BAB V
HASIL DAN PEMBAHASAN

Pada penelitian bagian hasil dan pembahasan yang sesuai dengan perancangan yang telah dibuat. Berikut penjelasan dari hasil dan pembahasan yang dilakukan.

5.1 Hasil

Sesuai perencanaan yang telah dibuat berdasarkan *flowchart* atau diagram alur terhadap pengujian algoritma, berikut pembahasan hasil yang telah dilakukan meliputi perancangan sistem, perancangan K-NN, dan perancangan *Random Forest*.

5.1.1 Hasil Perencanaan

Terdapat langkah-langkah yang dilakukan dalam perancangan, berikut langkah-langkah yang dilakukan :

1. Langkah pertama dalam perancangan sistem yaitu *import library python.* Berikut *libraries* yang digunakan dalam penelitian :
    - Pandas

      Pandas merupakan salah satu *library python* yang digunakan dalam pengerjaan dengan *dataset.* Pandas berfungsi untuk melakukan analisa, *cleaning,* eksplorasi data dan memanipulasi data (DQLab, Belajar Pandas untuk Tingkatkan Kompetensi Bahasa Pemrograman Pythonmu!, 2020).
    - Numpy

      Numpy *(Numerical Python)* merupakan salah satu *library python* yang digunakan dalam komputasi sains. Berfungsi dalam mengolah data menjadi *array, list* dan lainnya (Rohman, 2019).
    - Matplotlib

      Merupakan *library python* yang berfungsi dalam visualisasi data seperti membuat *plot* grafik (Rohman, 2019).
    - *Scikit-learn*

Scikit-learn merupakan *library* yang berfokus pada *machine learning* seperti pembuatan model data. Proses yang mencangkup dari *library* ini antara lain pemuatan, manipulasi, visualisasi, dan penanganan data (DQLab, 2020).

2. Setelah *import library,* dilakukan pemuatan *dataset* yang digunakan. *Dataset* yang digunakan didapatkan dari situs Mendeley dengan nama *Diabetes Dataset* dengan Ahlam Rashid sebagai kontributornya. Data dikumpulkan dari laboratorium di *Medical City Hospital* dan di-*publish* pada 18 Juli 2020 yang datanya diambil dari 1000 pasien dan memiliki tiga kelas. Berikut dataset yang digunakan.

Tabel 5.1 *Dataset* asli

id	no_pation	gender	age	urea	cr	hba1c	chol	tg
502	17975	F	50	4.7	46	4.9	4.2	0.9
735	34221	M	26	4.5	62	4.9	3.7	1.4
420	47975	F	50	4.7	46	4.9	4.2	0.9
680	87656	F	50	4.7	46	4.9	4.2	0.9
504	34223	M	33	7.1	46	4.9	4.9	1.0

hdl	ldl	Vldl	bmi	class
2.4	1.4	0.5	24.0	N
1.1	2.1	0.6	23.0	N
2.4	1.4	0.5	24.0	N
2.4	1.4	0.5	24.0	N
0.8	2.0	0.4	21.0	N

3. Langkah selanjutnya yaitu dilakukan *pre-processing* dan normalisasi data. Hal ini dilakukan untuk memaksimalkan pemodelan algoritma yang digunakan nantinya. Berdasarkan eksplorasi pada data yang dilakukan, perlu melakukan perubahan pada seluruh nilai pada atribut *gender* dengan nilai F atau *Female* menjadi angka 0, sementara M atau *Male* menjadi angka 1. Kemudian mengubah seluruh nilai pada atribut *class* terdapat tiga kelas yaitu N, Y, dan P dengan nilai N atau *Non-Diabetic* menjadi 0, Y atau *Diabetic* menjadi 1, dan P atau *Predict-Diabetic* menjadi 2. Hal ini dilakukan agar keseluruhan atribut dapat diproses dalam pemodelan nantinya. Setelah mengubah nilai pada atribut-atribut tersebut, dilakukan data *cleansing* untuk membuang data yang tidak lengkap. Setelah dilakukan *pre-processing,* data yang tadinya sebanyak 1000 menjadi 994 data.

Tabel 5.2 *Dataset* setelah *pre-processing*

Id	no_pation	gender	age	urea	cr	hba1c	chol	tg	hdl
502	17975	0	50	4.7	46	4.9	4.2	0.9	2.4
735	34221	1	26	4.5	62	4.9	3.7	1.4	1.1
420	47975	0	50	4.7	46	4.9	4.2	0.9	2.4
680	87656	0	50	4.7	46	4.9	4.2	0.9	2.4
504	34223	1	33	7.1	46	4.9	4.9	1.0	0.8

Ldl	vldl	bmi	class
1.4	0.5	24.0	0
2.1	0.6	23.0	0
1.4	0.5	24.0	0
1.4	0.5	24.0	0
2.0	0.4	21.0	0

Kemudian dilakukan seleksi fitur yang nantinya akan di-normalisasi agar nilai pada data yang digunakan *balance* satu dengan lainnya. Normalisasi dilakukan menggunakan *Min Max Scale* dengan skala 0 sampai 1.

