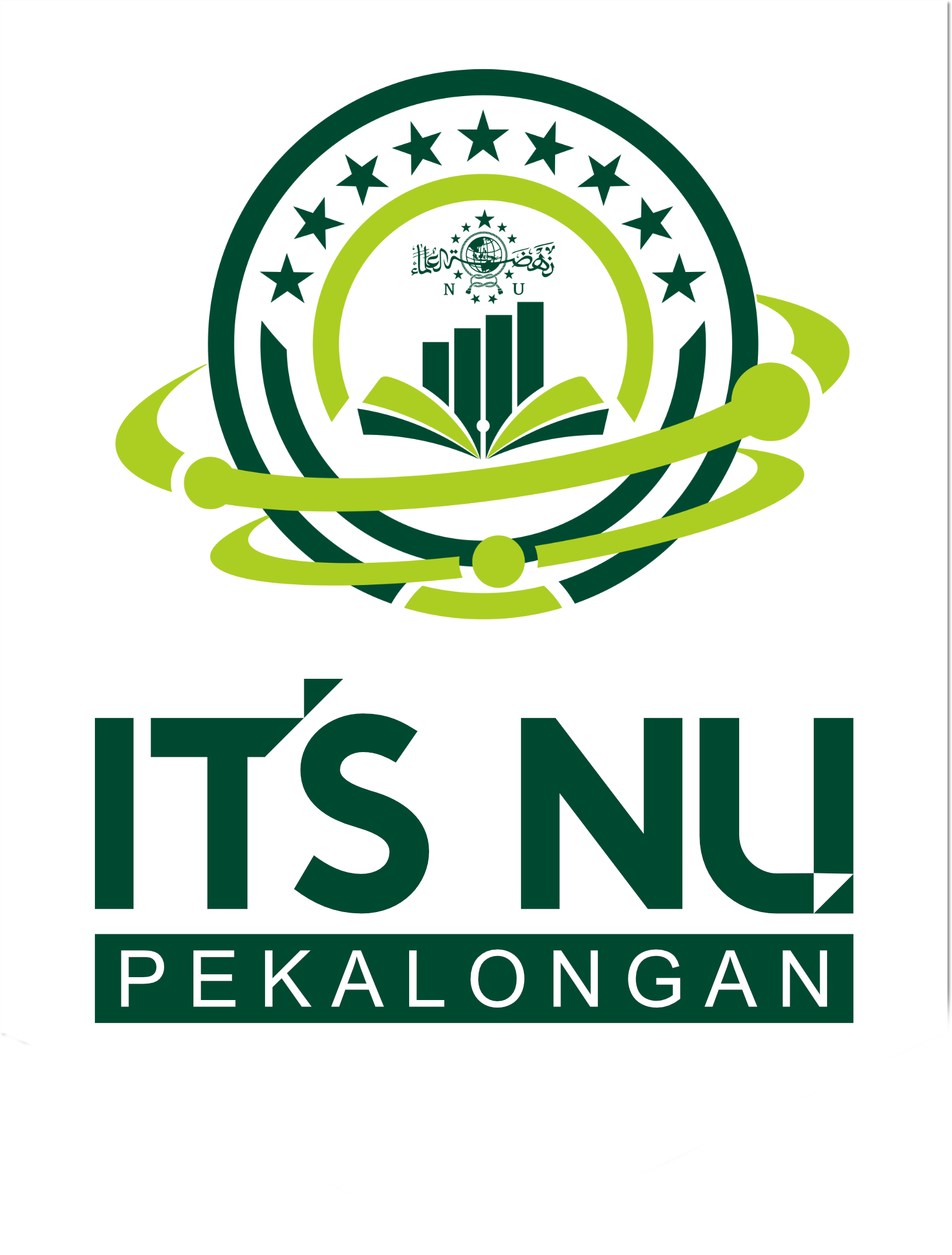
**Knowledge Management**

**Implementasi Support Vector Machine (SVM) untuk Prediksi Risiko Diabetes Berdasarkan Data Kesehatan**

(Disusun untuk memenuhi tugas mata kuliah

Knowledge Management)



|  |  |  |
| --- | --- | --- |
|  | Oleh Kelompok 1 TIF21D: |  |
| 1. | Randi Afif | 101210072 |
| 2. | M. Mustaqim | 101210075 |
| 3. | Alya Safitri | 101210079 |
| 4. | Rizqi Mau'ida | 101210086 |
| 5. | Fachrudin FA | 101210088 |

