Rancang Bangun Aplikasi dengan Perbandingan Metode K-Nearest Neighbor (KNN) dan Naive Bayes dalam Klasifikasi Penderita Penyakit Diabetes

Wirawan Dwi Prasetya[1], Bambang Sujatmiko[2]
[1,2] Jurusan Teknik Informatika, Fakultas Teknik, Universitas Negeri Surabaya
[1]wirawan.18051204026@mhs.unesa.ac.id
[2]bambangsujatmiko@unesa.ac.id

***Abstrak*— Diabetes adalah penyakit yang tidak menular dan termasuk cukup serius bagi manusia dikarenakan pankreas tidak mampu menghasilkan insulin secara optimal. *Internasional Diabetes Federation* (IDF) memperkirakan sedikitnya terdapat 463 juta orang pada usia 20-79 tahun di dunia menderita diabetes. Prevelensi diabetes diperkirakan meningkat seiring penambahan umur penduduk menjadi 19,9% juta pada tahun 2030 dan 700 juta pada tahun 2024. Oleh karena itu dibutuhkan sistem yang bertujuan untuk mendeteksi penderita penyakit diabetes. Penelitian ini menggunakan dua algoritma yaitu KNN dan *Naive Bayes*. Hal ini untuk memandingkan kedua algoritma yang memiliki tingkat akurasi yang terbaik. Algoritma KNN adalah algoritma yang digunakan untuk mengklasifikasi objek baru berdasarkan objek terdekatnya. Adapun Algoritma *Naive Bayes* adalah salah satu algoritma yang digunakan untuk klasifikasi sistematika yang dapat digunakan untuk memprediksi probabilitas keanggotaan dalam suatu *class*. Pada penelitian ini proses klasifikasi dilakukan dengan cara memasukkan data ke dalam *tools Jupyter Notebook* dan membuat rancangan proses penelitian. Dataset yang diambil oleh ibu Saptarum di Klinik Bidan Saptarum Maslahah Kabupaten Jombang dengan jumlah 50 data akan diolah dengan Algoritma KNN dan Naive Bayes. Tahap akhir menjadikan file dalam bentuk *DataPickle* agar dapat direalisasikan kedalam sistem. Adapun hasil nilai akurasi Algoritma KNN dengan K=3 memiliki nilai sebesar 93%, sedangkan algoritma Naive Bayes memiliki akurasi sebesar 95%.**

***Kata Kunci*— Diabetes, Klasifikasi, Algoritma KNN, Algoritma Naive Bayes, Data Pickle.**

I. PENDAHULUAN

Diabetes adalah penyakit yang tidak menular dan termasuk cukup serius bagi manusia dikarenakan pankreas tidak mampu menghasilkan insulin secara optimal [1]. Diabetes dapat disebabkan oleh berbagai variabel antara lain tekanan darah, tinggi, obesitas, riwayat keluarga yang terkena diabetes, umur, gaya hidup dan makanan yang tidak sehat [2]. Salah satu penyebab terjadinya penyakit diabetes, orang yang mempunyai riwayat keluarga diabetes jauh lebih besar beresiko menderita diabetes dibandingkan yang tidak memiliki riwayat diabetes. Dalam menyikapi permasalahan tersebut, perlu dilakukan pendeteksian sejak dini terhadap penyakit diabetes. Deteksi sejak dini diharapkan mampu menurunkan resiko komplikasi penderita diabetes. Dengan menganalisa pasien diabetes sejak dini, maka pencatatan terhadap penyakit dilakukan sebagai langkah pencegahan. Diabetes termasuk jenis penyakit adanya garis keturunan. Hal tersebut bisa dikatakan bahwa seorang anak memiliki resiko sebesar 15% apabila dari salah satu orang tuanya menderita diabetes dan resiko tersebut akan meningkat hingga 75% jika keduanya mengalami diabetes [3]. Seseorang yang didalam keluarganya memiliki penyakit diabetes, maka kemungkinan besar orang tersebut terkena diabetes [4]. Gejala awal diabetes memang seringkali tidak terlihat. Dampak yang ditimbulkan penyakit diabetes antara lain kebutaan, amputasi, dan gagal ginjal dikarenakan kurangnya pemahaman terhadap masyarakat mengenai bahaya penyakit diabetes [5]. Salah satu pencatatan dapat dilakukan dengan memanfaatkan teknik klasifikasi menggunakan *data mining* [6]. *Data mining* adalah sebuah metode yang digunakan guna melakukan akuisisi pengetahuan. Oleh sebab itu data mining membuat informasi-informasi menjadi implisit dan berharga dari sebuah data supaya dapat diekstrak. Adapun metode yang biasanya operasikan pada data mining antara lain: deskripsi atau penggambaran, prediksi, atau ramalan, clustering, klasifikasi dan asosiasi, dan estimasi.

Klasifikasi adalah pengelompokkan suatu objek ke dalam data kelas dengan menngambil *data training* untuk membuat sebuah model. Model yang telah dirancang digunakan untuk memprediksi label dalam suatu kelas data terbaru yang belum di ketahui [7]. Pada penelitian tersebut, maka klasifikasi dapat dimanfaatkan untuk memprediksi pasien yang menderita penyakit diabetes maupun tidak diabetes. Adapun algoritma digunakan untuk melakukan pengujian pada proses klasifikasi. Adapun algoritma klasifikasi adalah Algoritma *K-Nearest Neighbor* (KNN) dan *Naive Bayes*. KNN adalah metode yang dilakukan dalam klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek. Keunggulan dari KNN yaitu lebih efektif di data training yang besar sehingga data yang dihasilkan lebih akurat. Pada penelitian sebelumnya sudah ada yang menggunakan data mining terkait deteksi diabetes retinopati. Tingkat akurasi yang diperoleh dengan menggunakan metode KNN memiliki akurasi dengan hasil terbaik dengan parameter nilai K=9 sebesar 65% [8]. Adapun pengertian dari Algoritma *Naive Bayes* adalah metode yang membagi suatu permasalahan ke dalam berbagai jenis kelas berdasarkan persamaan dan perbedaan menentukan statistik guna memprediksi nilai probabilitas sebuah kelas [9]. *Naive Bayes* memiliki beberapa kelebihan dalam pelatihan dan penggunaan, maka dari itu akurasi yang dihasilkan relatif tinggi dan metode tersebut membutuhkan jumlah *data training* yang kecil agar dapat memastikan estimasi parameter diperlukan dalam proses pengklasifikasian. *Naive Bayes* memiliki nilai akurasi paling tinggi dalam memprediksi penyakit diabetes

