

Klasifikasi Penyakit Diabetes Melitus Menggunakan Metode Support Vector Machine (SVM)

Pungky Rosalya Putri¹, Ronggo Alit²

^{1,2} Jurusan Teknik Informatika/Program Studi S1 Teknik Informatika, Universitas Negeri Surabaya

¹pungky.20002@mhs.unesa.ac.id

³rongoalit@unesa.ac.id

Abstrak— Diabetes Melitus (DM) adalah salah satu penyakit kronis dimana tubuh pasien tidak mampu mengontrol meningkatnya kadar gula darah. Pada penelitian ini akan menunjukkan apakah pasien di diagnosa positif diabetes atau negative diabetes dengan menggunakan algoritma Support Vector Machine (SVM) termasuk dalam algoritma dari machine learning. Penyakit diabetes penting untuk di deteksi dini sebagai penanganan dan pencegahan kondisi medis. Penelitian ini bertujuan untuk membangun model prediktif sehingga mampu mengidentifikasi risiko diabetes terhadap pasien berdasarkan data medis. Penelitian ini memanfaatkan data sekunder yaitu mengambil data dari website *Kaggle*, dataset tersebut berjudul *Pima Indians Diabetes*, yang berisi informasi medis dengan 8 atribut yang digunakan yaitu *Kehamilan*, *Glukosa*, *Tekanan darah*, *Ketebalan kulit*, *Insulin*, *BMI*, *DiabetesPedigreeFunction*, dan *Umur*. Dalam penelitian ini melibatkan proses *Preprocessing data* untuk melakukan *pembersihan outliers*, *missing data value*, *cleaning data*, dan *menstandarisasi data*, serta *split data* dengan membagi data *training* (80%) dan data *testing* (20%).

Evaluasi model dilakukan menggunakan metrik akurasi, *precision*, *recall*, *f1-score*, *support* dan *confusion matrix* untuk mengukur seberapa andal kemampuan model dalam mendeteksi kasus diabetes melitus dengan baik. Selanjutnya hasil prediksi akan di implementasikan pada aplikasi sederhana menggunakan *Streamlit*, untuk memudahkan pengguna dalam menginput data pasien.

Hasil penelitian mampu menunjukkan bahwa model prediksi yang dihasilkan mampu mengidentifikasi pasien dengan ketepatan yang baik, sehingga memiliki potensi dalam mendukung deteksi dini penyakit diabetes melitus. Penelitian ini mampu memberikan kontribusi dalam pemanfaatan data medis khususnya pada penyakit diabetes melitus dengan harapan dapat membantu pengambilan keputusan dalam praktik kesehatan.

Kata Kunci— Diabetes Melitus, Support Vector Machine (SVM), *Pima Indians Diabetes*, *Prediksi*, *Evaluasi Model*.

I. PENDAHULUAN

Diabetes Melitus (DM) dianggap sebagai masalah kesehatan global yang signifikan. Secara umum DM merupakan penyakit kronis terjadinya peningkatan kadar gula/glukosa dalam tubuh (Muhammad Hilmy Haidar Aly, 2024). Beberapa dekade terakhir jumlah penderita diabetes melitus meningkat secara drastis dan di perkirakan akan terus

meningkat pada masa yang akan datang. Penyakit diabetes dianggap serius apabila resiko komplikasi tidak ditangani dengan baik. Identifikasi dini dan pengelolaan yang efektif menjadi krusial dalam menangani penyakit diabetes melitus.

Adapun beberapa faktor genetika/keturunan bahkan gaya hidup lingkungan yang tidak sehat sangat rentan terjadinya penyakit diabetes melitus. Diabetes melitus memiliki gejala meningkatnya kadar gula dengan rentan waktu yang lama pada sekelompok gangguan metabolisme. Tubuh pasien yang terdiagnosa diabetes tidak mampu memproduksi hormon insulin dengan baik, sehingga kadar gula dalam aliran darah mengalami peningkatan (Gunawan, Sugiarto dan Mardinto, 2020). Diabetes melitus dianggap sebagai penyakit kronis meningkatnya prevalensi komplikasi kronis dengan cepat, yang disebabkan oleh menurunnya *relative sensitivitas sel* terhadap insulin atau kondisi *hiperglikemia* karena ketiadaan absolut insulin, bahkan kematian seorang yang didiagnosa diabetes melitus sering disebabkan oleh komplikasi (Purnama et al., 2019).

