

## Klasifikasi Diagnosis untuk Penyakit Kanker Serviks Menggunakan Algoritma Extreme Gradient Boosting (XGBoost)

Opitasari\*<sup>1</sup>, Fauzan Natsir<sup>2</sup>, Ega Shela Marsiani<sup>3</sup>

<sup>1,2,3</sup>Program Studi Teknik Informatika, Fakultas Teknik dan Ilmu Komputer, Universitas Indraprasta PGRI

<sup>1</sup>opitasari@gmail.com; <sup>2</sup>fauzan.natsir@gmail.com; <sup>3</sup>egashela@gmail.com

\*) Corresponding author

(received: 26-03-24, revised: 11-05-24, accepted: 30-05-24)

### Abstract

Cervical cancer, also commonly known as cervical mouth cancer, is one of the deadliest cancers in women after breast cancer. According to the WHO survey, of the total cancer cases in Indonesia, 9.2% of them are cervical cancer with 36,633 cases. It is difficult to determine the early symptoms of cervical cancer because the symptoms that arise are invisible, so there are many cases of late treatment in patients with this disease. Research was conducted using the XGBoost method to classify early symptoms of cervical cancer using datasets taken from the UCI Repository. XGBoost performs optimization with boosting techniques to minimize the loss function value. Evaluation of this model obtained an accuracy value of 86%. Accuracy calculations are used to find out how precisely the model makes predictions for each instance, Metric Precision is used to find the value of the classification model in returning relevant instances, the precision value in the research is 100%, Metric Recall is used to find out how well the model "remembers" or identifies the value the relevant instance, the recall value in this study is 82% and the F1-Score or also known as F Measure is the average comparison and balance value of the Precision and Recall f1-score of 90% in the XGBoost model so it can be concluded that XGBoost is very good at carrying out classification. cervical cancer.

**Keyword:** Cervical Cancer, XGBoost, Phyton.

### Abstrak

Kanker serviks yang juga biasa dikenal dengan kanker mulut leher rahim merupakan satu dari beberapa jenis penyakit kanker yang mematikan pada wanita setelah kanker payudara. Menurut survei WHO, dari total kasus kanker di Indonesia, 9,2% kasus di antaranya adalah kanker serviks dengan jumlah 36.633 kasus. Sulitnya menentukan gejala awal pada kanker serviks dikarenakan gejala yang timbul tidak kasat mata sehingga banyak sekali kasus terlambat penanganan pada pasien penderita penyakit ini. Penelitian dilakukan dengan metode XGBoost untuk mengklasifikasi gejala awal penyakit kanker serviks dengan menggunakan dataset yang diambil dari UCI Repository. XGBoost melakukan optimasi dengan teknik boosting untuk meminimalisir nilai *loss function*. Jika nilai *loss function* menghasilkan nilai yang tinggi itu menandakan bahwa model yang dihasilkan sangat buruk, begitu sebaliknya. Evaluasi model ini didapatkan perolehan nilai accuracy sebesar 86%. Perhitungan Accuracy digunakan untuk mencari seberapa tepat model melakukan prediksi dari masing-masing instance, Metric Precision digunakan untuk mencari nilai model klasifikasi dalam mengembalikan instance yang relevan nilai precision pada penelitian 100%, Metric Recall digunakan untuk mengetahui seberapa baik model "mengingat" atau mengidentifikasi nilai instance yang relevan nilai recall pada penelitian ini 82% dan F1-Score atau bisa disebut juga F Measure merupakan nilai rata-rata perbandingan dan keseimbangan dari Precision dan Recall f1-score 90% pada model XGBoost sehingga dapat disimpulkan bahwa XGBoost sangat baik dalam melakukan klasifikasi penyakit kanker serviks.

**Kata Kunci:** Kanker Serviks, XGBoost, Phyton

### I. Pendahuluan

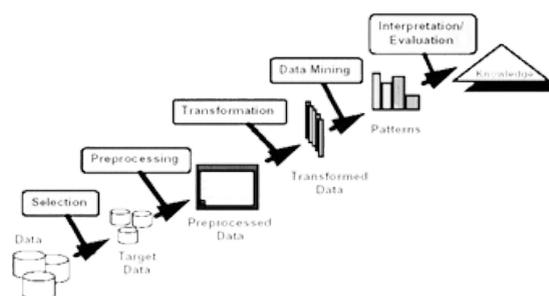
Penyakit kanker serviks yang juga biasa dikenal dengan kanker mulut leher rahim, yang merupakan satu dari beberapa jenis penyakit kanker yang mematikan pada wanita setelah kanker payudara[1]. Menurut survei WHO,

