Prediksi Akurasi Keganasan Kanker Prostat dengan Klasifikasi Machine Learning

Reyhan Eldwin Maulana[1)], Moch. Rikza Lucky Ardiansyah[2)], Angela Aivel Alan Putri[3)], Airlangga Dwi Satrio[4)]

Abstrak

Kanker prostat adalah salah satu kanker paling umum di antara pria di seluruh dunia, dimana kanker prostat sendiri diidentifikasi menjadi 2, yaitu kanker prostat ganas (maligna) dan kanker prostat jinak (benigna) yang keduanya berpotensi menyebabkan kematian. Oleh karena itu, penting untuk melakukan penelitian pada data kanker prostat dengan tujuan untuk mengklasifikasikan, memprediksi, dan mendiagnosis ganas tidaknya kanker prostat. Dalam penelitian ini, dengan bantuan *machine learning* kami menggunakan dua jenis pengolahan data dengan tiga algoritma pembelajaran mesin yang populer untuk mengklasifikasikan dataset *Cancer Prostate*, yaitu *Naïve Bayes Classifier*, *Random Forests*, *Decision Trees,* dan *Support Vector Machines (SVM)* guna mencari akurasi tertinggi dari ketiga algoritma yang ada. Setelah mendapatkan hasil akurasi klasifikasi dari kedua pengolahan data, kami mendapatkan hasil bahwa akurasi perhitungan paling besar dari pengolahan data adalah menggunakan pengolahan data tanpa transformasi dan seleksi fitur berupa *Random Forests* sebesar 95%.

Kata Kunci: Machine Learning, Naive Bayes, Decision Tree, Random Forest, support vector machines (SVM), Cancer, Prostate Cancer

Abstract

Prostate cancer is one of the most common cancers among men worldwide, with two identified types: malignant and benign, both of which can potentially cause death. Therefore, it is important to conduct research on prostate cancer data with the aim of classifying, predicting, and diagnosing whether prostate cancer is malignant or benign. In this study, with the help of machine learning, we used two types of data processing with three popular machine learning algorithms to classify the Prostate Cancer dataset, namely Naïve Bayes Classifier, Random Forests, Decision Trees, and Support Vector Machines (SVM) to find the highest accuracy of the three existing algorithms. After obtaining the classification accuracy results from both data processing methods, we found that the highest calculation accuracy was obtained from data processing without feature transformation and selection, using Random Forests, with an accuracy of 95%.

Keywords: Machine Learning, Naive Bayes, Decision Tree, Random Forest, Cancer, Prostate Cancer

I. INTRODUCTION

Kanker adalah penyakit kronis dan tidak menular yang masih menjadi masalah kesehatan masyarakat global yang signifikan. Kematian akibat kanker diproyeksikan meningkat menjadi 11 juta kematian setiap tahun pada tahun 2030, dengan sebagian besar terjadi di wilayah dunia dengan kapasitas paling sedikit untuk merespons[1]. Kanker dianggap sebagai salah satu penyakit paling berbahaya di dunia karena bertanggung jawab atas sekitar 13% dari semua kematian di dunia[2]. Indonesia sendiri mencatat kanker sebagai penyebab kematian nomor tujuh di Indonesia dan menjadi penyebab kematian nomor dua di dunia. Angka penderita kanker selalu meningkat setiap tahun, bahkan di tahun 2012 sebanyak 8,2 juta kematian penyebabnya adalah kanker[3]. Ada banyak sekali jenis kanker di dunia ini, salah satunya adalah kanker prostat. Kanker prostat adalah salah satu kanker paling umum di antara pria di seluruh dunia. Menurut *American Cancer Society*, kira-kira satu dari delapan pria akan didiagnosis menderita kanker prostat selama hidup mereka[4]. Hal ini menjadi lebih umum di seluruh dunia, dengan perkiraan populasi lebih dari 1,4 juta kasus baru dan lebih dari 370.000 kematian pada tahun 2020[5]. Penyebab kanker prostat belum dipahami dengan baik, namun faktor genetik dan lingkungan telah diidentifikasi sebagai faktor risiko. Penelitian terus berfokus untuk mengidentifikasi pengobatan baru dan lebih

baik untuk kanker prostat[4]. Kanker prostat sendiri diidentifikasi menjadi 2, yaitu kanker prostat ganas (maligna) dan kanker prostat jinak (benigna).

Hampir 20 tahun memasuki abad ke-21, dunia kita telah dibentuk ulang secara dramatis oleh komputasi modern, dan seluruh industri kini didirikan berdasarkan penerapan kecerdasan buatan (AI)[6]. Pembelajaran mesin (ML) adalah sub bidang AI yang melibatkan pengembangan dan penerapan algoritme dinamis untuk menganalisis data dan memfasilitasi identifikasi pola yang rumit[6]. Pembelajaran mesin diharapkan dapat meningkatkan bidang perawatan kesehatan, terutama dalam spesialisasi medis, seperti radiologi diagnostik, kardiologi, oftalmologi, dan patologi[7]. Penelitian ini sendiri menggunakan tiga algoritma pembelajaran mesin yang populer untuk mengklasifikasikan dataset kanker prostat. Tiga algoritma yang digunakan adalah *Naïve Bayes Classifier*, *Random Forests*, *Decision Trees*, dan *Support Vector Machines* (SVM) . Penulis melakukan penelitian pada data kanker prostat menggunakan bantuan tiga algoritma pembelajaran mesin populer dengan tujuan untuk mengklasifikasikan, memprediksi, dan mendiagnosis ganas tidaknya kanker prostat.