Algoritma Support Vector Machine (SVM) mampu menangani data yang tidak linear dengan baik, oleh karena itu SVM dipilih sebagai metode klasifikasi yang menjadi karakteristik data medis termasuk data yang digunakan dalam diagnosis DM. Metode klasifikasi pada SVM memiliki nilai yang tinggi untuk memprediksi klasifikasi pada data. Algoritma tersebut memiliki 2 metode yaitu klasifikasi (*Support Vector Classification*) dan regresi (*Support Vector Regression*) (Hovi et al., 2022).

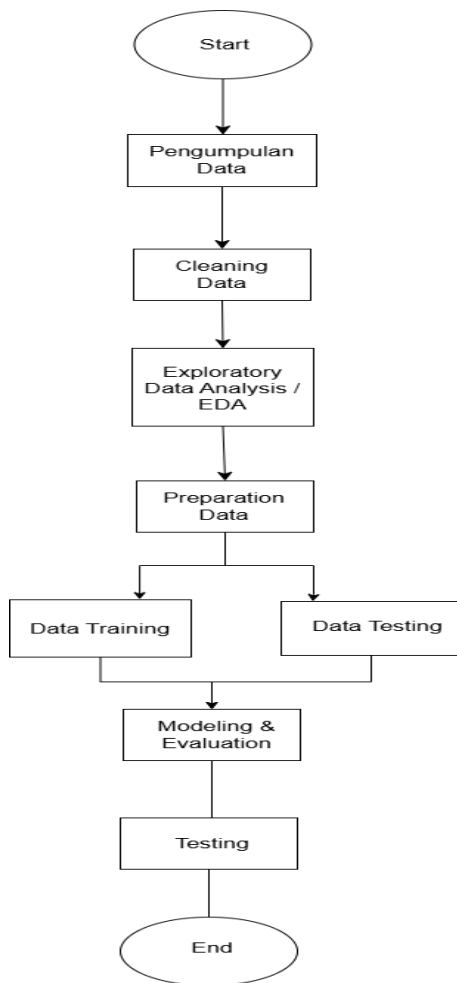
Klasifikasi merupakan pengelompokan data menjadi beberapa kelas, yang sebelumnya sudah menentukan label dan target (Cahyani & Basuki, 2019). SVM telah digunakan secara luas dalam bidang kedokteran untuk klasifikasi serta mampu menangani data yang tidak linear serta kontribusi yang signifikan untuk pengembangan sistem diagnosis yang lebih efisien dan canggih. Pemeriksaan untuk mendiagnosa dilakukan dengan pemeriksaan klinis menggunakan alat *glucometer* untuk memeriksa kadar gula pada pasien penderita diabetes (Aldama & Nasir, 2023).

Algoritma SVM pada klasifikasi DM menjadi relevan dan layak untuk dibahas lebih lanjut dengan mengangkat permasalahan yang diambil oleh penulis yang memiliki dampak langsung pada praktek klinis serta pengembangan

sistem pendukung dalam penanganan penyakit DM. Komplikasi penyakit diabetes melitus dianggap serius jika tidak ditangani dengan baik.

II. METODOLOGI PENELITIAN

Penelitian ini memanfaatkan data sekunder yaitu data yang berhubungan dengan informasi dari sumber yang telah ada sebelumnya sebagai pelengkap kebutuhan data penelitian. Adapun alur penelitian dapat dilihat pada gambar 1. Pada alur berikut dapat membantu dalam memahami metodologi dan pendekatan pada penelitian (Algoritma et al., 2024).



Gbr.1 Alur Penelitian

A. Data set

Penulis menggunakan dataset yang berasal dari *National Institute of Diabetes and Digestive and Kidney Diseases*. Dengan judul dataset Pima Indians Diabetes diperoleh dari UCI Machine Learning Repository (Agatsa et al., 2020). Dataset tersebut sering digunakan dalam berbagai penelitian

antara lain : data mining dan analisis statistic, namun untuk penelitian ini berfokus pada bidang kesehatan.