yang dimana tingkat akurasinya sebesar 75% dengan menggunakan pengujian *confusion matriks* [10]. Oleh sebab itu dengan melakukan pegujian diabetes menggunakan KNN dan Naïve Bayes dapat menghasilkan dataset dengan hasil tingkat akurasi yang terbaik.

Internasional Diabetes Federation (IDF) memperkirakan 2019 sekitar 463 juta orang dengan rentang usia 20 hingga 79 tahun terkena diabetes dengan nilai prevalensi sebesar 9,3% dari total penduduk dengan umur yang sama. Menurut jenis kelamin, IDF mempunyai perkiraan prevalensi diabates melittus tahun 2019 pada perempuan sebesar 9% dan pada laki-laki sebesar 9,65%. Prevalensi diabetes diprediksi akan terus naik seiring peningkatan umur penduduk menjadi 19,9% juta di tahun 2030 dan 700 juta tahun 2045 [11]. Berdasarkan permasalahan diatas, maka dilakukan penelitian supaya membantu mengatasi permasalahan dalam mengklasifikasi penyakit diabetes sehingga dibutuhkan sebuah metode dapat mengolah dataset yang sudah ada. Penggunaan data mining algoritma KNN dan *Naive Bayes* termasuk langkah untuk klasifikasi diabetes melittus guna menjadi alternatif pilihan yang tepat dalam melakukan pengujian, akan tetapi masih belum diketahuhi manakan algoritma paling tepat dalam mendeteksi penyakit diabetes [9]. Pada penelitian tersebut akan melakukan komparasi atau membandingkan *data mining* algortima KNN dan *Naive Bayes* agar dapat mengetahui algoritma mana yang mempunyai tingkat akurasi yang tinggi dalam klasifikasi penyakit diabetes. Berdasarkan suatu hal yang sudah dijelaskan, maka peneliti memberikan judul "Rancang Bangun Aplikasi Klasifikasi Penderita Penyakit Diabetes Berbasis Machine Learning Menggunakan Algoritma KNN dan Naive Bayes".

II. METODOLOGI PENELITIAN

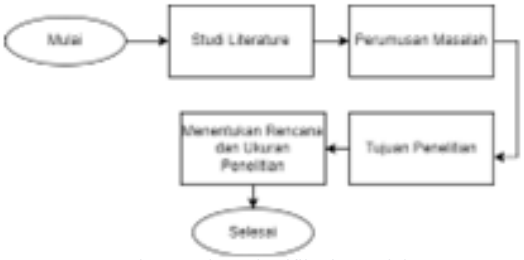
Pendekatan utama penelitian yaitu dengan menggunakan pendekatan kualitatif dan pendekatan kuantitatif. Tujuan penelitian adalah melakukan pengujian klasifikasi dan evaluasi model algoritma KNN dan Naive Bayes untuk mengetahui akurasi dari kedua algoritma dalam mengklasifikasikan penyakit diabetes.

Gbr. 1. Tahap Penelitian

Penelitian ini memiliki berbagai tahapan pada prosesnya agar mendapatkan hasil dalam suatu penelitian yang berhasil seperti diuraikan pada Gbr. 1. Adapun penjelasan tiap tahapan akan dijelaskan sebagai berikut:

A. Identifikasi Masalah

Tahap pertama dalam penelitian merupakan identifikasi masalah yaitu penulis menganalisis permasalahan yang ada di sekitar seperti data penderita penyakit diabetes. Permasalahan data tersebut yang sudah diberikan oleh ahli pakar akan diolah oleh penulis untuk menentukan data tersebut termasuk diabetes atau tidak. Menindaklanjuti dari permasalahan tersebut maka akan lebih baiknya ada sebuah sistem rekomendasi untuk klasifikasi penyakit diabetes agar tidak sampai terjadi kesalahan dalam melakukan pencatatan penderita penyakit diabetes. Adapun flowchart tahap identifikasi sebagai berikut.



Gbr. 2. Tahap Identifikasi Masalah

B. Penentuan Algoritma Penelitian

Tahap kedua yaitu penentuan algoritma penelitian. Dalam penelitian akan membandingkan kedua metode yaitu Algoritma K-Nearest Neighbor dan Algoritma Naive Bayes.