II. Literature Review

Kanker prostat adalah jenis kanker yang mempengaruhi kelenjar prostat pada pria. Hal ini dapat menyebabkan berbagai gejala yang tidak nyaman dan bahkan dapat menyebabkan kematian. Oleh karena itu, penting untuk melakukan penelitian pada data kanker prostat dengan tujuan untuk mengklasifikasikan, memprediksi, dan mendiagnosis ganas tidaknya kanker prostat. Dalam penelitian ini, penulis menggunakan tiga metode yaitu *decision tree*, *naive bayes*, *random forest*, dan *support vector machines*.

Penelitian sebelumnya telah dilakukan untuk mendiagnosis kanker prostat menggunakan berbagai metode pembelajaran mesin. Salah satu penelitian yang menarik adalah penelitian oleh Wang et al.[11] yang menggunakan deep learning untuk mengklasifikasikan kanker prostat. Penelitian ini menunjukkan bahwa deep learning dapat digunakan untuk menghasilkan hasil yang lebih akurat dalam mendiagnosis kanker prostat.

Penelitian lain yang menarik adalah penelitian oleh Zhang et al.[12] yang menggunakan ensemble learning untuk memprediksi risiko metastasis kanker prostat. Hasil penelitian ini menunjukkan bahwa ensemble learning dapat meningkatkan akurasi prediksi risiko metastasis kanker prostat.

Dalam penelitian ini, penulis menghasilkan hasil yang menarik dengan menggunakan *decision tree*, *naive bayes*, *random forest*, dan *support vector machines* untuk mengklasifikasikan, memprediksi, dan mendiagnosis ganas tidaknya kanker prostat. Hasil ini dapat digunakan untuk membantu dokter dalam mendiagnosis kanker prostat dengan lebih akurat dan efisien.

III. Methods

Dalam penelitian ini, kami menggunakan tiga algoritma pembelajaran mesin yang populer untuk mengklasifikasikan dataset *Cancer Prostate*. Tiga algoritma yang digunakan adalah *Naïve Bayes Classifier*, *Random Forests*, *Decision Trees*, dan *Support Vector Machines (SVM)*. Dataset berisi sekitar seratus (100) pasien, setiap instance terdiri dari sepuluh (10) atribut termasuk label kelas yang menunjukkan bahwa hasil diagnosis instance adalah ganas atau jinak. Kedua istilah ini menggambarkan tingkat kanker. Jika daerah kanker terlokalisir, memiliki batas yang jelas, dan tidak menyebar maka disebut jinak. Jika daerah kanker bersifat umum, memiliki bentuk dan batas yang tidak normal, dan menyebar dengan cepat maka disebut ganas. Atribut dari dataset ini antara lain id (kode unik pasien), diagnosis_result (hasil diagnosis), radius, texture, preimeter, area, smoothness, compactness, symmetry, dimensi_fraktal.

```
# Mengimport data
data = pd.read_excel("Prostate_Cancer.xlsx")
df = pd.DataFrame(data)
# Data preview
print("Data Preview : \n", df)
# Mengecek missing value
print("Mengecek missing value : \n", df.isnull().sum())
# Menghilangkan kolom id dan mengganti diagnosis result menjadi boolean
df = df.drop(['id'], axis=1)
df['diagnosis_result'].replace({'M':1,'B':0},inplace=True)
```

Fig1. Syntax Importing Dataset

Atribut id tidak digunakan atau tidak masuk perhitungan karena menunjukkan kode pasien, lalu mengubah value diagnosis result menjadi boolean, M(*malignant*) menjadi 1 dan B(*balign*) menjadi 0, artinya apabila hasil diagnosa kanker ganas akan menghasilkan nilai 1 dan kanker jinak akan menghasilkan nilai 0.

```
# Grouping data
x = df.iloc[:,1:9]
y = df.iloc[:, 0]
print("Data x : \n", x)
print("Data y : \n", y)
```

Fig2. Syntax Grouping Data

Mengelompokkan atribut sesuai dengan variabel input dan variabel output. Variabel inputnya terdiri dari radius, texture, preimeter, area, smoothness, compactness, symmetry, dimensi_fraktal. Variabel outputnya diagnosis_result.

A. Data Preprocessing

Sebelum menerapkan machine learning algorithms, yang harus dilakukan pertama adalah melakukan preprocessing data, preprocessing dilakukan agar data yang diproses nantinya adalah data yang benar dan akurat, tidak ada data yang hilang dan data yang menyimpang[16]. Preprocessing dengan mendeteksi missing value dan outlier value. Setelah dideteksi, diketahui bahwa tidak ada missing dan terdapat outlier di beberapa variabel. Outlier yang terdeteksi kemudian diganti dengan nilai rata-rata dari tiap variabel.[17]

```
# Mendeteksi outliers
print("Deteksi outlier : \n")
outliers=[]
def detect_outlier(data):
    threshold=3
    mean = np.mean(data)
    std = np.std(data)

    for x in data:
        z_score = (x-mean)/std
        if np.abs(z_score)>threshold:
            outliers.append(x)
    return outliers
variabel = ['diagnosis_result','radius','area','smoothness','compactness','symmetry', 'fractal_dimension', 'texture', 'perimeter']
for var in variabel:
  outlier_datapoints = detect_outlier(df[var])
  print("Outlier ", var, " = ", outlier_datapoints)

df.isna().sum()

# Mengganti Outlier value dengan mean
for i in variabel:
  df[i] = df[i].fillna(df[i].mean)

print("Data telah di preprocessing : \n",df)
```