Tabel 5.3 Normalisasi Fitur

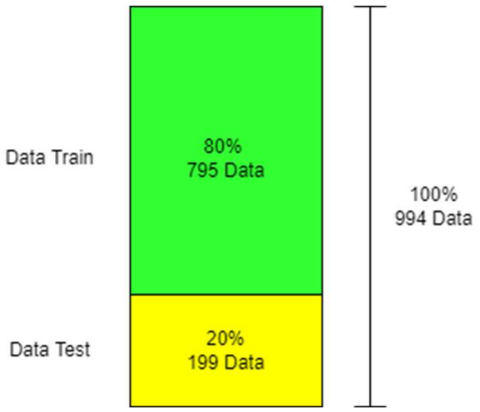
Gender	age	urea	cr	hba1c	chol	tg
0	0.508475	0.109375	0.050378	0.264901	0.407767	0.044444
1	0.101695	0.104167	0.070529	0.264901	0.359223	0.081481
0	0.508475	0.109375	0.050378	0.264901	0.407767	0.044444
0	0.508475	0.109375	0.050378	0.264901	0.407767	0.044444
1	0.220339	0.171875	0.050378	0.264901	0.475728	0.051852

Hdl	ldl	vldl	bmi
0.226804	0.114583	0.011461	0.173913
0.092784	0.187500	0.014327	0.139130
0.226804	0.114583	0.011461	0.173913
0.226804	0.114583	0.011461	0.173913
0.061856	0.177083	0.008596	0.069565

4. Setelah dilakukan *pre-processing* dan normalisasi, dilakukan pembagian data menjadi dua bagian secara acak, yaitu data *training* dan data *test.* Penelitian ini menggunakan rasio 80:20 dimana data *training* sejumlah 80% dari total keseluruhan data (994) menjadi 795 data dan data *test* 20% dari total keseluruhan data (994) menjadi 199 data.

Tabel 5.4 *Split* Data

Data	Total	Rasio	Jumlah
*Train*ing	994	80	795
Test		20	199

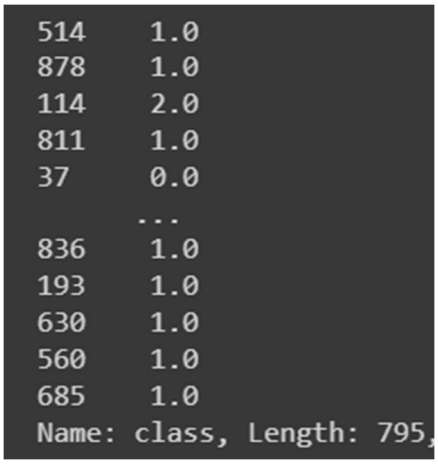


Gambar 5.1 Perbandingan *Split data*

	gender	age	urea	cr	hbalc	chol	tg	hdl	ldl	vldl	bmi
513	0.0	0.610169	0.119792	0.083123	0.476821	0.427184	0.133333	0.092784	0.229167	0.022923	0.243478
877	1.0	0.576271	0.072917	0.050378	0.602649	0.417476	0.096296	0.092784	0.177083	0.048711	0.556522
113	1.0	0.491525	0.078125	0.066751	0.337748	0.388350	0.133333	0.123711	0.166667	0.022923	0.173913
810	1.0	0.542373	0.088542	0.085642	0.715232	0.679612	0.607407	0.144330	0.114583	0.054441	0.278261
37	1.0	0.440678	0.039062	0.060453	0.298013	0.359223	0.074074	0.061856	0.218750	0.014327	0.069565
...	...	...	...	...	...	...	...	...	...	...	...
835	0.0	0.576271	0.140625	0.085642	0.761589	0.533981	0.074074	0.144330	0.145833	0.051576	0.278261
192	1.0	0.338983	0.098958	0.083123	0.403974	0.388350	0.111111	0.072165	0.218750	0.346705	0.730435
629	1.0	0.627119	0.179688	0.046599	0.536424	0.378641	0.081481	0.051546	0.239583	0.014327	0.243478
559	1.0	0.593220	0.054688	0.052897	0.721854	0.495146	0.066667	0.082474	0.343750	0.011461	0.486957
684	1.0	0.677966	0.135417	0.085642	0.781457	0.436893	0.133333	0.072165	0.156250	0.022923	0.591304

