

Aplikasi DIATECT Untuk Prediksi Penyakit Diabetes Menggunakan SVM Berbasis Web

Ganis Sanhaji¹, Anisa Febrianti², Hidayat³

^{1,2,3}Fakultas Teknik, Program Studi Teknik Elektro, Universitas Islam Nusantara, Kota Bandung, Indonesia
Email: ganissanhaji90@gmail.com, nisaf5544@gmail.com, jajangh475@gmail.com

Abstrack– Penelitian ini menampilkan dampak serius dari Penyakit diabetes. Aplikasi ini bertujuan untuk mendeteksi gejala diabetes. Pada Pendeskripsian masalah yang diteliti berfokus pada peningkatan jumlah penderita diabetes. Menurut data International Diabetes Federation (IDF) diprediksi bahwa jumlah orang dewasa yang menderita diabetes dalam rentang usia 20 hingga 79 tahun mencapai 415 juta individu di seluruh dunia. Penelitian ini memperkenalkan aplikasi berbasis web yang disebut DIATECT (Diabetes Prediction using Support Vector Machine on the Web) yang dirancang untuk memprediksi risiko penyakit diabetes. Aplikasi ini memanfaatkan algoritme Support Vector Machine (SVM) untuk mengolah data kesehatan yang luas dan beragam. DIATECT dapat memberikan prediksi yang akurat tentang kemungkinan seseorang mengetahui diabetes berdasarkan data yang dimasukkan. Penelitian ini menggunakan metode kuantitatif dengan data sekunder sebanyak 768 data, terdiri dari terkena diabetes dan tidak. Hasil uji coba menunjukkan bahwa DIATECT menggunakan algoritma SVM memiliki performa yang baik dalam mendeteksi penyakit diabetes dengan tingkat akurasi sebesar 77% sehingga model yang mampu memprediksi dengan benar pada 77% kasus. Sisanya, sekitar 23%, mungkin merupakan prediksi yang tidak tepat. Hasil penelitian menunjukkan bahwa aplikasi ini memiliki performa yang kuat, dengan presisi mencapai 77%, recall mencapai 91%, dan F1-score mencapai 91. Aplikasi ini memiliki potensi besar untuk mendukung masyarakat dalam mendeteksi dan mengidentifikasi penyakit melalui platform web yang mudah diakses.

Kata Kunci: SVM, Diabetes, Deteksi, Aplikasi, DIATECT

***Abstrack** – This research shows the serious impact of diabetes. This application aims to detect diabetes symptoms. In the description of the problem studied, it focuses on the increasing number of diabetes sufferers. According to data from the International Diabetes Federation (IDF), it is predicted that the number of adults suffering from diabetes between the ages of 20 and 79 years will reach 415 million individuals worldwide. This research introduces a web-based application called DIATECT (Diabetes Prediction using Support Vector Machine on the Web) which is designed to predict the risk of diabetes. This application utilizes the Support Vector Machine (SVM) algorithm to process extensive and varied health data. DIATECT can provide accurate predictions about a person's likelihood of developing diabetes based on the data entered. This research used quantitative methods with secondary data totaling 768 data, consisting of whether they had diabetes or not. The trial results show that DIATECT using the SVM algorithm has good performance in detecting diabetes with an accuracy rate of 77% so that the model is able to predict correctly in 77% of cases. The remainder, around 23%, may be inaccurate predictions. Research results show that this application has strong performance, with precision reaching 77%, recall reaching 91%, and F1-score reaching 91. This application has great potential to support community in detecting and identifying diseases through an easily accessible web platform.*

Keywords: SVM, Diabetes, Deteksi, Aplikasi, DIATECT

1. PENDAHULUAN

Diabetes telah menjadi tantangan bagi kesehatan global dengan tingkat kejadian yang terus meroket. Menurut laporan dari Organisasi Kesehatan Dunia (WHO), jumlah individu yang menderita diabetes telah mengalami lonjakan yang signifikan dalam beberapa dekade belakangan ini, dan proyeksi ke depan menunjukkan peningkatan yang terus menerus.[1]

