

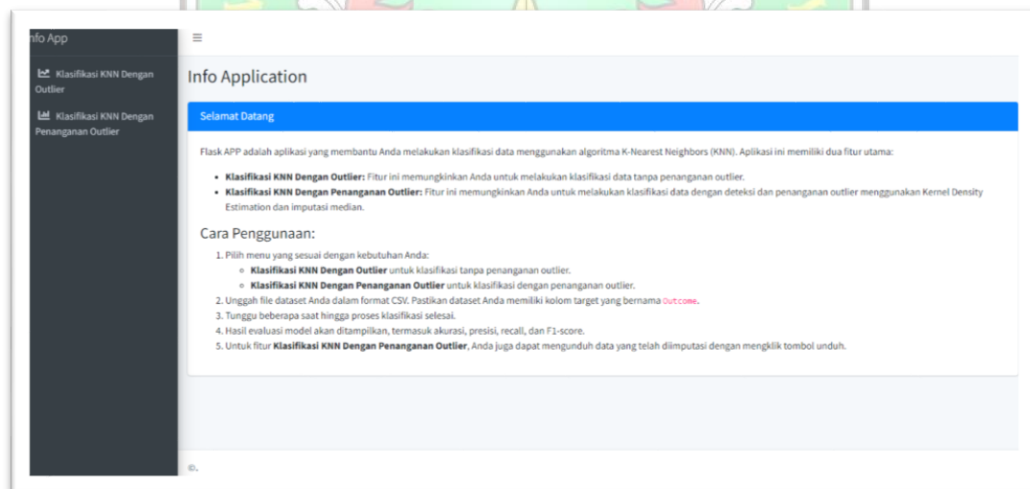
BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi Sistem

Implementasi sistem yang dibangun dalam penelitian ini menggunakan *flask* terdiri dari tiga menu utama. Pertama, yaitu informasi aplikasi yang akan menampilkan teks atau petunjuk penggunaan aplikasi ini. Menu kedua, menu Klasifikasi menggunakan *K-Nearest Neighbors* memungkinkan pengguna untuk mengunggah dataset dan melakukan klasifikasi data menggunakan algoritma *K-Nearest Neighbors*. Menu ketiga, Klasifikasi menggunakan *K-Nearest Neighbors* dan penanganan *outlier* menggunakan metode *Kernel Density Estimation*, memungkinkan pengguna untuk memilih dataset dan melakukan proses klasifikasi dan penanganan *outlier* untuk mengidentifikasi dataset yang terdeteksi *outlier*. Berikut ini adalah beberapa gambar hasil dari implementasinya.

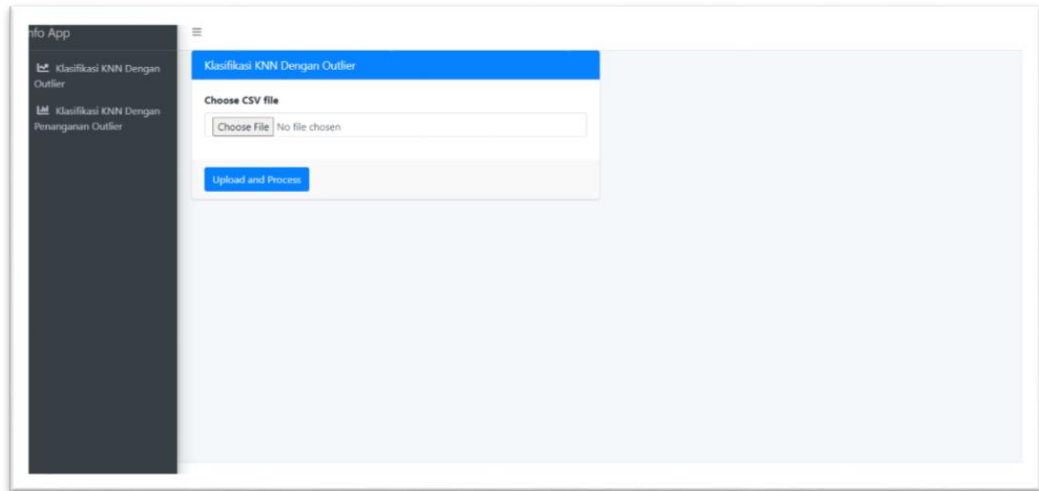
1. Halaman Informasi Aplikasi



Gambar 4. 1 Tampilan Halaman Informasi Aplikasi

Gambar 4.1 merupakan hasil implementasi *user interface* dari halaman informasi aplikasi. Informasi aplikasi ini merupakan halaman yang pertama kali akan muncul pada saat *user* membuka sistem klasifikasi yang berupa *website*. Halaman ini memberikan informasi kepada *user* mengenai informasi aplikasi dan cara penggunaan aplikasi ini.

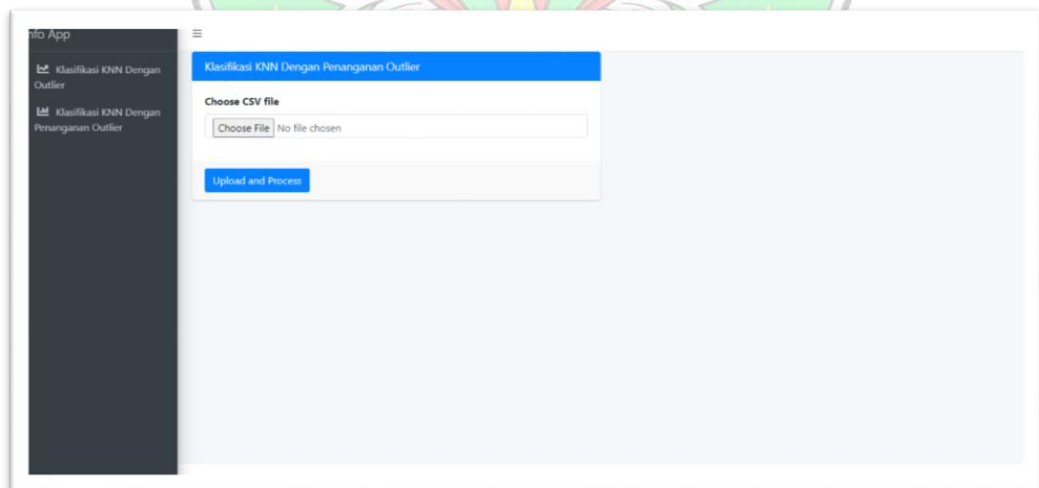
2. Halaman *Upload Dataset* Klasifikasi KNN dengan *Outlier*



Gambar 4. 2 Tampilan Halaman *Upload Dataset* Klasifikasi KNN Dengan *Outlier*

Gambar 4.2 merupakan hasil implementasi *user interface* dari halaman *upload dataset* klasifikasi knn dengan *outlier*. Halaman ini merupakan halaman untuk menguploadkan *dataset* yang berupa *file csv* lalu pada saat tombol diklik akan menuju kehalaman hasil klasifikasi *dataset* dengan *outlier*.