795 rows × 11 columns

Gambar 5.2 Fitur Data *Train*



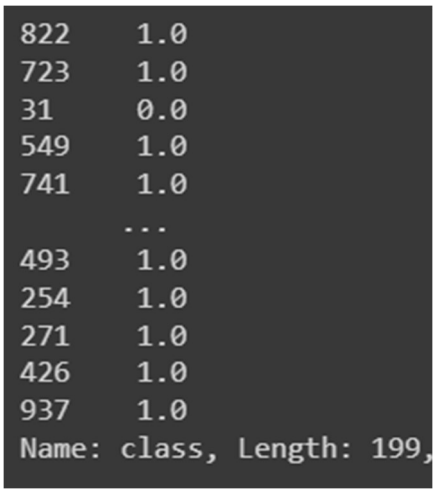
```
514    1.0
878    1.0
114    2.0
811    1.0
37     0.0
      ...
836    1.0
193    1.0
630    1.0
560    1.0
685    1.0
Name: class, Length: 795,
```

Gambar 5.3 Target Data *Train*

	gender	age	urea	cr	hba1c	chol	tg	hdl	ldl	vldl	bmi
821	1.0	0.677966	0.091146	0.090680	0.874172	0.398058	0.074074	0.164948	0.343750	0.054441	0.486957
722	1.0	0.627119	0.052083	0.061713	0.801325	0.611650	0.511111	0.103093	0.312500	0.077364	0.556522
31	1.0	0.457627	0.106771	0.061713	0.225166	0.446602	0.037037	0.072165	0.406250	0.114613	0.173913
548	1.0	0.593220	0.156250	0.083123	0.430464	0.427184	0.074074	0.082474	0.229167	0.022923	0.556522
740	1.0	0.525424	0.070313	0.069270	0.847682	0.339806	0.051852	0.072165	0.114583	0.040115	0.278261
...	...	...	...	...	...	...	...	...	...	...	...
492	1.0	0.593220	0.093750	0.100756	0.602649	0.475728	0.029630	0.103093	0.322917	0.008596	0.382609
253	0.0	0.593220	0.057292	0.057935	0.470199	0.446602	0.103704	0.082474	0.270833	0.020057	0.382609
270	1.0	0.593220	0.119792	0.090680	0.721854	0.436893	0.111111	0.092784	0.250000	0.361032	0.434783
425	1.0	0.711864	0.093750	0.071788	0.569536	0.349515	0.355556	0.072165	0.229167	0.022923	0.382609
936	1.0	0.593220	0.166667	0.093199	0.589404	0.708738	0.066667	0.134021	0.135417	0.011461	0.347826

199 rows × 11 columns

Gambar 5.4 Fitur Data *Test*



```
822    1.0
723    1.0
31     0.0
549    1.0
741    1.0
      ...
493    1.0
254    1.0
271    1.0
426    1.0
937    1.0
Name: class, Length: 199,
```

Gambar 5.5 Target Data *Test*

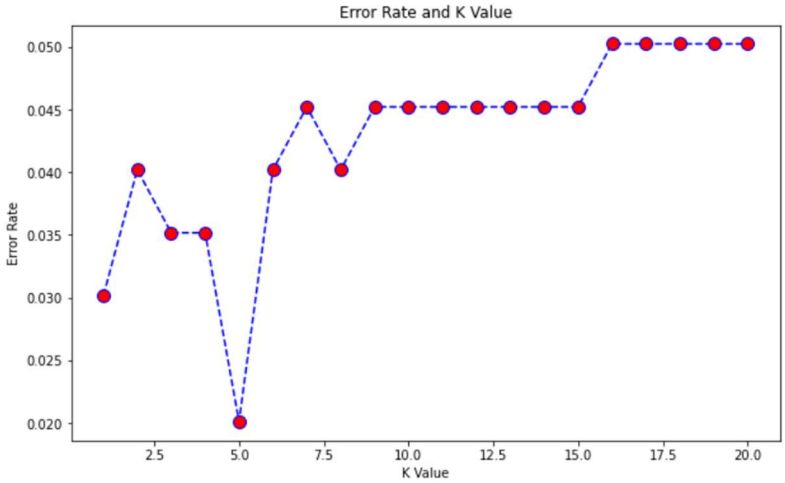
5.1.2 Hasil *K-Nearest Neighbors*

Masuk ke tahapan pemodelan algoritma *machine learning,* yang pertama yaitu KNN. Sebelum masuk ke tahapan pembuatan model, dilakukan *elbow method* dalam menentukan K-Optimal yang akan digunakan untuk menjadi K atau *nearest neighbors* pada algoritma.