C. Pengumpulan Dataset

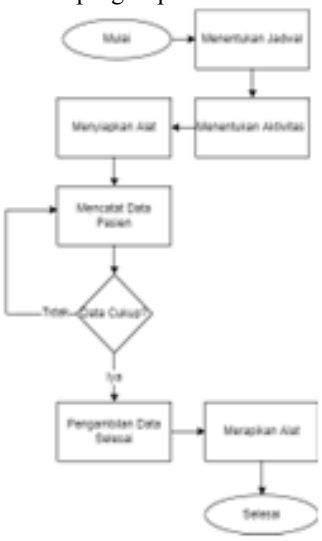
Tahapan selanjutnya yaitu pengumpulan *dataset*. Sumber data pada penelitian adalah menggunakan *dataset* Klinik Bidan Sapratum Maslahah Jombang yang telah diuji oleh Ibu Saptarum Maslahah Amd. Keb sebagai ahli pakar. Variabel

yang akan digunakan dalam penelitian sebanyak 8 variabel dari 50 pasien. Dasar dalam penentuan mengambil 8 variabel yang digunakan sebagai atribut penilaian sesuai penelitian yang sudah dilakukan oleh Indrayanti, Devi Sugianti, dan M. Adib Al Karomi pada tahun 2017 [12]. Adapun detail atribut dataset Klinik Bidan Saptarum Maslahah ditunjukkan berikut :

TABEL I
ATRIBUT DATASET

Atribut	Deskripsi	Tipe Data
Pregnancies	Jumlah Kehamilan	Integer
Glucose	Kadar Glukosa	Integer
BloodPressure	Tekanan Darah	Integer
SkinThickness	Ketebalan Kulit	Integer
Insulin	Insulin	Integer
BMI	Berat Tubuh	Integer
Riwayat Diabetes	Riwayat keturunan	Integer
Age	Tahun	Integer

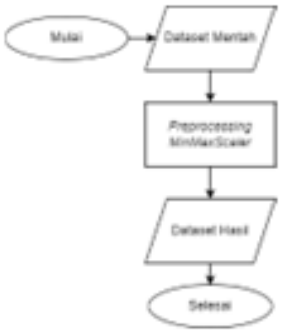
Pengambilan dataset yang sudah dilakukan bulan Februrari 2022 di Klinik Bidan Saptarum Maslahah kecamatan Jombang. Adapun flowchart dari pengumpulan dataset sebagai berikut:



Gbr. 3. Tahap Pengumpulan Data

D. *Preprocessing*

Tahap *preprocessing* adalah langkah awal yang dilakukan untuk mendapatkan kualitas data agar hasilnya lebih maksimal. *Preprocessing* ini dilakukan dengan cara normalisasi *feature scaling* menggunakan *MinMaxScaler*. Tujuan *preprocessing* yaitu supaya mudah mengelola data dalam proses pengujian, identifikasi dan pemilihan atribut, pemrosesan atribut yang hilang. *Dataset* tersebut akan dinormalisasi dan diubah ke range 0 – 1 karena nilainya sangat beragam [13]. Jadi sebab dari normalisasi tersebut agar memperoleh nilai antara 0 dan 1. Adapun rumus persamaan dalam menghitung tahapan *minmax* sebagai berikut :



Gbr. 4. Tahap Preprocessing

E. *Metode K-Nearest Neighbor*

Konsep KNN dapat digunakan guna mengklasifikasikan objek berdasarkan nilai k. Metode ini memerlukan ukuran jarak untuk menentukan suatu kedekatan pada objek, sehingga objek data uji akan diklasifikasikan dengan tetangga yang jaraknya lebih dekat. KNN dapat memberikan keputusan dalam pengelompokan data latih dalam jumlah besar agar dapat mendapatkan hasil yang baik. Adapun *tools* yang digunakan yaitu menggunakan *GridSearchCV* untuk diterapkan dalam dataset baru yang ingin ditraining maupun ditesting menggunakan *algoritma KNN*. Metode dalam penelitian ini menggunakan *Euclidean Distance* dan *Manhattan Distance*. Adapun cara dalam menghitung *Euclidean Distance* sesuai dengan persamaan (1) dan *Manhattan Distance* dengan persamaan (2).

$$distance = \sqrt{\sum_{i=1}^{n} (X^i_{training} - X_{testing})} \quad (1)$$

$$distance = \sum_{i=1}^{n} |X^i_{training} - X_{testing}| \quad (2)$$

Keterangan:

$X^i_{training}$ = data train pertama

$X_{testing}$ = data test

i = record (baris) ke-i dalam tabel

n = jumlah data train

Adapun diagram alur dari *metode K-Nearest Neighbor* akan ditunjukkan pada Gbr. 5.

Gbr. 5. Metode K-Nearest Neighbor

F. *Metode Naive Bayes*

Metode berikutnya yaitu *Metode Naive Bayes* dengan menggunakan pendekatan *GaussianNB*. Metode tersebut menggunakan nilai probabilitas dan statistik seperti dipaparkan ahli ilmuan Thomas Bayes. Adapun *tools* dalam membuat model *Naive Bayes* menggunakan fungsi *GaussianNB* untuk menerapkan pengujian pada dataset baik di *training* maupun di *testing*. Adapun rumus dalam menghitung Naive Bayes seperti persamaan (3) berikut.

$$P(H|X) = \frac{P(X|H).P(H)}{P(X)} \quad (3)$$

Keterangan:

X	: Data *dass* belum ditemukan
H	: Hipotesis data
P (H | X)	: Probabilitas hipotesis berdasarkan kondisi x
P (X | H)	: Probabilitas berdasarkan kondisi hipotesis
P (H)	: Probabilitas hipotesis H
P (X)	: Probabilitas X

Adapun diagram alur metode *Naive Bayes* akan ditunjukkan Gbr. 6.

Gbr. 6. Metode Naive Bayes

G. *Pengembangan Sistem*

Pengembangan sistem merupakan tahapan dalam membuat sistem yang akan dirancang menuju pada tujuan dari pembuatan artikel ilmiah ini sehingga dapat menjadikan sebuah sistem deteksi diabtes. Pada tahap pengembangan sistem terdapat analisis sistem dan desain sistem.

1) *Analisis Sistem*

a. Identifikasi Masalah

Berdasarkan permasalahan tersebut, maka dapat diketahui bagaimana cara merealisasikan sistem deteksi diabetes menggunakan metode *K-Nearest Neighbor* dengan pendekatan *GridSearchCV* dan menggunakan metode *Naive Bayes* pendekatan *GaussianNB*.