3. Halaman *Upload Dataset* Klasifikasi KNN dengan penanganan *Outlier*

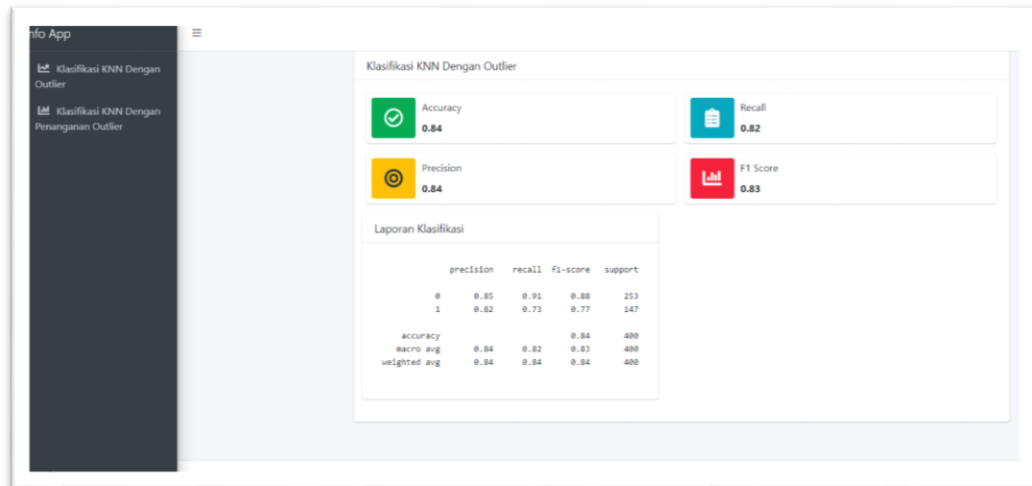


Gambar 4. 3 Tampilan Halaman *Upload Dataset* Klasifikasi KNN Dengan Penanganan *Outlier*

Gambar 4.3 merupakan hasil implementasi *user interface* dari halaman *Upload dataset* KNN dengan penanganan *outlier*. Halaman ini merupakan halaman

untuk menginputkan *dataset* yang berupa file csv lalu pada saat tombol diklik akan menuju kehalaman hasil klasifikasi *dataset* dengan penanganan *outlier*.

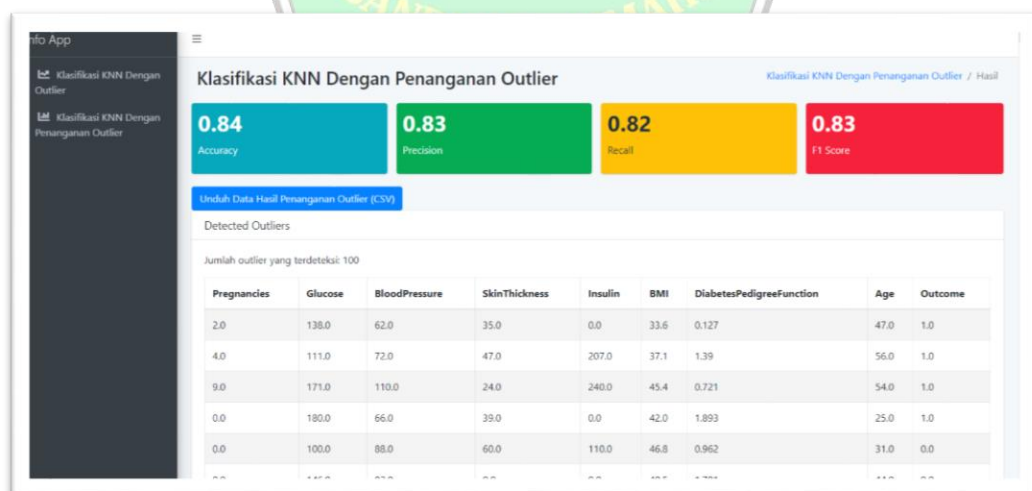
4. Halaman Hasil Klasifikasi KNN dengan *Outlier*

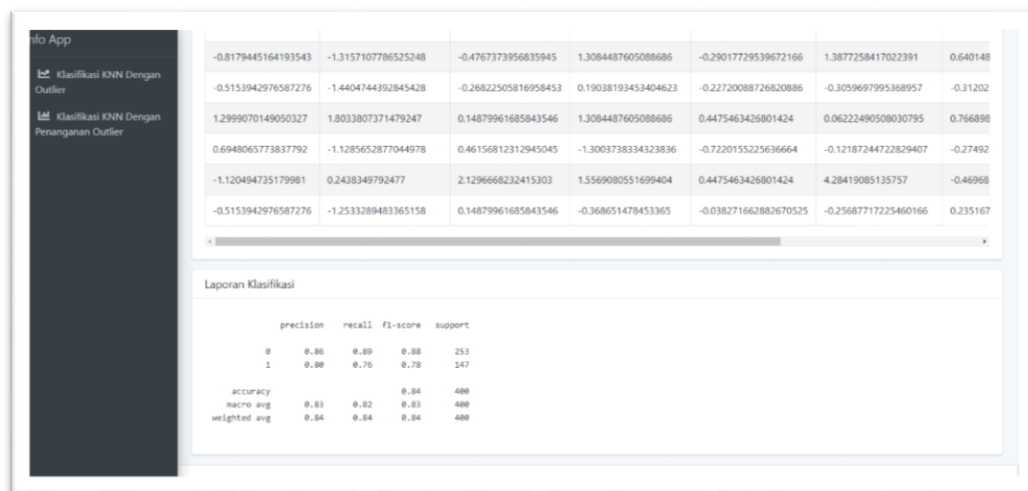


Gambar 4. 4 Tampilan Halaman Hasil Klasifikasi KNN Dengan *Outlier*

Gambar 4.4 merupakan hasil implementasi *user interface* dari halaman hasil klasifikasi KNN dengan *outlier*. Halaman ini menunjukkan hasil klasifikasi model KNN yang disertai laporan klasifikasi berupa *precision*, *recall*, *f1-score*, dan *support*.