Tabel 5.5 Nilai *error* pada 20 *clusters*

n	*Error*
1	0.03015
2	0.0402
3	0.03518
4	0.03518
5	0.0201
6	0.0402
7	0.04523
8	0.0402
9	0.04523
10	0.04523
11	0.04523
12	0.04523
13	0.04523
14	0.04523
15	0.04523
16	0.05025
17	0.05025
18	0.05025
19	0.05025
20	0.05025

Tabel 5.5 menunjukan nilai *error* yang dihasilkan dari pengujian sebanyak 20 kali pemodelan. Kemudian ditunjukan pada gambar 5.6 terhadap nilai-nilai yang didapatkan.



Gambar 5.6 Grafik *Elbow Method*

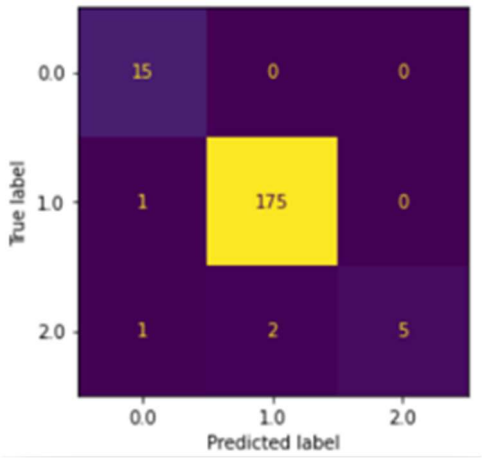
Pada gambar, didapatkan nilai *error* paling rendah yaitu pada K = 5 dengan nilai 0.0201. Langkah selanjutnya yaitu pembuatan model dari data yang telah dimuat, dilakukan *pre-processing.* Normalisasi, dan *splitting* data dengan nilai K-Optimal yang telah didapatkan dari *elbow method*. Namun untuk mendukung bahwa K = 5 lebih akurat dibandingkan lainnya, maka dilakukan pengujian sebenyak empat kali terhadap empat nilai K dari 4 s/d 7 dalam pengujian lebih lanjut. Setelah dilakukan pemodelan, hasil evaluasi dari prediksi dituangkan sesuai dengan metode yang telah ditentukan dalam bentuk tabel sebagai berikut.

Tabel 5.6 Hasil evaluasi prediksi KNN

n	*Class*	*Precision*	*Recall*
4	0	0.83	1.00
	1	0.98	0.98
	2	0.80	0.50
	Accuracy	0.96	
5	0	0.88	1.00
	1	0.99	0.99
	2	1.00	0.62
	Accuracy	0.98	
6	0	0.79	1.00
	1	0.98	0.98
	2	0.75	0.38
	Accuracy	0.96	
7	0	0.79	0.88
	1	0.98	0.98

	2	0.60	0.46
	Accuracy	0.95	

Berdasarkan tabel 5.6, nilai K = 5 memiliki nilai angka akurasi yang tertinggi mencapai sebesar 98%.



Gambar 5.7 *Confusion matrix* pada model KNN

Kemudian berdasarkan gambar 5.7 yang merupakan evaluasi *confusion matrix* terhadap model KNN, dilakukan penelusuran lebih lanjut terhadap data *test* pada kelas prediksi dan aktual. Berdasarkan matriks tersebut berdasarkan penjelasan pada sub-sub-bab 2.2.7 tentang *confusion matrix,* didapatkan nilai TP, FN, FP, dan TN serta pengukuran dari presisi dan *recall* dari masing-masing kelas serta pengukuran dari nilai akurasi dan presisinya.

Nilai-nilai dari kelas 0 *(non-diabetic)* antara lain yaitu :

1. TP = 15 data
2. FN = 0 data
3. FP = 2 data
4. TN = 182 data
5. Presisi = 88%
6. Recall = 100%

Nilai-nilai dari kelas 1 *(diabetic)* antara lain yaitu :

1. TP = 175 data
2. FN = 1 data
3. FP = 2 data
4. TN = 22 data
5. Presisi = 99%
6. Recall = 99%

Nilai-nilai dari kelas 2 *(predict-diabetic)* antara lain yaitu :

1. TP = 5 data
2. FN = 3 data
3. FP = 0
4. TN = 191
5. Presisi = 100%
6. Recall = 62%

Setelah evaluasi terhadap model yang telah dibuat, dilakukan pengujian terhadap model KNN yang dibuat dengan nilai K = 5 dengan memprediksi lebih lanjut dengan menggunakan lima data pertama terhadap data aktual pada data *test.*

Tabel 5.7 Hasil prediksi KNN data aktual dan prediksi

No	Gender	age	urea	cr	hba1c	chol	tg
1	1.0	0.677966	0.091146	0.090680	0.874172	0.398058	0.074074
2	1.0	0.627119	0.052083	0.061713	0.801325	0.611650	0.511111
3	1.0	0.457627	0.106771	0.061713	0.225166	0.446602	0.037037
4	1.0	0.593220	0.156250	0.083123	0.430464	0.427184	0.074074
5	1.0	0.525424	0.070313	0.069270	0.847682	0.339806	0.051852

