

IF3270 Pembelajaran Mesin

# **PENGEMBANGAN MODEL PEMBELAJARAN MESIN UNTUK PREDIKSI PENYAKIT DIABETES**

Laporan Praktikum

Disusun untuk memenuhi tugas mata kuliah IF3270 Pembelajaran Mesin  
pada Semester 2 (dua) Tahun Akademik 2023/2024



Oleh

Chiquita Ahsanunnisa	13521129
Rava Maulana Azzikri	13521149

**PROGRAM STUDI TEKNIK INFORMATIKA  
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA  
INSTITUT TEKNOLOGI BANDUNG  
BANDUNG**

**2024**

## DAFTAR ISI

<b>DAFTAR ISI</b>	<b>1</b>
<b>1 HASIL ANALISIS DATA</b>	<b>3</b>
1.1 Deskripsi Data	3
1.2 Analisis Data dan Penanganannya	4
1.2.1 Data Duplikat	4
1.2.2 Missing Value	5
1.2.3 Outlier	5
1.2.4 Balance of Data	7
<b>2 PENANGANAN DATA DAN JUSTIFIKASI</b>	<b>8</b>
2.1 Penanganan Data Duplikat	8
2.2 Penanganan Missing Value	8
2.3 Penanganan Outlier	8
2.4 Penanganan Imbalanced Dataset	8
2.5 Penanganan Lainnya	8
<b>3 PERUBAHAN JAWABAN BAGIAN 1</b>	<b>9</b>
3.1 Nomor 2	9
3.2 Nomor 3	9
3.3 Nomor 5	9
<b>4 DESAIN EKSPERIMEN</b>	<b>9</b>
4.1 Tujuan	9
4.2 Variabel	10
4.2.1 Variabel Independen	10
4.2.2 Variabel Dependen	10
4.3 Metrik Kinerja Model	11
4.4 Desain Eksperimen	11
4.4.1 Strategi Eksperimen	11
4.4.2 Skema Eksperimen	11
<b>5 HASIL EKSPERIMEN</b>	<b>12</b>
5.1 K-Fold Cross Validation	12
5.1.1 Model Logistic Regression	12
5.1.2 Model Random Forest	12
5.1.3 Model Multilayer Perceptron (MLP)	12
5.1.4 Model XGBoost	13
5.1.5 Model CatBoost	13
5.2 Grid Search	14
<b>6 ANALISIS HASIL EKSPERIMEN</b>	<b>15</b>
<b>7 SIMPULAN</b>	<b>16</b>
<b>8 PEMBAGIAN TUGAS</b>	<b>17</b>

# 1 HASIL ANALISIS DATA

## 1.1 Deskripsi Data

Dataset yang diberikan merupakan dataset terstruktur yang terdiri atas 50736 baris dan 20 fitur. Berikut merupakan detail fitur yang ada pada dataset.

Tabel 1.1.1 Deskripsi Data Setiap Fitur

No.	Fitur	Deskripsi	Tipe Data
1	HighBP	Memiliki tekanan darah tinggi (BP: Blood Pressure) atau tidak	Kualitatif, kategorikal, biner
2	HighChol	Kolesterol tinggi atau tidak	Kualitatif, kategorikal, biner
3	BMI	Besaran Body Mass Index	Kuantitatif, rasio
4	Smoker	Perokok atau bukan perokok	Kualitatif, kategorikal, biner
5	Stroke	Pernah mengalami struk atau tidak	Kualitatif, kategorikal, biner
6	HeartDiseaseorAttack	Memiliki riwayat penyakit antara jantung koroner dan serangan jantung atau tidak sama sekali	Kualitatif, kategorikal, biner
7	PhysActivity	Aktif secara fisik dalam 30 hari terakhir atau tidak	Kualitatif, kategorikal, biner
8	Fruits	Mengonsumsi buah setiap hari atau tidak	Kualitatif, kategorikal, biner
9	Veggies	Mengonsumsi sayur setiap hari atau tidak	Kualitatif, kategorikal, biner
10	HvyAlcoholConsump	Peminum berat alkohol atau bukan	Kualitatif, kategorikal, biner
11	AnyHealthcare	Memiliki perlindungan kesehatan atau tidak, contohnya memiliki asuransi kesehatan	Kualitatif, kategorikal, biner
12	GenHlth	Evaluasi mandiri terhadap kesehatan, skala 1-5 (1: Sangat baik, 2: Cukup Baik, 3: Baik, 4: Biasa saja, 5: Buruk)	Kualitatif, ordinal
13	MentHlth	Jumlah hari keadaan mental buruk dalam 30 hari terakhir (skala 0-30 hari)	Kuantitatif, rasio
14	PhysHlth	Jumlah hari keadaan fisik buruk dalam 30 hari terakhir (skala 0-30 hari)	Kuantitatif, rasio
15	DiffWalk	Memiliki kesulitan berjalan atau menaiki tangga	Kualitatif, kategorikal, biner
16	Sex	(M) Male atau (F) Female	Kualitatif, kategorikal, biner
17	Age	13 kategori umur (1: 18-24 tahun, 9: 60-64 tahun, 13: 80 tahun ke atas)	Kualitatif, ordinal