5. Halaman Hasil Klasifikasi KNN dengan penanganan *Outlier*





	-0.8179445164193543	-1.3157107786525248	-0.4767373956835945	1.3084487605088686	-0.29017729539672166	1.3877258417022391	0.640148
	-0.5153942976587276	-1.4404744392845428	-0.26822505816958453	0.19038193453404623	-0.22720088726820886	-0.3059697995368957	-0.31202
	1.2999070149050327	1.8033807371479247	0.14879961685843546	1.3084487605088686	0.4475463426801424	0.06222490508030795	0.766898
	0.6948065773837792	-1.1285652877044978	0.46156812312945045	-1.3003738334323836	-0.7220155225636664	-0.12187244722829407	-0.27492
	-1.120494735179981	0.2438349792477	2.129668232415303	1.5569080551699404	0.4475463426801424	4.28419085135757	-0.46968
	-0.5153942976587276	-1.2533289483365158	0.14879961685843546	-0.368651478453365	-0.038271662882670525	-0.25687717225460166	0.235167

	precision	recall	f1-score	support
0	0.86	0.89	0.88	253
1	0.88	0.76	0.78	147
accuracy			0.84	400
macro avg	0.83	0.82	0.83	400
weighted avg	0.84	0.84	0.84	400

Gambar 4. 5 Tampilan Halaman Hasil Klasifikasi KNN Dengan Penanganan Outlier

Gambar 4.5 merupakan hasil implementasi *user interface* dari halaman hasil klasifikasi KNN dengan penanganan *outlier*. Halaman ini merupakan halaman untuk menunjukkan hasil klasifikasi model knn dan deteksi *outlier* menggunakan kde lalu menampilkan data yang telah di imputasi dengan *median* lalu akan mengeluarkan laporan klasifikasi berupa *precision*, *recall*, *f1-score*, dan *support*.

4.2 Pengujian Model KDE

KDE digunakan untuk mendeteksi *outlier* dengan nilai *bandwidth* 0.5, *bandwidth* yang lebih kecil memberikan estimasi kepadatan yang lebih tajam, membuat model lebih sensitif terhadap fluktuasi lokal dalam data.

- Ambang Batas (*Threshold*)

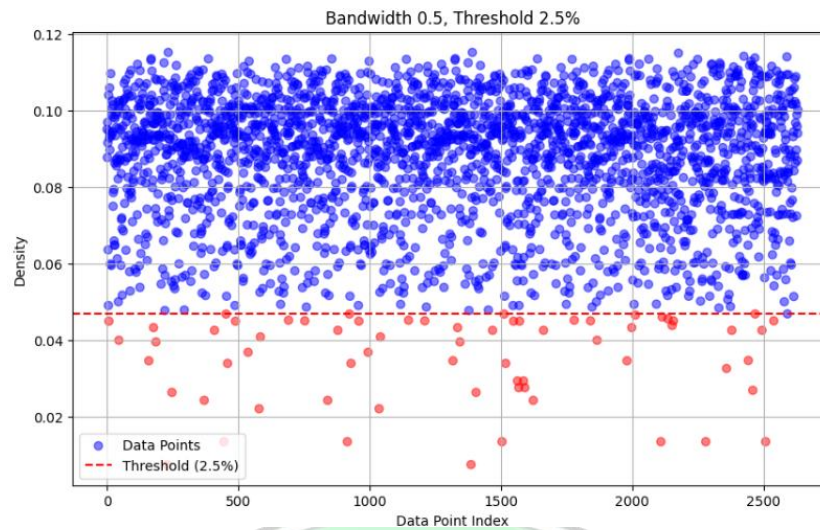
Ambang Batas 2.5%: Menandai titik-titik data yang memiliki kepadatan log di bawah persentil 2.5 sebagai *outlier*.

Ambang Batas 5%: Menandai titik-titik data yang memiliki kepadatan log di bawah persentil 5 sebagai *outlier*.

- Deteksi *Outlier*

Hasil deteksi outlier berdasarkan *threshold*:

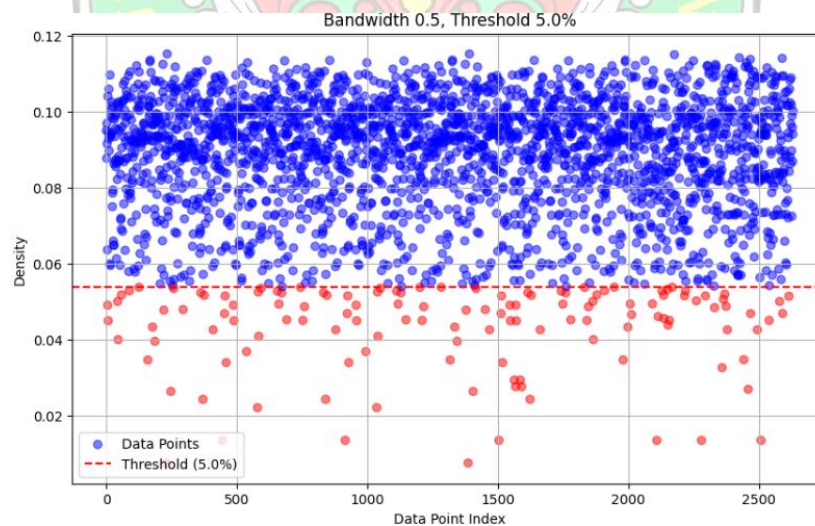
1. Threshold 2.5% ditunjukkan pada Gambar 4.6



Gambar 4. 6 Deteksi Outlier Threshold 2.5%

Bandwidth 0.5 dan Threshold 2.5%: Jumlah outlier yang terdeteksi adalah 66 data. Bandwidth ini memberikan estimasi kepadatan yang lebih tajam, sehingga lebih sensitif terhadap outlier.

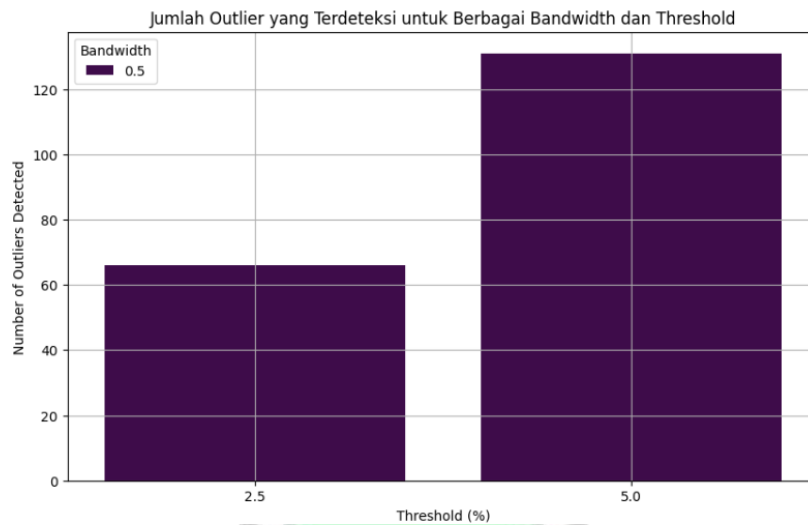
2. Threshold 2.5% ditunjukkan pada Gambar 4.7



Gambar 4. 7 Deteksi Outlier Threshold 5%

Bandwidth 0.5 dan Threshold 5%: Jumlah outlier yang terdeteksi 131 data. Dengan threshold yang lebih tinggi.

