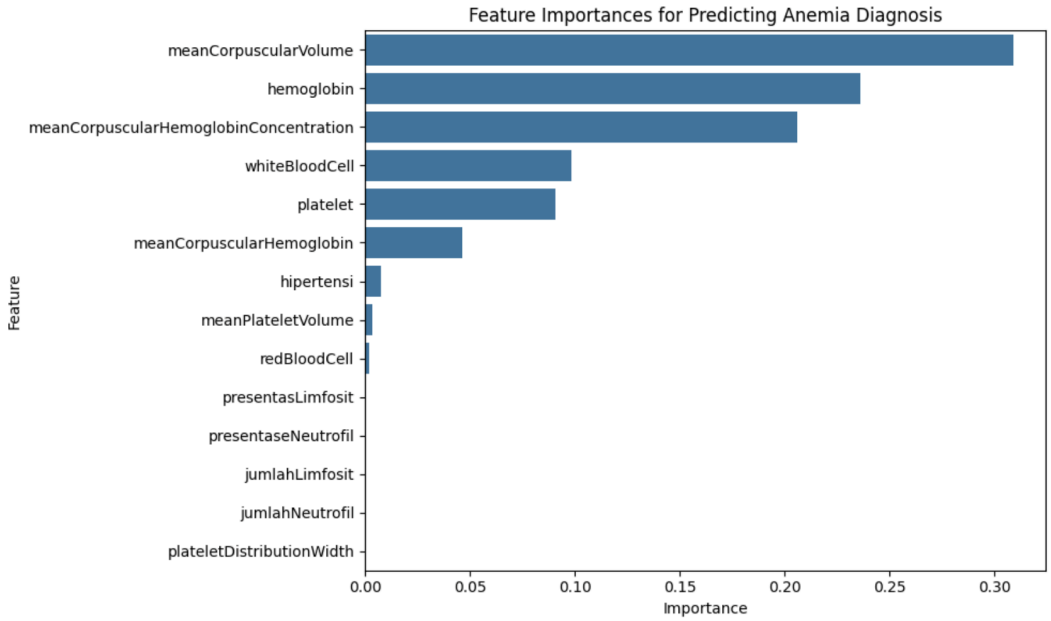
c. Pentingnya Fitur Untuk Memprediksi Diagnosis Anemia Menggunakan Barplot

```
importances = model.featureImportances
FeatureSelect = ["whiteBloodCell", "presentasLimfosit", "presentaseNeutrofil", "jumlahLimfosit",
                                   "jumlahNeutrofil", "redBloodCell", "hemoglobin", "hipertensi", "meanCorpuscularVolume",
                                   "meanCorpuscularHemoglobin", "meanCorpuscularHemoglobinConcentration", "platelet",
                                   "plateletDistributionWidth", "meanPlateletVolume"]
feature_importances = pd.DataFrame(importances.toArray(), index=FeatureSelect, columns=["Importance"]).sort_values(by="Importance", ascending=False)

feature_importance_dict = dict(zip(FeatureSelect, importances))
sorted_feature = sorted(feature_importance_dict.items(), key=lambda x: x[1], reverse=True)
for feature, importance in sorted_feature:
  print(f"{feature}: {round(importance * 100, 2)}%")

plt.figure(figsize=(10, 6))
sns.barplot(x=feature_importances.Importance, y=feature_importances.index)
plt.title('Feature Importances for Predicting Anemia Diagnosis')
plt.xlabel('Importance')
plt.ylabel('Feature')
plt.tight_layout()
plt.show()
```

```
meanCorpuscularVolume: 30.89%
hemoglobin: 23.64%
meanCorpuscularHemoglobinConcentration: 20.62%
whiteBloodCell: 9.82%
platelet: 9.08%
meanCorpuscularHemoglobin: 4.62%
hipertensi: 0.78%
meanPlateletVolume: 0.36%
redBloodCell: 0.2%
presentasLimfosit: 0.0%
presentaseNeutrofil: 0.0%
jumlahLimfosit: 0.0%
jumlahNeutrofil: 0.0%
plateletDistributionWidth: 0.0%
```