18	Education	Level edukasi skala 1-6 (1: Tidak pernah sekolah atau hanya TK, 2: SD, dst)	Kualitatif, ordinal
19	Income	Skala pendapatan 1-8	Kualitatif, ordinal
20	Diabetes	Apakah mengalami diabetes atau tidak (Fitur target)	Kualitatif, kategorikal, biner

Dalam representasi Python-nya, berikut merupakan tipe data yang tersimpan pada *notebook*. Beberapa data kategorikal yang nilainya biner disimpan dalam tipe *float*, dengan nilai 1.0 untuk True dan 0.0 untuk False. Selain itu, masih ada fitur kategorikal biner yang masih bertipe *object*, yaitu fitur “Sex”.

```

HighBP          float64
HighChol        float64
BMI             float64
Smoker          float64
Stroke          float64
HeartDiseaseorAttack float64
PhysActivity     float64
Fruits          float64
Veggies         float64
HvyAlcoholConsump float64
AnyHealthcare   float64
GenHlth         float64
MentHlth        float64
PhysHlth        float64
DiffWalk        float64
Sex             object
Age            float64
Education       float64
Income          float64
Diabetes        bool
dtype: object

```

Gambar 1.1.1 Tipe Data Setiap Fitur

Karena fitur “Sex” yang masih bertipe *object*, perlu dilakukan *encoding* agar model dapat dibangun. Fitur ini bertipe kategorikal, dengan dua nilai yang mungkin, yaitu "M" (*male*) dan "F" (*female*). Oleh karena itu, dilakukan *label encoding* dengan meng-*encode* nilai "M" menjadi 0 dan "F" menjadi 1.

## 1.2 Analisis Data dan Penanganannya

Di awal, data telah dibagi menjadi tiga bagian, yaitu data latih, data validasi, dan data uji. Oleh karena itu, analisis data dilakukan secara terpisah untuk ketiga bagian data tersebut.

### 1.2.1 Data Duplikat

Analisis terhadap data duplikat dilakukan untuk menemukan nilai duplikat pada data. Hal ini dilakukan karena data duplikat tidak memberikan wawasan tambahan. Data yang duplikat bahkan terkadang memberikan pengaruh negatif untuk model yang dibangun.

Berdasarkan analisis yang telah dilakukan, terdapat 1135 nilai duplikat pada data latih, 98 nilai duplikat pada data validasi, dan 155 nilai duplikat pada data uji.

```
Duplicate values (train set): 1135
Duplicate values (validation set): 98
Duplicate values (test set): 155
```

Gambar 1.2.1.1 Data Duplikat

## 1.2.2 Missing Value

Analisis terhadap *missing value* dilakukan untuk menemukan jumlah data yang hilang untuk setiap fitur pada dataset. Hal ini dilakukan karena *missing value* memerlukan penanganan khusus karena tidak semua model memiliki teknik bawaan untuk menerima *missing value* pada data.

Berdasarkan analisis yang telah dilakukan, tidak ditemukan data yang hilang untuk setiap fitur pada setiap bagian data.

Missing values (train set):	Missing values (validation set):	Missing values (test set):	
HighBP	0	HighBP	0
HighChol	0	HighChol	0
BMI	0	BMI	0
Smoker	0	Smoker	0
Stroke	0	Stroke	0
HeartDiseaseorAttack	0	HeartDiseaseorAttack	0
PhysActivity	0	PhysActivity	0
Fruits	0	Fruits	0
Veggies	0	Veggies	0
HvyAlcoholConsump	0	HvyAlcoholConsump	0
AnyHealthcare	0	AnyHealthcare	0
GenHlth	0	GenHlth	0
MentHlth	0	MentHlth	0
PhysHlth	0	PhysHlth	0
DiffWalk	0	DiffWalk	0
Sex	0	Sex	0
Age	0	Age	0
Education	0	Education	0
Income	0	Income	0
Diabetes	0	Diabetes	0
dtype: int64	dtype: int64	dtype: int64	