Hdl	ldl	vldl	bmi	aktual	prediksi
0.164948	0.343750	0.054441	0.486957	1	1
0.103093	0.312500	0.077364	0.556522	1	1
0.072165	0.406250	0.114613	0.173913	0	0
0.082474	0.229167	0.022923	0.521739	1	1
0.072165	0.114583	0.040115	0.278261	1	1

5.1.3 Hasil Random Forest

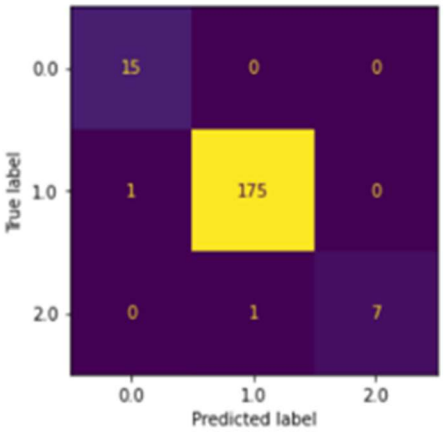
Algoritma kedua yang digunakan yaitu *Random Forest.* Berdasarkan perancangan yang telah dibuat setelah dilakukan data *pre-processing* dan normalisasi dan *split* data, dilakukan pemodelan *Random Forest* dengan parameter

n_estimators dan *max_depth.* Pengujian dibuat dengan beberapa kali pengujian dengan *n_estimators* dan *max_depth* yang berbeda. Berikut hasil yang telah didapatkan.

Tabel 5.8 Hasil evaluasi prediksi *Random Forest*

Tree	*Max Depth*	*Class*	*Precision*	*Recall*
100	3	0	0.94	1.00
		1	0.97	0.99
		2	1.00	0.25
	Accuracy		0.96	
	4	0	0.94	1.00
		1	0.97	0.99
		2	1.00	0.50
	Accuracy		0.96	
	5	0	0.94	1.00
		1	0.97	0.99
		2	1.00	0.38
	Accuracy		0.96	
Tree	*Max Depth*	*Class*	*Precision*	*Recall*
200	3	0	0.94	1.00
		1	0.97	0.99
		2	1.00	0.25
	Accuracy		0.96	
	4	0	0.94	1.00
		1	0.98	0.99
		2	1.00	0.62
	Accuracy		0.98	
	5	0	0.94	1.00
		1	0.99	0.99
		2	1.00	0.88
	Accuracy		0.99	
Tree	*Max Depth*	*Class*	*Precision*	*Recall*
300	3	0	0.94	1.00
		1	0.97	0.99
		2	1.00	0.25
	Accuracy		0.96	
	4	0	0.94	1.00
		1	0.98	0.99
		2	1.00	0.62
	Accuracy		0.98	
	5	0	0.94	1.00
		1	0.99	0.99
		2	1.00	0.88
	Accuracy		0.99	

Berdasarkan tabel 5.8, pengujian pada jumlah *tree* = 200 dan *max depth* = 5 mendapatkan hasil yang paling tinggi pada tingkat akurasi hingga mencapai 99%.



Gambar 5.8 *Confusion matrix* pada model *random forest*

Kemudian berdasarkan gambar 5.7 yang merupakan evaluasi *confusion matrix* terhadap model *random forest,* dilakukan penelusuran lebih lanjut terhadap data *test* pada kelas prediksi dan aktual. Berdasarkan matriks tersebut berdasarkan penjelasan pada sub-sub-bab 2.2.7 tentang *confusion matrix,* didapatkan nilai TP, FN, FP, dan TN serta pengukuran dari presisi dan *recall* dari masing-masing kelas serta pengukuran dari nilai akurasi dan presisinya.

Nilai-nilai dari kelas 0 *(non-diabetic)* antara lain yaitu :

1. TP = 15 data
2. FN = 0 data
3. FP = 1 data
4. TN = 183 data
5. Presisi = 98%
6. Recall = 100%

Nilai-nilai dari kelas 1 *(diabetic)* antara lain yaitu :

1. TP = 175 data
2. FN = 1 data
3. FP = 1 data
4. TN = 23 data
5. Presisi = 99%
6. Recall = 99%

Nilai-nilai dari kelas 2 *(predict-diabetic)* antara lain yaitu :

1. TP = 7 data
2. FN = 1 data
3. FP = 0
4. TN = 192
5. Presisi = 100%
6. Recall = 88%

Setelah model dievaluasi menggunakan *confusion matrix* maka perlu dilakukan prediksi terhadap data aktual yang didapatkan dari data *test.* Prediksi dilakukan sebanyak 5 kali pengujian terhadap 5 data pertama data *test* yang ditunjukan pada tabel 5.7.