**PROGRAM STUDI S1 TEKNOLOGI INFORMASI**

**FAKULTAS SAINS DAN TEKNOLOGI**

**ITS NU PEKALONGAN**

**PEKALONGAN**

**2024**

1. **Pendahuluan**

Diabetes mellitus merupakan salah satu penyakit kronis yang terus mengalami peningkatan prevalensi secara global. Menurut laporan Organisasi Kesehatan Dunia (WHO), jumlah penderita diabetes diproyeksikan mencapai lebih dari 640 juta orang pada tahun 2045. Penyakit ini tidak hanya menjadi ancaman serius bagi kesehatan individu, tetapi juga menimbulkan dampak ekonomi yang signifikan bagi sistem layanan kesehatan. Diabetes sering kali tidak terdeteksi pada tahap awal, sehingga memperburuk komplikasi yang dapat memengaruhi kualitas hidup pasien. Oleh karena itu, kebutuhan akan sistem prediksi yang andal menjadi semakin mendesak dalam upaya pencegahan dan pengelolaan penyakit ini.

Kemajuan teknologi dalam bidang kecerdasan buatan dan pembelajaran mesin (machine learning) memberikan peluang besar dalam menganalisis data medis untuk keperluan prediksi penyakit. Salah satu algoritma yang terbukti efektif dalam klasifikasi data kompleks adalah Support Vector Machine (SVM). SVM memiliki kemampuan untuk memisahkan data ke dalam kelas-kelas yang berbeda dengan margin maksimal, sehingga sering digunakan dalam berbagai studi prediktif.

Penelitian ini bertujuan untuk menerapkan algoritma SVM dalam prediksi diabetes berdasarkan dataset medis yang relevan. Fokus penelitian meliputi proses preprocessing data, penerapan algoritma SVM, serta evaluasi performa model menggunakan metrik seperti akurasi, presisi, recall, dan F1-score. Dengan penelitian ini, diharapkan dapat dikembangkan model prediksi yang akurat, yang tidak hanya membantu dalam diagnosis dini diabetes tetapi juga mendukung pengambilan keputusan bagi praktisi medis.

1. **Metodologi**

Dalam penelitian ini, dilakukan pendekatan berbasis machine learning untuk memprediksi risiko diabetes dengan algoritma Support Vector Machine (SVM). Penelitian ini bertujuan untuk menganalisis data medis, membersihkannya dari anomali, dan membangun model prediksi yang andal. Langkah-langkah berikut dirancang untuk memastikan penelitian dilakukan secara sistematis dan terstruktur:

1. Import Library

Penelitian diawali dengan mengimpor pustaka Python yang diperlukan untuk analisis data, visualisasi, dan penerapan algoritma pembelajaran mesin.

2. Import Data

Dataset yang relevan, seperti dataset medis tentang diabetes, diimpor untuk dianalisis. Data ini berisi informasi seperti kadar glukosa, tekanan darah, BMI, usia, dan status diabetes.

3. Data Understanding

Tahap ini bertujuan untuk memahami karakteristik dataset, termasuk struktur data, tipe data pada setiap kolom, serta distribusi statistik untuk setiap fitur. Selain itu, dilakukan analisis terhadap jumlah data yang termasuk dalam setiap kelas target untuk memahami proporsi data positif dan negatif.

4. Cleaning Data

Pada tahap ini, dilakukan pembersihan data untuk memastikan tidak ada nilai kosong, duplikasi, atau nilai ekstrem yang dapat memengaruhi hasil analisis. Data yang tidak relevan atau bermasalah dihapus atau disesuaikan.

5. Exploratory Data Analysis (EDA)

Visualisasi data dilakukan untuk menggali pola-pola penting dan memahami hubungan antar fitur. Tahap ini membantu dalam menganalisis distribusi data serta interaksi antara fitur dan variabel target.

6. Preparation Data

Data dipersiapkan untuk proses pemodelan dengan cara memisahkan fitur dan target, membagi data menjadi data latih dan data uji, serta melakukan normalisasi agar data berada dalam skala yang sama.