Gambar 1.2.2.1 Missing Value

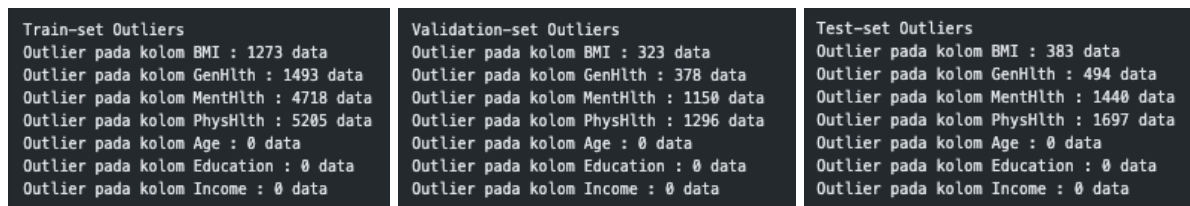
## 1.2.3 Outlier

Analisis terhadap *outlier* dilakukan untuk menemukan data yang mempunyai nilai berbeda dengan data-data lainnya. *Outlier* dapat menandakan sebuah anomali pada data. Ada beberapa metode yang dapat digunakan untuk mengidentifikasi *outlier* pada data, salah satunya—dan yang digunakan saat ini—yaitu metode *interquartile range* (IQR).

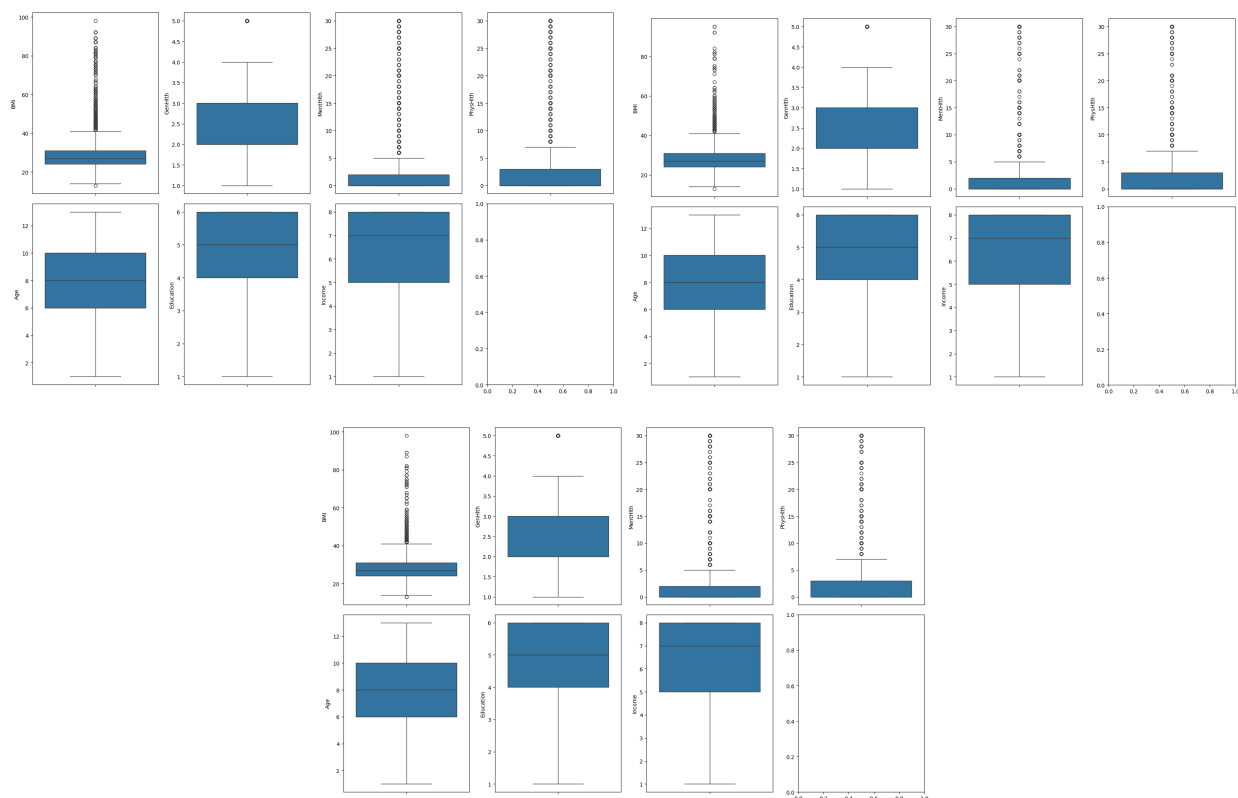
Pada metode IQR, ditentukan sebuah batas bawah dan batas atas. Batas bawah dihitung dengan rumus  $Q1 - 1.5IQR$  dan batas atas dihitung dengan rumus  $Q3 + 1.5IQR$ , dengan  $Q1$  adalah kuartil 1 dari data,  $Q3$  adalah kuartil 3 dari data, dan  $IQR$  adalah selisih antara kuartil 3 dan kuartil 1. Data yang bernilai di bawah batas bawah atau di atas batas atas dikategorikan sebagai *outlier*.