Tabel 5.9 Hasil prediksi *Random Forest* data aktual dan prediksi

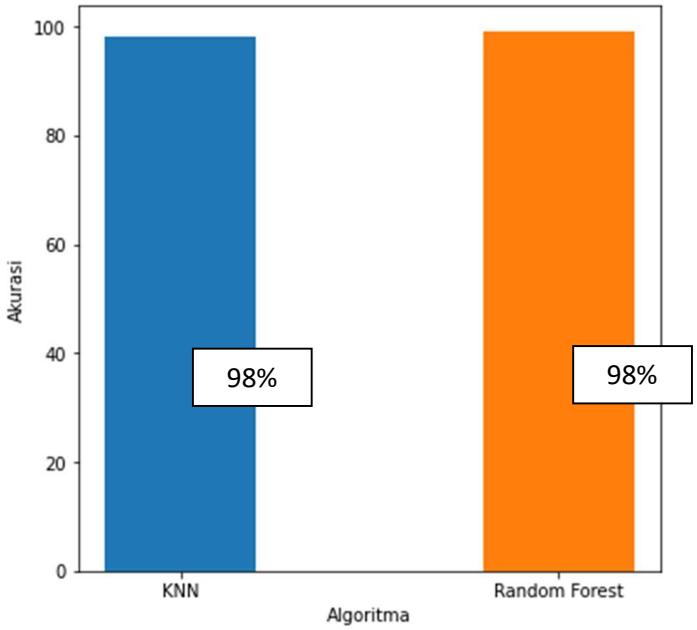
No	gender	Age	urea	cr	hba1c	chol	tg
1	1.0	0.677966	0.091146	0.090680	0.874172	0.398058	0.074074
2	1.0	0.627119	0.052083	0.061713	0.801325	0.611650	0.511111
3	1.0	0.457627	0.106771	0.061713	0.225166	0.446602	0.037037
4	1.0	0.593220	0.156250	0.083123	0.430464	0.427184	0.074074
5	1.0	0.525424	0.070313	0.069270	0.847682	0.339806	0.051852

hdl	ldl	vldl	bmi	aktual	prediksi
0.164948	0.343750	0.054441	0.486957	1	1
0.103093	0.312500	0.077364	0.556522	1	1
0.072165	0.406250	0.114613	0.173913	0	0
0.082474	0.229167	0.022923	0.521739	1	1
0.072165	0.114583	0.040115	0.278261	1	1

Berdasarkan tabel 5.9, pengujian terhadap 5 data pertama didapatkan hasil yang sama dan akurat sesuai dengan tingkat akurasi yang didapatkan.

5.1.4 Hasil Perbandingan Algoritma KNN dan *Random Forest*

Perbandingan model algoritma KNN dan *random forest* terhadap prediksi penyakit diabetes mellitus ditampilkan pada suatu bagan. Gambar 5.9 menampilkan bagan terhadap perbandingan kedua algoritma dengan indikator nilai akurasi keduanya.



Gambar 5.9 Bagan perbandingan kedua algoritma

Pada bagan tersebut, kedua algoritma terlihat cukup identik satu dengan yang lainnya. Jika ditelusuri lebih lanjut dari evaluasi *confusion matrix* dari kedua algoritma masing-masing pada gambar 5.7 dan gambar 5.8, keduanya sedikit memiliki perbedaan pada nilai-nilai dan pengukuran *confusion matrix.*

5.2 Pembahasan

Bagian pembahasan, dilakukan pemrograman model *machine learning* dalam memprediksi terhadap diabetes menggunakan algoritma KNN dan *Random Forest.* Pemrograman dilakukan berdasarkan perencanaan yang telah dibuat untuk kemudian diuji untuk melihat apakah hasil yang didapatkan sesuai dengan spesifikasi yang dibutuhkan menggunakan metode pengujian *whitebox.* Berikut pengujian yang dilakukan.