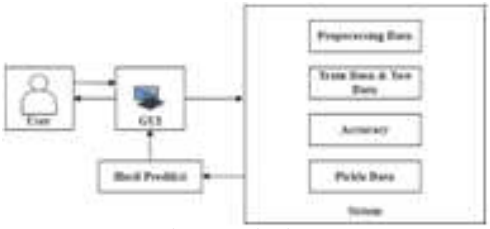
b. Analisis Kebutuhan Fungsional

Sistem untuk mendeteksi diabetes dalam penelitian ini dapat melakukan beberapa hal seperti:

- Pengguna dapat mengisi form deteksi diabetes dengan memasukkan nilai banyaknya jumlah melahirkan.
- Pengguna dapat mengisi form deteksi diabetes dengan memasukkan nilai kadar glukosa.
- Pengguna dapat mengisi form deteksi diabetes dengan memasukkan nilai tekanan darah.
- Pengguna dapat mengisi form deteksi diabetes dengan memasukkan nilai tebal kulit.
- Pengguna dapat mengisi form deteksi diabetes dengan memasukkan nilai kadar insulin.
- Pengguna dapat mengisi form deteksi diabetes dengan memasukkan nilai BMI.
- Pengguna dapat mengisi form deteksi diabetes dengan memasukkan nilai riwayat diabetes.
- Pengguna dapat mengisi form deteksi diabetes dengan memasukkan nilai usia.

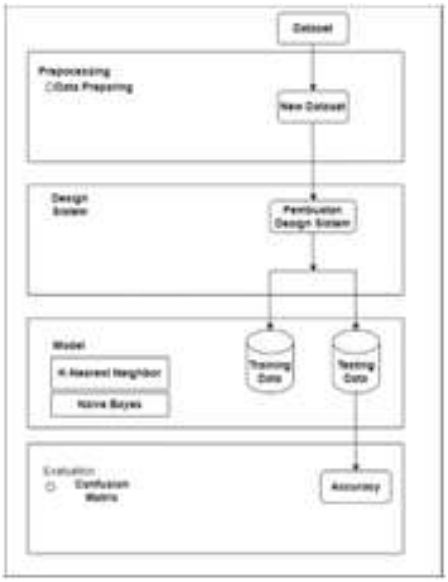
- Sistem dapat menampilkan 2 hasil prediksi yang berbeda prediksi penderita penyakit diabetes.

2) *Desain Sistem*



Gbr. 7 Desain Sistem

Dari Gbr. 7 dijelaskan bahwasanya proses awal dengan pengguna diarahkan ke GUI atau menampilkan program yang berisi *input* banyaknya melahirkan, kadar glukosa, tekanan darah, tebal kulit, kadar insulin, bmi, riwayat diabetes, dan umur. Setelah pengguna memasukkan data, sistem akan memproses data masukan pengguna dengan model *GridSearchCV* dan *GaussianNB* yang telah dilatih. Setelah memasukkan dataset, maka dapat menampilkan hasil prediksi dari kedua algoritma. Adapun bagian dari berjalannya proses sistem ada di Gbr. 8. Berikut *flowchart* dari proses sistem.



Gbr. 8 Diagram Alur Sistem

Gbr. 8 merupakan *flowchart* menggambarkan tentang proses sistem yang akan dijalankan oleh *user*. Jadi, proses berlangsung dengan diawali *import* dataset yang akan diolah dahulu pada tahap *preprocessing* untuk mendapatkan kualitas data agar *maksimal*. Setelah selesai dari tahap *preprocessing,* proses pengujian dilanjutkan dengan membagi *dataset* tergolong 2 bagian yaitu *data training* dan *data testing*. Dalam pengambilan *data training* yang diambil yaitu jumlah melahirkan, kadar glukosa, tekanan darah, tebal kulit, kadar insulin, bmi, dan riwayat diabetes. Dalam pengambilan *data testing* yang di ambil hanya *outcome* atau hasil. Setelah melakukan pembagian dataset, maka data akan diproses dengan proses normalisasi *featur scaling* menggunakan *MinMaxScale*. Selanjutnya masuk ke algoritma dari machine learning KNN dan Naive Bayes agar dapat mengetahui akurasi dari kedua metode tersebut. Setelah itu melakukan *confusion matrix* untuk mengukur hasil akurasi sistem menggunakan persamaan sebagai berikut:

$$Akurasi = \frac{TP+TN}{TP+FP+TN+FN} \times 100\% \quad (4)$$

Selanjutnya penelitian yang sudah dilakukan menggunakan *tools jupyter notebook* dengan bahasa *pyhton* akan disimpan dalam model *pickle* agar dapat menerapkan *machine learning* kedua algoritma ke dalam sistem yang sudah dibuat.

III. HASIL DAN PEMBAHASAN

Penelitian menghasilkan Sistem Deteksi Penyakit Diabetes menggunakan Metode KNN dengan pendekatan *GridSearchCV* dan Metode *Naive Bayes* dengan pendekatan *GaussianNB* dimana proses terbagi menjadi 4 yaitu *preprocessing,* proses dari *metode K-Nearest Neighbor*, proses dari metode *Naive Bayes*, serta pengujian sistem. Berikut penjelasan hasil dari 4 proses tersebut:

A. *Hasil Preprocessing*

Dataset yang diperoleh masih dalam data mentah berisi 50 *dataset* penderita penyakit diabetes seperti jumlah kehamilan, kadar glukosa, tekanan darah, tebal kulit, insulin, bmi, riwayat diabetes, dan umur yang mana masih perlu perbaikan pada beberapa atributnya agar data dapat diproses oleh algoritma. Data yang masih kosong memiliki beberapa kolom yang perlu diubah dan memiliki tipe yang belum tepat sebagai berikut:

TABEL III
DATA YANG BELUM DIPROSES

No	Kehamilan	Glukosa	Tekanan Darah	Tebal Kulit	Insulin	BMI	Riwayat	Umur
1	1	126	56	29	152	28.7	0.801	21
2	1	96	122	0	0	22.4	0.207	27
3	4	144	58	28	140	29.5	0.287	37
4	3	83	58	31	18	34.3	0.336	25
5	3	195	85	25	36	37.4	0.247	47

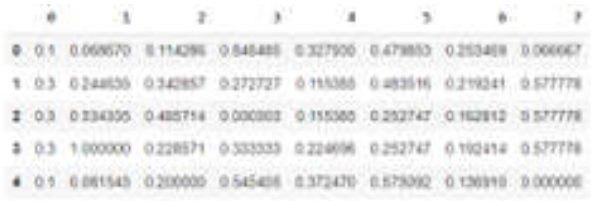
Dari Tabel II diketahui bahwa masih terdapat data yang perlu diubah sehingga perlu melakukan tahap pengujian data dengan *tools jupyter notebook* menggunakan bahasa pemrograman *python*. Proses pertama dari *preprocessing* adalah normalisasi *featur scaling* menggunakan *MinMaxScaler*. Berikut hasil dari *preprocessing* tahap *featur scaling* dengan menggunakan *MinMaxScaler* ditunjukkan pada Gbr. 7.

```
from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler(feature_range = (0,1))
x = scaler.fit_transform(x)
x = pd.DataFrame(x)
x.head()
```

Gbr. 9 Pre-processing Tahap Featur Scaling

Setelah itu tahap mengganti tipe data dalam atribut harga dari string menjadi integer. Berikut hasil dari preprocessing tahap ubah tipe data ditunjukkan pada Gbr. 10.



Gbr. 10. Pre-processing Tahap Ubah Tipe Data Atribut

Setelah itu tahap pembagian dataset yang berjumlah 50 data terbagi menjadi 2 bagian yaitu *data training* dan *data testing*. Adapun pembagian *data training* sebesar 90% dengan jumlah 45 data, sedangkan *data testing* sebesar 10% dengan perbandingan jumlah data uji yang lebih kecil dibanding data latih yaitu berjumlah 5 data. Tujuan dari pembagian data training lebih besar dari data testing supaya sistem dapat mengenali permodelan yang dimiliki oleh data tersebut. Berikut hasil dari tahap pembagian data ditunjukkan pada Gbr. 11.

Gbr. 11 Tahap Pembagian Data

B. Hasil *K-Nearest Neighbor*

Tahap awal yaitu melakukan pengujian *cross validation* untuk menentukan jumlah K dari beberapa variasi K yang digunakan hingga mendapatkan akurasi terbaik. Tahap ini perlu dilakukan untuk mendapatkan model *K-Nearest Neighbor* dengan menggunakan pengujian K yang mempunyai nilai data training lebih besar daripada data uji. Pada pengujian K ditentukan nilai K dari 2-fold hingga 10-*fold*. Hasil pengujian K-fold *cross validation* dilakukan dalam pengujian metode *K-Nearest Neighbor* dapat ditampilkan pada tabel berikut.

TABEL III
HASIL PENGUJIAN K-NEAREST NEIGHBOR

K	Akurasi Data Train (%)	Akurasi Data Test (%)
2	95 %	80 %
3	95 %	80 %
4	86 %	80 %
5	88 %	80 %
6	84 %	80 %
7	88 %	80 %
8	84 %	100 %
9	86 %	80 %

Berdasarkan data pengujian menggunakan banyak jumlah parameter, model KNN dengan nilai K=3 memperoleh nilai akurasi paling tinggi (95%). Sehingga model yang akan digunakan adalah KNN dengan K=3. Model ini yang akan diberikan proses *training* dan disimpan untuk digunakan dalam perhitungan rekomendasi di aplikasi deteksi diabetes. Adapun *script* dari proses dalam menggunakan algoritma *K-Nearest Neighbor* tahap perhitungan *GridSearchCV* yang sudah pernah dijelaskan oleh Tuomas Tanner & Hannu Toivonen tahun 2021 [14] dengan rumus sebagai berikut:

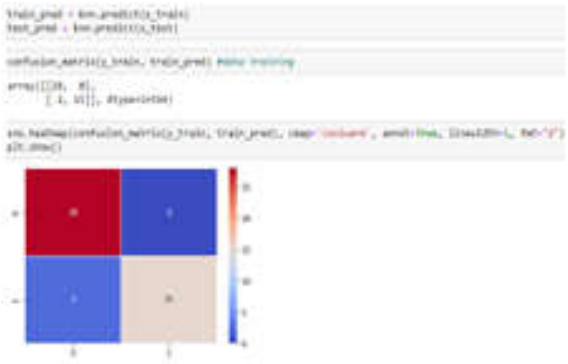
```
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(x_train,y_train)

knn_acc = accuracy_score(y_test,knn.predict(x_test))
```

Setelah memasukkan rumus berikut, maka dapat menampilkan hasil prediksi dengan script berikut.

Gbr. 12. Script K-Nearest Neighbor

Setelah melakukan proses *K-Nearest Neighbor* dengan perhitungan *GridSearchCV* untuk menentukan hasil prediksi, maka dilakukan tahap uji terakhir yaitu *confusion matrixs* supaya dapat menentukan jumlah data pengujian telah sesuai diklasifikasikan dan tidak sesuai diklasifikasikan. Berikut hasil *confusion matrix* dari *data training* dan *data testing* dari proses KNN pada Gbr. 13.