dari total kasus kanker di Indonesia, 9,2% kasus di antaranya adalah kanker serviks dengan jumlah 36.633 kasus dan menempati urutan ke 2 terbanyak dari jenis kanker lainnya[2]. Penyebab meningkatnya jumlah kematian akibat kanker serviks antara lain adalah keterlambatan mendiagnosis penyakit tersebut. Sulitnya menentukan gejala awal pada kanker serviks dikarenakan gejala yang timbul tidak kasat mata, sehingga banyak sekali kasus terlambat penanganan pada pasien penderita penyakit tersebut. Permasalahan umum yang terjadi di masyarakat menengah ke bawah adalah bahwa mereka akan mencari perawatan ketika gejala yang dirasakan sudah cukup parah sehingga kanker sudah menyebar cepat dan sulit untuk ditangani[3]. Untungnya, hampir setiap kasus kanker serviks berpotensi dapat dicegah dan dilakukan pengobatan dini dengan dilakukan *Screening Test* (*pap smear* dan IVA test) dan vaksinasi HPV. Namun, lagi-lagi dengan adanya *screening test* tersebut pun masih belum sepenuhnya masyarakat berminat dan mengerti untuk melakukan *screening test* tersebut dikarenakan keterbatasan pengetahuan dan ketidaknyamanan proses yang dilakukan[4]. Dari segi vaksinasi HPV pun belum diketahui secara pasti mengenai jangka waktu kemanjurannya.

Langkah pencegahan awal dapat dilakukan dengan mendeteksi perilaku sosial dan gaya hidup pasien mengingat perilaku seksual menjadi penyebab utama naiknya angka kasus dari penyakit kanker serviks ini. Penelitian yang dilakukan oleh Danang Wijayanto dengan [5] menggunakan algoritma K-Nearest Neighbor (KNN) dan Decision Tree C4.5 pada proses klasifikasi dan menghasilkan akurasi sebesar 96,49% pada algoritma C4.5 dan 94,73% pada algoritma K-Nearest Neighbor (KNN). Pada algoritma C4.5 metode imputasi tidak berdampak signifikan pada hasil akurasi yang diperoleh. Berdasarkan permasalahan tersebut, maka dilakukan penelitian untuk mengklasifikasi penyakit kanker serviks dengan menggunakan metode *Algoritma Extreme Gradient Boosting (XGBoost)*. Penulis menggunakan metode ini dikarenakan XGBoost merupakan metode dengan tingkat fleksibilitas yang baik dan mampu menangani kasus real dalam skala kecil maupun besar dengan tingkat akurasi 10 kali lebih cepat dibandingkan algoritma *Tree Learning* populer lainnya[6]. Untuk algoritma pembandingnya penelitian ini menggunakan metode Random Forest Decision Trees pada penelitian ini didukung oleh Denisko dan Hoffman [7] yang menyimpulkan bahwa dalam penggunaan klasifikasi di industri medis, metode Random Forest Decision Trees memiliki kelebihan seperti performa yang tinggi, bisa ditemukannya kepentingan dari setiap fitur, dan interpretasi yang mudah. Pengambilan fitur yang digunakan pada penelitian ini juga didasarkan dari penelitian yang dilakukan Sholih [8] yang menjabarkan gejala-gejala dan faktor risiko kanker paru-paru di Indonesia. namun pada hasilnya gejala antara resiko kanker paru paru dan kanker serviks memiliki perbedaan yang akhirnya metode ini dirasa masih sedikit kurang akurat. Penelitian dilakukan dengan metode XGBoost untuk mengklasifikasi gejala awal penyakit kanker serviks dengan menggunakan dataset yang diambil dari UCI Repository. Data ini berisi 72 pasien yang dibagi dengan nilai hasil 1 dan 0. Nilai 1 diindikasikan sebagai pasien mengidap kanker serviks dan 0 diindikasikan sebagai pasien tidak mengidap kanker serviks. Data pasien memiliki 19 atribut dan 1 kelas. Kemudian XGBoost akan dilatih dengan dataset yang digunakan sehingga dapat mengidentifikasi dengan tepat apakah pasien terindikasi dengan mengidap kanker serviks atau tidak mengidap kanker serviks dapat tervalidasi dan terklasifikasi dengan tepat dan benar. Harapan dari penelitian menggunakan XGBoost ini adalah metode ini dapat menghasilkan nilai performa yang baik untuk kasus ini sehingga dapat dijadikan suatu sistem yang optimal untuk mendeteksi penyakit kanker serviks.

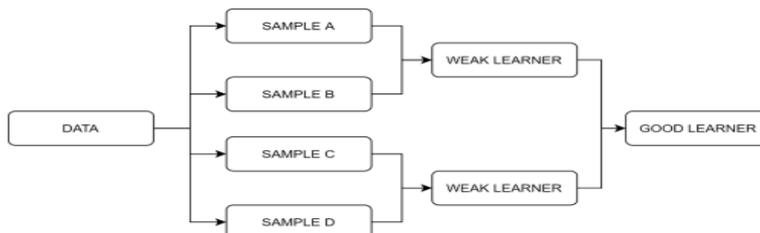
## II. Metodologi Penelitian

Pada Gambar 1 merupakan tahapan dalam penelitian ini. Langkah – langkah pada penelitian ini adalah melalui tahapan *selection* merupakan langkah awal menseleksi data yang akan digunakan untuk penelitian, kemudian *Preprocessing* memastikan bahwa data data yang akan dilanjutkan terhindar dari data yang redundan serta *missing value*, *Transformation* memastikan data siap diolah menyesuaikan dengan metode yang digunakan, *Data Mining* merupakan tahapan uji metode dengan data untuk mengetahui akurasi hasil, dan *Evaluation* mengevaluasi hasil yang akan dipresentasikan.