Fig3. Syntax Data Preprocessing

B. Normalisasi: Z-Score

Setelah melakukan preprocessing selanjutnya melakukan transformasi data. Tahap ini mengubah skala pengukuran data menjadi bentuk lain yang nantinya dapat memenuhi. Sering kali variabel dataset memiliki nilai rentang yang besar, oleh karena itu perlu dilakukan transformasi dengan normalisasi dengan mengubah nilai rentangnya jadi tidak terlalu besar[15]. Normalisasi Z-Score membuat nilai atribut akan diubah berdasarkan mean dan standar deviasi, menskalakan selisih antara nilai pada data dan rata-ratanya dengan nilai standar deviasinya[18].

```
# Normalisasi data dengan Z-Score
Zscore = StandardScaler()
x = Zscore.fit_transform(x)
```

Fig4. Syntax Normalisasi Z-Score

Data value yang dinormalisasikan dengan z-score adalah data variabel input

C. *Principal Component Analysis*

Seleksi fitur adalah mengambil fitur yang paling signifikan dari dataset yang menggambarkan karakteristik dari data tersebut. Seleksi fitur menggunakan metode PCA(*Principal Component Analysis*), dimana PCA merupakan metode untuk mereduksi dimensi dengan membentuk variabel baru yang disebut *Principal Components*. PCA dalam mereduksi data, juga mempertahankan informasi sebanyak-banyaknya dan meminimalkan error.[14]

```
# Seleksi Fitur Dengan PCA
x = PCA(n_components=5).fit_transform(x)
```

Fig5. Syntax PCA

D. *Data Training and Data Testing*

Data yang telah dinormalisasikan akan dibagi menjadi data training dan data testing ditetapkan dengan rasio pembagian 80% data training dan 20% data testing. Testing digunakan untuk mengevaluasi model prediksi.[13]

```
# Spliting data menjadi data training dan data testing
x_1, x_2, y_train, y_test = train_test_split(x,y, test_size=0.2, random_state=0)

# Mengklasifikasikan data variable menjadi 3 bagian
kbins = KBinsDiscretizer(n_bins=3, encode='ordinal', strategy='uniform', random_state=0)
x_train = kbins.fit_transform(x_1)
x_test = kbins.transform(x_2)


print("Data training variable :", x_train)
print("Data training class :", *y_train)
print("Data testing variable :", x_test)
print("Data testing class :", *y_test)
```

Fig6. Syntax data training data testing

- Pertama, data x dan y dibagi menjadi dua bagian yaitu data training (x_1, y_train) dan data testing (x_2, y_test) dengan menggunakan fungsi train_test_split() dari library Scikit-Learn. Data testing akan digunakan untuk menguji performa model pada data yang belum pernah dilihat sebelumnya, sedangkan data training akan digunakan untuk melatih model
- Selanjutnya, data training (x_1) diolah dengan menggunakan fungsi KBinsDiscretizer() dari Scikit-Learn untuk mengklasifikasikan data menjadi 3 bagian. Proses ini dilakukan dengan membagi data menjadi beberapa interval atau range nilai, kemudian menentukan label atau kelas untuk setiap interval tersebut[19]
- Hasil pengolahan data training kemudian disimpan pada variabel x_train, sedangkan data testing (x_2) diolah dengan menggunakan fungsi transform() yang sama dengan KBinsDiscretizer() yang telah dilakukan pada data training. Hasil pengolahan data testing kemudian disimpan pada variabel x_test.

Dalam proses ini, n_bins menentukan jumlah interval yang diinginkan, encode menentukan cara encoding label/kelas (dalam hal ini menggunakan encoding ordinal), strategy menentukan metode pembagian interval

(dalam hal ini menggunakan metode uniform), dan random_state menentukan seed yang digunakan untuk menghasilkan nilai acak yang sama setiap kali program dijalankan sehingga menghasilkan output yang konsisten.[20]

E. *Random Forests*

Klasifikasi pertama yaitu *Random Forests* yang merupakan algoritma pembelajaran mesin yang serbaguna dan kuat yang dapat menangani tugas klasifikasi dan regresi. Mereka telah menjadi alat yang populer di bidang ilmu data karena kemampuannya untuk mencapai kinerja canggih pada berbagai masalah. Kami mengimplementasikan algoritma menggunakan *library scikit-learn* dengan *Python*[8].

```
# Random Forest
print("Random Forest")
Forest = RandomForestClassifier(random_state=0)
Forest.fit(x_train, y_train)
y_predictionRF = Forest.predict(x_test)
accuracy_RF = round(accuracy_score(y_test, y_predictionRF)* 100, 2)
acc_DecisionRF = round(Forest.score(x_train, y_train)* 100, 2)
print("Prediksi Random Forest : ", y_predictionRF)


# Confusion Matrix Random Forest
CMRF = confusion_matrix(y_test, y_predictionRF)
accuracyRF = accuracy_score(y_test, y_predictionRF)
precisionRF = precision_score(y_test, y_predictionRF)
recallRF = recall_score(y_test, y_predictionRF)
f1RF = f1_score(y_test, y_predictionRF)

TNRF = CMRF[1][1] * 1.0
FNRF = CMRF[1][0] * 1.0
TPRF = CMRF[0][0] * 1.0
FPRF = CMRF[0][1] * 1.0
total = TNRF + TPRF + FPRF + FNRF
sensitivityRF = TNRF / (TNRF + FPRF)* 100
specificityRF = TPRF / (TPRF + FNRF)* 100

print("Akurasi Random Forest: ", accuracyRF * 100, "%")
print("Sensitivity Random Forest: ", sensitivityRF, "%")
print("Specificity Random Forest: ", + specificityRF, "%")
print("Precision Random Forest: ", + precisionRF)

# Menampilkan Confusion Matrix Random Forest
cm_displayRF=ConfusionMatrixDisplay(confusion_matrix=CMRF)
print('Confusion matrix for Random Forest\n',CMRF)
f, ax = plt.subplots(figsize=(8,5))
sns.heatmap(confusion_matrix(y_test, y_predictionRF), annot=True, fmt=".0f", ax=ax)
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()
```