Dataset tersebut terdiri dari 768 baris dengan 8 fitur medis (Kehamilan, Glukosa, Tekanan Darah, Ketebalan Kulit, Insulin, BMI, DiabetesPedigreeFunction, Umur) dan 1 fitur target (Hasil) untuk mengidentifikasi diagnosa bahwa pasien negative atau positif diabetes.

kehamilan	glukosa	tekanan darah	ketebalan kulit	insulin	BMI	diabetes pedigree function	umur.	Hasil
6	148	72	35	0	33.6	0.627	50	1
1	85	66	29	0	26.6	0.351	31	0
8	183	64	0	0	23.3	0.672	32	1
1	89	66	23	94	28.1	0.167	21	0
0	137	40	35	168	43.1	2.288	33	1
5	116	74	0	0	25.6	0.201	30	0
3	78	50	32	88	31	0.248	26	1
10	115	0	0	0	35.3	0.134	29	0
2	197	70	45	543	30.5	0.158	53	1
8	125	96	0	0	0	0.232	54	1

Gbr.2 Data Frame Dataset

Dataset akan diperiksa lebih lanjut untuk informasi dasar termasuk jumlah baris dan kolom, tipe data, serta mengenali data untuk mengetahui apakah data memiliki nilai kosong/missing value dan data outlier yang dapat mempengaruhi hasil analisis data.

B. Pre-processing Data

Berdasarkan data yang digunakan penulis menggunakan data sekunder yaitu untuk notabene data bukan dari sumber peneliti secara langsung maka diperlukan adanya pengecekan mengenai data ekstream, data outlier dan missing value yang dapat mempengaruhi hasil analisis data yang tidak sesuai (Septiana et al., 2024).

1. Missing Data Value

Mengisi data yang hilang dengan mengetahui nilai mean dan median. Menentukan missing data value dengan mempertimbangkan beberapa hal, nilai mean dari pengisian kolom cenderung memiliki distribusi data normal dan data yang hilang tidak terlalu banyak. Sedangkan nilai median dari kolom memiliki distribusi data yang cenderung abnormal dan mempunyai banyak data yang hilang. Tujuan dari missing value adalah untuk mempertahankan ukuran sampel, dan meningkatkan keakuratan data. Gambar dibawah ini menunjukkan penerapan missing value.

```

df[['kehamilan', 'glukosa', 'tekanan darah', 'ketebalan kulit', 'insulin', 'bmi', 'diabetespedigreefunction', 'umur']] = df[['kehamilan', 'glukosa', 'tekanan darah', 'ketebalan kulit', 'insulin', 'bmi', 'diabetespedigreefunction', 'umur']].fillna(df.mean())

df.head()

```

	kehamilan	glukosa	tekanan darah	ketebalan kulit	insulin	bmi	diabetespedigreefunction	umur	hasil
0	6.000000	148.0	72.0	35.000000	155.548223	33.6	0.627	50	1
1	1.000000	85.0	66.0	29.000000	155.548223	26.6	0.351	31	0
2	8.000000	183.0	64.0	29.15342	155.548223	23.3	0.672	32	1
3	1.000000	89.0	66.0	23.000000	94.000000	28.1	0.167	21	0
4	4.494673	137.0	40.0	35.000000	168.000000	43.1	2.288	33	1

Gbr.3 Pengisian Missing Data Value

2. Pembersihan Data Outlier

Data outlier merupakan nilai ekstrem atau nilai yang tidak mungkin dalam suatu data. Pembersihan outlier ini bertujuan untuk mengurangi pengaruh outlier pada model yang akan dibangun, meningkatkan akurasi data, dan stabilitas pada pelatihan (Mucholladin et al., 2021). Gambar dibawah ini menunjukkan implementasi data outlier pada setiap parameter.

```
from scipy.stats import zscore

# Menghitung z-score dan membuang outlier
z_scores = np.abs(zscore(df.select_dtypes(include='number')))
df = df[(z_scores < 3).all(axis=1)]
```

Gbr.4 Data Outlier

C. Statistik Deskriptif

Untuk memahami dataset pada tahap pemrosesan dan analisis lebih lanjut dengan menyediakan informasi awal mengenai distribusi data, penyebaran kolom, rentang nilai, dan penyebaran di setiap kolom, dan mengidentifikasi bagaimana potensi masalah data seperti nilai dari mean atau outlier yang mampu memengaruhi hasil analisis lebih lanjut. Apabila data perlu dinormalisasi untuk ketidakseimbangan data sebelum lanjut ke tahap pemodelan. Berikut data frame yang harus ditampilkan adalah:

	Kehamilan	Glukosa	Tekanan Darah	Ketebalan Kulit	Insulin
Count	719.000000	719.000000	719.000000	719.000000	719.000000
Mean	4.466575	120.047891	72.290370	28.768993	146.020735
Std	2.892202	29.642216	38.000000	8.154910	56.968123
Min	1.000000	44.000000	64.000000	7.000000	15.000000
25 %	2.000000	99.000000	62.000000	25.000000	120.000000
50 %	4.494673	115.000000	72.000000	29.1543420	155.548223
75 %	6.000000	138.000000	80.000000	32.000000	155.548223
Max	13.000000	199.000000	108.000000	54.000000	402.000000