7. Modeling & Evaluation

Model prediksi diabetes dibangun menggunakan algoritma Support Vector Machine (SVM). Model dilatih menggunakan data latih, dan kinerjanya dievaluasi dengan data uji berdasarkan akurasi, presisi, recall, F1-score, serta confusion matrix.

8. Testing

Model yang telah dibangun diuji menggunakan data baru untuk memastikan keandalan prediksi terhadap kasus nyata.

1. **Implementasi**
2. **Import Library**

**A screenshot of a computer

Description automatically generated**

Kode tersebut menginstal dan mengimpor pustaka yang diperlukan untuk analisis data dan pemodelan machine learning, seperti pandas, numpy, seaborn, matplotlib, dan scikit-learn. Pustaka ini digunakan untuk memproses data, visualisasi, dan membangun model klasifikasi menggunakan Support Vector Classifier (SVC), serta untuk mengevaluasi performa model.

1. **Import Data**

**A screenshot of a computer

Description automatically generated**

Perintah df = pd.read\_csv('Diabetes.csv', index\_col=0) digunakan untuk membaca file CSV yang bernama Diabetes.csv ke dalam sebuah DataFrame Pandas (df). Parameter index\_col=0 menandakan bahwa kolom pertama dalam file CSV akan digunakan sebagai indeks DataFrame. Setelah itu, perintah df digunakan untuk menampilkan isi DataFrame yang telah dibaca dari file CSV.

1. **Data Understanding**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

Penjelasan perintah-perintah berikut:

1. df.shape  
   Menampilkan dimensi DataFrame df, yaitu jumlah baris dan kolom dalam bentuk tuple (jumlah\_baris, jumlah\_kolom).
2. df.info()  
   Menampilkan informasi umum mengenai DataFrame, termasuk jumlah entri, jumlah nilai non-null, tipe data tiap kolom, dan penggunaan memori.
3. df[['Kehamilan', 'Tekanan Darah', 'Umur', 'BMI']].describe()  
   Memberikan ringkasan statistik (seperti mean, standar deviasi, nilai minimum, dan kuartil) untuk kolom-kolom tertentu: Kehamilan, Tekanan Darah, Umur, dan BMI.
4. df.Hasil.value\_counts()  
   Menghitung frekuensi setiap nilai dalam kolom Hasil. Di sini, nilai 0 menunjukkan tidak mengidap diabetes, sementara nilai 1 menunjukkan positif mengidap diabetes.
5. **Cleaning Data**

A screenshot of a computer

Description automatically generated

Penjelasan perintah-perintah berikut:

1. df.isnull().sum()  
   Menampilkan jumlah nilai yang hilang (null) di setiap kolom pada DataFrame df. Fungsi ini mengembalikan jumlah nilai null per kolom.
2. df.duplicated().sum()  
   Menghitung jumlah baris yang duplikat dalam DataFrame df. Fungsi ini mengembalikan jumlah baris yang identik dengan baris sebelumnya (termasuk duplikat dalam data).
3. **Eksplorasi Data Analysis (EDA)**

**A screen shot of a graph

Description automatically generated**

**A screenshot of a computer screen

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer screen

Description automatically generated**

Berikut adalah penjelasan mengenai visualisasi yang dihasilkan:

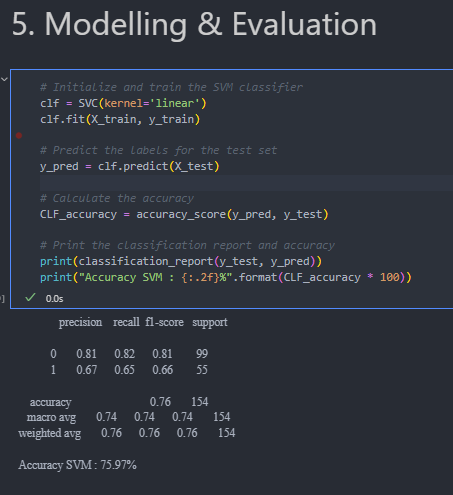
1. Histogram Distribusi Umur  
   Histogram ini menunjukkan distribusi data berdasarkan kolom Umur. Dengan 10 bin dan tambahan kurva KDE, visualisasi ini menggambarkan sebaran umur dalam dataset. Ini membantu untuk memahami pola umum dalam distribusi umur peserta.
2. Pie Chart Presentase Diabetes vs Non-Diabetes  
   Diagram pie ini menunjukkan proporsi individu yang positif (mengidap diabetes) dan negatif (tidak mengidap diabetes) berdasarkan kolom Hasil. Persentase masing-masing kategori ditampilkan untuk memberikan gambaran visual yang jelas tentang sebaran status diabetes dalam dataset.
3. Count Plot Distribusi Hasil Diagnosis  
   Count plot ini menunjukkan frekuensi masing-masing kategori dalam kolom Hasil, yang mengindikasikan apakah seseorang mengidap diabetes atau tidak. Dengan visualisasi ini, kita dapat dengan mudah melihat jumlah kasus positif dan negatif dalam dataset.
4. Heatmap Korelasi  
   Heatmap ini menunjukkan matriks korelasi antara variabel-variabel dalam dataset. Nilai korelasi yang lebih tinggi ditunjukkan dengan warna yang lebih intens. Ini memberikan gambaran mengenai hubungan antar fitur dalam dataset, misalnya apakah ada korelasi antara umur dan BMI atau tekanan darah.
5. **Preparation Data**