Fig7. Syntax Random Forests

F. Decision Trees

Klasifikasi kedua yaitu *Decision Trees* yang merupakan algoritma pembelajaran mesin yang banyak digunakan yang dapat menangani data kategorik dan kontinu. Namun, mereka rentan terhadap overfitting dan mungkin tidak bekerja sebaik algoritma lain pada kumpulan data yang kompleks. Meskipun demikian, mereka populer karena mudah ditafsirkan. Kami mengimplementasikan algoritma menggunakan *library scikit-learn* dengan *Python*[9].

```
# Decision Tree
print("Decision Tree")
Decision = DecisionTreeClassifier(random_state=0)
Decision.fit(x_train, y_train)
y_predictionDT = Decision.predict(x_test)
accuracy_DT = round(accuracy_score(y_test, y_predictionDT)* 100, 2)
acc_DecisionDT = round(Decision.score(x_train, y_train)* 100, 2)
print("Prediksi Decision Tree : ", y_predictionDT)

# Confusion Matrix Decision Tree
CMDT = confusion_matrix(y_test, y_predictionDT)
accuracyDT = accuracy_score(y_test, y_predictionDT)
precisionDT = precision_score(y_test, y_predictionDT)
recallDT = recall_score(y_test, y_predictionDT)
f1DT = f1_score(y_test, y_predictionDT)

TNDT = CMDT[1][1] * 1.0
FNDT = CMDT[1][0] * 1.0
TPDT = CMDT[0][0] * 1.0
FPDT = CMDT[0][1] * 1.0
total = TNDT + TPDT + FPDT + FNDT
sensitivityDT = TNDT / (TNDT + FPDT)* 100
specificityDT = TPDT / (TPDT + FNDT)* 100

print("Akurasi Decision Tree: ", accuracyDT * 100, "%")
print("Recall Decision Tree: ", recallDT * 100, "%")
print("Precision Decision Tree: ", + precisionDT)

# Menampilkan Confusion Matrix Decision Tree
cm_displayDT=ConfusionMatrixDisplay(confusion_matrix=CMDT)
print('Confusion matrix for Decision Tree\n',CMDT)
f, ax = plt.subplots(figsize=(8,5))
sns.heatmap(confusion_matrix(y_test, y_predictionDT), annot=True, fmt=".0f", ax=ax)
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()
```

Fig8. Syntax Decision Trees

G. *Naïve Bayes*

Klasifikasi ketiga yaitu *Naïve Bayes* yang merupakan algoritma klasifikasi dasar yang mengasumsikan bahwa kelas untuk klasifikasi bersifat independen, meskipun hal ini jarang benar. Namun, *Naïve Bayes* telah terbukti memberikan hasil yang cukup baik dalam contoh kehidupan nyata. Kami mengimplementasikan algoritma menggunakan *library scikit-learn* dengan *Python*[10].

```
# Naive Bayes
print("Naive Bayes")
gaussian = GaussianNB()
gaussian.fit(x_train, y_train)
y_predictionNB = gaussian.predict(x_test)
accuracy_nb = round(accuracy_score(y_test, y_predictionNB)* 100, 2)
acc_gaussianNB = round(gaussian.score(x_train, y_train)* 100, 2)
print("Prediksi Naive Bayes : ", y_predictionNB)

# Confusion Matrix Naive Bayes
CMNB = confusion_matrix(y_test, y_predictionNB)
accuracyNB = accuracy_score(y_test, y_predictionNB)
precisionNB = precision_score(y_test, y_predictionNB)
recallNB = recall_score(y_test, y_predictionNB)
f1NB = f1_score(y_test, y_predictionNB)

TNNB = CMNB[1][1] * 1.0
FNNB = CMNB[1][0] * 1.0
TPNB = CMNB[0][0] * 1.0
FPNB = CMNB[0][1] * 1.0
total = TNNB + TPNB + FPNB + FNNB
sensitivityNB = TNNB / (TNNB + FPNB)* 100
specificityNB = TPNB / (TPNB + FNNB)* 100

print("Akurasi Naive Bayes: ", accuracyNB * 100, "%")
print("Recall Naive Bayes: ", recallNB*100, "%")
print("Precision Naive Bayes: ", + precisionNB)

# Menampilkan Confusion Matrix Naive Bayes
cm_displayNB=ConfusionMatrixDisplay(confusion_matrix=CMNB)
print('Confusion matrix for Naive Bayes\n',CMNB)
f, ax = plt.subplots(figsize=(8,5))
sns.heatmap(confusion_matrix(y_test, y_predictionNB), annot=True, fmt=".0f", ax=ax)
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()
```