Gambar 1. Tahapan Penelitian Extreme Gradient Boosting

Penelitian ini menggunakan algoritma *Extreme Gradient Boosting (XGBoost)*. Metode tersebut merupakan salah satu varian dari *Tree Gradient Boosting*[9]. Dalam penelitian tersebut peneliti menyebutkan bahwa XGBoost dapat digunakan untuk menangani kasus real dalam skala kecil maupun besar dengan menggunakan sumber data yang minimal sekalipun. Peneliti juga menambahkan bahwa XGBoost ini berjalan 10 kali lebih cepat daripada algoritma *Tree Learning* populer lainnya karena komputasi algoritma ini dilakukan secara paralel dan terdistribusi. Algoritma ini juga mampu menyesuaikan fleksibilitas dalam suatu kasus. Ilustrasi XGBoost dapat dilihat dari ilustrasi gambar 2.



Gambar 2. Ilustrasi algoritma Extreme Gradient Boosting

Penelitian yang dilakukan oleh Kurnia, dkk [10] mengatakan bahwa XGBoost melakukan optimasi dengan teknik boosting untuk meminimalisir nilai *loss function*. *Loss function* merupakan mekanisme evaluasi atas model. Jika nilai *loss function* menghasilkan nilai yang tinggi itu menandakan bahwa model yang dihasilkan sangat buruk, begitu sebaliknya. Ketika melakukan proses ini, setiap iterasi dibuat agar nilai rata-rata pada *loss function* menjadi sekecil mungkin berdasarkan fungsi awalnya  $F_0(x)$ . Secara umum, algoritma Gradient Boosting menggunakan persamaan 2.2 yang ditulis:

$$\{y_{hm}\} = \operatorname{argmin} \sum L(y_i, f^{(m-1)}(x_i) + y_{hm}(x_i)) \quad M \quad m=1 \dots\dots (1)$$

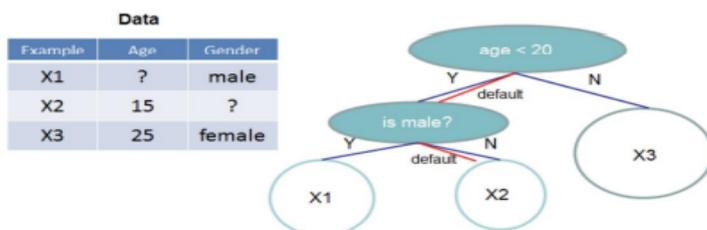
Seperti metode boosting pada umumnya penggunaan nilai residu  $y - F(x)$  pada model berupa nilai gradient negatif. Sehingga singkatnya gradient boosting adalah algoritma gradient descent yang mengurangi nilai *loss* secara diferensiabel sehingga mengarah secara gradient[11]. Tujuan akhirnya untuk mencapai fungsi paling mendekati  $F(x)$  terhadap fungsi-fungsi pembangunnya  $f(x)$  dengan meminimalkan nilai *loss function*  $L(y, f(x))$  sehingga bentuk persamaan dari fungsi tersebut dapat dilihat pada persamaan 2.3 di bawah ini:

$$F = \operatorname{argmin} f_{Ex,y}[L(y, f(x))] \dots\dots\dots (2)$$

Seperti metode *boosting* pada umumnya penggunaan nilai residu  $y - F(x)$  pada model berupa nilai gradient negatif. Sehingga singkatnya *gradient boosting* adalah algoritma *gradient descent* yang mengurangi nilai *loss* secara diferensiabel sehingga mengarah secara gradient[12]. Tujuan akhirnya untuk mencapai fungsi paling mendekati  $F(x)$  terhadap fungsi-fungsi pembangunnya  $f(x)$  dengan meminimalkan nilai *loss function*  $L(y, f(x))$  sehingga bentuk persamaan dari fungsi tersebut dapat dilihat pada persamaan 2.4 di bawah ini:

$$F = \operatorname{argmin} f_{Ex,y}[L(y, f(x))] \dots\dots\dots (3)$$

Penelitian dari D. Septhya [13] juga menambahkan bahwa menurut mereka, sebagian besar algoritma *Tree Learning* lainnya hanya dioptimalkan untung data yang benar-benar bersih atau membutuhkan prosedur khusus untuk mengatasi kasus yang terbatas seperti *categorical encoding*. Tetapi tidak untuk XGBoost, XGBoost dapat menangani *missing value* pada suatu dataset dalam kondisi apapun. Dalam kata lain XGBoost memiliki cara tersendiri untuk mengatasi *missing value* yaitu pada saat pelatihan data, XGBoost akan mempelajari secara otomatis bagaimana cara untuk melakukan imputasi data.