Indonesia adalah negara peringkat keenam di dunia setelah Tiongkok, India, Amerika Serikat, Brazil dan Meksiko dengan jumlah penyandang Diabetes usia 20-79 tahun sekitar 10,3 juta orang. Riset Kesehatan Dasar (Riskesdas) memperlihatkan peningkatan angka prevalensi Diabetes yang cukup signifikan, yaitu dari 6,9% di tahun 2013 menjadi 8,5% di tahun 2018; sehingga estimasi jumlah penderita di Indonesia mencapai lebih dari 16 juta orang yang kemudian berisiko terkena penyakit lain, seperti: serangan jantung, stroke, kebutaan dan gagal ginjal bahkan dapat menyebabkan kelumpuhan dan kematian.[3]

International Diabetes Federation (IDF) menyatakan bahwa status Indonesia terhadap diabetes adalah serius karena Indonesia berada di urutan ke-7 dari 10 negara dengan jumlah pasien diabetes tertinggi. Jumlah pasien diabetes di Indonesia mencapai 6,2% pada tahun 2020, atau 10,8 juta orang, dan terus meningkat setiap tahunnya. Diabetes juga dapat disebabkan oleh makanan yang tidak sehat. Penyakit diabetes masih dapat dicegah sebanyak 80%, sehingga jumlah penderitanya tidak meningkat setiap tahun dan kualitas hidup akan meningkat.[2]

Diabetes adalah penyakit kronis yang ditandai dengan glukosa yang tinggi atau di atas normal. [4]glukosa adalah sumber energi utama bagi seluruh sistem tubuh manusia, namun untuk bisa diakses dan dimanfaatkan oleh sel-sel tubuh, glukosa memerlukan kehadiran hormon kunci yang disebut insulin. Pada penderita diabetes, entah

karena kurangnya produksi insulin atau ketidakmampuan tubuh untuk memanfaatkan insulin dengan efektif, terjadi disfungsi dalam regulasi glukosa yang mengakibatkan peningkatan kadar glukosa dalam darah.

Ada banyak jenis diabetes, dengan diabetes tipe 1 dan tipe 2 yang paling umum orang dengan diabetes tipe 1 mengalami peningkatan rasa haus, yang dikenal sebagai polidipsia, lebih sering buang air kecil, dan lebih banyak rasa lapar, yang dikenal sebagai polifagia. Gejala mungkin berubah dari minggu ke bulan. Ketoasidosis diabetik adalah kondisi yang dapat menyebabkan kehilangan kesadaran dan sangat sakit jika tidak diobati. Mereka yang menderita diabetes tipe 2 dan yang sedang hamil mungkin tidak memiliki gejala apa pun. Satu pengukuran glukosa darah tinggi dapat menunjukkan diagnosis atau mungkin memerlukan beberapa pengukuran glukosa darah tambahan.[5]

Diabetes merupakan salah satu penyakit tidak menular (PTM) yang menyebabkan peningkatan kadar glukosa dalam darah karena tubuh tidak dapat memproduksi insulin. Penelitian telah menunjukkan bahwa dalam jangka panjang, peningkatan kadar glukosa dalam darah menyebabkan kerusakan organ, yang dapat menyebabkan kegagalan fungsi jaringan dan organ. Diabetes dapat dibagi menjadi diabetes tipe 1, diabetes tipe 2, dan diabetes gestasional. Beberapa peneliti menemukan bahwa diabetes gestasional hanya terjadi pada wanita, yaitu selama kehamilan karena perubahan hormon. [12]

Penyakit yang tidak dapat menyebar dari satu orang ke orang lain disebut penyakit tidak menular. Penyakit kardiovaskuler, kanker, penyakit pernapasan kronis, dan diabetes adalah empat jenis utama penyakit tidak menular. Penyakit degeneratif, yang merupakan penyakit yang disebabkan oleh penurunan fungsi organ tubuh, semakin meningkat dan berbahaya karena gaya hidup modern, termasuk pola makan, merokok, konsumsi alkohol, dan penggunaan obat-obatan.[6]