4. Model Building and Evaluation

```
# Bikin classifier Decision Tree
dt = DecisionTreeClassifier(featuresCol="scaledFeatures", labelCol="Diagnosis")

# Training model
model = dt.fit(trainingData)

# Membuat prediksi dengan testingData
predictions = model.transform(testingData)
predictions.show()

# Evaluasi model Decision Tree
evaluator = MulticlassClassificationEvaluator(labelCol="Diagnosis", predictionCol="prediction", metricName="accuracy")
accuracy = evaluator.evaluate(predictions)
print("Decision Tree Accuracy: {:.2f}%".format(accuracy * 100))
```

whiteBloodCell	presentaseLimfosit	presentaseNeutrofil	jumlahLimfosit	jumlahNeutrofil	redBloodCell	hemoglobin	hipertensi	meanCorpuscularVolume	meanCorpuscularHemoglobin	meanCorpuscularHemoglobinConcentration	plate	
0.8	45.3	48.4	1.7	1.9	3.99	7.3	28.1	70.5	18.2	25.9	14	
2.5	36.8	55.3	0.9	1.4	3.55	8.6	28.5	80.3	24.2	30.1	1	
2.6	25.845	77.511	1.88076	5.14094	4.62	14.7	46.1526	93.4	31.9	34.1	15	
2.6	25.845	77.511	1.88076	5.14094	5.84	14.9	46.1526	75.0	25.5	25.5	16	
2.7	43.4	1.2	0.2	4.77	13.1	41.0	2.0	86.4	27.4	31.7	16	
2.7	43.4	49.5	1.2	1.3	4.77	13.1	41.2	86.4	27.4	31.1	16	
3.03	25.845	77.511	1.88076	5.14094	3.85	11.1	46.1526	95.6	28.8	30.2	6	
3.1	48.5	45.5	1.5	1.4	5.05	9.6	34.6	68.6	18.8	27.1	26	
3.2	35.6	56.2	1.1	1.8	5.22	14.9	45.8	87.8	28.5	32.5	13	
3.2	35.6	56.2	1.1	1.8	5.22	14.9	45.8	87.8	28.5	32.5	13	
3.4	23.9	70.6	0.4	2.4	4.95	11.3	38.0	76.9	22.8	29.7	19	
3.5	31.0	56.3	1.0	2.1	5.07	14.7	43.7	86.3	28.9	33.6	11	
3.5	31.0	59.4	1.1	2.1	5.07	14.7	43.7	86.3	28.9	33.6	11	
3.6	41.0	52.0	1.5	1.8	4.64	11.6	37.4	80.8	25.0	31.0	15	
3.7	21.9	65.9	0.8	2.4	3.95	9.4	32.5	82.5	23.7	28.9	16	
3.7	25.845	77.511	1.88076	5.14094	3.5	12.0	46.1526	102.3	34.3	33.5	25	
3.8	30.0	63.2	1.1	2.4	4.48	11.8	36.9	82.4	26.3	31.9	18	
3.8	36.6	54.7	1.4	2.1	4.34	10.9	35.2	81.2	25.1	30.9	11	
3.8	36.6	54.7	1.4	2.1	10.9	35.2	81.2	25.1	10.9	30.9	11	
3.92	25.845	77.511	1.88076	5.14094	4.97	13.9	46.1526	88.1	28.0	31.7	22	

only showing top 20 rows

Decision Tree Accuracy: 96.07%

- Kami menggunakan model Decision Tree untuk mempelajari relasi antara fitur-fitur (kolom) yang tersedia dalam dataset.
- Kemudian akan kami latih model tersebut menggunakan dataset training.
- Setelah di training, model kami melakukan prediksi anemia menggunakan dataset testing.
- Kami gunakan MulticlassClassificationEvaluator untuk mengevaluasi hasil akurasi model tersebut.
- Setelah testing dan evaluasi, model kami mendapat skor akurasi 96.07%