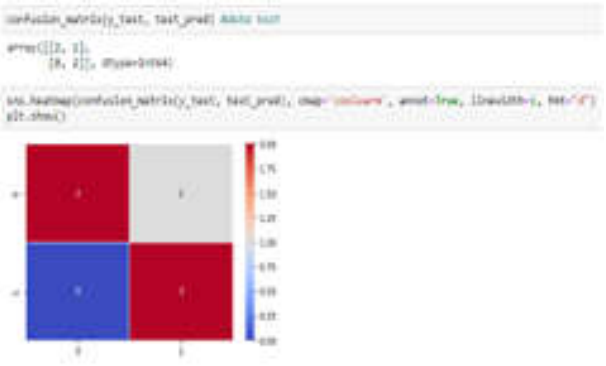


Gbr. 13 Hasil Confusion Matrix Data Train K-Nearest Neighbor

Berdasarkan diagram Gbr. 13. diketahui terdapat 42 data (TP+TN) diklasifikasikan valid (accurate) dan terdapat 3 data

yang diklasifikasikan tidak valid (error) dari total 45 data. Maka dapat dihitung nilai *accuracy* dan *error* sebagai berikut:

- Accuracy = (TP + TN) / Total Data
  = (28 + 15) / 45
  = (43) /45
  = 0.95 x 100%
  = 95 %
- MR = (FP + FN) / Total Data
  = (2 + 0) / 45
  = (2) / 45
  = 0.04 x 100%
  = 4 %



Gbr. 14. Hasil Confusion Matrix Data Test K-Nearest Neighbor

Berdasarkan diagram Gbr. 14. diketahui terdapat 4 data (TP+TN) diklasifikasikan valid (accurate) dan terdapat 1 data yang diklasifikasikan tidak valid (error) dari total 5 data. Maka dapat dihitung nilai *accuracy* dan *error* sebagai berikut:

- Accuracy = (TP + TN) / Total Data
  = (2 + 2) / 5
  = (4) / 5
  = 0.8 x 100 %
  = 80 %
- MR = (FP + FN) / Total Data
  = (0 + 1) / 5
  = (1) / 5
  = 0.2 x 100%
  = 20 %

C. Hasil Algoritma *Naive Bayes*

Tahap yang dilakukan tidak jauh berbeda seperti penelitian sebelumnya yang menggunakan metode K-Nearest Neighbor, akan tetapi untuk hasil akurasi dari metode Naive Bayes akan memiliki nilai yang berbeda dengan metode sebelumnya. Tingkat akurasi yang di peroleh menggunakan metode *Naive Bayes* memiliki akurasi hasil terbaik sebesar 93 %. Berikut ini hasil dari proses menggunakan algoritma *Naive Bayes* tahap perhitungan *GaussianNB* yang sudah pernah dijelaskan dan digunakan oleh Nanda Harsana Octavya tahun 2021 dengan rumus sebagai berikut:

```
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

gnb = GaussianNB()
gnb.fit(x_train,y_train)

gnb_acc = accuracy_score(y_test,gnb.predict(x_test))
```

Setelah memasukkan rumus diatas, maka dapat menampilkan hasil prediksi dengan script berikut.

```
print("Train Set Accuracy:"+str(accuracy_score(y_train,gnb.predict(x_train))*100))
print("Test Set Accuracy:"+str(accuracy_score(y_test,gnb.predict(x_test))*100))

Train Set Accuracy:93.33333333333333
Test Set Accuracy:100.0
```

Gbr. 15. Script Metode Naive Bayes

Setelah melakukan proses *Naïve Bayes* dengan perhitungan *GaussianNB* untuk menentukan hasil prediksi, maka dilakukan tahap pengujian yang terakhir yaitu *confusion matrixs* supaya dapat menyatakan jumlah data uji yang telah benar diklasifikasikan dan djumlah data uji yang salah diklasifikasikan. Berikut hasil *confusion matrix* dari *data training* dan *data testing* dari proses *Naïve Bayes* pada Gbr. 16.

Gbr. 16. Hasil Confusion Matrix Data Train Naive Bayes

Berdasarkan diagram Gbr. 16. diketahui terdapat 42 data (TP+TN) diklasifikasikan valid (accurate) dan terdapat 3 data yang diklasifikasikan tidak valid (error) dari total 45 data. Maka dapat dihitung nilai *accuracy* dan *error* sebagai berikut:

- Accuracy = (TP + TN) / Total Data
  = (27 + 15) / 45
  = (42) /45
  = 0.93 x 100 %

= 93 %

- MR = (FP + FN) / Total Data
  = (1 + 2) / 45
  = (3) / 45
  = 0.06 x 100 %
  = 6.6 %

Gbr. 17. Hasil Confusion Matrix Data Test Naive Bayes

Berdasarkan diagram Gbr. 17. diketahui terdapat 5 data (TP+TN) diklasifikasikan valid (accurate) dan tidak terdapat data yang diklasifikasikan tidak valid (error) dari total 5 data. Maka dapat dihitung nilai *accuracy* dan *error* sebagai berikut:

- Accuracy = (TP + TN) / Total Data
  = (3 + 2) / 5
  = (5) / 5
  = 1 x 100 %
  = 100 %
- MR = (FP + FN) / Total Data
  = (0 + 0) / 5
  = (0) / 5
  = 0 x 100%
  = 0 %

D. *Hasil Pengujian Sistem*

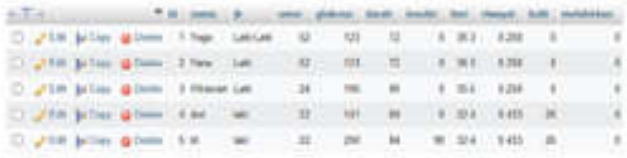
Sebelum melakukan tahap pengujian, maka *file* penelitian yang sudah dikerjakan menggunakan *Google Colabs* akan di *export* menjadi *file pickle* agar bisa diterapkan dalam suatu *system* yang sudah dibuat. Cara tersebut sangat sederhana dengan menyimpan model hasil training pada suatu file yang mana file tersebut dapat digunakan kembali di masa depan sebagai otak dari aplikasi yang telah dikembangkan. Setelah model terbentuk maka dibuat tampilan antarmuka yang dapat menghubungkan pengguna (*user*) dengan proses yang berlangsung. Adapun sistem dibentuk dengan menggunakan bahasa pemrograman *python*. *Visual Studio Code* menjadi platform yang penulis gunakan untuk mengembangkan aplikasi deteksi penyakit diabetes yang berbasis *python*. Berikut adalah tampilan dari halaman beranda sistem deteksi penyakit diabetes:

Gbr. 18. Beranda Sistem Deteksi Diabetes

Pada tampilan sistem di Gbr. 18. mulai memasuki beranda sistem deteksi diabetes dimana terdapat form untuk *user* mengisi proses *input* data.