Metode IQR dapat divisualisasikan dengan *box plot*. Pada *box plot*, ditampilkan batas bawah dan batas atas dari sebuah fitur. *Outlier* berarti titik-titik yang berada di atas batas atas atau di bawah batas bawah pada *box plot*.

Berdasarkan analisis yang dilakukan, untuk setiap bagian data, ditemukan empat fitur yang memiliki *outlier*, yaitu BMI, GenHlth, MentHlth, dan PhysHlth. Berikut merupakan detailnya.



Gambar 1.2.3.1 *Outlier* pada Data Latih, Data Validasi, dan Data Uji

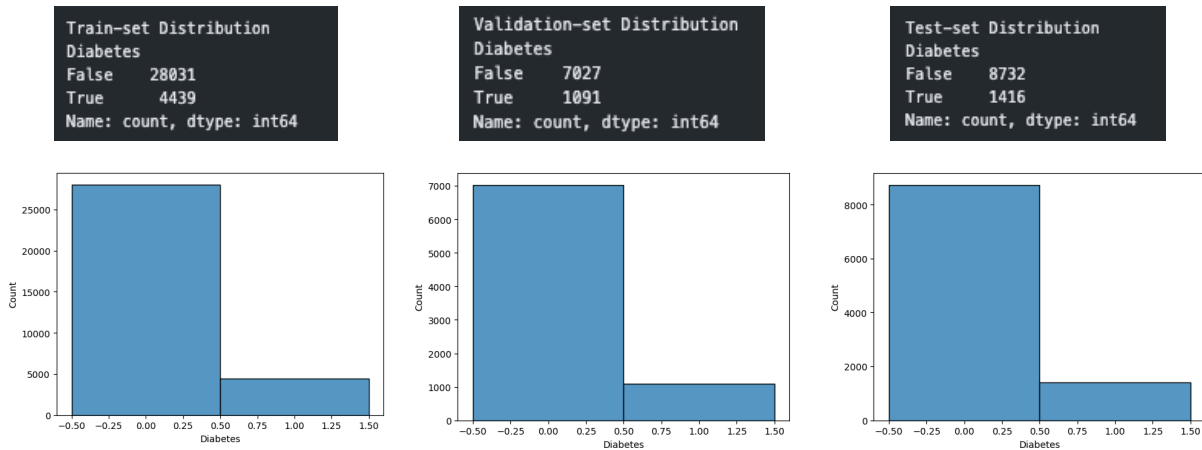


Gambar 1.2.3.2 Visualisasi *Outlier* pada Data Latih, Data Validasi, dan Data Uji

### 1.2.4 Balance of Data

Analisis terhadap keseimbangan data dilakukan untuk menentukan keseimbangan distribusi dari kelas target. Jika distribusi kelas target tidak seimbang (*imbalanced*), diperlukan penanganan khusus agar model tidak bias ke kelas dengan distribusi lebih tinggi (kelas mayoritas).

Berdasarkan analisis yang dilakukan, ditemukan bahwa dataset tidak seimbang untuk setiap bagian data, dengan kelas mayoritas bernilai False (tidak mengidap diabetes).



Gambar 1.2.4.1 Distribusi Fitur Target untuk Data Latih, Data Validasi, dan Data Uji

## 2 PENANGANAN DATA DAN JUSTIFIKASI

### 2.1 Penanganan Data Duplikat

Data yang duplikat dapat langsung dihapus saja. Hal ini bertujuan untuk menghasilkan proses pelatihan model yang lebih efisien.

```
df_train.drop_duplicates(inplace=True)
df_val.drop_duplicates(inplace=True)
```

Gambar 2.1.1 Penanganan Data Duplikat

### 2.2 Penanganan *Missing Value*

Karena tidak ada *missing value*, tidak ada penanganan khusus yang perlu dilakukan di sini.