BAB 3. EVALUASI DAN DETAIL ALUR KERJA

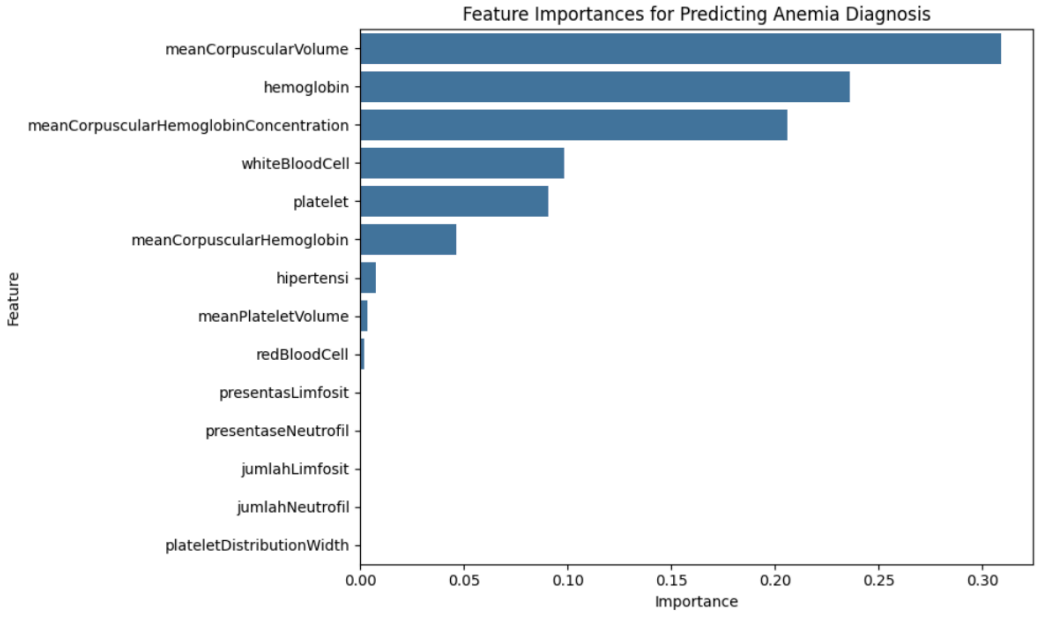
Dari model yang kami bangun menggunakan algoritma Decision Tree, kami mendapati hasil akurasi 96.07%. Dengan itu, kita bisa memprediksi tipe penyakit anemia berdasarkan beberapa fitur, yakni: Hemoglobin, Platelet, White Blood Cell, Red Blood Cell, Mean Corpuscular Volume, Mean Corpuscular Hemoglobin, Mean Corpuscular Hemoglobin Concentration, Platelet Distribution Width, Procalcitonin, Limfosit, Hipertensi, dan Neutrofil.

```
importances = model.featureImportances
FeatureSelect = ["whiteBloodCell", "presentasLimfosit", "presentaseNeutrofil", "jumlahLimfosit",
                                   "jumlahNeutrofil", "redBloodCell", "hemoglobin", "hipertensi", "meanCorpuscularVolume",
                                   "meanCorpuscularHemoglobin", "meanCorpuscularHemoglobinConcentration", "platelet",
                                   "plateletDistributionWidth", "meanPlateletVolume"]
feature_importances = pd.DataFrame(importances.toArray(), index=FeatureSelect, columns=["Importance"]).sort_values(by="Importance", ascending=False)

feature_importance_dict = dict(zip(FeatureSelect, importances))
sorted_feature = sorted(feature_importance_dict.items(), key=lambda x: x[1], reverse=True)
for feature, importance in sorted_feature:
  print(f"{feature}: {round(importance * 100, 2)}%")

plt.figure(figsize=(10, 6))
sns.barplot(x=feature_importances.Importance, y=feature_importances.index)
plt.title('Feature Importances for Predicting Anemia Diagnosis')
plt.xlabel('Importance')
plt.ylabel('Feature')
plt.tight_layout()
plt.show()
```

```
meanCorpuscularVolume: 30.89%
hemoglobin: 23.64%
meanCorpuscularHemoglobinConcentration: 20.62%
whiteBloodCell: 9.82%
platelet: 9.08%
meanCorpuscularHemoglobin: 4.62%
hipertensi: 0.78%
meanPlateletVolume: 0.36%
redBloodCell: 0.2%
presentasLimfosit: 0.0%
presentaseNeutrofil: 0.0%
jumlahLimfosit: 0.0%
jumlahNeutrofil: 0.0%
plateletDistributionWidth: 0.0%
```