Sampai saat ini, diabetes masih tercatat sebagai penyebab kematian paling umum di dunia, dengan prevalensi yang terus meningkat secara signifikan setiap tahun. Gejala yang ditandai dengan sering makan (polifagi), sering minum (polidipsi), dan sering kencing (poliuri) adalah akibat dari umumnya dikenal sebagai diabetes, adalah salah satu penyakit kronis yang menjadi beban kesehatan global yang semakin meningkat.. Oleh karena itu, pencegahan, deteksi dini, dan manajemen yang efektif sangat penting dalam mengatasi masalah ini. Teknologi informasi dan pembelajaran mesin telah memainkan peran yang semakin penting dalam pencegahan dan manajemen diabetes. Dalam upaya ini, penelitian ini memperkenalkan aplikasi berbasis web yang inovatif dan canggih, dikenal sebagai DIATECT (Diabetes Prediction using Support Vector Machine on the Web). Aplikasi ini bertujuan untuk membantu dalam prediksi risiko seseorang terkena diabetes dengan menggunakan algoritme Support Vector Machine (SVM) yang telah terbukti akurat dalam analisis data kesehatan. Support Vector Machine (SVM) memilih hiperplane margin maksimum dalam ruang input yang telah diubah untuk memisahkan kelas contoh, dengan tujuan memaksimalkan jarak ke contoh yang terpisah dengan bersih. Parameter dari hiperplane diperoleh melalui optimasi pemrograman kuadrat. Contoh sederhana mencakup metode klasifikasi yang memisahkan dua kelas contoh dalam ruang fitur dengan hiperplane yang wajar melewati antara rata-rata kelas. Hiperplane ini dapat dilihat sebagai permukaan keputusan yang menetapkan titik baru ke kelas yang rata-ratanya lebih dekat dengannya, dengan sudut antara vektor yang menghubungkan rata-rata kelas. Penggunaan fungsi kernel secara implisit mendefinisikan permukaan keputusan dalam hal kemiripan antara titik baru dan rata-rata kluster. Dalam teknik yang lebih canggih, SVM hanya bergantung pada subset contoh pelatihan yang disebut sebagai vektor dukungan, memungkinkan penyesuaian lebih fleksibel terhadap distribusi data dan pengurangan biaya komputasi. Penggunaan trik kernel memungkinkan SVM memiliki model linear dan sejumlah fungsi keputusan non-linear yang kaya. Dalam konteks SVM, trik kernel memproyeksikan data ke dalam ruang fitur yang lebih tinggi, memungkinkan pembentukan hiperplane yang efektif. Mercer's condition memastikan bahwa kernel memenuhi syarat matematika sebagai perkalian titik di suatu ruang. Meskipun pemilihan kernel terbaik masih menjadi subjek penelitian aktif, beberapa kernel umum telah diidentifikasi. Penting untuk diingat bahwa SVM dapat menangani kasus non-linear dengan efektif berkat trik kernel, memungkinkan pemodelan yang lebih fleksibel dan akurat terhadap distribusi data yang kompleks.[7] Melalui penggunaan SVM dan integrasi data kesehatan yang luas, DIATECT memberikan prediksi yang andal tentang kemungkinan seseorang terjuangkit penyakit diabetes.

Penelitian ini menguraikan pengembangan dan implementasi DIATECT, serta membahas manfaat potensialnya dalam mendukung upaya manajemen penyakit diabetes. DIATECT adalah alat prediksi, tetapi juga sumber informasi dan panduan untuk individu yang sudah menderita diabetes. Dengan akses melalui platform web yang mudah diakses, aplikasi ini memiliki potensi besar untuk memengaruhi kesehatan masyarakat dengan memberikan pemahaman yang lebih baik tentang risiko diabetes.

2. METODE PENELITIAN

2.1. Metodologi penelitian

Gambar 1 menunjukkan prosedur yang digunakan untuk melakukan penelitian tentang klasifikasi penderita diabetes menggunakan metode SVM.