3. Grafik dari dua *scenario threshold* diatas ditunjukkan pada Gambar 4.8



Gambar 4. 8 Grafik Deteksi Outlier

Hasil dari grafik 4.8 menunjukan jumlah *outlier* yang terdeteksi pada *threshold* 2.5% yaitu sebanyak 66 data sedangkan jumlah *outlier* yang terdeteksi pada *threshold* 5% yaitu sebanyak 131 data. Dan berdasarkan eksperimen yang dilakukan sebelumnya akan digunakan *threshold* 2.5% memiliki 66 data yang terindikasi *outlier* untuk memvisualisasikan tahap pengujian model selanjutnya yaitu model KNN.

4.3 Pengujian Model KNN

Pengujian model *K-Nearest Neighbors* menggunakan dua skenario yaitu pengujian model tanpa penanganan *outlier* dan pengujian model dengan menggunakan penanganan *outlier*. Pengujian dilakukan dengan *dataset* dengan perbandingan 90%:10%, 80%:20%, dan 70%:30%, Hasil pengujian tersebut menunjukkan nilai akurasi, presisi, *recall*, dan *f1-score* dari model yang diuji. Hasil dari evaluasi model pada setiap skenario pembagian data latih dan data uji akan memberikan gambaran tentang performa model dalam berbagai kondisi [40]. dan dengan nilai *K* yang di variasikan antara 1, 3, 5, dan 7.

4.2.1 Skenario 1 Dataset 90% : 10%

Hasil pengujian pertama dilakukan menggunakan dataset dengan perbandingan 90%:10% dan dengan variasi nilai $K = 1, 3, 5$, dan 7 . Performa model dapat dilihat pada Tabel 4.1.

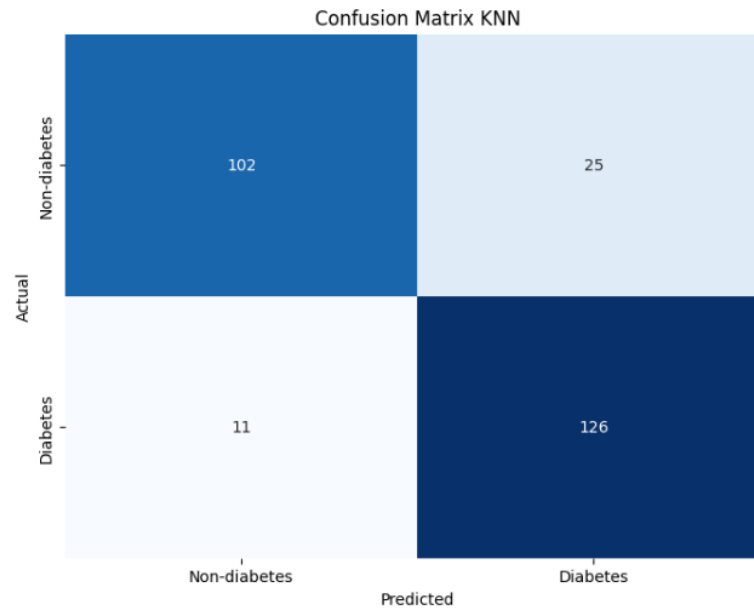
Tabel 4. 1 Pengujian Dataset 90/10

Performa	KNN				KNN dan <i>Imputasi Outlier</i>			
	K=1	K=3	K=5	K=7	K=1	K=3	K=5	K=7
Akurasi	0.98	0.94	0.87	0.86	0.96	0.94	0.87	0.87

Berdasarkan Tabel 4.1, pada dataset rasio 90%:10% KNN tanpa imputasi *outlier* mencapai akurasi tertinggi dengan $k=1$ (0.98). Hal ini menunjukkan bahwa model KNN dengan nilai k yang sangat kecil mampu memprediksi dengan sangat baik pada dataset ini. Namun, seiring peningkatan nilai k , akurasi menurun. Hal ini dapat disebabkan oleh fakta bahwa dengan meningkatnya k , model menjadi lebih rentan terhadap *outlier* dalam data. KNN dengan imputasi *outlier* menunjukkan akurasi yang sedikit lebih rendah pada $k=1$ (0.96) dibandingkan KNN tanpa imputasi *outlier*. Namun, pada $k=7$, akurasi KNN dengan imputasi *outlier* (0.87) tetap lebih baik daripada KNN tanpa imputasi *outlier* (0.86). Ini menunjukkan bahwa terdapat peningkatan akurasi pada KNN dengan penanganan *outlier*, khususnya saat nilai k lebih besar, yang dapat mengurangi dampak dari *outlier* dalam data. Secara keseluruhan, model KNN dengan imputasi *outlier* menunjukkan performa yang lebih konsisten dan tahan terhadap *outlier*, terutama pada nilai k yang lebih besar, dibandingkan dengan model KNN tanpa penanganan *outlier*.

Confusion matrix adalah alat yang sangat berguna dalam evaluasi kinerja model klasifikasi. Matrix ini memberikan informasi tentang jumlah prediksi yang benar dan salah yang dilakukan oleh model, dengan rincian yang ditunjukkan pada Gambar 4.9.

- *Confusion Matrix*



Gambar 4. 9 Confusion Matrix KNN Dataset 90/10

Pada Gambar 4.9 menunjukkan bahwa pembagian kelas positif dan negatifnya tidak seimbang, padahal pada dataset sudah dilakukan penyeimbangan data menggunakan SMOTE. Hal itu dikarenakan pembagian data train dan test pada penelitian ini tidak terlalu representatif sehingga menyebabkan ketidakseimbangan antara kelas positif dan negatifnya. Gambaran mengenai performa model dalam memprediksi penyakit diabetes dengan *Confusion Matrix* KNN, yaitu:

True Negative (TN): 102

- Jumlah orang yang benar-benar tidak menderita diabetes dan diprediksi tidak menderita diabetes oleh model.

False Positive (FP): 25

- Jumlah orang yang tidak menderita diabetes tetapi diprediksi menderita diabetes oleh model.

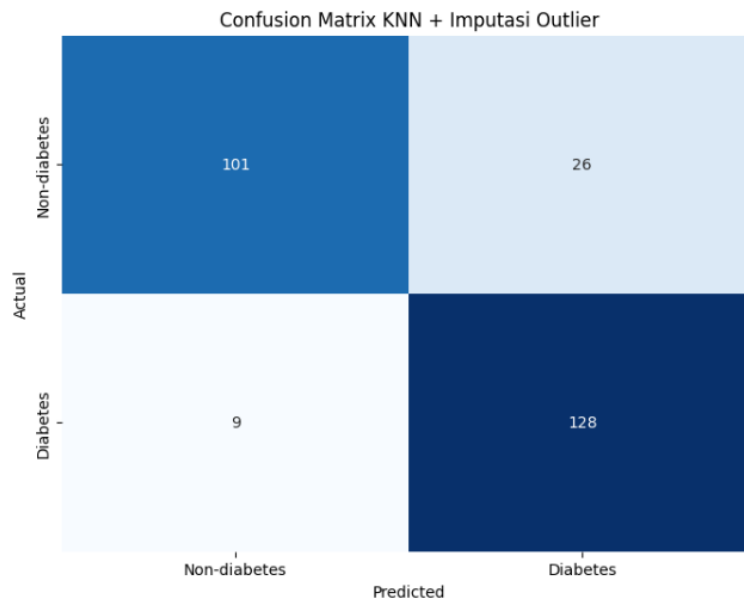
False Negative (FN): 11

- Jumlah orang yang menderita diabetes tetapi diprediksi tidak menderita diabetes oleh model.