Tabel 5.10 Pengujian kode program menggunakan *whitebox*

No	Kode Program	Hasil yang diharapkan	Hasil Pengujian

1	`import pandas as pd`
	`import numpy as np`
	`import matplotlib.pyplot as plt`
	`from sklearn.metrics import mean_squared_error as mse`
	`from sklearn.model_selection import train_test_split`
	`from sklearn.preprocessing import MinMaxScaler`
	`from sklearn.neighbors import KNeighborsClassifier as KNN`
	`from sklearn.metrics import classification_report`
	`from sklearn.metrics import confusion_matrix`
	`from sklearn.ensemble import RandomForestClassifier as RFC`
	`from sklearn.metrics import accuracy_score`
		Import Seluruh library yang dibutuhkan.	Valid

2	`from google.colab import drive`
	`drive.mount('/content/drive')`
	`path = '/content/drive/MyDrive/data-ai/diabetes_v2.csv'`
	`data = pd.read_csv(path)`
	`data.describe()`
		Memuat dataset asli dan melihat data asli.	Valid

3	`data_columns = ['id', 'no_pation', 'gender', 'age',`
	`'urea', 'cr', 'hba1c', 'chol', 'tg',`
	`'hdl', 'ldl', 'vldl', 'bmi', 'class']`
	`data.columns = data_columns`
		Inisiasi kolom pada dataset	Valid

4	`# Mengubah nilai pada gender fitur Female (F) -> 0 dan`
	`# Male (M) -> = 1`
	`data['gender'].replace(to_replace='F',`
	`value=0, inplace=True)`
	`data['gender'].replace(to_replace='M',`
	`value=1, inplace=True)`
	
	`# Mengubah nilai fitur gender menjadi numerik`
	`data['gender'] = pd.to_numeric(data['gender'],`
	`downcast='integer',`
	`errors='coerce')`
	`data['gender'].head()`
		Mengubah nilai pada fitur *gender*	Valid

5	`# Mengubah fitur Class menjadi 1,2,0`
	`data['class'].replace(to_replace='Y',`
	`value=1, inplace=True) # Diabetic`
	`data['class'].replace(to_replace='P',`
	`value=2, inplace=True) # Predict-Diabetic`
	`data['class'].replace(to_replace='N',`
	`value=0, inplace=True) # Non-Diabetic`
	
	`# Mengubah nilai fitur Class menjadi numerik`
	`data['class'] = pd.to_numeric(data['class'],`
	`downcast='integer',`
	`errors='coerce')`
		Mengubah nilai pada fitur *class*	Valid

6	`# Cek apakah ada data error atau tidak` `data.isnull().sum()`	Mengecek apakah terdapat data *error* atau tidak	Valid
7	`# Data cleaning` `data.dropna(axis=0, inplace=True)`	Melakukan data *cleaning* terhadap data yang teridentifikasi *error*	Valid

8	`# Mengecek kembali apakah data sudah`
	`#tidak terdapat error`
	`data.isna().sum()`
		Mengecek kembali apakah data sudah tidak terdapat *error* atau tidak	Valid

9	`# Inisialisasi fitur dan target prediksi`
	
	`data_train = data.iloc[:, 2:13]`
	`features = ['gender', 'age', 'urea', 'cr',`
	`'hba1c', 'chol', 'tg', 'hdl', 'ldl', 'vldl', 'bmi']`
	`X = data_train[features]`
	`y = data.iloc[:, 13]`
		Melakukan inisialisasi fitur dan target prediksi yang digunakan	Valid

9	`# Inisialisasi fitur dan target prediksi`
	
	`data_train = data.iloc[:, 2:13]`
	`features = ['gender', 'age', 'urea', 'cr',`
	`'hba1c', 'chol', 'tg', 'hdl', 'ldl', 'vldl', 'bmi']`
	`X = data_train[features]`
	`y = data.iloc[:, 13]`
		Melakukan inisialisasi fitur dan target prediksi yang digunakan	Valid

10	`# Normalisasi Data`
	
	`scaler = MinMaxScaler()`
	`X = scaler.fit_transform(X)`
	`scaled_X = pd.DataFrame(X, columns = features)`
		Normalisasi fitur data	Valid

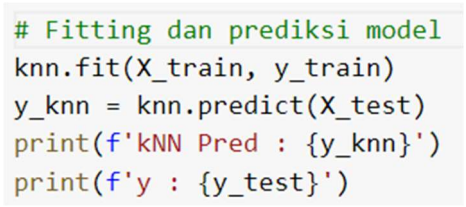
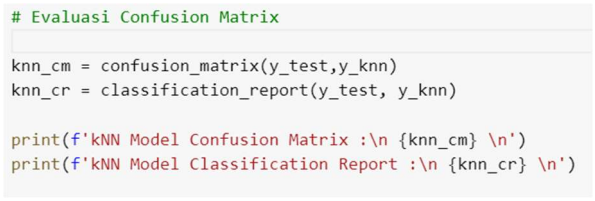
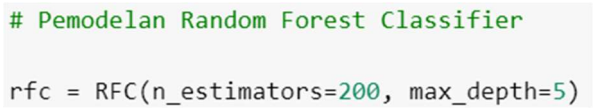
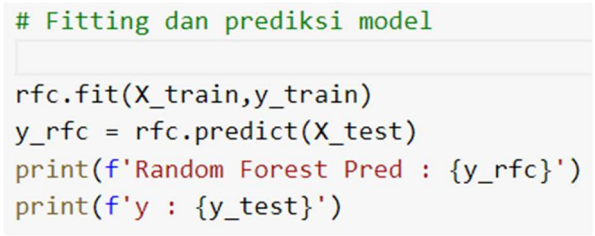
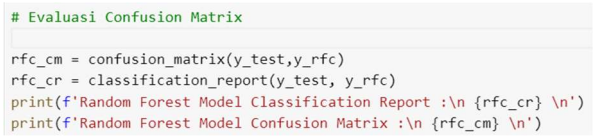
11	`# Split Data`
	