Gambar 1. Flowchart Penelitian

Dapat dipahami Gambar 1 bahwa tahap pertama adalah proses menentukan ide penelitian yang akan dilakukan dan dilanjutkan dengan proses pencarian data yang sesuai dengan ide penelitian. Selanjutnya pengumpulan data yang kami peroleh dari kaggle dalam bentuk CSV. Data yang diperoleh berupa data orang yang terkena diabetes sebanyak 268, dan data orang yang tidak terkena sebanyak 500 data. Jika dijumlahkan semuanya sebanyak 768 data. Selanjutnya proses load dataset melibatkan file dataset diabetes perlu disiapkan dan diakses oleh aplikasi, biasanya dalam format CSV atau Excel. Standarisasi data, mengubah setiap nilai ke dalam skala z-score, distribusi data menjadi terstandarisasi dengan mean 0 dan deviasi standar 1. Klasifikasi data training dan testing proses menjelaskan atau membedakan ide atau kelas data berdasarkan tujuan tertentu data training diambil sebesar 614 data dan data testing sebesar 154 data. Algoritma SVM proses dimana memisahkan data kedua kelas secara maksimal, dengan margin yang paling baik, yang merupakan jarak garis pemisah dengan anggota-anggota terdekat kedua kelas. Terakhir proses modelling dilakukan dengan menggunakan dataset yang relevan untuk memastikan kualitas dan akurasi model dimana Tahapan modeling pada algoritma SVM mencakup langkah-langkah khusus yang disesuaikan dengan karakteristik dan prinsip kerja SVM..

2.2. Analisis Data

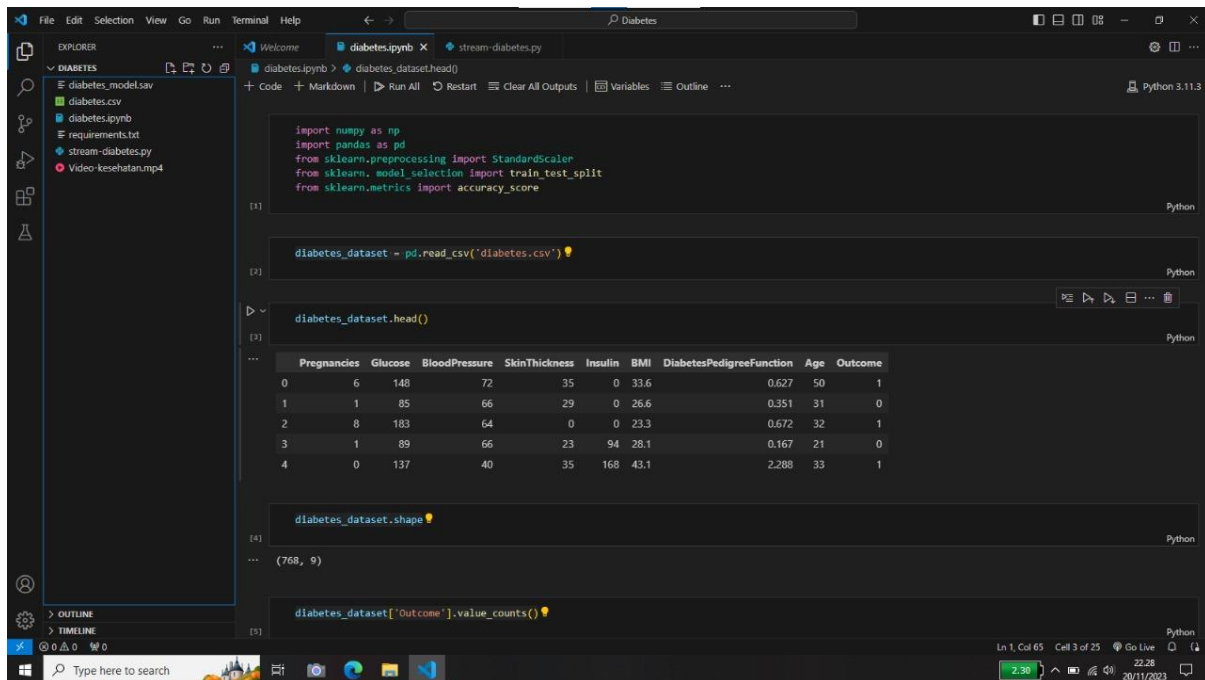
2.2.1. Pengumpulan Data

Untuk mendukung penelitian ini dilakukan pengumpulan data untuk mendapatkan sumber data set berupa data sekunder berasal dari web penyedia yaitu kaggle berjudul diabetes data set. Data sekunder merupakan data yang didapatkan dari penelitian sebelumnya tanpa harus terjun langsung ke lapangan. Jumlah dataset pada penelitian ini sebanyak 768 terdiri dari 9 atribut terdiri dari Pregnancies, Glucose, Blood Pressure, Skin Thickness, Insulin, BMI, Diabetes Pedigree Function, Age dan Outcome.