### 2.3 Penanganan *Outlier*

Berdasarkan analisis lebih lanjut, ternyata *outlier* justru banyak ditemukan pada data dengan label kelas target True (data orang yang mengidap diabetes). Padahal, berdasarkan analisis keseimbangan data, kelas ini termasuk kelas minoritas. Oleh karena itu, *outlier* pada data ini tidak perlu dihilangkan karena akan menghilangkan data kelas untuk pengidap diabetes.

### 2.4 Penanganan *Imbalanced Dataset*

Untuk permasalahan ini, dilakukan *oversampling* dengan metode *synthetic minority over sampling* (SMOTE), yang melakukan *oversampling* kelas minoritas dengan memanfaatkan graf k-NN-nya. Hal ini dilakukan karena *oversampling* tidak membuang data atau informasi yang sudah ada.

```
X_train, y_train = SMOTE(random_state=42).fit_resample(df_train_kf.drop(["Diabetes"], axis=1), df_train_kf["Diabetes"])
```

Gambar 2.4.1 Penanganan *Imbalanced Dataset*

### 2.5 Penanganan Lainnya

Sebelum data digunakan, dilakukan *feature scaling*. Hal ini bertujuan untuk menstandarisasi nilai-nilai yang ada sehingga pembangunan model dapat dilakukan secara efektif dan efisien.

```
X_train_scaled = scaler.fit_transform(X_train)
X_val_scaled = scaler.transform(X_val)
```

Gambar 2.5.1 *Feature Scaling*



### **3 PERUBAHAN JAWABAN BAGIAN 1**

#### **3.1 Nomor 2**

Terdapat kesalahan pengetikan pada deskripsi bagian analisis duplikat data. Deskripsi yang seharusnya sesuai pada laporan ini.

#### **3.2 Nomor 3**

Setelah dilakukan analisis lebih lanjut, penanganan yang dipilih sesuai pada laporan ini.

#### **3.3 Nomor 5**

Ditambahkan percobaan untuk model XGBoost dan CatBoost.

### **4 DESAIN EKSPERIMEN**

#### **4.1 Tujuan**

Tujuan dari eksperimen ini yaitu membuat model *classifier* yang dapat memprediksi (mengklasifikasikan) apakah seseorang mengidap diabetes atau tidak berdasarkan data-data:

1. Kepemilikan tekanan darah tinggi
2. Kepemilikan kolesterol tinggi
3. Body Mass Index (BMI)
4. Status perokok
5. Riwayat struk
6. Riwayat penyakit jantung koroner dan serangan jantung
7. Keaktifan fisik dalam 30 hari terakhir
8. Konsumsi buah setiap hari
9. Konsumsi sayur setiap hari
10. Konsumsi alkohol berat
11. Kepemilikan perlindungan kesehatan
12. Evaluasi mandiri terhadap kesehatan
13. Jumlah hari keadaan mental buruk dalam 30 hari terakhir
14. Jumlah hari keadaan fisik buruk dalam 30 hari terakhir
15. Kesulitan berjalan
16. Jenis kelamin
17. Usia
18. Tingkat edukasi

## 19. Pendapatan

### 4.2 Variabel

#### 4.2.1 Variabel Independen

Terdapat 19 variabel independen pada eksperimen ini. Variabel tersebut berupa fitur-fitur berikut.

1. HighBP: Memiliki tekanan darah tinggi (BP: Blood Pressure) atau tidak
2. HighChol: Kolesterol tinggi atau tidak
3. BMI: Besaran Body Mass Index
4. Smoker: Perokok atau bukan perokok
5. Stroke: Pernah mengalami stroke atau tidak
6. HeartDiseaseorAttack: Memiliki riwayat penyakit antara jantung koroner dan serangan jantung atau tidak sama sekali
7. PhysActivity: Aktif secara fisik dalam 30 hari terakhir atau tidak
8. Fruits: Mengonsumsi buah setiap hari atau tidak
9. Veggies: Mengonsumsi sayur setiap hari atau tidak
10. HvyAlcoholConsump: Peminum berat alkohol atau bukan
11. AnyHealthcare: Memiliki perlindungan kesehatan atau tidak, contohnya memiliki asuransi kesehatan
12. GenHlth: Evaluasi mandiri terhadap kesehatan, skala 1-5 (1: Sangat baik, 2: Cukup Baik, 3: Baik, 4: Biasa saja, 5: Buruk)
13. MentHlth: Jumlah hari keadaan mental buruk dalam 30 hari terakhir (skala 0-30 hari)
14. PhysHlth: Jumlah hari keadaan fisik buruk dalam 30 hari terakhir (skala 0-30 hari)
15. DiffWalk: Memiliki kesulitan berjalan atau menaiki tangga
16. Sex: (M) Male atau (F) Female
17. Age: 13 kategori umur (1: 18-24 tahun, 9: 60-64 tahun, 13: 80 tahun ke atas)
18. Education: Level edukasi skala 1-6 (1: Tidak pernah sekolah atau hanya TK, 2: SD, dst)
19. Income: Skala pendapatan 1-8

Fitur-fitur di atas merupakan fitur nontarget.