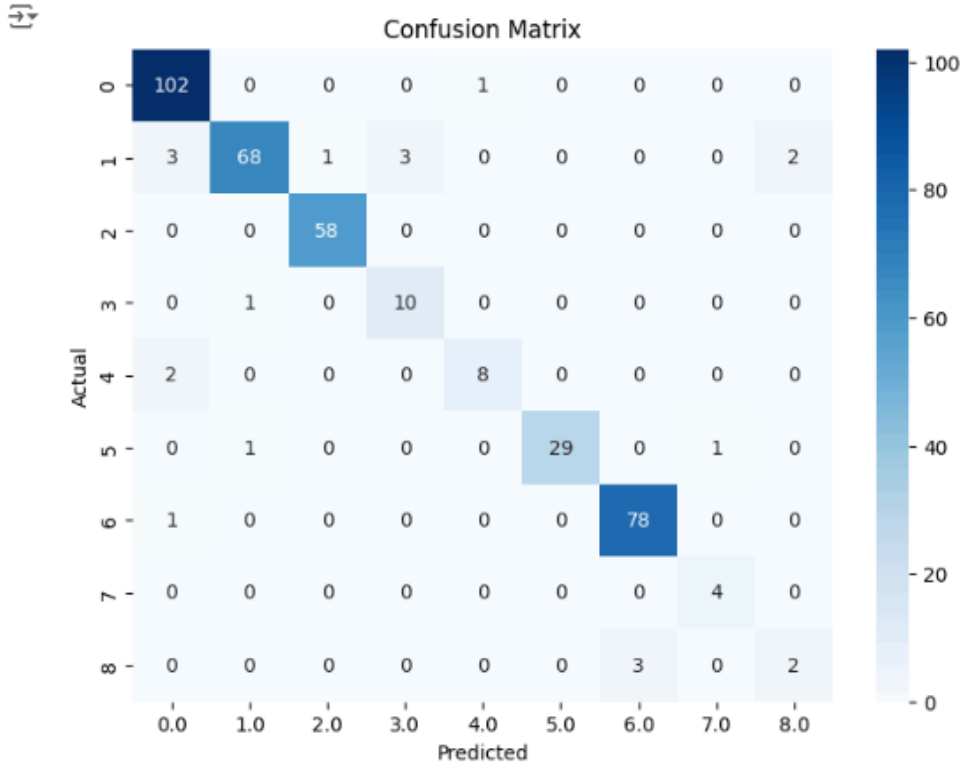


Berdasarkan hasil chart level of importance yang telah dibuat, terbukti bahwa fitur yang paling berpengaruh terhadap terhadap hasil akurasi prediksi tipe anemia adalah “Mean corpuscular volume” yakni sebesar 30.89%. Diikuti oleh “hemoglobin” sebesar 23.64%, “mean corpuscular hemoglobin concentration” sebesar 20.62%, “white blood cell” sebesar 9.82%, “platelet” sebesar 9.06%, “mean corpuscular hemoglobin” sebesar 4.62%, “hipertensi” sebesar 0.78%, “mean platelet volume” sebesar 0.36%, “red blood cell” sebesar 0,2% dan 6 fitur lainnya tidak terlalu berpengaruh dalam memprediksi tipe penyakit Anemia.

Untuk membantu kami memastikan kebenaran akurasi model, kami menggunakan visualisasi menggunakan Confusion Matrix untuk mengevaluasi kinerja model kami, dan memberikan gambaran menyeluruh tentang hasil prediksi yang diberikan model.

```
conf_matrix = predictions.groupBy("Diagnosis").pivot("prediction").count().fillna(0).orderBy("Diagnosis")
conf_matrix_pd = conf_matrix.toPandas()

plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix_pd.iloc[:, 1:], annot=True, cmap="Blues", fmt='g')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()
```



Confusion matrix kami berukuran 9x9, karena terdapat 9 klasifikasi tipe penyakit anemia yang berbeda berdasarkan dataset yang kami gunakan (multi class).

```
# Mengganti Column Diagnosis menjadi Integer
from pyspark.sql.functions import when
anemiaDataset = anemiaDataset.withColumn("Diagnosis", when(anemiaDataset["Diagnosis"] == "Healthy", 0)
                                        .when(anemiaDataset["Diagnosis"] == "Normocytic hypochromic anemia", 1)
                                        .when(anemiaDataset["Diagnosis"] == "Iron deficiency anemia", 2)
                                        .when(anemiaDataset["Diagnosis"] == "Other microcytic anemia", 3)
                                        .when(anemiaDataset["Diagnosis"] == "Leukemia", 4)
                                        .when(anemiaDataset["Diagnosis"] == "Thrombocytopenia", 5)
                                        .when(anemiaDataset["Diagnosis"] == "Normocytic normochromic anemia", 6)
                                        .when(anemiaDataset["Diagnosis"] == "Leukemia with thrombocytopenia", 7)
                                        .otherwise(8))
```

Setiap kategori “Aktual” pada Confusion matrix mewakili setiap pembagian kelas dataset yang disetting saat Transform Data.

Contoh cara baca:

Misal untuk baris pertama (kategori Healthy), disini model kami berhasil memprediksi 102 kondisi healthy yang benar, namun terjadi 1 kesalahan prediksi pada kolom Leukimia, yang seharusnya memprediksinya tetap pada kolom Healthy.

Misal untuk baris kedua (kategory Normocytic hypochromic anemia), disini model kami berhasil memprediksi 68 kondisi Normocytic hypochromic anemia dengan benar, namun terjadi 3 kesalahan pada kolom Healthy, 1 kesalahan pada kolom Iron deficiency anemia, 3 kesalahan pada Other microcytic anemia, dan 2 kesalahan pada kolom otherwise, yang mana seharusnya model kami memprediksi tetap pada kolom Normocytic hypochromic anemia.