Gambar 3. Ilustrasi XGBoost Mengatasi Masalah pada Missing Value

Penelitian ini sangat penting sekali jika sebuah algoritma itu aware atau mengetahui pola dari kekosongan nilai dalam suatu dataset. Karena hal itu mereka mengusulkan untuk menambahkan sebuah arah default pada masing-masing simpul (*node*) dalam pohon algoritma tersebut. Pada gambar 3 digambarkan ilustrasi mengenai bagaimana penambahan arah *default* pada simpul pohon algoritma tersebut. Penjelasananya ketika sebuah data *x* tidak memiliki nilai dalam suatu atribut, maka data atribut tersebut akan diklasifikasikan menuju arah default. Pada masing-masing cabang (*branch*) pohon algoritma akan memiliki dua pilihan arah *default* tersebut[14]. Lalu untuk menentukan arah *default* yang paling optimal XGBoost mempelajari tersebut dari data yang tersedia. Namun, pada dataset yang diambil untuk penelitian ini tidak terdapat *missing value* sehingga tidak diperlukan adanya proses *data cleaning* pada tahap *pre-processing* dan tahap pemodelan XGBoost pada penelitian ini dapat langsung memasuki tahap selanjutnya.

### III. Hasil dan Pembahasan

Pada penelitian ini, acuan dataset yang digunakan diambil dari UCI Respiratory. Dataset terdiri dari 72 pasien. 72 data tersebut terdiri dari 22 pasien positif penyakit kanker serviks (0 = has cervical cancer) dan 50 pasien negatif kanker serviks (1 = no cervical cancer). Masing-masing data pasien memiliki 20 fitur dan atribut. fitur ke 20 adalah class yang menentukan pasien tersebut positif (0 = has cervical cancer) dan negatif (1 = no cervical cancer) dengan tipe data nominal. Lalu 19 atribut lain nya memiliki fitur dengan tipe data numerik. Fitur atau atribut dalam dataset dapat dilihat pada Tabel 1.

Tabel 1. Atribut Dataset

NO	Nama Atribut	Tipe Data
1	behavior_sexualRisk	numerik
2	behavior_eating	numerik
3	behavior_personalHygine	numerik
4	intention_aggregation	numerik
5	intention_commitment	numerik
6	attitude_consistency	numerik
7	attitude_spontaneity	numerik
8	norm_significantPerson	numerik
9	norm_fulfillment	numerik
10	perception_vulnerability	numerik
11	perception_severity	numerik
12	motivation_strength	numerik
13	motivation_willingness	numerik
14	socialSupport_emotionality	numerik
15	socialSupport_appreciation	numerik
16	socialSupport_instrumental	numerik
17	empowerment_knowledge	numerik
18	empowerment_abilities	numerik
19	empowerment_desires	numerik
20	ca_cervix	nominal

Pada tabel 2 adalah contoh data pasien yang tersedia pada dataset penelitian ini. atribut yang tercantum hanya sebagai contoh dari beberapa atribut yang lain nya yang terdapat nilai pasien positif kanker serviks (Ca\_Cervix bernilai 0) dan nilai pasien negatif kanker serviks (Ca\_Cervix bernilai 1).

Tabel 2. Atribut Dataset

behavior_sexualRisk	behavior_eating	behavior_personalHygine	intention_aggregation	s/d	ca_cervix
10	12	12	8	s/d	1
10	15	15	4	s/d	0

Pada tahap *pre-processing data* ini, dataset yang digunakan pada penelitian akan dilakukan tahap *pre-processing* terlebih dahulu. Pada tahap ini dilakukan proses *data splitting* yaitu proses untuk membagi dataset menjadi data latih dan data uji yang akan digunakan dalam menentukan performansi XGBoost dalam klasifikasi penyakit kanker. Skema pembagian dataset akan dibagi 70% data latih dan 30% data uji.

Pemodelan metode XGBoost terhadap penelitian ini dibuat dengan bahasa pemrograman Phyton yang akan dijalankan di dalam Jupyter Notebook. Jupyter Notebook ini nantinya akan digunakan untuk menjalankan model

yang telah dibuat sesuai dengan dataset yang digunakan dalam penelitian klasifikasi penyakit kanker serviks. *Output* dalam hasil pemodelan ini akan menentukan nilai Accuracy, Precision, Recall, dan F1- Score dari model XGBoost. Untuk menghitung evaluasi model performa klasifikasi, nilai ukur performa model XGBoost dalam mengklasifikasi penyakit kanker serviks yang digunakan memiliki 4 metric, yaitu Accuracy, Precision, Recall dan F1-Score. Hasil dari data uji nantinya akan dimasukkan ke dalam Confusion Matrix terlebih dahulu agar dapat mencari nilai *metric* yang diperoleh model. Ilustrasi dari Confusion Matrix dapat dilihat pada Tabel 3.