	BMI	D.Pedigree Function	Umur	Hasil
Count	719.000000	719.000000	719.000000	719.000000
Mean	32.139245	0.44922	32.922114	0.336579
Std	18.200000	0.28279	11.309623	0.472868
Min	27.400000	0.07800	21.000000	0.000000
25 %	27.300000	0.24200	24.000000	0.000000
50 %	32.100000	0.36200	29.000000	0.000000
75 %	36.100000	0.60000	40.000000	1.000000
Max	52.900000	1.46100	68.000000	1.000000

Gbr.5 Statistik Deskriptif

- count: Jumlah nilai dalam kolom (non-null).
- mean: Rata-rata dari nilai-nilai dalam kolom.
- std: Standar deviasi, mengukur seberapa jauh nilai-nilai menyebar dari mean.
- min: Nilai minimum dalam kolom.
- 25% (kuartil pertama): Batas 25% dari data.

- 50% (median): Batas 50% atau nilai tengah dari data.
- 75% (kuartil ketiga): Batas 75% dari data.
- max: Nilai maksimum dalam kolom.

D. Standarisasi Data

Algoritma SVM memiliki sensitifitas pada skala fitur maka perlu dilakukan standarisasi data untuk memastikan bahwa fitur yang ada memiliki skala yang sama. Standarisasi data adalah proses mengubah skala data sehingga memiliki distribusi nilai dengan rata-rata 0 dan standar deviasi 1.

Hasil standarisasi yang disimpan dalam variabel standardized data yang dihasilkan dengan mengambil data pada variabel x yang telah difitkan oleh scaler. Hasilnya adalah standardized_data yang memiliki rata-rata nol dan deviasi standar satu pada setiap atributnya.

```
x = standardized_data
y = diabetes_dataset ['Hasil']
print(x)
print(y)
```

```
[30]
... [[ 0.50625491  0.86510807 -0.03351824 ... 0.16629174  0.46849198
      1.4259954 ]
      [-1.17528945 -1.20616153 -0.52985903 ... -0.85253118 -0.36506078
      -0.19067191]
      [ 1.17887265  2.0158134  -0.69530596 ... -1.33283341  0.60439732
      -0.10558415]
      ...
      [ 0.16994604 -0.0225789 -0.03351824 ... -0.91074963 -0.68519336
      -0.27575966]
      [-1.17528945  0.14180757 -1.02619983 ... -0.34311972 -0.37110101
      1.17073215]
      [-1.17528945 -0.94314317 -0.19896517 ... -0.29945588 -0.47378505
      -0.87137393]]
0  1
1  0
2  1
3  0
4  1
..
763 0
764 0
```

Gbr.6 Standarisasi Data

E. Split Data

Proses membagi yaitu dengan membagi 80% data latih dan 20% data uji. pada model machine learning, untuk memastikan bahwa model mampu generalisasi dengan baik dari data yang belum pernah dilihat sebelumnya (Teknologi & Informatika, 2024).

Data latih digunakan untuk membuat model, sementara data uji digunakan untuk memeriksa apakah model berfungsi dengan benar.

```
#Memisahkan data training dan data testing
x_train, x_test, y_train, y_test = train_test_split(x,y, test_size=0.2, stratify=y, random_state=42)
print(x.shape, x_train.shape, x_test.shape)
```

```
(768, 8) (614, 8) (154, 8)
```

Gbr.7 Split Data

F. Analysis Model Machine Learning

Analisis model pada proses evaluasi untuk memahami bagaimana performa model kekurangan dan kelebihan yang telah dilatih dengan tujuan untuk menentukan apakah model mampu bekerja dengan baik untuk mengidentifikasi peluang perbaikan atau mungkin ada masalah tertentu.