Fig9. Syntax Naïve Bayes

H. *Support Vector Machine (SVM)*

SVM bekerja dengan menemukan garis atau hiperbidang terbaik yang memisahkan dua kelompok data yang berbeda[21]. Garis atau hiperbidang ini ditemukan dengan memaksimalkan jarak antara garis atau hiperbidang dan data yang terdekat dari setiap kelompok. Dalam kasus data yang tidak dapat dipisahkan secara linier, SVM dapat menggunakan kernel untuk mentransformasikan data ke dimensi yang lebih tinggi sehingga dapat dipisahkan secara linier.

```
# SVM
print("SVM")
SVM = SVC(random_state=0)
SVM.fit(x_train, y_train)
y_predictionSVM = SVM.predict(x_test)
accuracy_SVM = round(accuracy_score(y_test, y_predictionSVM)* 100, 2)
acc_DecisionSVM = round(SVM.score(x_train, y_train)* 100, 2)
print("Prediksi SVM : ", y_predictionSVM)

# Confusion Matrix SVM
CMSVM = confusion_matrix(y_test, y_predictionSVM)
accuracySVM = accuracy_score(y_test, y_predictionSVM)
precisionSVM = precision_score(y_test, y_predictionSVM)
recallSVM = recall_score(y_test, y_predictionSVM)
f1SVM = f1_score(y_test, y_predictionSVM)

TNSVM = CMDT[1][1] * 1.0
FNSVM = CMDT[1][0] * 1.0
TPSVM = CMDT[0][0] * 1.0
FPSVM = CMDT[0][1] * 1.0
total = TNSVM + TPSVM + FPSVM + FNSVM
sensitivityDT = TNSVM / (TNSVM + FPSVM)* 100
specificityDT = TPSVM / (TPSVM + FNSVM)* 100

print("Akurasi SVM: ", accuracySVM * 100, "%")
print("Recall SVM: ", recallSVM*100, "%")
print("Precision SVM: ", + precisionSVM)
```

Fig10. Syntax SVM

I. Evaluation

Untuk mengevaluasi kinerja setiap algoritma, kami menghitung akurasi, presisi, dan recall setiap algoritma klasifikasi yang digunakan. Kami juga melakukan perbandingan hasil yang diperoleh dari masing-masing algoritma untuk mengidentifikasi yang paling efektif untuk dataset kami. Data yang diolah dengan transformasi dan seleksi fitur dibandingkan dengan data yang diolah tanpa menggunakan hal tersebut, dari hasil kedua tersebut, diperoleh hasil perbandingan antara akurasi yang tertinggi dan akan digunakan untuk mengecek prediksi.

IV. RESULTS

Dataset berisi sekitar seratus (100) pasien, setiap instance terdiri dari sepuluh (10) atribut termasuk label kelas yang menunjukkan bahwa hasil diagnosis instance adalah ganas atau jinak. Kedua istilah ini menggambarkan tingkat kanker. Jika daerah kanker terlokalisir, memiliki batas yang jelas, dan tidak menyebar maka disebut jinak. Jika daerah kanker bersifat umum, memiliki bentuk dan batas yang tidak normal, dan menyebar dengan cepat maka disebut ganas. Atribut dari dataset ini antara lain id (kode unik pasien), diagnosis result (hasil diagnosis), radius, tekstur, keliling, luas, kehalusan, kekompakan, simetri, dimensi_fraktal.

```
Data Preview :
      id diagnosis_result  radius  texture  perimeter  area  smoothness  compactness  symmetry  fractal_dimension
0      1                M      23       12        151   954       0.143        0.278     0.242              0.079
1      2                B       9       13        133  1326       0.143        0.079     0.181              0.057
2      3                M      21       27        130  1203       0.125        0.160     0.207              0.060
3      4                M      14       16         78   386       0.070        0.284     0.260              0.097
4      5                M       9       19        135  1297       0.141        0.133     0.181              0.059
..   ...              ...     ...      ...        ...   ...         ...          ...       ...                ...
95    96                M      23       16        132  1264       0.091        0.131     0.210              0.056
96    97                B      22       14         78   451       0.105        0.071     0.190              0.066
97    98                B      19       27         62   295       0.102        0.053     0.135              0.069
98    99                B      21       24         74   413       0.090        0.075     0.162              0.066
99   100                M      16       27         94   643       0.098        0.114     0.188              0.064

[100 rows x 10 columns]
```

Fig11. Data Preview

A. Data tanpa transformasi dan seleksi fitur

Pengolahan data tanpa normalisasi z-score dan seleksi fitur dengan PCA. Dari tiga algoritma klasifikasi, didapatkan prediksi, akurasi, recall, precision, dan confusion matrix masing-masing algoritma. Pengolahan data tanpa transformasi dan seleksi fitur menghasilkan nilai akurasi paling besar pada *Random Forests* sebesar 95%.

```
Naive Bayes
Prediksi Naive Bayes :  [1 0 1 0 1 0 1 0 0 1 1 0 1 1 0 1 1 1 1 0]
Akurasi Naive Bayes:  75.0 %
Recall Naive Bayes:  70.58823529411765 %
Precision Naive Bayes:  1.0
Confusion matrix for Naive Bayes
 [[ 3  0]
 [ 5 12]]
```

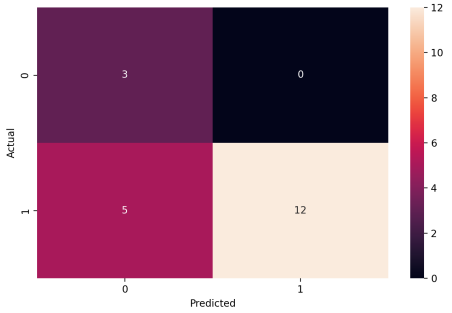

Fig12. Hasil Naïve Bayes tanpa transformasi

```
Decision Tree
Prediksi Decision Tree :  [1 0 1 0 1 0 1 1 0 1 1 1 1 1 0 1 1 1 1 0]
Akurasi Decision Tree:  75.0 %
Recall Decision Tree:  76.47058823529412 %
Precision Decision Tree:  0.9285714285714286
Confusion matrix for Decision Tree
 [[ 2  1]
 [ 4 13]]
```