Tabel 3. Contoh *Confusion Matrix*

	Data Aktual Positif	Data Aktual Negatif
Terklasifikasi Positif Kanker Serviks	<i>True Positive (TP)</i>	<i>False Positive (FP)</i>
Terklasifikasi Negatif Kanker Serviks	<i>False Negative (FN)</i>	<i>True Negative (TN)</i>

Di dalam Confusion Matrix tersebut terdapat 4 nilai yang menentukan hasil klasifikasi yang telah dijalankan. *True Positive (TP)* adalah data pasien positif ginjal kronis dan terklasifikasi sebagai positif. Kemudian, *False Negative (FN)* adalah sebuah data yang salah diklasifikasikan oleh model, yang artinya data pasien positif ginjal kronis tetapi diklasifikasi oleh model sebagai data pasien yang negatif ginjal kronis. Lalu *False Positive (FP)* adalah data pasien negatif ginjal kronis namun terklasifikasi sebagai positif. Lalu *True Negative (TN)* adalah data pasien negatif ginjal kronis dan berhasil terklasifikasi oleh model sebagai negatif. Perhitungan *accuracy* digunakan untuk mencari seberapa tepat model melakukan prediksi dari masing-masing *instance*. Dimana data pasien positif yang berhasil terklasifikasi (TP) dan data pasien negatif yang berhasil terklasifikasi *True Negative (TN)* akan ditambah, kemudian dibagi dengan jumlah seluruh data yang ada pada *confusion matrix*.

Pengembangan model *Machine Learning Extreme Gradient Boosting* untuk mengklasifikasi penyakit kanker serviks berdasarkan tahapan dan metode yang diterapkan selama pengembangan ini. Bahasa yang digunakan yaitu bahasa Python yang telah diimplementasi ke dalam tools Jupyter Notebook dan beberapa *library*, seperti pada Gambar 4. Berikut ini merupakan *library* yang digunakan:

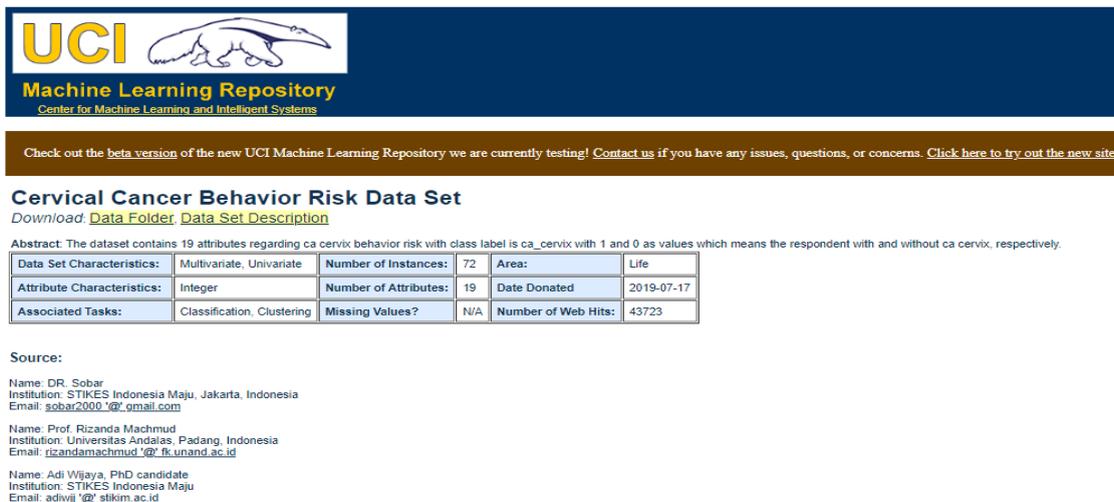
- Pandas*, ini adalah *library* untuk menganalisis dataset, mengkonversi *raw data* ke dalam bentuk yang lebih rapi untuk dianalisis.
- Seaborn* dan *Matplotlib*, *library* ini digunakan untuk memvisualisasikan *confusion matrix* yang digunakan saat evaluasi model.
- Scikit-learn*, ini merupakan *library* inti yang dipakai untuk membangun model machine learning pada penelitian ini. Banyak algoritma pembelajaran yang tersedia dalam *library* ini, salah satu nya adalah klasifikasi. Fungsi yang digunakan adalah `train_test_split()` untuk membagi skema data latih dan data uji.
- XGBoost*, ini merupakan *library* model Extreme Gradient Boosting (XGBoost) yang digunakan dalam program[15].

```
import numpy as np import pandas as pd
import matplotlib.pyplot as plt import seaborn as sns
import pickle
import xgboost as xgb

from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
from sklearn.metrics import accuracy_score
from xgboost import XGBClassifier
```

Gambar 4. Ilustrasi XGBoost Mengatasi Masalah pada Missing Value

Pada Gambar 5 merupakan acuan dataset yang digunakan diambil dari UCI Respiratory. Dataset terdiri dari 72 pasien. 72 data tersebut terdiri dari 22 pasien positif penyakit kanker serviks (0 = *has cervical cancer*) dan 50 pasien negatif kanker serviks (1 = *no cervical cancer*). masing-masing data pasien memiliki 20 fitur dan atribut. fitur ke 20 adalah class yang menentukan pasien tersebut positif (0 = *has cervical cancer*) dan negatif (1 = *no cervical cancer*) dengan tipe data nominal. Lalu 19 atribut lainnya memiliki fitur dengan tipe data numerik.