Algoritma SVM dengan kernel linier memiliki potensi yang baik untuk digunakan dalam klasifikasi risiko diabetes, meskipun masih terdapat ruang untuk peningkatan kinerja. Model ini memberikan prediksi yang cukup akurat dan dapat digunakan sebagai dasar dalam pengembangan aplikasi prediksi risiko diabetes.

```
#Testing
new_data = {'Kehamilan' : [1.0],
            'Glukosa' : [85.0],
            'Tekanan Darah' : [66.0],
            'Ketebalan Kulit' : [29.0],
            'Insulin' : [155.5],
            'BMI' : [26.6],
            'DiabetesPedigreeFunction' : [0.351],
            'Umur' : [31]}

new_data = pd.DataFrame(new_data)
new_data

[37]
...
Kehamilan  Glukosa  Tekanan Darah  Ketebalan Kulit  Insulin  BMI  DiabetesPedigreeFunction  Umur
0           1.0    85.0           66.0           29.0    155.5    26.6           0.351    31

scaled_new_data = scaler.transform(new_data)
y_pred_new = clf.predict(scaled_new_data)

print("Diagnosis Data Baru : ", y_pred_new)

[38]
...
Diagnosis Data Baru : [0]
```

Gbr.8 Model Machine Learning

III. HASIL DAN PEMBAHASAN

A. Implementasi Support Vector Machine

Pemodelan menggunakan algoritma SVM dan mengevaluasi kinerja model untuk memprediksi risiko diabetes. Evaluasi model yang dilakukan adalah mengidentifikasi nilai confusion matrix, accuracy score dan classification report.

- *Accuracy*, menunjukkan bagaimana akurasi model klasifikasi yang digunakan.

$$accuracy = \left(\frac{TP+TN}{TP+FP+TN+FN} \right) \times 100\%$$

- *Precision*, hasil prediksi yang dibuat oleh model menggunakan akurasi data yang diminta.

$$precision = \left(\frac{TP}{TP+FP} \right) \times 100\%$$

- *Recall*, menemukan kembali informasi dari hasil keberhasilan model.

$$recall = \left(\frac{TP}{TP+FN} \right) \times 100\%$$

- *F-1 score*, hasil dari recall yang dibobotkan dan perbandingan rata-rata precision.

$$f1 - score = \left(\frac{2 \times precision \times recall}{precision+recall} \right) \times 100\%$$

Berikut adalah hasil pengukuran kinerja dari classification report yang diukur berdasarkan precision, recall, f1-score, accuracy, dan lainnya.

	precision	recall	f1-score	support
0	0.75	0.83	0.79	100
1	0.60	0.48	0.54	54
accuracy			0.71	154
macro avg	0.68	0.66	0.66	154
weighted avg	0.70	0.71	0.70	154

Akurasi SVM : 70.78%

Gbr.9 Akurasi Data

Dapat disimpulkan dari pengukuran kinerja

1. Akurasi Model:

Akurasi keseluruhan dari model SVM adalah **70.78%**, yang menunjukkan bahwa model dapat membuat prediksi yang benar pada data uji.

2. Kinerja Berdasarkan Kelas:

Kelas 0 (Non-diabetes):

- *Precision*: 0,75
- *Recall*: 0,83
- *F1-score*: 0.79
- *Support*: 100

Kelas 1 (Diabetes):

- *Precision*: 0.60
- *Recall*: 0.48
- *F1-score*: 0.54

3. Confusion Matrix

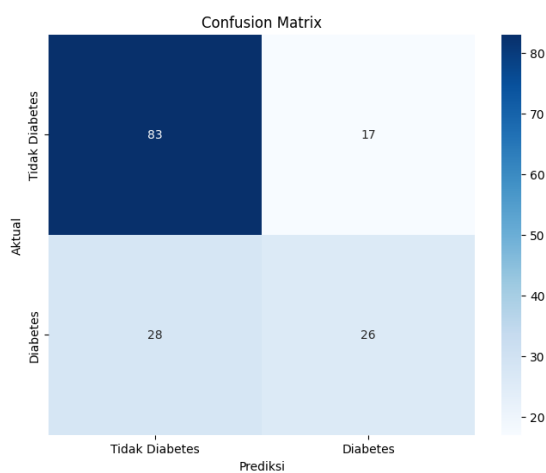
Proses mengevaluasi performa model dengan menyajikan jumlah prediksi benar dan salah untuk setiap kelas dalam format matriks. Confusion matrix adalah pengujian untuk mengetahui bagaimana kinerja algoritma pada machine learning (Pradana et al., 2022). Klasifikasi machine learning dengan 2 class output atau bisa lebih disebut sebagai confusion matrix. Terdapat 4 kombinasi tabel dengan nilai prediksi dan nilai aktual.