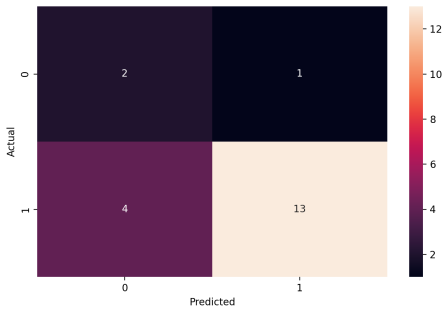

Fig13. Hasil Desicion Tree tanpa transformasi

```
Random Forest
Prediksi Random Forest :  [1 1 1 0 1 0 1 1 1 1 1 0 1 1 0 1 1 1 1 1]
Akurasi Random Forest:  95.0 %
Recall Random Forest:  94.11764705882352 %
Precision Random Forest:  1.0
Confusion matrix for Random Forest
 [[ 3  0]
 [ 1 16]]
```

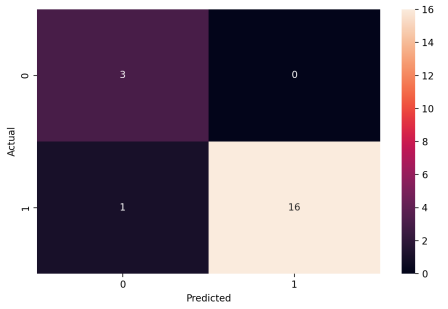

Fig14. Hasil Random Forest tanpa transformasi

```
SVM
Prediksi SVM :  [1 1 1 0 1 0 1 1 1 1 1 0 1 1 0 1 1 1 1 1]
Akurasi SVM:  95.0 %
Recall SVM:  94.11764705882352 %
Precision SVM:  1.0
Confusion matrix for SVM
 [[ 3  0]
 [ 1 16]]
```

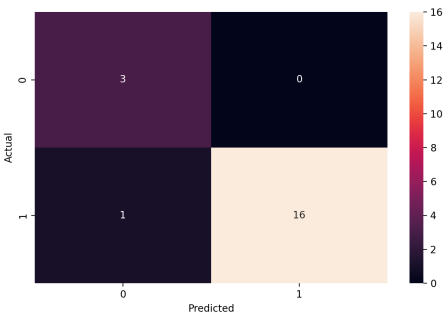

Fig15. Hasil SVM tanpa transformasi

```
Perbandingan Akurasi Tanpa Transformasi dan Seleksi Fitur :
 Naive Bayes :  75.0 %
 Decision Tree :  75.0 %
Random Forest :  95.0 %
SVM :  95.0 %
```

Fig16. Perbandingan Akurasi tanpa transformasi

B. Data dengan tranformasi dan seleksi fitur

Dari tiga algoritma klasifikasi, didapatkan prediksi, akurasi, recall, precision, dan confusion matrix masing-masing algoritma. Pengolahan data dengan transformasi dan seleksi fitur menghasilkan nilai akurasi paling besar pada *Naive Bayes* sebesar 80%.

```
Naive Bayes
Prediksi Naive Bayes :  [1 0 1 0 1 0 1 0 0 1 1 1 1 1 1 1 1 1 1]
Akurasi Naive Bayes:  80.0 %
Recall Naive Bayes:  82.35294117647058 %
Precision Naive Bayes:  0.9333333333333333
Confusion matrix for Naive Bayes
 [[ 2  1]
 [ 3 14]]
```

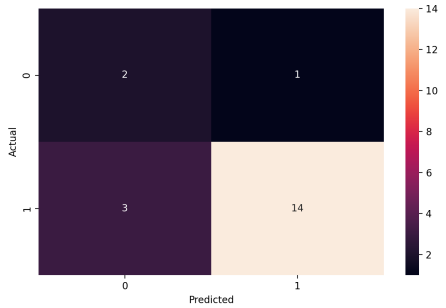

Fig17. Hasil Naïve Bayes dengan transformasi

```
Decision Tree
Prediksi Decision Tree :  [1 0 0 0 1 0 1 0 0 1 1 1 1 1 1 1 1 1 0]
Akurasi Decision Tree:  70.0 %
Recall Decision Tree:  70.58823529411765 %
Precision Decision Tree:  0.9230769230769231
Confusion matrix for Decision Tree
 [[ 2  1]
 [ 5 12]]
```

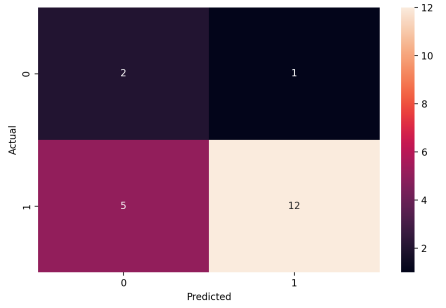

Fig18. Hasil Desicion Tree dengan transformasi

```
Random Forest
Prediksi Random Forest :  [1 0 1 0 1 0 1 0 0 1 1 1 1 1 1 1 1 1 0]
Akurasi Random Forest:  75.0 %
Recall Random Forest:  76.47058823529412 %
Precision Random Forest:  0.9285714285714286
Confusion matrix for Random Forest
 [[ 2  1]
 [ 4 13]]
```