#### 4.2.2 Variabel Dependen

Variabel dependen pada eksperimen ini yaitu fitur “Diabetes”, yang menandakan apakah seseorang mengidap diabetes atau tidak. Fitur ini merupakan fitur target.

### 4.3 Metrik Kinerja Model

*Task* yang dilakukan pada eksperimen ini adalah *task* klasifikasi. Oleh karena itu, metrik kinerja model yang dipilih yaitu akurasi, presisi, *recall*, dan skor F1.

### 4.4 Desain Eksperimen

#### 4.4.1 Strategi Eksperimen

Dibangun model-model sebagai berikut.

1. Model *baseline*, yaitu *logistic regression*.
2. Model lainnya, yaitu *random forest*, *artificial neural network* (ANN) atau *multilayer perceptron* (MLP), XGBoost, dan CatBoost.

Setelah itu, dilakukan skema validasi untuk mencari model yang paling baik. Lalu, dilakukan *grid search* untuk mendapatkan *hyperparameter* terbaik untuk model terbaik.

```
grid_search = GridSearchCV(estimator=MLPClassifier(hidden_layer_sizes=(4,2)), param_grid=param_grid, n_jobs=-1, cv=3, scoring="f1_macro")
grid_search.fit(X_train_scaled, y_train)
```

Gambar 4.4.1.1 *Grid Search*

#### 4.4.2 Skema Eksperimen

Skema eksperimen yang digunakan adalah *k-fold cross validation*. Dataset dibagi menjadi data latih dan data uji. Lalu, untuk setiap model, dilakukan *k-fold cross validation* dengan cara mempartisi data train menjadi k partisi, lalu mengambil 1 partisi sebagai data validasi dan k-1 partisi sebagai data latih. Lalu, setelah dihitung, akan didapatkan model dengan performa paling baik. Model tersebut akan dites terhadap data uji. Pada eksperimen ini, dipilih nilai k (jumlah partisi) sebanyak lima partisi.

```
kf = KFold(n_splits=5, shuffle=True, random_state=42)
```

Gambar 4.4.2.1 *K-Fold Cross Validation*

Namun, di bagian awal, dataset telah dipisahkan menjadi data latih, data validasi, dan data uji, yang tidak sesuai dengan skema *k-fold cross validation*. Oleh karena itu, nantinya, data validasi dan data latih akan digabungkan menjadi data latih.

```
df_train = pd.concat([df_train, df_val])
```

Gambar 4.4.2.2 Penyatuan Data Latih dan Data Validasi

## 5 HASIL EKSPERIMEN

### 5.1 K-Fold Cross Validation

#### 5.1.1 Model Logistic Regression

Berikut merupakan hasil *k-fold cross validation* untuk model *logistic regression*.

FOLD 1					FOLD 2					FOLD 3							
		precision	recall	f1-score	support			precision	recall	f1-score	support			precision	recall	f1-score	support
	False	0.95	0.74	0.83	6724		False	0.95	0.75	0.83	6819		False	0.94	0.73	0.82	6743
	True	0.33	0.76	0.46	1147		True	0.31	0.73	0.43	1052		True	0.31	0.72	0.44	1128
accuracy				0.74	7871	accuracy				0.74	7871	accuracy				0.73	7871
macro avg		0.64	0.75	0.64	7871	macro avg		0.63	0.74	0.63	7871	macro avg		0.63	0.73	0.63	7871
weighted avg		0.86	0.74	0.77	7871	weighted avg		0.86	0.74	0.78	7871	weighted avg		0.85	0.73	0.77	7871

FOLD 4					FOLD 5						
		precision	recall	f1-score	support			precision	recall	f1-score	support
	False	0.95	0.74	0.83	6749		False	0.94	0.73	0.82	6798
	True	0.33	0.75	0.45	1122		True	0.30	0.72	0.42	1073
accuracy				0.74	7871	accuracy				0.73	7871
macro avg		0.64	0.75	0.64	7871	macro avg		0.62	0.72	0.62	7871
weighted avg		0.86	0.74	0.78	7871	weighted avg		0.85	0.73	0.77	7871

Gambar 5.1.1.1 K-Fold Cross Validation Model Logistic Regression

#### 5.1.2 Model Random Forest

Berikut merupakan hasil *k-fold cross validation* untuk model *random forest*.