Tabel 1. Jenis keterangan yang digunakan

Keterangan	label	Jumlah
Terkena	1	268
Tidak terkena	0	500

Setelah mengumpulkan data yang kami koversi ke file CSV. Kami menentukan library yang digunakan dan meload dataset yang telah dibuat. Berikut langkah pemanggilan library dan meload data yang dilakukan terdapat pada Gambar 2.



Gambar 2. Memanggil Library Dan Meload Data Set

2.2.2. Pelabelan atribut data

Dalam penelitian itu peneliti melakukan pelabelan secara manual dimana kami menggunakan label 0 dan 1. Label 0 di kategorikan dengan keterangan pasien tidak terkena diabetes sedangkan label 1 dikategorikan dengan keterangan pasien terkena diabetes. Label – label tersebut dapat di lakukan dan dapat di lihat teks beserta label yang di masukan kedalam tabel 2. Berikut contoh teks beserta label yang tercantum pada tabel 2 :

Tabel 2. Jenis prediksi beserta Labelnya

Keterangan	Label
Tidak terkena diabetes	0
Terkena diabetes	1

2.2.3. Preprocessing Data

Sebelum proses pengklasifikasian dimulai, tahap preprocessing diperlukan untuk membersihkan, menghilangkan, dan mengubah sumber data.[8] Tujuan untuk memastikan bahwa data yang digunakan selama proses pengklasifikasian menjadi lebih optimal. Preprocessing data penelitian dilakukan dalam dua tahap: pembersihan (pembersihan data) dan transformasi (transformasi data). Setelah pengumpulan data, pembersihan data dilakukan untuk menghilangkan data yang tidak relevan atau tidak diperlukan; contohnya, data kosong dan data yang tidak penting diambil secara manual. Selanjutnya, data diubah dari tipe kategori menjadi tipe binary. Setelah tahap pembersihan data, 768 data baru, dan jumlah atribut yang dihasilkan setelah tahap transformasi data adalah 9 atribut.

2.2.4. Klasifikasi Data Training dan Data Testing

Klasifikasi adalah upaya untuk menemukan model atau fungsi yang menjelaskan atau membedakan ide atau kelas data berdasarkan tujuan tertentu. untuk memperkirakan kelas darisuat objek, umumnya selama proses klasifikasi memiliki dua tahapan, yaitu

1. Proses pelatihan: dalam proses pelatihan digunakan instruksi yang telah diketahui labelnya untukkonstruksi model atau fungsi.
2. Proses pengujian: untuk memastikan keakuratan model atau fungsi yang akan dibuat selama proses pelatihan, maka digunakan data yang dikenal sebagai set pengujian. Untuk mengetahui labelnya.[9]

Data yang di sajikan sebanyak 768 data. Data tersebut akan dibagi menjadi Data training dan Data Testing. Data yang dihasilkan tercantum dalam tabel di bawah ini.

Tabel 3. Data Training dan Data Testing

Pembagian	Jumlah
Data latih	614
Data uji	154
Jumlah	768

Dapat dilihat dari tabel 2 terdapat 2 kelompok yaitu Data latih (training) dan Data uji (testing). Kelompok data latih 80% atau sebanyak 614 data sedangkan data uji 20% atau sebanyak 154 data yang dihasilkan.