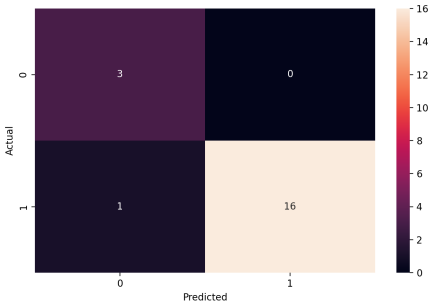

Fig19. Hasil Random Forest dengan transformasi

```
SVM
Prediksi SVM :  [1 0 1 0 1 0 1 0 0 1 1 1 1 1 1 1 1 1 1]
Akurasi SVM:  80.0 %
Recall SVM:  82.35294117647058 %
Precision SVM:  0.9333333333333333
Confusion matrix for SVM
 [[ 2  1]
 [ 3 14]]
```

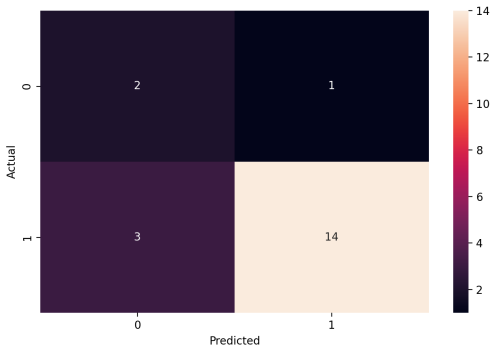

Fig20. Hasil SVM dengan transformasi

```
Perbandingan Akurasi Dengan Transformasi dan Seleksi Fitur:
 Naive Bayes :  80.0 %
 Decision Tree :  70.0 %
 Random Forest :  75.0 %
 SVM :  80.0 %
```

Fig21. Perbandingan Akurasi

Setelah mendapatkan hasil akurasi klasifikasi dari kedua pengolahan data, kami mendapatkan hasil bahwa akurasi perhitungan paling besar dari pengolahan data adalah menggunakan pengolahan data tanpa transformasi dan seleksi fitur berupa *Random Forests* dan *Support Vector Machines* sebesar 95% seperti yang dapat dilihat pada Tabel 1.

TABLE 1
THE SIGNIFICANCE OF THE RELATIONSHIPS IN THE MODEL

Klasifikasi	Accuracy(%)	Recall(%)	Precision
Data tanpa tranformasi dan seleksi fitur			
Random Forests	95	94.1	1
Decision Trees	75	76.47	0.92
Naïve Bayes	75	70.58	1
Support Vector Machine (SVM)	95	94.1	1
Data dengan tranformasi dan seleksi fitur			
Random Forests	75	76.47	0.92
Decision Trees	70	70.58	0.92
Naïve Bayes	80	82.35	0.93
Support Vector Machine (SVM)	80	82.35	0.93

V. DISCUSSION

Setelah menyeleksi akurasi tertinggi dari dua skema, ditemukan bahwa akurasi random forest dan support vector machines skema satu memiliki akurasi terbesar dibandingkan metode lainnya dengan nilai akurasi sebesar 95%. Setelah itu kami mencoba untuk membuat tiga data testing baru dengan memasukan nilai parameter dan hasil dari pembelajaran mesin dengan metode random forest sebagai berikut:

TABLE 2
INPUT DATA TESTING RANDOM FOREST

Parameter	Data 1	Data 2	Data 3
Radius	27	5	4
Texture	20	40	10
Perimeter	132	182	132
Area	1334	1000	1334
Smoothness	0.15	0.253	0.126
Compactness	0.231	0.131	0.111
Symmetry	0.321	0.421	0.221
Fractal Dimension	0.069	0.169	0.043
Diagnosis Result	Maligna	Maligna	Benigna

```
===Input Data 1===
Dari data yang diinput sebagai berikut :  {'radius': 27, 'texture': 20, 'perimeter': 132, 'area': 1334, 'smoothness': 0.15, 'compactne
ss': 0.231, 'symmetry': 0.321, 'fractal_dimension': 0.069}
Didapatkan hasil diagnosis :  [1]
Orang tersebut mengidap kanker prostat ganas (maligna)

===Input Data 2===
Dari data yang diinput sebagai berikut :  {'radius': 5, 'texture': 40, 'perimeter': 182, 'area': 1000, 'smoothness': 0.253, 'compactne
ss': 0.131, 'symmetry': 0.421, 'fractal_dimension': 0.169}
Didapatkan hasil diagnosis :  [1]
Orang tersebut mengidap kanker prostat ganas (maligna)

===Input Data 3===
Dari data yang diinput sebagai berikut :  {'radius': 4, 'texture': 10, 'perimeter': 132, 'area': 1334, 'smoothness': 0.126, 'compactne
ss': 0.111, 'symmetry': 0.221, 'fractal_dimension': 0.043}
Didapatkan hasil diagnosis :  [0]
Orang tersebut mengidap kanker prostat jinak (benigna)
```

Fig22. Output Data Testing Random Forest

Dapat dilihat bahwa hasil data testing terbaru pada Table 2 menunjukkan bahwa dari ketiga data testing baru, data satu dan data dua mengalami kanker prostat ganas dan data tiga mengalami kanker prostat jinak.

VI. Conclusions

Dalam penelitian ini, kami melakukan pengolahan data menggunakan berbagai teknik transformasi dan seleksi fitur, serta metode klasifikasi berbeda untuk menghasilkan model klasifikasi yang akurat. Berdasarkan hasil yang diperoleh, dapat disimpulkan bahwa pengolahan data tanpa transformasi dan seleksi fitur menggunakan metode klasifikasi *Random Forests* dan ***Support Vector Machines*** memberikan akurasi terbaik sebesar 95%. Hal ini menunjukkan bahwa pengolahan data yang dilakukan sudah cukup baik dalam menentukan fitur yang relevan untuk digunakan dalam proses klasifikasi dan menghasilkan model klasifikasi yang cukup kuat. Penggunaan metode klasifikasi *Random Forests* dan ***Support Vector Machines*** juga terbukti efektif dalam menghasilkan prediksi yang akurat dan dapat digunakan dalam proses pengambilan keputusan yang tepat. Oleh karena itu, pengolahan data tanpa transformasi dan seleksi fitur berupa *Random Forests* dan ***Support Vector Machines*** dapat dijadikan pilihan yang baik untuk menghasilkan model klasifikasi yang akurat dalam berbagai aplikasi kecerdasan buatan.