True Positive (TP): 126

- Jumlah orang yang benar-benar menderita diabetes dan diprediksi menderita diabetes oleh model.

Selanjutnya performa model dalam memprediksi penyakit diabetes, dengan *Confusion Matrix* KNN dengan Imputasi *Median*, yang ditunjukkan pada Gambar 4.10



Gambar 4. 10 *Confusion Matrix* KNN dan Outlier Dataset 90/10

Pada Gambar 4.10 menunjukkan bahwa pembagian kelas positif dan negatifnya tidak seimbang, padahal pada dataset sudah dilakukan penyeimbangan data menggunakan SMOTE. Hal itu dikarenakan pembagian data train dan test pada penelitian ini tidak terlalu representatif sehingga menyebabkan ketidakseimbangan antara kelas positif dan negatifnya. Gambaran *Confusion Matrix* KNN dengan Imputasi *Median*, dengan data sebagai berikut:

True Negative (TN): 101

Jumlah orang yang benar-benar tidak menderita diabetes dan diprediksi tidak menderita diabetes oleh model.

False Positive (FP): 26

- Jumlah orang yang tidak menderita diabetes tetapi diprediksi menderita diabetes oleh model.

False Negative (FN): 9

- Jumlah orang yang menderita diabetes tetapi diprediksi tidak menderita diabetes oleh model.

True Positive (TP): 128

- Jumlah orang yang benar-benar menderita diabetes dan diprediksi menderita diabetes oleh model.

Memahami distribusi ini berarti dapat dihitung beberapa metrik kinerja penting seperti akurasi, presisi, *recall*, dan *F1-score* yang membantu dalam mengevaluasi sejauh mana model klasifikasi berhasil dalam tugasnya.

Laporan klasifikasi (*classification report*) untuk model *K-Nearest Neighbors* (KNN) yang digunakan dalam klasifikasi diabetes memberikan beberapa metrik kinerja penting untuk setiap kelas (Negatif dan Positif) yang diberikan pada Tabel 4.2 dan Tabel 4.3.

- Clasification Report

Tabel 4. 2 Clasification Report KNN

KNN				
Kelas	Presisi	Recall	F1-Score	Support
Negatif	0.90	0.80	0.85	127
Positif	0.83	0.92	0.88	137

Tabel 4. 3 Clasification Report KNN dan Outlier

KNN dan Imputasi Outlier				
Kelas	Presisi	Recall	F1-Score	Support
Negatif	0.92	0.80	0.85	127
Positif	0.83	0.93	0.88	137

- Performa Kelas Negatif:

Tabel 4.2 menunjukkan presisi 0.90, *recall* 0.80, dan *F1-score* 0.85.

Tabel 4.3 menunjukkan presisi 0.92, *recall* 0.80, dan *F1-score* 0.85.

- Performa Kelas Positif:

Tabel 4.2 menunjukkan presisi 0.83, *recall* 0.92, dan *F1-score* 0.88.

Tabel 4.3 menunjukkan presisi 0.83, *recall* 0.93, dan *F1-score* 0.88.

- Support:

Support adalah jumlah sebenarnya dari sampel di kelas yang diberikan dalam dataset. Ini memberikan gambaran tentang distribusi data di kelas-kelas tersebut.

Kelas Negatif memiliki support sebanyak 127, yang berarti ada 127 sampel dalam kelas negatif.

Kelas Positif memiliki support sebanyak 137, yang berarti ada 137 sampel dalam kelas positif.

4.2.2 Skenario 2 Dataset 80% : 20%

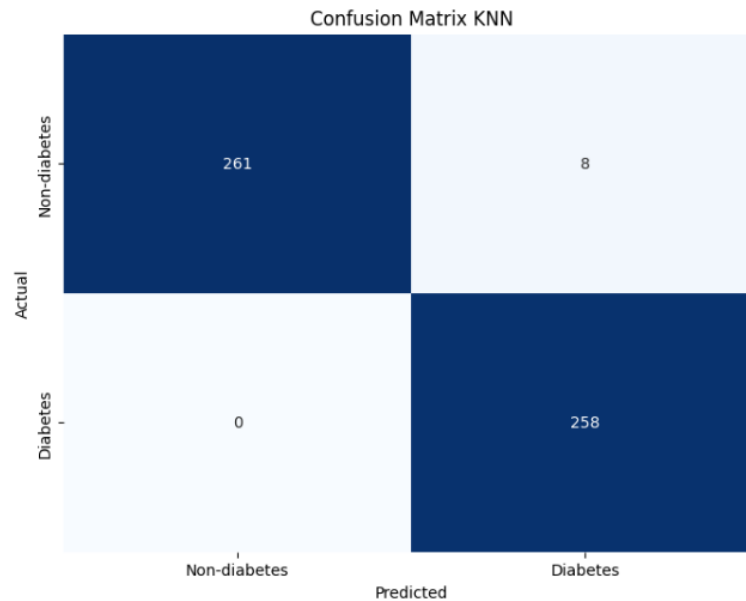
Hasil pengujian kedua dilakukan menggunakan dataset dengan perbandingan 80%:20% dan dengan variasi nilai $K = 1, 3, 5, \text{ dan } 7$. Performa model dapat dilihat pada Tabel 4.4.

Tabel 4. 4 Pengujian Dataset 80/20

Performa	KNN				KNN dan <i>Imputasi Outlier</i>			
	K=1	K=3	K=5	K=7	K=1	K=3	K=5	K=7
Akurasi	0.98	0.93	0.86	0.83	0.96	0.91	0.84	0.81

Berdasarkan tabel 4.4 tersebut pada dataset dengan rasio 80%:20%, KNN tanpa imputasi outlier mencapai akurasi tertinggi dengan $k=1$ (0.98). Hal ini menunjukkan bahwa model KNN dengan nilai k yang sangat kecil mampu memprediksi dengan sangat baik pada dataset ini. Namun, seiring peningkatan nilai k , akurasi menurun. Hal ini dapat disebabkan oleh fakta bahwa dengan meningkatnya k , model menjadi lebih rentan terhadap *outlier* dalam data. KNN dengan imputasi *outlier* menunjukkan akurasi yang sedikit lebih rendah pada $k=1$ (0.96) dibandingkan KNN tanpa imputasi outlier. Namun, pada $k=3$, $k=5$, dan $k=7$, akurasi KNN dengan imputasi *outlier* tetap lebih rendah dibandingkan KNN tanpa imputasi *outlier*. Ini menunjukkan bahwa meskipun imputasi *outlier* membantu dalam menstabilkan performa, model KNN tanpa imputasi *outlier* tetap menunjukkan performa yang lebih baik dalam kasus ini.