	`X_train, X_test, y_train, y_test = train_test_split(`
	`scaled_X,y, random_state=0, test_size = 0.2)`
		Membagi data menjadi data *train* dan data *test* rasio 80:20	Valid

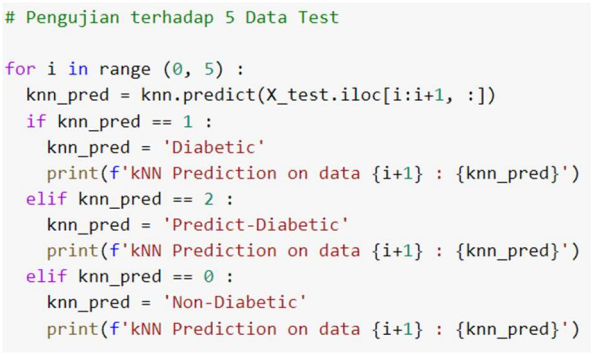
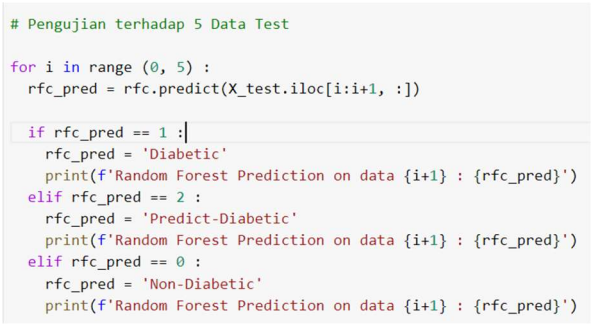
12	`print(len(X_train))`
	`print(len(y_train))`
	`print(len(X_test))`
	`print(len(y_test))`
		Menampilkan banyaknya data *train* dan data *test*	Valid

13	`# Elbow Method`
	`err_val = []`
	
	`for i in range(1,21):`
	`knn = KNN(n_neighbors = i)`
	`knn.fit(X_train, y_train)`
	`pred_i = knn.predict(X_test)`
	`err_ = round(np.mean(pred_i != y_test.values),5)`
	`err_val.append(err_)`
	`print(err_val)`
		Melakukan *elbow method* sebanyak 20 *clusters* untuk mendapatkan nilai K-Optimal pada pemodelan KNN	Valid

14	`plt.figure(figsize=(10,6))`
	`plt.plot(range(1,21),err_val,color='blue',`
	`linestyle='dashed', marker='o',`
	`markerfacecolor='red', markersize=10)`
	`plt.title('Error Rate and K Value')`
	`plt.xlabel('K Value')`
	`plt.ylabel('Error Rate')`
		Menampilkan grafik *elbow method* untuk visualisasi *cluster* 1 s/d. 20	Valid

15	`# Pemodelan KNN`
	`from sklearn.neighbors import KNeighborsClassifier as KNN`
	`knn = KNN(n_neighbors = 5)`
		Pemodelan KNN menggunakan K-Optimal yang telah didapatkan	Valid

16		Melakukan *fitting* model KNN menggunakan data *train* dan melakukan prediksi terhadap data *test*	Valid
17		Melakukan evaluasi model KNN menggunakan *confusion matrix* dan *classification report* untuk mendapatkan perhitungan akurasi, presisi, dan recall	Valid
18		Melakukan pemodelan algoritma *Random Forest* dengan parameter *n_estimators* dan *max_depth*	Valid
19		Melakukan *fitting* model *Random Forest* menggunakan data *train* dan melakukan prediksi terhadap data *test*	Valid
20		Melakukan evaluasi model Random Forest menggunakan *confusion matrix* dan *classification report* untuk mendapatkan perhitungan akurasi, presisi, dan recall	Valid

21		Pengujian model KNN terhadap 5 data *test*	Valid
22		Pengujian model *Random Forest* terhadap 5 data *test*	Valid
23		Visualisasi perbandingan antara KNN dan *Random Forest* menggunakan grafik batang	Valid