Gambar 5. UCI Respirotory

Pada tahap ini dataset yang digunakan pada penelitian akan dilakukan tahap *pre-processing* terlebih dahulu. Pada tahap ini dilakukan proses *data splitting* yaitu proses untuk membagi dataset menjadi data latih dan data uji yang akan digunakan dalam menentukan performansi XGBoost dalam klasifikasi penyakit kanker serviks [16]. skema pembagian dataset akan dibagi menjadi 70% dan 30%. *Data splitting* ini menggunakan perintah `train_test_split` yang disediakan oleh *library scikit-learn*.

```
X, y = df.iloc[:, Xpd.get_dummies(X) X_train, X_test, y_train, y_test --1], df.iloc[:, -1]
train_test_split(x, y, test_size=0.3, random_state=42)
```

Gambar 6. Source Code Data Splitting

Pada gambar 6 terlihat terdapat beberapa proses yang dimana proses pertama adalah `df.iloc` yang bertujuan untuk menyeleksi *frame dataset* terakhir dari variabel X dan Y. Setelah itu dilanjutkan dengan proses `get_dummies()` atau bisa disebut *One-Hot Encoding*. *One-Hot Encoding* ini perlu dilakukan karena model tidak dapat membaca tipe data string [10]. Tugas dari encoding ini sendiri adalah dengan mengubah tipe data string tadi menjadi angka biner agar bisa dibaca oleh model. Lalu proses terakhir ialah *data splitting* yang dilakukan secara otomatis oleh perintah `train_test_split` dengan beberapa parameter pendukung seperti `test_size`. `Test_size` ini merupakan parameter untuk memasukan jumlah data uji, yakni pada penelitian ini ialah 0.3 atau 30%. Adapun hasil dari *data splitting* dapat dilihat pada gambar 7.

```
print("Dataset Training", X_train.shape)
print("Dataset Testing", X_test.shape)
Dataset Training - (50, 18) Dataset Testing = (22, 18)
```

Gambar 7. Parameter XGBoost

Pada penentuan tersebut dapat dilihat bahwa terdapat beberapa parameter dalam model XGBoost. Namun, parameter yang dilakukan dilakukan *tuning* secara manual dalam model XGBoost yang dibuat adalah *objective*, *n\_jobs*, *seed* dan *verbosity*. Parameter *objective* merupakan parameter untuk meminimalisir loss function. Nilai yang digunakan dalam parameter ini adalah *binary logistic* dikarenakan hasil akhir kasus klasifikasi ini berupa probabilitas yaitu *Has Cervical Cancer (0)* dan *No Cervical Cancer (1)* atau Positif dan Negatif. Kemudian parameter selanjutnya adalah *n\_jobs*. Parameter ini ialah parameter untuk mengatur jumlah core pada CPU yang digunakan saat pembuatan model. Dalam model yang dibuat ini, nilai parameter *n\_jobs* adalah -1 yang artinya adalah semua core pada CPU digunakan saat pembuatan model. Lalu berikutnya adalah parameter *seed*, parameter ini merupakan parameter untuk mengatur proses *reproducibility* yang artinya ketika proses dijalankan kembali, hasil atau output-nya akan tetap sama seperti hasil dari proses sebelumnya. *Verbose* digunakan untuk memunculkan *report* atau laporan dari model ketika melakukan pelatihan untuk melihat apakah ada kekeliruan atau tidak dan juga waktu eksekusi model.

Pada tahap ini dijelaskan bagaimana proses pemodelan XGBoost pada penelitian ini yang dibuat dengan bahasa pemrograman Python yang akan dijalankan di dalam Jupyter Notebook. Jupyter Notebook ini digunakan untuk menjalankan model yang telah dibuat sesuai dengan dataset yang digunakan dalam penelitian klasifikasi penyakit kanker serviks. *Output* dalam hasil pemodelan ini akan menentukan nilai *Accuracy*, *Precision*, *Recall*, dan *F1-Score* dari model XGBoost. Pada Gambar 8 merupakan source code yang digunakan untuk menghitung *Training Accuracy Score*.

```
print("Training Accuracy Score: ", model_xgb.score(X_train, y_train)) print()
print("Test Accuracy Score: ", model_xgb.score(X_test, y_test))
print()
pred = model_xgb.predict(X_test)
print(classification_report(y_test, pred))
print()
print('Accuracy:', accuracy_score(y_test, pred))
print()
```

Gambar 8. Source Code *Training Accuracy Score*

Variabel *train\_accuracy* adalah variabel untuk menampilkan nilai hasil *training*. Sedangkan variabel *test\_accuracy score* merupakan variabel untuk menampilkan hasil testing. Variabel *pred* adalah variabel untuk melakukan prediksi terhadap data uji. Lalu dihasilkan tampilan *classification report* berupa nilai *accuracy* dari nilai testing dan prediksi terhadap data uji.