Confusion matrix adalah alat yang sangat berguna dalam evaluasi kinerja model klasifikasi. Matrix ini memberikan informasi tentang jumlah prediksi yang benar dan salah yang dilakukan oleh model, dengan rincian yang ditunjukkan pada Gambar 4.11.



Gambar 4. 11 Confusion Matrix KNN Dataset 80/20

Pada Gambar 4.11 menunjukkan bahwa pembagian kelas positif dan negatifnya tidak seimbang, padahal pada dataset sudah dilakukan penyeimbangan data menggunakan SMOTE. Hal itu dikarenakan pembagian data train dan test pada penelitian ini tidak terlalu representatif sehingga menyebabkan ketidakseimbangan antara kelas positif dan negatifnya.. Gambaran mengenai performa model dalam memprediksi penyakit diabetes dengan *Confusion Matrix* KNN, yaitu:

True Negative (TN): 261

- Jumlah orang yang benar-benar tidak menderita diabetes dan diprediksi tidak menderita diabetes oleh model.

False Positive (FP): 8

- Jumlah orang yang tidak menderita diabetes tetapi diprediksi menderita diabetes oleh model.

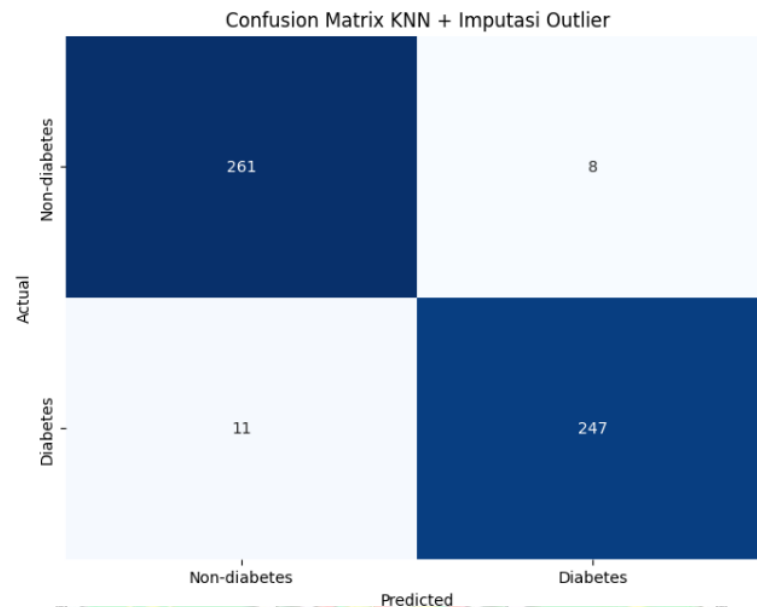
False Negative (FN): 0

- Jumlah orang yang menderita diabetes tetapi diprediksi tidak menderita diabetes oleh model.

True Positive (TP): 258

- Jumlah orang yang benar-benar menderita diabetes dan diprediksi menderita diabetes oleh model.

Selanjutnya performa model dalam memprediksi penyakit diabetes, dengan *Confusion Matrix* KNN dengan Imputasi *Median*, yang ditunjukkan pada Gambar 4.12



Gambar 4. 12 *Confusion Matrix* KNN dan Outlier Dataset 80/20

Pada Gambar 4.12 menunjukkan bahwa pembagian kelas positif dan negatifnya tidak seimbang, padahal pada dataset sudah dilakukan penyeimbangan data menggunakan SMOTE. Hal itu dikarenakan pembagian data train dan test pada penelitian ini tidak terlalu representatif sehingga menyebabkan ketidakseimbangan antara kelas positif dan negatifnya. Gambaran *Confusion Matrix* KNN dengan Imputasi *Median*, dengan data sebagai berikut:

True Negative (TN): 261

- Jumlah orang yang benar-benar tidak menderita diabetes dan diprediksi tidak menderita diabetes oleh model.

False Positive (FP): 8

- Jumlah orang yang tidak menderita diabetes tetapi diprediksi menderita diabetes oleh model.

False Negative (FN): 11

- Jumlah orang yang menderita diabetes tetapi diprediksi tidak menderita diabetes oleh model.

True Positive (TP): 247

- Jumlah orang yang benar-benar menderita diabetes dan diprediksi menderita diabetes oleh model.

Memahami distribusi ini berarti dapat dihitung beberapa metrik kinerja penting seperti akurasi, presisi, *recall*, dan *F1-score* yang membantu dalam mengevaluasi sejauh mana model klasifikasi berhasil dalam tugasnya.

Laporan klasifikasi (*classification report*) untuk model *K-Nearest Neighbors* (KNN) yang digunakan dalam klasifikasi diabetes memberikan beberapa metrik kinerja penting untuk setiap kelas (Negatif dan Positif) yang diberikan pada Tabel 4.5 dan Tabel 4.6.

- Clasification Report

Tabel 4. 5 Clasification Report KNN

KNN				
Kelas	<i>Presisi</i>	<i>Recall</i>	<i>F1-Score</i>	<i>Support</i>
Negatif	1.00	0.97	0.98	269
Positif	0.97	1.00	0.98	258

Tabel 4. 6 Clasification Report KNN dan Outlier

KNN dan <i>Imputasi Outlier</i>				
Kelas	<i>Presisi</i>	<i>Recall</i>	<i>F1-Score</i>	<i>Support</i>
Negatif	0.96	0.97	0.96	269
Positif	0.97	0.96	0.96	258

- Performa Kelas Negatif:

Tabel 4.5 menunjukkan presisi 1.00, *recall* 0.97, dan *F1-score* 0.98.

Tabel 4.6 menunjukkan presisi 0.96, *recall* 0.97, dan *F1-score* 0.96.

- Performa Kelas Positif:

Tabel 4.5 menunjukkan presisi 0.97, *recall* 1.00, dan *F1-score* 0.98.

Tabel 4.6 menunjukkan presisi 0.97, *recall* 0.96, dan *F1-score* 0.96.

- Support:

Support adalah jumlah sebenarnya dari sampel di kelas yang diberikan dalam dataset. Ini memberikan gambaran tentang distribusi data di kelas-kelas tersebut.

Kelas Negatif memiliki support sebanyak 269, yang berarti ada 269 sampel dalam kelas negatif.

Kelas Positif memiliki support sebanyak 258, yang berarti ada 258 sampel dalam kelas positif.

4.2.3 Skenario 3 Dataset 70% : 30%

Hasil pengujian ketiga dilakukan menggunakan dataset dengan perbandingan 70%:30% dan dengan variasi nilai $K = 1, 3, 5$, dan 7 . Performa model dapat dilihat pada Tabel 4.7.