REFERENCES

[1] Akinnuwesi, B. A., Kehinde A. O., Benjamin S. A., Stephen G. F., Elliot M., Moses O., Patrick O. (2023). Application of support vector machine algorithm for early differential diagnosis of prostate cancer. Data Science and Management, Volume 6, Issue 1, 2023, Pages 1-12, ISSN 2666-7649. https://doi.org/10.1016/j.dsm.2022.10.001.

[2] Win, S. L., Hatike, Z. Z., Yusof, F., & Noorbatcha, I. A. (2014, Mei). Cancer Recurence Prediction using Machine Learning,. *International Journal of Computational Science and Information Technology (IJCSITY)*, *2*(2).

[3] Rahayuwati, L., Rizal, I. A., Pahria, T., Lukman, M., & Juniarti, N. (2020). Pendidikan kesehatan tentang pencegahan penyakit kanker dan menjaga kualitas kesehatan. Media Karya Kesehatan, 3(1).

[4] Coletti, R., Leonardelli, L., Parolo, S., & Marchetti, L. (2020). A QSP model of prostate cancer immunotherapy to identify effective combination therapies. Scientific reports, 10(1), 1-18.

[5] Bray, F., Ferlay, J., Soerjomataram, I., Siegel, R. L., Torre, L. A., & Jemal, A. (2018). GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries. Ca Cancer J Clin, 68(6), 394-424.

[6] Goldenberg, S. L., Nir, G., & Salcudean, S. E. (2019). A new era: artificial intelligence and machine learning in prostate cancer. Nature Reviews Urology, 16(7), 391-403.

[7] Ahuja, A. S. (2019). The impact of artificial intelligence in medicine on the future role of the physician. PeerJ, 7, e7702.

[8] Raschka, S., & Mirjalili, V. (2019). Python machine learning: Machine learning and deep learning with Python, scikit-learn, and TensorFlow 2. Packt Publishing Ltd.

[9] Mao, L., & Zhang, W. (2021). Analysis of entrepreneurship education in colleges and based on improved decision tree algorithm and fuzzy mathematics. Journal of Intelligent & Fuzzy Systems, 40(2), 2095-2107..

[10] Taheri,S. & Mammadov,M.(2013).Learning the naive Bayes classifier with optimization models. International Journal of Applied Mathematics and Computer Science,23(4) 787-795. https://doi.org/10.2478/amcs-2013-0059

[11] Wang, S., Guo, X., Xu, L., Feng, G., Liu, M., Hu, Z., & Zhang, Y. (2020). Deep learning-based diagnosis of prostate cancer using histopathology images: A review. Frontiers in Oncology, 10, 1021.

[12] Zhang, J., Yuan, Y., Zhang, X., & Xu, J. (2021). An ensemble learning method for predicting metastasis risk of prostate cancer. BMC Cancer, 21(1), 1-9.

[13] Kusnadi, A., Pane, I. Z., Yaman Khaeruzzaman, V., & Clara, C. (2022). Ekstrasi Fitur Dan Pengenalan Wajah Konsep Dan Aplikasinya. CV Literasi Nusantara Abadi.

[14] Liu, H., Sun, J., Liu, L., & Zhang, H. (2019). Feature selection with dynamic mutual information. Pattern Recognition, 42(7), 1330-1339.

[15] Hafemeister, C., & Satija, R. (2019). Normalization and variance stabilization of single-cell RNA-seq data using regularized negative binomial regression. Genome biology, 20(1), 296.

[16] Guo, W., Che, L., Shahidehpour, M., & Wan, X. (2021). Machine-Learning based methods in short-term load forecasting. The Electricity Journal, 34(1), 106884.

[17] Rustum, R., & Adeloye, A. J. (2017). Replacing outliers and missing values from activated sludge data using Kohonen self-organizing map. Journal of Environmental Engineering, 133(9), 909-916.

[18] Henderi, H., Wahyuningsih, T., & Rahwanto, E. (2021). Comparison of Min-Max normalization and Z-Score Normalization in the K-nearest neighbor (kNN) Algorithm to Test the Accuracy of Types of Breast Cancer. International Journal of Informatics and Information Systems, 4(1), 13-20.

[19] Kenett, R., Zacks, S., & Gedeck, P. (2022). Modern Analytic Methods: Part I. In Modern Statistics: A Computer-Based Approach with Python (pp. 361-393). Cham: Springer International Publishing.

[20] Kietzmann, P., Schmidt, T. C., & Wählisch, M. (2021). A guideline on pseudorandom number generation (PRNG) in the IoT. ACM Computing Surveys (CSUR), 54(6), 1-38.

[21] Fauzi, A., Setiawan, E. B., & Baizal, Z. K. A. (2019, March). Hoax news detection on twitter using term frequency inverse document frequency and support vector machine method. In Journal of Physics:

[22] https://www.kaggle.com/datasets/sajidsaifi/prostate-cancerConference Series (Vol. 1192, No. 1, p. 012025). IOP Publishing.