Akurasi berdasarkan Confusion Matrix dihitung dengan:

Akurasi = **Σ** True Positive / **Σ** Total Instances

= (102 + 68 + 58 + 10 + 8 + 29 + 78 + 4 + 2) / (Total semua)

= 353 / 376

= 0.9388

= 93.88 %

Berdasarkan hasil perhitungan tersebut, didapat nilai akurasi berdasarkan confusion matrix sebesar 93.88%. Hasil ini masih dapat ditingkatkan lagi dengan melakukan pemangkasan data dengan lebih baik, karena algoritma Decision Tree, terkadang mengalami overfitting, yang

membuatnya hanya bekerja sesuai dengan data pelatihan, tidak dengan data yang baru. Selain itu karena data medis cukup banyak dan kompleks, sebetulnya membutuhkan teknik preprocessing yang lebih baik dan kompleks dibandingkan Decision Tree untuk menangkap pola yang ada dalam data.

BAB 4. KESIMPULAN

Penyakit Anemia dapat disebabkan oleh banyak hal, dimana jika tidak segera diatasi dengan dapat menyebabkan komplikasi penyakit yang lebih berbahaya. Penyakit anemia sangat dipengaruhi oleh kondisi darah penderita, dimana dengan menyediakan beberapa informasi seperti Hemoglobin, Platelet, White Blood Cell, Red Blood Cell, Mean Corpuscular Volume, Mean Corpuscular Hemoglobin, Mean Corpuscular Hemoglobin Concentration, Platelet Distribution Width, Procalcitonin, Limfosit, dan Neutrofil, kita dapat memprediksi tipe penyakit anemia yang diidap pasien, berdasarkan model prediksi yang dibuat menggunakan algoritma Decision Tree. “Mean corpuscular volume”, “Hemoglobin”, dan “Mean Corpuscular Hemoglobin Concentration” adalah informasi yang sangat berperan dan dibutuhkan untuk memprediksi tipe penyakit anemia. Fitur lainnya, seperti “White Blood Cell”, “Platelet”, “Mean Corpuscular Hemoglobin”, “Hipertensi”, “Mean Platelet Volume”, “Red Blood Cell”, “Limfosit”, “Neutrofil”, “Platelet Distribution Width” menjadi faktor pendukung yang dapat membantu pendeteksian adanya penyakit anemia. Namun, hasil ini masih harus ditingkatkan lebih tinggi lagi, karena hasil yang diberikan COnfusion Matrix masih perlu ditingkatkan lagi, yakni 93.88%, dengan mengubahnya dengan algoritma yang lebih sesuai dengan dataset yang digunakan. Oleh karena itu, untuk menjaga agar kita terhindar dari penyakit Anemia yang bukan bawaan dari lahir, kita harus menjaga pola hidup kita, seperti mengonsumsi cukup air setiap harinya, mengonsumsi makanan ber zat besi tinggi, bergizi, dan lain - lain.

DAFTAR PUSTAKA

Dataset: Anemia Types Classification - Kaggle: https://www.kaggle.com/datasets/ehababoelnaga/anemia-types-classification/data

NCBI. 2023. Normochromic Normocytic Anemia. URL: https://www.ncbi.nlm.nih.gov/books/NBK565880/. Diakses Pada 10 Juni 2024.

Wikipedia. 2023. Hypochromic Anemia. URL: https://en.wikipedia.org/wiki/Hypochromic_anemia. Diakses Pada 10 Juni 2024.

Nareza, M, T. 2023. Anemia Defisiensi Besi. URL: https://www.alodokter.com/anemia-defisiensi-besi. Diakses Pada 10 Juni 2024.

Cleveland Clinic. 2022. Microcytic Anemia. URL: https://my.clevelandclinic.org/health/diseases/23015-microcytic-anemia. Diakses Pada 10 Juni 2024.

Patient Power. 2023. Anemia and Leukemia: How Are They Linked?. URL: https://www.patientpower.info/leukemia/anemia-and-leukemia. Diakses Pada 10 Juni 2024.

Alodokter. 2022. Trombositopenia. URL: https://www.alodokter.com/trombositopenia. Diakses Pada 10 Juni 2024.

IHME. 2023. The Lancet: New Study Reveals Global Anemia Cases Remain Persistently High Among Women and Children. Anemia Rates Decline for Men. https://www.healthdata.org/news-events/newsroom/news-releases/lancet-new-study-reveals-global-anemia-cases-remain-persistently. Diakses Pada 12 Juni 2024.