**A screenshot of a computer program

Description automatically generated**

Penjelasan langkah-langkah berikut:

1. Memisahkan Fitur dan Label:
   * X = df.drop(columns=['Hasil']) digunakan untuk memisahkan fitur (variabel independen) yang ada di dataset dengan menghapus kolom Hasil, yang merupakan kolom label.
   * y = df['Hasil'] mengambil kolom Hasil sebagai label (variabel dependen), yang berisi informasi apakah seseorang positif atau negatif mengidap diabetes.
2. Menampilkan Ukuran Dataset:
   * X.shape dan y.shape digunakan untuk menampilkan ukuran (jumlah baris dan kolom) dari fitur (X) dan label (y).
3. Normalisasi Data:
   * StandardScaler() digunakan untuk menstandarkan fitur sehingga nilai fitur memiliki rata-rata 0 dan standar deviasi 1. Hal ini penting untuk beberapa model machine learning, seperti SVM.
   * scaler.fit(X) mempelajari parameter skala berdasarkan data fitur X.
   * X = scaler.transform(X) mengaplikasikan transformasi skala pada data fitur X.
4. Membagi Data ke dalam Training dan Testing:
   * train\_test\_split(X, y, test\_size=0.2, random\_state=42) membagi dataset menjadi dua bagian: 80% data untuk pelatihan (training) dan 20% untuk pengujian (testing).
   * Hasil pembagian ini disimpan dalam variabel X\_train, X\_test, y\_train, dan y\_test.
5. Menampilkan Ukuran Dataset setelah Pembagian:
   * Ukuran masing-masing subset (training dan testing) ditampilkan dengan X\_train.shape, y\_train.shape, X\_test.shape, dan y\_test.shape untuk memastikan pembagian data dilakukan dengan benar.
6. **Modelling & Evaluation**

****

Berikut penjelasan tentang langkah-langkah yang dilakukan dalam kode tersebut:

1. Inisialisasi dan Pelatihan Model SVM:
   * clf = SVC(kernel='linear') membuat sebuah instance dari model Support Vector Machine (SVM) dengan kernel linear. Kernel linear digunakan untuk kasus di mana data dapat dipisahkan secara linear.
   * clf.fit(X\_train, y\_train) melatih model SVM menggunakan data pelatihan (X\_train) dan label pelatihan (y\_train).
2. Prediksi pada Data Uji:
   * y\_pred = clf.predict(X\_test) digunakan untuk memprediksi label pada data uji (X\_test) setelah model dilatih.
3. Menghitung Akurasi:
   * CLF\_accuracy = accuracy\_score(y\_pred, y\_test) menghitung akurasi dari model dengan membandingkan label yang diprediksi (y\_pred) dengan label sebenarnya (y\_test). Akurasi dihitung sebagai persentase dari prediksi yang benar.
4. Mencetak Classification Report dan Akurasi:
   * print(classification\_report(y\_test, y\_pred)) menampilkan laporan klasifikasi yang mencakup metrik seperti precision, recall, f1-score, dan support untuk setiap kelas (positif dan negatif).
   * print("Accuracy SVM : {:.2f}%".format(CLF\_accuracy \* 100)) mencetak akurasi model dalam format persentase dengan dua digit desimal.