3. HASIL DAN PEMBAHASAN

3.1 Proses Load Data Set

Dalam pengembangan Aplikasi DIATECT untuk Prediksi Penyakit Diabetes menggunakan SVM Berbasis Web dengan Python, proses load dataset melibatkan beberapa langkah yang terperinci. Pertama, file dataset diabetes perlu disiapkan dan diakses oleh aplikasi, biasanya dalam format CSV atau Excel. Selanjutnya, kita memilih library atau fungsi dalam Python yang sesuai untuk membantu dalam proses membaca dataset, seperti pandas atau openpyxl untuk membaca file CSV atau Excel. Selanjutnya, kita membuka dan membaca file dataset menggunakan library yang telah dipilih. Sebagai contoh, dengan menggunakan pandas, langkah ini dapat diimplementasikan dengan beberapa baris kode:

```

import numpy as np
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

diabetes_dataset = pd.read_csv('diabetes.csv')

diabetes_dataset.head()

diabetes_dataset.shape

diabetes_dataset['Outcome'].value_counts()
    
```

The screenshot shows a Jupyter Notebook interface with the following code and output:

```

import numpy as np
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

diabetes_dataset = pd.read_csv('diabetes.csv')

diabetes_dataset.head()

diabetes_dataset.shape

diabetes_dataset['Outcome'].value_counts()
    
```

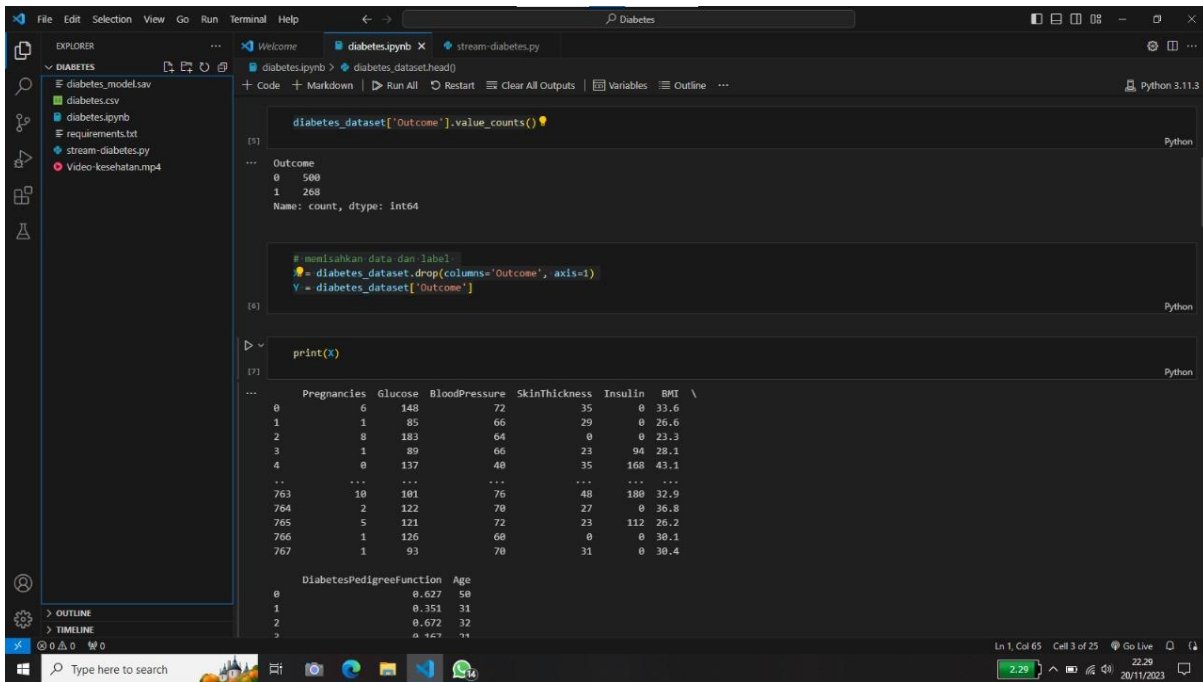
The output for `diabetes_dataset.head()` is:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

The output for `diabetes_dataset.shape` is `(768, 9)`.