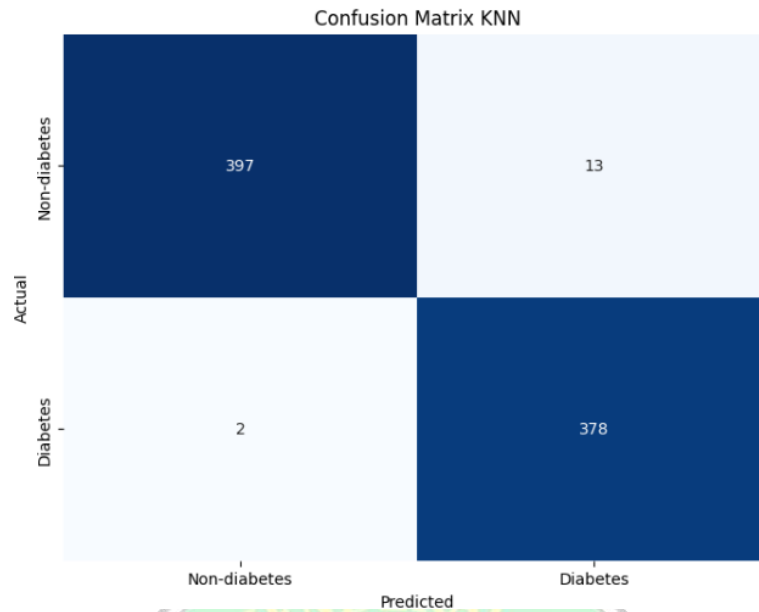
Tabel 4. 7 Pengujian Dataset 70/30

Performa	KNN				KNN dan <i>Imputasi Outlier</i>			
	K=1	K=3	K=5	K=7	K=1	K=3	K=5	K=7
Akurasi	0.98	0.91	0.86	0.82	0.98	0.91	0.85	0.82

Berdasarkan tabel 4.7 performa, pada dataset dengan rasio 70%:30%, terlihat bahwa KNN tanpa imputasi *outlier* menunjukkan akurasi tertinggi pada $K=1$ (0.98), dengan penurunan akurasi seiring meningkatnya nilai k KNN dengan imputasi *outlier* memiliki akurasi yang sama dengan KNN tanpa imputasi *outlier* pada $K=1$ dan $K=3$. Namun, pada $K=5$ dan $K=7$, akurasi KNN dengan imputasi *outlier* sedikit lebih rendah, tetapi pada $K=7$ akurasi keduanya sama (0.82). KNN dengan dan tanpa imputasi *outlier* menunjukkan performa yang serupa. Imputasi *outlier* tidak memberikan keuntungan signifikan dalam meningkatkan akurasi pada dataset ini, terutama pada nilai k yang lebih besar.

Confusion matrix adalah alat yang sangat berguna dalam evaluasi kinerja model klasifikasi. Matrix ini memberikan informasi tentang jumlah prediksi yang benar dan salah yang dilakukan oleh model, dengan rincian yang ditunjukkan pada Gambar 4.13

- Confusion Matrix



Gambar 4. 13 Confusion Matrix KNN Dataset 70/30

Pada Gambar 4.13 menunjukkan bahwa pembagian kelas positif dan negatifnya tidak seimbang, padahal pada dataset sudah dilakukan penyeimbangan data menggunakan SMOTE. Hal itu dikarenakan pembagian data train dan test pada penelitian ini tidak terlalu representatif sehingga menyebabkan ketidakseimbangan antara kelas positif dan negatifnya. Gambaran mengenai performa model dalam memprediksi penyakit diabetes dengan *Confusion Matrix* KNN, yaitu:

True Negative (TN): 397

- Jumlah orang yang benar-benar tidak menderita diabetes dan diprediksi tidak menderita diabetes oleh model.

False Positive (FP): 13

- Jumlah orang yang tidak menderita diabetes tetapi diprediksi menderita diabetes oleh model.

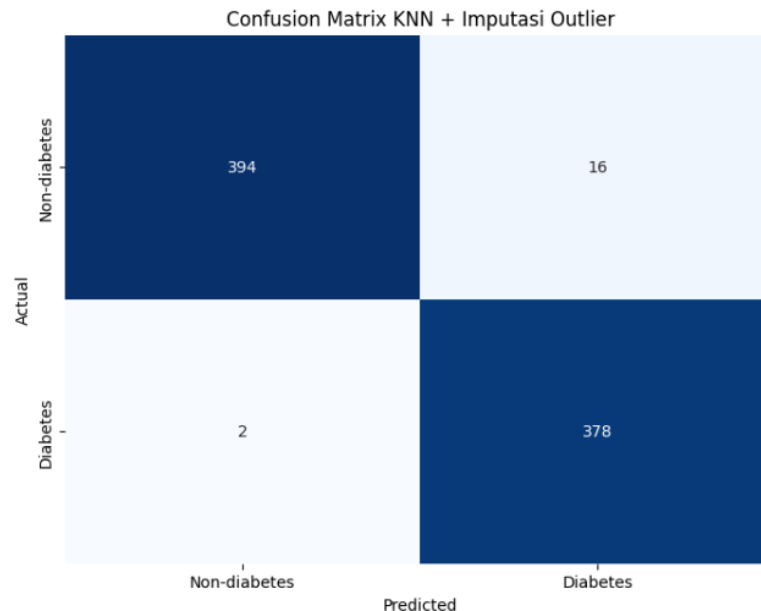
False Negative (FN): 2

- Jumlah orang yang menderita diabetes tetapi diprediksi tidak menderita diabetes oleh model.

True Positive (TP): 378

- Jumlah orang yang benar-benar menderita diabetes dan diprediksi menderita diabetes oleh model.

Selanjutnya performa model dalam memprediksi penyakit diabetes, dengan *Confusion Matrix* KNN dengan Imputasi *Median*, yang ditunjukkan pada Gambar 4.14



Gambar 4. 14 *Confusion Matrix* KNN dan Outlier Dataset 70/30

Pada Gambar 4.14 menunjukkan bahwa pembagian kelas positif dan negatifnya tidak seimbang, padahal pada dataset sudah dilakukan penyeimbangan data menggunakan SMOTE. Hal itu dikarenakan pembagian data train dan test pada penelitian ini tidak terlalu representatif sehingga menyebabkan ketidakseimbangan antara kelas positif dan negatifnya.. Gambaran *Confusion Matrix* KNN dengan Imputasi *Median*, dengan data sebagai berikut:

True Negative (TN): 394

- Jumlah orang yang benar-benar tidak menderita diabetes dan diprediksi tidak menderita diabetes oleh model.

False Positive (FP): 16

- Jumlah orang yang tidak menderita diabetes tetapi diprediksi menderita diabetes oleh model.

False Negative (FN): 2

- Jumlah orang yang menderita diabetes tetapi diprediksi tidak menderita diabetes oleh model.