Gbr. 19.Proses Input Sistem Deteksi Diabetes

Pada tampilan sistem di Gbr. 19. merupakan proses pengujian berlangsung dimana *user* dapat memasukkan data pasien penderita penyakit diabetes. Setelah itu *user* akan mendapatkan hasil prediksi penderita penyakit diabetes yang nantinya akan tersambung kedalam database seperti pada Gbr. 20.



Gbr. 20. Hasil database dari input sistem

Berikut ini adalah tampilan dari halaman hasil prediksi penderita penyakit diabetes sesuai dengan inputan pengguna seperti pada Gbr. 21.

Gbr. 21. Tampilan Hasil Deteksi Diabetes

Pada tampilan hasil rekomendasi di Gbr. 21. ditunjukkan hasil deteksi penderita penyakit diabetes berdasarkan input dari pengguna. Beberapa hasil pengujian deteksi diabetes juga ditunjukkan pada tabel IV berikut.

TABEL IV
HASIL PENGUJIAN SISTEM DETEKSI DIABETES MENGGUNAKAN KNN DAN NAIVE BAYES

No.	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10
1.	1	108	60	46	178	35,5	0,415	24	T	T
2.	3	190	76	27	0	35.6	0.378	47	T	T
3.	3	325	86	19	0	29.3	0.317	47	T	T
4.	3	542	68	29	127	29.3	0.349	47	T	T
5.	1	114	66	36	200	38.1	0.289	21	T	T
6.	5	117	86	30	105	39.1	0.251	42	T	T
7.	1	111	94	0	0	32.8	0.265	45	T	T
8.	4	112	78	40	0	39.4	0.236	38	T	T
9.	1	116	78	29	180	36.1	0.496	25	T	T
10.	0	141	84	26	0	32.4	0.433	22	T	T

11.	2	175	88	0	0	22.9	0.326	22	F	T
12.	2	92	52	0	0	30.1	0.141	22	T	T
13.	3	225	78	23	79	28.4	0.323	34	T	T
14.	2	240	86	0	0	28.4	0.259	22	T	T
15.	2	174	88	37	120	44.5	0.646	24	T	T
16.	2	106	56	27	165	29	0.426	22	T	T
17.	2	105	75	0	0	23.3	0.56	53	T	T
18.	4	95	60	32	0	35.4	0.284	28	T	T
19.	0	126	86	27	120	27.4	0.515	21	T	T
20.	1	325	90	51	220	49.7	0.325	47	T	T
21.	3	190	72	0	0	39	1.222	47	T	T
22.	1	151	60	0	0	26.1	0.179	22	T	T
23.	0	125	96	0	0	22.5	0.262	21	T	T
24.	1	81	72	18	40	26.6	0.283	24	T	T
25.	2	85	65	0	0	39.6	0.93	27	T	T
26.	1	126	56	29	152	28.7	0.801	21	T	T
27.	1	96	122	0	0	22.4	0.207	27	T	T
28.	4	144	58	28	140	29.5	0.287	37	T	T
29.	3	83	58	31	18	34.3	0.336	25	T	T
30.	3	195	85	25	36	37.4	0.247	47	T	T
31.	3	171	72	33	15	33.3	0.199	24	T	T
32.	8	155	62	26	495	34	0.543	46	T	T
33.	1	89	76	34	37	31.2	0.192	23	T	T
34.	4	76	62	0	0	34	0.391	25	T	T
35.	7	160	54	32	175	30.5	0.588	39	F	T
36.	0	181	88	44	510	43.4	0.222	26	T	T
37.	8	154	78	32	0	32.4	0.443	45	T	T
38.	1	128	88	39	110	36.5	1.057	37	F	T
39.	3	137	90	41	0	32	0.391	39	T	T
40.	0	123	72	0	0	36.3	0.258	52	T	F
41.	1	106	76	0	0	37.5	0.197	26	T	T
42.	6	190	92	0	0	35.5	0.278	66	T	T
43.	2	88	58	26	16	28.4	0.766	22	T	T
44.	9	170	74	31	0	44	0.403	43	T	T
45.	9	89	62	0	0	22.5	0.142	33	T	T
46.	10	101	76	48	180	32.9	0.171	63	T	F
47.	2	122	70	27	0	36.8	0.34	27	T	T
48.	5	121	72	23	112	26.2	0.245	30	T	T
49.	1	126	60	0	0	30.1	0.349	47	T	F
50.	1	93	70	31	0	30.4	0.315	23	T	T

Keterangan:

- X1 = Banyak Kehamilan
- X2 = Kadar Glukosa
- X3 = Tekanan Darah
- X4 = Tebal Kulit
- X5 = Insulin
- X6 = BMI
- X7 = Riwayat Diabetes
- X8 = Umur
- X9 = Hasil KNN
- X10 = Haisl Naive Bayes
- T = True (Prediksi Benar)
- F = False (Prediksi Salah)

Dari tampilan tabel IV terdapat 2 hasil pengujian dataset memiliki prediksi yang berbeda terkait deteksi penderita penyakit diabetes. Terdapat persentase tingkat prediksi dari masing-masing data yang merupakan hasil prediksi terhadap input dari pengguna. Adapun metode *K-Nearest Neighbor* memiliki hasil akurasi sebesar 95% daripada metode *Naive Bayes* yang hanya memiliki nilai akurasi sebesar 93%.