Gambar 3. Proses Load Data Set

Setelah membaca dataset, langkah berikutnya adalah memproses baris-baris dataset. Setiap baris dataset diproses untuk mendapatkan nilai-nilai atribut dan label yang dibutuhkan. Misalnya, jika dataset memiliki kolom-kolom seperti “Usia”, “Tekanan Darah”, dan “Label Diabetes”, kita dapat mengakses nilai-nilai ini sebagai berikut:

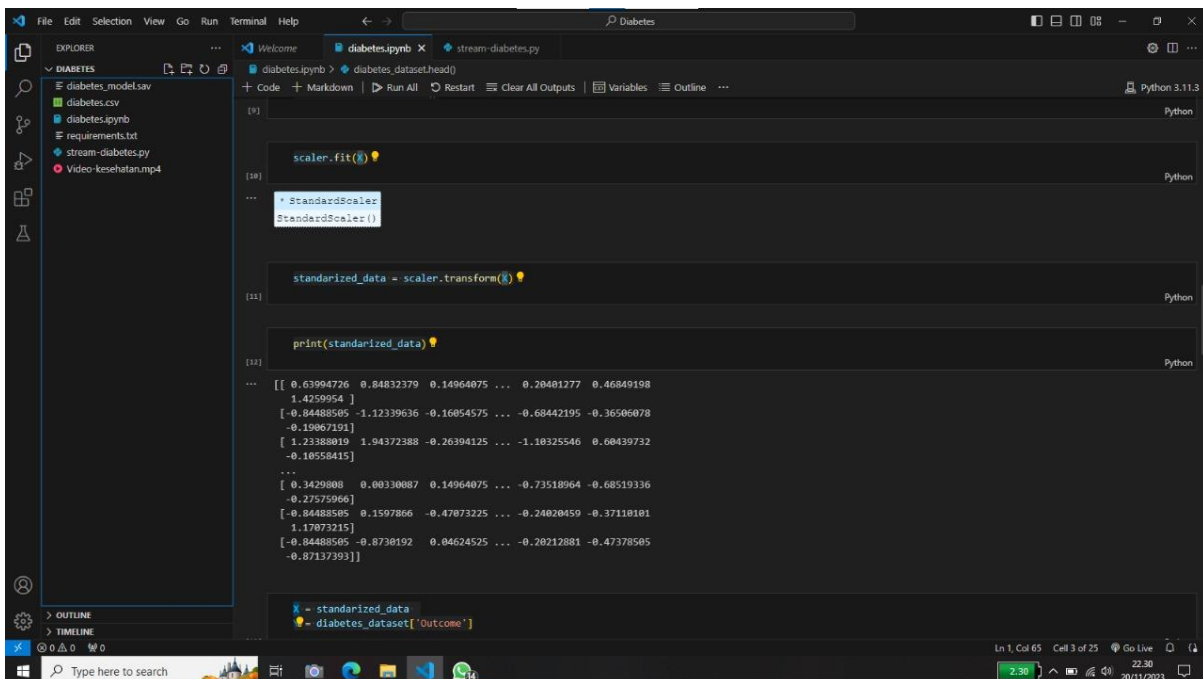


Gambar 4. Proses Baris Data Set

Setelah memproses nilai-nilai dari setiap baris dataset, kita dapat memuat data ke dalam struktur yang sesuai untuk digunakan dalam proses pelatihan dan pengujian model SVM. Selanjutnya, aplikasi dapat melanjutkan proses pengembangan, seperti membangun model SVM, melatihnya dengan dataset yang dimuat, dan mengintegrasikannya ke dalam aplikasi web untuk prediksi penyakit diabetes.

3.2 Standarisasi Data

Tahapan standarisasi data pada pembuatan Aplikasi DIATECT untuk prediksi penyakit diabetes menggunakan algoritma Support Vector Machine (SVM) berbasis web melibatkan perhitungan z-score untuk setiap fitur dalam dataset. [10] Dengan mengubah setiap nilai ke dalam skala z-score, distribusi data menjadi terstandarisasi dengan mean 0 dan deviasi standar 1. Langkah ini memastikan bahwa model SVM dapat memproses data dengan efisien, menghindari dominasi oleh fitur-fitur dengan rentang nilai yang besar, dan menghasilkan prediksi yang lebih konsisten. Tahapan ini memainkan peran kunci dalam meningkatkan akurasi dan stabilitas model.