		True Values	
		True	False
Prediction	True	TP Correct result	FP Unexpected result
	False	FN Missing result	TN Correct absence of result

Gbr.10 Tabel Confusion Matrix

Berikut hasil visualisasi dari confusion matriks:

- **True Positive (TP):** Kasus "Diabetes" yang terdeteksi dengan benar = 83.
- **True Negative (TN):** Kasus "Tidak Diabetes" yang terdeteksi dengan benar = 26.
- **False Positive (FP):** Kasus "Tidak Diabetes" yang salah diprediksi sebagai "Diabetes" = 17.
- **False Negative (FN):** Kasus "Diabetes" yang salah diprediksi sebagai "Tidak Diabetes" = 28.



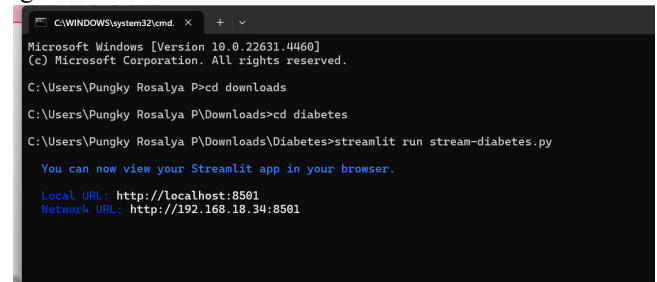
Gbr.11 Hasil Confusion Matrix

Hasil evaluasi berdasarkan nilai-nilai matriks dan interpretasi dalam konteks klasifikasi penyakit diabetes. Setelah melakukan proses pengujian maka didapatkan bahwa model SVM mampu mendapatkan hasil yang cenderung lebih optimal dengan menghasilkan representasi hasil output yang cenderung seimbang dan dapat menangani dimensi data yang kompleks.

Hyper parameter juga telah disesuaikan dengan model yang lebih optimal untuk menangani klasifikasi penyakit diabetes melitus. Berdasarkan metrik-metrik tersebut membantu bagaimana kinerja evaluasi model menggunakan algoritma SVM untuk klasifikasi apakah pasien di diagnosa sebagai penderita diabetes (1) dan bukan penderita diabetes (0). Dari hasil penelitian berikut menunjukkan bahwa algoritma SVM dapat diandalkan sebagai pengelolaan penyakit diabetes, pencegahan dan juga sebagai alat untuk pendeteksi dini untuk penyakit diabetes melitus.

B. Implementasi Pembuatan Aplikasi Streamlit

Membangun aplikasi sederhana sebagai langkah selanjutnya untuk proses pemodelan yang telah dibuat sebelumnya. Untuk pembuatan aplikasi streamlit juga membuat code menggunakan software visual studio code dengan format "pickle" untuk mengekspor pada jupyter notebook yang berperan sebagai program pengolahan data untuk mengembangkan program aplikasi berbasis web. Adapun gambar dibawah ini menunjukkan bagaimana memanggil aplikasi streamlit sehingga dapat terhubung dengan baik.



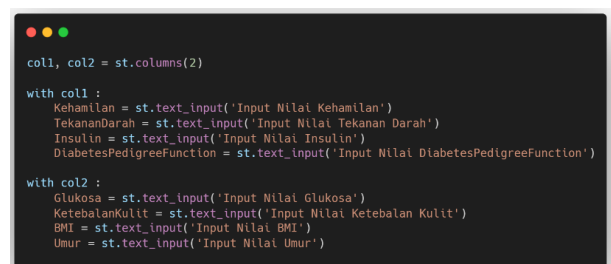
Gbr.12 Run Streamlit

C. Membuat Aplikasi dengan Python

Dengan membuat file stream-diabetes dengan ekstensi file(.py) . Adapun proses dari mengimplementasikan aplikasi streamlit sebagai berikut:

1. Input Data

Bagian ini meminta pengguna untuk memasukkan data melalui antarmuka Streamlit yang berarti mengumpulkan nilai input dari pengguna untuk diteruskan ke model. `st.text_input()` komponen Streamlit yang memungkinkan pengguna untuk memasukkan nilai. Variabel/atribut, input yang diperlukan oleh model untuk membuat prediksi. Nilai ini biasanya sesuai dengan fitur data yang digunakan saat model dilatih.