IV. KESIMPULAN

Dari penelitian yang dilakukan dan berdasarkan pembahasan sebelumnya diperoleh kesimpulan sebagai berikut:

1. Pembuatan rancang bangun aplikasi deteksi diabetes dengan tahap pengujian metode KNN perhitungan *GridSearchCV* dan metode *Naive Bayes* perhitungan *GaussianNB*. Adapun *tools* yang digunakan yaitu menggunakan *Goolge Colab* dan *Visual Studio Code* dengan bahasa pemrograman *python* dalam pembuatan sistem deteksi diabtes.
2. Pembuatan aplikasi yang sudah dilakukan dengan metode KNN dan Naive Bayes memiliki hasil prediksi yang akurat sehingga menunjukkan akurasi KNN memiliki akurasi sebesar 93%, sedangkan Naive Bayes dengan akurasi 95%.

V. SARAN

Berdasarkan penelitian yang dilakukan tidak dipungkiri bahwa pasti ada kekurangan, sehingga perlu adanya saran. Adapun saran yang diberikan oleh penulis sebagai berikut:

1. Sistem deteksi penderita diabtes bisa dikembangkan lagi menggunakan metode lain dengan tujuan mendapatkan hasil akurasi yang lebih tinggi atau akurat.
2. Memperbanyak pengumpulan data dalam dataset karena jika variasi data training lebih banyak maka akan memperkuat hasil akurasi.

UCAPAN TERIMA KASIH

Terimakasih penulis mengucapkan kepada Tuhan Yang Maha Esa, atas karunia-Nya penulis dapat mengerjakan laporan penelitian dengan baik dan lancar. Tidak lupa mengucapkan banyak terimakasih ini ditujukkan kepada kedua orang tua, dan semua pihak yang sudah terlibat dalam terselesaikannya penelitian ini.

REFERENSI

[1] Y. Safitri & I. K. A. Nurhayati. (2019). Pengaruh Pemberian Sari Pati Bengkuang Terhadap kadar Glukosa Darah Pada Penderita Diabetes Melittus Tipe II usia 40-50 tahun di Kelurahan Bangkinang. Jurnal Ners. 3(1). 69-81

[2] L. I. Umikulsum, Y. N. Anis, & N. Cahyunu. (2021). Penerapan Metode *K-Nearest Neighbor* untuk Sistem Pendukung Keputusan Identifikasi Penyakit Diabetes Melittus. Jurnal Teknik Informatika dan Sistem Informasi. Vol 8. No. 5.

[3] Trisnawati, S. K., & Setyorogo, S. (2013). Faktor Risiko Kejadian Diabetes Melitus tipe II di Puskesmas Kecamatan Cengkareng Jakarta Barat Tahun 2012. Jurnal Ilmiah Kesehatan. 5(1). 6-11.

[4] Sukmaningsih, W.R., Heru Subariskasjono, S. K. M. & Werdani, K. E. (2016). Faktor Risiko Kejadian Diabetes Mellitus Tipe II di

Wilayah Kerja Puskesmas Purwodiningratan Surakarta. DISS. Universitas Muhammadiyah Surakarta.

[5] *WHO. (2021). 6 Facts On Diabetes. [Online]. Available : https://www.who.int/news-room/fact-sheets/detail/diabetes.* [Accessed : 11-Maret-2022]

[6] F. Maisa Hana. (2020). Klasifikasi Penderita Penyakit Diabetes Menggunakan Algoritma Decision Tree C4.5. Jurnal Sistem Komputer dan Kecerdasan Buatan. Vol. 4. No. 1.

[7] Y. Safitri & I. K. A. Nurhayati. (2019). Pengaruh Pemberian Sari Pati Bengkuang Terhadap kadar Glukosa Darah Pada Penderita Diabetes Melittus Tipe II usia 40-50 tahun di Kelurahan Bangkinang. Jurnal Ners. 3(1). 69-81.

[8] A. Suci, H. Sugondo, & N.R. Dadan. (2015). Analisis Perbandingan KNN dengan SVM untuk Klasifikasi Penyakit Diabetes Retinopati berdasarkan Citra Eksudat dan Mikroaneurisma. Jurnal Elkomika. Vol 3. No. 1.

[9] N. Nurdiana, & A. Abijar. (2020). Studi Komparasi Algoritma *Naive Bayes* untuk Klasifikasi Penyakit Diabetes Melittus. *Infotech Journal.*

[10] Fatmawati. (2016). Perbandingan Algoritma Klasifikasi *Data Mining Model C4.5* dan *Naive Bayes* untuk Prediksi Penyakit Diabetes. Jurnal Techno Nusa Mandiri, Vol 8. No. 1.

[11] *International Diabetes Federation (IDF) and DAR International Alliance.* (2021). *Diabetes and Ramadan: Practical Guidelines. Brussels. Belgium: International Diabetes Federation.*

[12] Indrayanti, Devi Sugianti, & M. Adib Al Karomi. (2017). Optimasi Parameter K Pada Algoritma K-Nearest Neighbour untuk Klasifikasi Penyakit Diabetes Mellitus. Prosiding SNATIF. Universitas Muria Kudus

[13] A. Ayu Dwi Sulistyawati & S. Mujiono. (2021). Penerapan Algoritma K-Medoids untuk Menentukan Segmentasi Pelanggan. Jurnal Sistem Informasi. Vol 10. No.3.

[14] T. Tanner & H. Toivonen. (2021). *Predicting anda preventing student failure – using the k-nearest neighbour method to predict student perfomance in an online course environment.* Helsinki: Findaland.