FOLD 1						FOLD 2						FOLD 3					
		precision	recall	f1-score	support			precision	recall	f1-score	support			precision	recall	f1-score	support
	False	0.88	0.96	0.92	6724		False	0.88	0.96	0.92	6819		False	0.88	0.96	0.92	6743
	True	0.49	0.20	0.29	1147		True	0.42	0.18	0.25	1052		True	0.45	0.18	0.26	1128
accuracy				0.85	7871	accuracy				0.86	7871	accuracy				0.85	7871
macro avg		0.68	0.58	0.60	7871	macro avg		0.65	0.57	0.58	7871	macro avg		0.66	0.57	0.59	7871
weighted avg		0.82	0.85	0.83	7871	weighted avg		0.82	0.86	0.83	7871	weighted avg		0.81	0.85	0.82	7871

FOLD 4						FOLD 5					
		precision	recall	f1-score	support			precision	recall	f1-score	support
	False	0.88	0.96	0.92	6749		False	0.88	0.96	0.92	6798
	True	0.46	0.19	0.27	1122		True	0.40	0.18	0.25	1073
accuracy				0.85	7871	accuracy				0.85	7871
macro avg		0.67	0.58	0.59	7871	macro avg		0.64	0.57	0.58	7871
weighted avg		0.82	0.85	0.83	7871	weighted avg		0.82	0.85	0.83	7871

Gambar 5.1.2.1 K-Fold Cross Validation Model Random Forest

#### 5.1.3 Model Multilayer Perceptron (MLP)

Berikut merupakan hasil *k-fold cross validation* untuk model *multilayer perceptron* (MLP).

FOLD 1						FOLD 2						FOLD 3					
		precision	recall	f1-score	support			precision	recall	f1-score	support			precision	recall	f1-score	support
	False	0.93	0.77	0.84	6724		False	0.92	0.85	0.89	6819		False	0.93	0.78	0.85	6743
	True	0.34	0.69	0.45	1147		True	0.36	0.54	0.43	1052		True	0.33	0.64	0.43	1128
	accuracy			0.76	7871		accuracy			0.81	7871		accuracy			0.76	7871
	macro avg	0.64	0.73	0.65	7871		macro avg	0.64	0.70	0.66	7871		macro avg	0.63	0.71	0.64	7871
	weighted avg	0.85	0.76	0.79	7871		weighted avg	0.85	0.81	0.83	7871		weighted avg	0.84	0.76	0.79	7871

FOLD 4						FOLD 5					
		precision	recall	f1-score	support			precision	recall	f1-score	support
	False	0.94	0.77	0.85	6749		False	0.93	0.79	0.85	6798
	True	0.33	0.69	0.45	1122		True	0.32	0.63	0.42	1073
	accuracy			0.76	7871		accuracy			0.77	7871
	macro avg	0.64	0.73	0.65	7871		macro avg	0.63	0.71	0.64	7871
	weighted avg	0.85	0.76	0.79	7871		weighted avg	0.85	0.77	0.80	7871

Gambar 5.1.3.1 *K-Fold Cross Validation Model Mutlilayer Perceptron (MLP)*

## 5.1.4 Model XGBoost

Berikut merupakan hasil *k-fold cross validation* untuk model XGBoost.

FOLD 1						FOLD 2						FOLD 3					
		precision	recall	f1-score	support			precision	recall	f1-score	support			precision	recall	f1-score	support
	False	0.87	0.97	0.92	6724		False	0.88	0.97	0.92	6819		False	0.88	0.97	0.92	6743
	True	0.48	0.19	0.27	1147		True	0.45	0.18	0.25	1052		True	0.48	0.19	0.27	1128
	accuracy			0.85	7871		accuracy			0.86	7871		accuracy			0.85	7871
	macro avg	0.68	0.58	0.60	7871		macro avg	0.67	0.57	0.59	7871		macro avg	0.68	0.58	0.59	7871
	weighted avg	0.82	0.85	0.82	7871		weighted avg	0.83	0.86	0.83	7871		weighted avg	0.82	0.85	0.83	7871

FOLD 4						FOLD 5					
		precision	recall	f1-score	support			precision	recall	f1-score	support
	False	0.88	0.97	0.92	6749		False	0.88	0.96	0.92	6798
	True	0.50	0.19	0.27	1122		True	0.46	0.19	0.27	1073
	accuracy			0.86	7871		accuracy			0.86	7871
	macro avg	0.69	0.58	0.60	7871		macro avg	0.67	0.58	0.60	7871
	weighted avg	0.82	0.86	0.83	7871		weighted avg	0.83	0.86	0.83	7871

Gambar 5.1.4.1 *K-Fold Cross Validation Model XGBoost*

## 5.1.5 Model CatBoost

Berikut merupakan hasil *k-fold cross validation* untuk model CatBoost.