Gbr.13 Membagi 2 Kolom Untuk Menginput Data

2. Proses Prediksi

Bagian ini akan menjalankan model machine learning dengan input dari pengguna untuk memprediksi hasil (diabetes atau tidak). `diab_diagnosis=` Variabel yang akan menyimpan hasil prediksi dari model. Nantinya, kode akan dilengkapi dengan pemanggilan fungsi

untuk menghasilkan prediksi berdasarkan input pengguna.

```
if st.button('Test Prediksi Diabetes'):  
    diab_prediction = diabetes_model.predict([[Kehamilan,Glukosa,  
        TekananDarah,KetebalanKulit,Insulin,BMI,DiabetesPedigreeFunction,Umur]])  
  
    if(diab_prediction[0] == 1):  
        diab_diagnosis = 'Pasien Positif Diabetes'  
    else:  
        diab_diagnosis = 'Pasien Negatif Diabetes'  
    st.success(diab_diagnosis)
```

Gbr.14 Membuat Button Test Prediksi Diabetes

D. Tampilan Antarmuka Aplikasi Streamlit

Setelah hasil testing dilakukan akan dilanjutkan untuk membangun aplikasi Streamlit untuk web interaktif secara sederhana, dengan menggunakan Bahasa pemrograman python sehingga model machine learning dapat dijalankan dengan baik. Nantinya pengguna akan meng-input 8 parameter untuk diprediksi oleh model. Machine learning tersebut nantinya dapat digunakan secara terbuka untuk kepentingan medis, Berikut tampilan "Prediksi Pasien Diabetes" aplikasi sederhana Streamlit yang berhasil dijalankan dengan baik.

Aplikasi antarmuka ini dirancang dengan sederhana dan intuitif. Pengguna dapat memasukkan data pasien dan akan menerima hasil prediksi status penyakit diabetes (Arminarahmah et al., 2024). Sistem akan memproses data masukan menggunakan model klasifikasi Support Vector Machine (SVM) yang telah dilatih sebelumnya. Model ini menggunakan dataset Pima Indians Diabetes sebagai dasar pelatihan. Aplikasi ini diharapkan dapat membantu tenaga medis atau pengguna lain dalam memberikan prediksi awal terkait risiko diabetes, yang kemudian dapat digunakan sebagai acuan untuk tindakan medis lebih lanjut.

Prediksi Pasien Diabetes

Input Nilai Kehamilan	Input Nilai Glukosa
<input type="text"/>	<input type="text"/>
Input Nilai Tekanan Darah	Input Nilai Ketebalan Kulit
<input type="text"/>	<input type="text"/>
Input Nilai Insulin	Input Nilai BMI
<input type="text"/>	<input type="text"/>
Input Nilai DiabetesPedigreeFunction	Input Nilai Umur
<input type="text"/>	<input type="text"/>
<input type="button" value="Test Prediksi Diabetes"/>	

Gbr.15 Tampilan Antarmuka

IV. KESIMPULAN

Berdasarkan hasil penelitian dapat diambil beberapa kesimpulan, sebagai berikut:

1. Algoritma SVM menggunakan bahasa pemrograman Python dengan menggunakan modul sklear yang memfasilitasi kelas pendukung dari berbagai jenis metode SVM. Kemudian dataset akan dianalisis sebagai model deteksi untuk lanjut pada proses split data yaitu dengan membagi 80% data latih dan 20% data uji. Melatih model SVM menggunakan kelas SVC dari sklearn selanjutnya menentukan parameter yang sesuai. Proses evaluasi model dengan mengukur akurasi pada data uji menggunakan score dan visualisasi hasil.
2. Implementasi aplikasi Streamlit dengan framework Python untuk memudahkan pengguna dalam menginput data pasien dan akan menghasilkan output prediksi apakah pasien didiagnosa penyakit diabetes (1) atau tidak diabetes (0) . Aplikasi web interaktif sebagai pembelajaran mesin dan analisis data.

V. SARAN

Berdasarkan hasil analisis, pembahasan, implementasi dan pengujian, maka saran yang dibuat penulis untuk pengembangan penelitian selanjutnya adalah:

1. Dapat menggunakan dataset pasien dengan jumlah lebih banyak, guna meningkatkan tingkat akurasi dari model deteksi penyakit diabetes yang akan dibuat.
2. Dapat menggunakan lebih dari satu kernel SVM untuk membandingkan hasil akurasi dan hasil pengujian terbaik.
3. Implementasi aplikasi dapat menggunakan bahasa pemrograman lain seperti HTML, CSS, JavaScript atau bahasa pemrograman lain untuk meluncurkan aplikasi berbasis mobile.