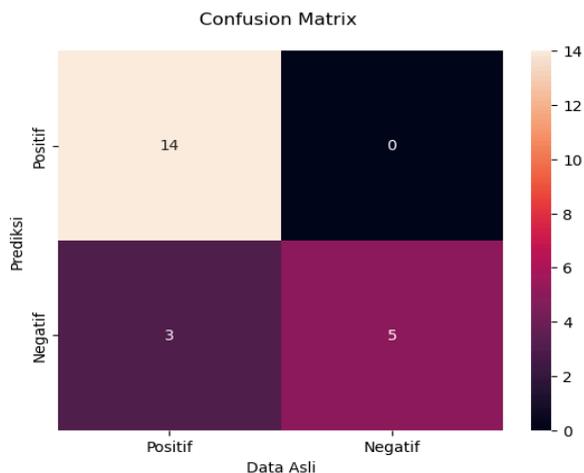
```
Training Accuracy Score : 1.0
Test Accuracy Score : 0.8636363636363636
```

	precision	recall	f1-score	support
0	0.82	1.00	0.90	14
1	1.00	0.62	0.77	8
accuracy			0.86	22
macro avg	0.91	0.81	0.84	22
weighted avg	0.89	0.86	0.85	22

```
Accuracy: 0.8636363636363636
```

Gambar 9. Hasil Klasifikasi Dataset

Pada gambar 9 menampilkan hasil *accuracy* dari data uji. Parameter yang terdapat pada code di gambar 8 adalah untuk menspesifikasikan nilai pada label apa yang ingin ditampilkan, pada kasus ini ialah “0 = *has cervical cancer*” (positif) atau “1 = *no cervical cancer*” (negatif) sesuai dengan label atau *class* pada dataset (Ca\_Cervix). Setelah dilakukan pengujian dataset penyakit ginjal kronis dengan model XGBoost yang telah dibuat, nilai *Accuracy* yang dihasilkan oleh model XGBoost sangat baik dengan nilai 0.86 atau 86%. Dari data uji 22 pasien yang divisualisasikan ke dalam bentuk *Confusion Matrix* yang diambil dari library Matplotlib dan Seaborn, dapat dilihat jumlah data yang berhasil terklasifikasi dengan benar maupun salah oleh model yang dibuat. Dikarenakan model dari kasus ini adalah sebuah klasifikasi, maka metode XGBoost yang digunakan diambil langsung dari library XGBoost yang tersedia, yaitu dengan fungsi *XGBClassifier*.



Gambar 10. Matrix Hasil Uji Model

Visualisasi *Confusion Matrix* ini dibuat guna mempermudah membaca tabel tersebut. Pada gambar 10 dapat dilihat data pasien positif yang terklasifikasi dengan benar atau termasuk kategori *True Positive* (TP) sebanyak 14 data. Kemudian data pasien negatif yang yang terklasifikasi dengan benar atau termasuk kategori *True Negative* (TN) sebanyak 5 data. Sebaliknya, data pasien negatif yang salah terklasifikasi sebagai pasien positif atau termasuk kategori *False Positive* (FP) berjumlah 0 data. Sedangkan data pasien positif yang salah terklasifikasi sebagai pasien negatif atau *False Negative* (FN) berjumlah 3. Sedangkan jika menggunakan *Random Forest Decision* didapatkan hasil dari *F1 Score*: 0.9174603174603173, *precision*: 0.9475308641975307 dan *recall*: 0.9444444444444444 yang dimana hasil tersebut berada di bawah *XGBoost*.

#### IV. Kesimpulan

Evaluasi model ini didapatkan perolehan nilai *accuracy* sebesar 86%. Perhitungan *Accuracy* digunakan untuk mencari seberapa tepat model melakukan prediksi dari masing-masing instance, *Metric Precision* digunakan untuk mencari nilai model klasifikasi dalam mengembalikan instance yang relevan nilai *precision* pada penelitian 100%, *Metric Recall* digunakan untuk mengetahui seberapa baik model “mengingat” atau mengidentifikasi nilai instance yang relevan nilai *recall* pada penelitian ini 82% dan *F1-Score* atau bisa disebut juga *F Measure* merupakan nilai rata-rata perbandingan dan keseimbangan dari *Precision* dan *Recall* *f1-score* 90% pada model *XGBoost* sehingga dapat disimpulkan bahwa *XGBoost* sangat baik dalam melakukan klasifikasi penyakit kanker serviks. Kesimpulan dari evaluasi model ini ialah dengan perolehan nilai *Accuracy* sebesar 86%, *Precision* 100%, *Recall* 82% dan *F1-Score* 90% pada model *XGBoost*, dapat disimpulkan bahwa *XGBoost* sangat baik dalam melakukan klasifikasi penyakit kanker serviks. Data tersebut jika dikomparasikan dengan algoritma *Random Forest Decision* didapatkan hasil dari *F1 Score*: 0.9174603174603173, *precision*: 0.9475308641975307 dan *recall*: 0.9444444444444444 yang dimana hasil tersebut berada di bawah *XGBoost*. Implementasi performa Algoritma *XGBoost* dapat lebih optimal jika *database* yang digunakan dalam jumlah yang banyak.