FOLD 1						FOLD 2						FOLD 3					
		precision	recall	f1-score	support			precision	recall	f1-score	support			precision	recall	f1-score	support
	False	0.88	0.97	0.92	6724		False	0.89	0.97	0.93	6819		False	0.88	0.97	0.92	6743
	True	0.53	0.20	0.29	1147		True	0.50	0.18	0.27	1052		True	0.49	0.19	0.27	1128
	accuracy			0.86	7871		accuracy			0.87	7871		accuracy			0.86	7871
	macro avg	0.70	0.58	0.60	7871		macro avg	0.69	0.58	0.60	7871		macro avg	0.69	0.58	0.59	7871
	weighted avg	0.83	0.86	0.83	7871		weighted avg	0.83	0.87	0.84	7871		weighted avg	0.82	0.86	0.83	7871

FOLD 4						FOLD 5					
		precision	recall	f1-score	support			precision	recall	f1-score	support
	False	0.88	0.97	0.92	6749		False	0.88	0.97	0.92	6798
	True	0.52	0.20	0.29	1122		True	0.46	0.17	0.25	1073
	accuracy			0.86	7871		accuracy			0.86	7871
	macro avg	0.70	0.58	0.60	7871		macro avg	0.67	0.57	0.58	7871
	weighted avg	0.83	0.86	0.83	7871		weighted avg	0.82	0.86	0.83	7871

Gambar 5.1.5.1 *K-Fold Cross Validation Model CatBoost*

## 5.2 Grid Search

Berdasarkan *k-fold cross validation* yang telah dilakukan untuk setiap model, dapat dilihat bahwa model yang memiliki performa terbaik (dilihat dari skor F1-nya) adalah model *multilayer perceptron* (MLP). Oleh karena itu, dilakukan *hyperparameter tuning* untuk model ini dengan metode *grid search*.

Berikut merupakan hasil untuk model *multilayer perceptron* (MLP) yang sudah diberikan *grid search*.

	precision	recall	f1-score	support
False	0.81	0.93	0.86	7587
True	0.61	0.34	0.44	2561
accuracy			0.78	10148
macro avg	0.71	0.63	0.65	10148
weighted avg	0.76	0.78	0.76	10148

Gambar 5.2.1 Hasil *Hyperparameter Tuning* pada Model *Mutlilayer Perceptron* (MLP)

## 6 ANALISIS HASIL EKSPERIMEN

Berdasarkan hasil eksperimen yang telah dilakukan, dapat dilihat bahwa model yang memiliki performa paling baik adalah model *multilayer perceptron* (MLP) dengan skor F1 sebesar 0.64–0.66 (sebelum diberikan *grid search*). Model *multilayer perceptron* yang digunakan adalah model dua layer dengan 4 *neuron* pada *layer* pertama dan 2 *neuron* pada *layer* kedua. Model ini sendiri memiliki performa yang lebih baik daripada model lainnya karena mampu menangkap non-linearitas pada data melalui fungsi aktivasi pada setiap *neuron*.

Setelah itu, dilakukan *grid search* untuk *hyperparameter tuning* pada model *multilayer perceptron* (MLP). Didapat skor F1 sebesar 0.65. Tahap *hyperparameter tuning* berhasil meningkatkan performa model sebanyak 0.01-0.02 dengan menyesuaikan parameter *alpha*, *learning\_rate\_init*, *beta\_1*, *beta\_2*, dan *learning\_rate* dari model *multilayer perceptron*.

## 7 SIMPULAN

Berdasarkan eksperimen yang telah dilakukan, dapat disimpulkan bahwa model yang paling baik di antara model yang diujikan pada eksperimen yaitu model *multilayer perceptron* (MLP) yang telah diberikan *grid search*.



## 8 PEMBAGIAN TUGAS

Berikut merupakan pembagian tugas pada praktikum ini.

Tabel 8.1 Pembagian Tugas

NIM	Nama	Tugas
13521129	Chiquita Ahsanunnisa	Baseline model, encoding, desain eksperimen, laporan
13521149	Rava Maulana Azzikri	Analisis data, penanganan data, implementasi eksperimen