True Positive (TP): 378

- Jumlah orang yang benar-benar menderita diabetes dan diprediksi menderita diabetes oleh model.

Memahami distribusi ini berarti dapat dihitung beberapa metrik kinerja penting seperti akurasi, presisi, *recall*, dan *F1-score* yang membantu dalam mengevaluasi sejauh mana model klasifikasi berhasil dalam tugasnya.

Laporan klasifikasi (*classification report*) untuk model *K-Nearest Neighbors* (KNN) yang digunakan dalam klasifikasi diabetes memberikan beberapa metrik kinerja penting untuk setiap kelas (Negatif dan Positif) yang diberikan pada Tabel 4.8 dan Tabel 4.9.

- Clasification Report

Tabel 4. 8 Clasification Report KNN

KNN				
Kelas	<i>Presisi</i>	<i>Recall</i>	<i>F1-Score</i>	<i>Support</i>
Negatif	0.99	0.97	0.98	410
Positif	0.97	0.99	0.98	380

Tabel 4. 9 Clasification Report KNN dan Outlier

KNN dan Imputasi Outlier				
Kelas	<i>Presisi</i>	<i>Recall</i>	<i>F1-Score</i>	<i>Support</i>
Negatif	0.99	0.96	0.98	410
Positif	0.96	0.99	0.98	380

- Performa Kelas Negatif:
Tabel 4.8 menunjukkan presisi 0.99, *recall* 0.97, dan *F1-score* 0.98.
Tabel 4.9 menunjukkan presisi 0.99, *recall* 0.96, dan *F1-score* 0.98.
- Performa Kelas Positif:
Tabel 4.8 menunjukkan presisi 0.97, *recall* 0.99, dan *F1-score* 0.98.
Tabel 4.9 menunjukkan presisi 0.96, *recall* 0.99, dan *F1-score* 0.98.
- Support:
Support adalah jumlah sebenarnya dari sampel di kelas yang diberikan dalam dataset. Ini memberikan gambaran tentang distribusi data di kelas-kelas tersebut.

Kelas Negatif memiliki support sebanyak 410, yang berarti ada 410 sampel dalam kelas negatif.

Kelas Positif memiliki support sebanyak 380, yang berarti ada 380 sampel dalam kelas positif.

4.4 Evaluasi Sistem Menggunakan *Confusion Matrix*

Dari hasil pengujian model KNN dengan Imputasi Outlier pada nilai $K=7$ menunjukkan akurasi tertinggi ketika menggunakan 90% data untuk pelatihan dan 10% data untuk pengujian. Berikut adalah rincian nilai dari laporan klasifikasi yang dihasilkan dari *confusion matrix* tersebut, yaitu:

- True Positive (TP): 128
- True Negative (TN): 101
- False Positive (FP): 26
- False Negative (FN): 9

1. Accuracy

Perhitungan *accuracy* ini dilakukan dengan menggunakan persamaan (2.5)

$$Accuracy = \frac{128 + 101}{128 + 101 + 26 + 9} = \frac{229}{264} = 0.867$$

2. Precision

Perhitungan *precision* ini dilakukan menggunakan persamaan (2.6)

$$Precision = \frac{128}{128 + 26} = \frac{128}{154} = 0.831$$

3. Recall

Perhitungan *recall* ini dilakukan menggunakan persamaan (2.7)

$$Recall = \frac{128}{128 + 9} = \frac{128}{137} = 0.934$$

4. F1-Score

Perhitungan *F1-Score* ini dilakukan menggunakan persamaan (2.8)

$$F1 - Score = 2 \times \frac{0.831 \times 0.934}{0.831 + 0.934} = 2 \times \frac{0.776}{1.765} = 0.88$$

■

4.4 Pengujian Sistem

Pada tahap pengujian sistem klasifikasi yang dibuat dalam penelitian tugas akhir ini menggunakan metode *Black Box Testing*. Tujuan dari pengujian ini adalah untuk mengamati hasil dari *input* dan *output* dari sistem yang telah dibuat tanpa mengetahui tentang detail dari struktur kode pemrogramannya. Pengujian ini untuk meninjau dan mengamati apakah di dalam sistem terdapat kesalahan atau *bug* serta meninjau apakah sistem sudah berjalan sesuai dengan yang diharapkan, yang diberikan pada Tabel 4.10.

Tabel 4. 10 Pengujian Black Box

Kode Uji	Skenario Pengujian	Kasus Pengujian	Hasil Pengujian	Keterangan
KU 01	Menampilkan Informasi aplikasi	Menampilkan informasi aplikasi dan cara penggunaannya pada sistem	Sesuai	Sukses
KU 02	Menampilkan form upload dataset untuk klasifikasi knn dengan outlier	Menampilkan halaman upload dataset untuk klasifikasi knn dengan outlier	Sesuai	Sukses
KU 03	Menampilkan form upload dataset untuk klasifikasi knn dengan penanganan outlier outlier	Menampilkan halaman upload dataset untuk klasifikasi knn dengan penanganan outlier	Sesuai	Sukses
KU 04	Menampilkan halaman hasil dari klasifikasi knn	Menampilkan halaman hasil dari klasifikasi knn	Sesuai	Sukses

	dengan outlier	dengan outlier pada saat tombol upload dataset pada halaman input dataset diklik pada sistem		
KU 05	Menampilkan halaman hasil dari klasifikasi knn dengan penanganan outlier	Menampilkan halaman hasil dari klasifikasi knn dengan penanganan outlier dan button download dataset pada saat tombol upload dataset pada halaman input dataset diklik pada sistem	Sesuai	Sukses

4.4.1 Evaluasi Hasil Pengujian

Berdasarkan hasil pengujian perangkat lunak yang tercatat dalam Tabel 4.3, kita dapat melakukan perhitungan untuk menentukan persentase kepatuhan sistem terhadap fungsinya sebagai berikut:

Total jumlah kasus uji: 5

Jumlah kasus uji yang sesuai: 5

Jumlah kasus uji yang tidak sesuai: 0

$$\text{Persentase} = \frac{(\text{Jumlah Kode Uji} - \text{Kode Uji Tidak Sesuai})}{\text{Jumlah Kode Uji}} \times 100\% \quad (4.1)$$

$$\text{Persentase} = \frac{5 - 0}{5} \times 100\% = 100\%$$

■

Berdasarkan hasil pengujian perangkat lunak yang tercatat dalam Tabel 4.10, dapat dinyatakan bahwa semua 5 kasus uji telah berhasil sesuai dengan harapan tanpa adanya kasus uji yang tidak sesuai. Ini menunjukkan bahwa sistem telah melewati pengujian dengan sukses dan memenuhi semua fungsi yang telah

ditetapkan. Persentase kesesuaian sistem dengan fungsinya adalah 100%. Kesimpulan ini menegaskan bahwa sistem siap untuk digunakan dan memiliki kinerja yang memenuhi harapan.