#### Daftar Pustaka

- [1] T. G. Pratama, A. Ridwan, and A. Prihandono, “Penerapan Algoritma C4. 5 untuk Klasifikasi Kanker Serviks Tingkat Awal,” *Flurecol Journal. Part E Eng.*, vol. 1, no. 1, pp. 1–6, 2021.
- [2] G. F. Ramadhan and E. Winarno, “Sistem Diagnosa Penyakit Ikan Menggunakan Metode Case Based Reasoning Dengan Algoritma Similaritas Sorgenfrei dan K-Nearest Neighbor,” *J. Ilm. Inform.*, vol. 10, no. 01, pp. 44–50, 2022.
- [3] R. A. Sihombing and F. Natsir, “Sistem Pakar Untuk Diagnosis Gangguan Pada Sistem Endokrin Berbasis Android,” *J. Sist. Inf. dan Teknol. Perad.*, vol. 2, no. 1, pp. 42–47, 2021.
- [4] R. A. Sihombing, F. Natsir, T. Indormatika, and S. Pakar, “Implementasi Metode Dempster Shafer dalam Sistem Pakar untuk Diagnosis Penyakit Kucing berbasis Android,” *J. Apl. Teknol. Inf. dan Manaj.*, vol. 4, no. 1, 2023.
- [5] N. M. Farhan and B. Setiaji, “Analisis Perbandingan Performa Algoritma *XGBoost* dan *LightGBM* pada Klasifikasi Kanker Payudara,” *Indones. J. Comput. Sci.*, vol. 12, no. 2, pp. 284–301, 2023,

- [Online]. Available: <http://ijcs.stmikindonesia.ac.id/ijcs/index.php/ijcs/article/view/3135>
- [6] A. Handayani, "Evaluasi Algoritma Extreme Gradient Boosting (XGBoost), Support Vector Machine (SVM) Dan Multi-Layer Perceptron (MLP) untuk Klasifikasi Jenis Tumor Payudara." Fakultas Teknik UAI, 2017.
- [7] R. D. Marzuq, S. A. Wicaksono, and N. Y. Setiawan, "Prediksi Kanker Paru-Paru menggunakan Algoritme Random Forest Decision Tree," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 7, no. 7, pp. 3448–3456, 2023.
- [8] M. R. Givari, M. R. Sulaeman, and Y. Umidah, "Perbandingan Algoritma SVM, Random Forest Dan XGBoost Untuk Penentuan Persetujuan Pengajuan Kredit," *Nuansa Inform.*, vol. 16, no. 1, pp. 141–149, 2022, doi: 10.25134/nuansa.v16i1.5406.
- [9] U. Rianantika, "Implementasi Metode Similarity untuk Pendukung Keputusan Diagnosis Kanker Serviks," 2013.
- [10] D. Kurnia, M. I. Mazdadi, D. Kartini, R. A. Nugroho, and F. Abadi, "Seleksi Fitur dengan Particle Swarm Optimization pada Klasifikasi Penyakit Parkinson Menggunakan XGBoost," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 10, no. 5, pp. 1083–1094, 2023.
- [11] R. J. Alfirdausy and S. Bahri, "Implementasi Algoritma K-Nearest Neighbor untuk Klasifikasi Diagnosis Penyakit Alzheimer," *Techno. Com*, vol. 22, no. 3, pp. 635–642, 2023.
- [12] N. M. Putry and B. N. Sari, "Komparasi Algoritma Knn Dan Naïve Bayes Untuk Klasifikasi Diagnosis Penyakit Diabetes Mellitus. EVOLUSI: Jurnal Sains Dan Manajemen, 10 (1)." 2022.
- [13] D. Septhya *et al.*, "Implementasi Algoritma Decision Tree dan Support Vector Machine untuk Klasifikasi Penyakit Kanker Paru: Implementation of Decision Tree Algorithm and Support Vector Machine for Lung Cancer Classification," *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 3, no. 1, pp. 15–19, 2023.
- [14] G. R. Fernandes, D. Wiguna, F. Natsir, Triyadi, N. Suwela, and A. Birowo, "Implementation of Face Detection to Count the Number of Mall Visitors," *2022 IEEE 8th Int. Conf. Comput. Eng. Des. ICCED 2022*, pp. 0–4, 2022, doi: 10.1109/ICCED56140.2022.10010472.
- [15] Y. Purbolingga, D. Marta, A. Rahmawatia, and B. Wajhi, "Perbandingan Algoritma CatBoost dan XGBoost dalam Klasifikasi Penyakit Jantung," *J. APTEK Vol. 15 No 2 126-133*, vol. 15, no. 2, pp. 126–133, 2023, [Online]. Available: <http://journal.upp.ac.id/index.php/aptek/article/download/1930/1163/4970>
- [16] G. Abdurrahman, H. Oktavianto, and M. Sintawati, "Optimasi Algoritma XGBoost Classifier Menggunakan Hyperparameter Gridesearch dan Random Search Pada Klasifikasi Penyakit Diabetes," *INFORMAL Informatics J.*, vol. 7, no. 3, p. 193, 2022, doi: 10.19184/isj.v7i3.35441.