UCAPAN TERIMA KASIH

Puji syukur ke hadirat Tuhan Yang Maha Esa atas limpahan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan artikel saya yang berjudul "Klasifikasi Penyakit Diabetes Melitus Menggunakan Metode Support Vector Machine (SVM)" dapat di selesaikan dengan baik dan tepat waktu. Penulis menyampaikan ucapan terima kasih yang sebesar-besarnya kepada pihak-pihak yang telah memberikan dukungan, bimbingan, dan motivasi selama proses penyelesaian artikel ini. Orang tua tercinta, atas doa, dukungan moral, dan materi yang tidak pernah putus selama masa studi hingga selesainya skripsi ini. Kepada Pak Ronggo Alit, M.M., M.T. sebagai Dosen Pembimbing, yang telah memberikan arahan, bimbingan, dan masukan yang sangat berharga dalam proses penelitian dan penulisan skripsi ini. Teman dan sahabat terimakasih atas kebersamaan, diskusi,

dan dukungan selama proses perkuliahan hingga penelitian ini dapat saya selesaikan dengan lancar sampai akhir.

REFERENSI

- [1] Agatsa, D. A., Rismala, R., & Wisesty, U. N. (2020). Klasifikasi Pasien Pengidap Diabetes menggunakan Metode Support Vector Machine. *E-Proceeding of Engineering*, Vol.7(No.1), 2517.
- [2] Aldama, C., & Nasir, M. (2023). Klasifikasi penyakit Diabetes menggunakan metode support vector machine pada Rumah Sakit Umum Prabumulih. *Jurnal Ilmiah Betrik*, 14(02), 376–383. <https://ejournal.pppmitpa.or.id/index.php/betrik/article/view/117>
- [3] Algoritma, I., Vector, S., Untuk, M., Status, K., & Balita, S. P. (2024). *G-Tech : Jurnal Teknologi Terapan*. 8(3), 2070–2079.
- [4] Arminarahmah, N., Mahalisa, G., & Informatika, T. (2024). *Implementasi Model Machine Learning pada Klasifikasi Status Penyakit Diabetes Berbasis Streamlit Implementation of Machine Learning Models in Diabetes Disease Status Classification Based on Streamlit*. 13(105), 470–475.
- [5] Cahyani, A. D., & Basuki, A. (2019). Klasifikasi Diabetes Mellitus Menggunakan Support Vector Machine (Studi Kasus: Puskesmas Modopuro, Mojokerto). *Rekayasa*, 12(2), 174–182. <https://doi.org/10.21107/rekayasa.v12i2.19763>
- [6] Hovi, H. S. W., Id Hadiana, A., & Rakhmat Umbara, F. (2022). Prediksi Penyakit Diabetes Menggunakan Algoritma Support Vector Machine (SVM). *Informatics and Digital Expert (INDEX)*, 4(1), 40–45. <https://doi.org/10.36423/index.v4i1.895>
- [7] Mucholladin, A. W., Abdurrachman Bachtiar, F., & Furqon, M. T. (2021). Klasifikasi Penyakit Diabetes menggunakan Metode Support Vector Machine. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 5(2), 622–633. <http://j-ptiik.ub.ac.id>
- [8] Muhammad Hilmy Haidar Aly. (2024). Klasifikasi Diabetes Menggunakan Algoritma Support Vector Machine Radial Basis Function. *Jurnal Teknik Informatika Dan Teknologi Informasi*, 4(1), 28–38. <https://doi.org/10.55606/jutiti.v4i1.3420>
- [9] Pradana, M. G., Saputro, P. H., & Wijaya, D. P. (2022). Komparasi Metode Support Vector Machine Dan Naïve Bayes Dalam Klasifikasi Peluang Penyakit Serangan Jantung. *Indonesian Journal of Business Intelligence (IJUBI)*, 5(2), 87. <https://doi.org/10.21927/ijubi.v5i2.2659>
- [10] Septiana, T., Muda, M. A., Budiyanto, D., Pratama, M., & Jaya, W. P. (2024). Analisis Penggunaan Support Vector Machine pada Deteksi Dini Penyakit Diabetes Melitus. *Jurnal Penelitian Inovatif*, 4(3), 1631–1640. <https://doi.org/10.54082/jupin.643>
- [11] Teknologi, M., & Informatika, D. (2024). *SUPPORT VECTOR MACHINE UNTUK KLASIFIKASI PENYAKIT DIABETES MELLITUS Media Teknologi dan Informatika*. 1, 147–152.