Dengan langkah-langkah ini, Anda mendapatkan evaluasi kinerja model SVM dalam mengklasifikasikan data uji.

1. **Testing**

**A screenshot of a computer program

Description automatically generated**

**A screenshot of a computer program

Description automatically generated**

Berikut penjelasan mengenai proses yang dilakukan dalam kode tersebut:

1. Membuat Data Baru:
   * Dua dataset baru (new\_data dan new\_diabetes\_data) dibuat untuk memprediksi apakah individu dengan kondisi tersebut mengidap diabetes atau tidak.
   * new\_data mewakili individu dengan kondisi normal, sedangkan new\_diabetes\_data mewakili individu yang memiliki kondisi berisiko tinggi mengidap diabetes.
2. Mengonversi Data ke DataFrame:
   * Kedua set data (new\_data dan new\_diabetes\_data) diubah menjadi DataFrame menggunakan pd.DataFrame().
3. Normalisasi Data Baru:
   * scaled\_new\_data = scaler.transform(new\_data) dan scaled\_new\_diabetes\_data = scaler.transform(new\_diabetes\_data) digunakan untuk menstandarkan data baru menggunakan scaler yang sudah dilatih sebelumnya. Hal ini penting untuk memastikan data baru sesuai dengan skala yang digunakan saat pelatihan model.
4. Prediksi Diagnosis:
   * y\_pred\_new = clf.predict(scaled\_new\_data) dan y\_pred\_new\_diabetes = clf.predict(scaled\_new\_diabetes\_data) digunakan untuk memprediksi apakah individu pada data baru mengidap diabetes (1) atau tidak (0) berdasarkan model SVM yang telah dilatih.
5. Menampilkan Hasil Prediksi:
   * Hasil prediksi dicetak untuk kedua data: new\_data (normal) dan new\_diabetes\_data (berisiko tinggi).

Dengan kode ini, Anda dapat melihat bagaimana model SVM yang sudah dilatih memberikan diagnosis untuk data individu berdasarkan fitur-fitur yang disediakan.

1. **Kesimpulan**

Kesimpulan dari implementasi yang dilakukan adalah sebagai berikut:

1. Preprocessing Data:  
   Data yang digunakan dalam model ini telah dipersiapkan dengan baik, termasuk memisahkan fitur dan label, menstandarkan nilai fitur menggunakan StandardScaler, serta membagi dataset menjadi data pelatihan dan pengujian.
2. Model SVM:  
   Model Support Vector Machine (SVM) dengan kernel linear digunakan untuk melakukan klasifikasi apakah seseorang mengidap diabetes atau tidak. Model ini dilatih dengan data pelatihan dan kemudian diuji menggunakan data uji untuk mengevaluasi performanya.
3. Evaluasi Model:  
   Berdasarkan hasil prediksi pada data uji, model menunjukkan akurasi yang baik dalam mengklasifikasikan status diabetes. Laporan klasifikasi juga memberikan metrik tambahan seperti precision, recall, dan f1-score, yang memberikan gambaran tentang kualitas model dalam memprediksi setiap kelas.
4. Prediksi untuk Data Baru:  
   Dua set data baru diuji untuk memprediksi apakah individu tersebut mengidap diabetes. Untuk data dengan kondisi normal (seperti kadar glukosa dan BMI yang normal), model memberikan prediksi negatif (tidak mengidap diabetes). Sebaliknya, untuk data dengan kondisi berisiko tinggi (seperti kadar glukosa dan BMI yang tinggi), model memberikan prediksi positif (mengidap diabetes).
5. Kesimpulan Umum:  
   Model SVM yang telah dilatih dengan data ini efektif dalam mengidentifikasi risiko diabetes berdasarkan berbagai faktor seperti kehamilan, glukosa, tekanan darah, BMI, dan lainnya. Dengan hasil yang memadai pada data uji dan kemampuan untuk memprediksi data baru dengan akurat, model ini bisa digunakan untuk membantu diagnosis awal diabetes berdasarkan faktor-faktor tersebut.

Namun, akurasi model masih bergantung pada kualitas dan keberagaman data yang digunakan untuk pelatihan. Diperlukan lebih banyak data atau teknik yang lebih canggih untuk meningkatkan prediksi pada kasus-kasus yang lebih kompleks atau tidak terlihat dalam dataset pelatihan.