Gambar 5. Proses Standarisasi Data

3.4 Pembagian Data

Pemisahan Data Latih dan Data Uji:

Pisahkan dataset menjadi dua subset utama: data latih (training data) dan data uji (testing data). Data latih digunakan untuk melatih model SVM, sedangkan data uji digunakan untuk menguji kinerja model yang telah dilatih. Model SVM dilatih dengan menggunakan data latih, dan parameter SVM disesuaikan melalui penyetelan hyperparameter. Evaluasi kinerja model dilakukan dengan menggunakan metrik evaluasi yang relevan. Opsi untuk validasi silang digunakan untuk memastikan konsistensi model. Akhirnya, model SVM yang telah dilatih disimpan untuk integrasi dengan Aplikasi DIATECT sehingga dapat memberikan prediksi yang akurat terhadap data baru. Tahapan ini membentuk dasar untuk memastikan kehandalan Aplikasi DIATECT dalam memprediksi penyakit diabetes.

```

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, stratify=Y, random_state=2)

print(X.shape, X_train.shape, X_test.shape)

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

from sklearn import svm
classifier = svm.SVC(kernel='linear')

classifier.fit(X_train, Y_train)

+ SVC
SVC(kernel='linear')

X_train_prediction = classifier.predict(X_train)
training_data_accuracy = accuracy_score(X_train_prediction, Y_train)

print('Akurasi data training adalah = ', training_data_accuracy)

Akurasi data training adalah = 0.7866449511480652
    
```

Gambar 6. Proses Pembagian Data

3.5 Algoritma SVM

Support Vector Machine (SVM) pertama kali dipublikasikan oleh Vapnik pada tahun 1992 bersama Bernhard Boser dan Isabelle Guyon. Untuk mengembalikan pengujian data awal ke ukuran yang lebih besar, algorithm support vector machine menggunakan pemetaan nonlinier. Tujuan dari algoritma pembelajaran mesin SVM adalah untuk membuat hyperlane yang mengatur dua kelas pada ruang input. Algoritma ini menggunakan prinsip minimisasi risiko struktural. Margin adalah ruang terbesar yang dapat digunakan untuk membedakan dua kelas. Dot product yang memiliki jarak terdekat dengan margin adalah hal yang paling penting dalam menemukan support vector. Untuk meminimalkan batasan kesalahan umum, SVM juga memiliki tujaun. Salah satu kelebihan algorithm ini, yaitu penggunaan metode SVM, dapat dipelajari secara teoretis dengan menggunakan teori pembelajaran komputasi.[11]

Prinsip utama SVM adalah mencari fungsi garis pemisah, atau hyperplane, yang dapat memisahkan data kedua kelas secara maksimal. Maksimal yang dimaksud adalah garis pemisah dapat memisahkan data kedua kelas dengan margin yang paling baik, yang merupakan jarak garis pemisah dengan anggota-anggota terdekat kedua kelas. Garis pemisah optimal disebut sebagai margin.[12]

Hyperplane merupakan suatu fungsi matematis yang digunakan untuk membedakan antara fitur-fitur dalam ruang dimensi yang berbeda. Pada dimensi dua, hyperplane ini dapat diidentifikasi sebagai suatu garis, yang bertindak sebagai pemisah antara dua kelompok atau kelas fitur. Seiring dengan peningkatan dimensi, konsep hyperplane berkembang menjadi entitas yang lebih kompleks dan sangat penting dalam konteks klasifikasi data.

Dalam dimensi tiga, hyperplane akan menjadi suatu bidang yang berfungsi untuk mengklasifikasikan fitur-fitur yang ada dalam ruang tersebut. Sama halnya, dalam dimensi yang lebih tinggi, hyperplane terus berkembang, dan fungsi ini dapat digunakan untuk memisahkan dan mengklasifikasikan titik-titik dalam ruang dimensi yang lebih kompleks.

Pentingnya hyperplane terletak pada kemampuannya untuk memberikan kerangka kerja matematis yang kuat untuk pemisahan dan klasifikasi fitur dalam berbagai konteks. Dalam dunia pembelajaran mesin dan statistika, konsep hyperplane sering digunakan dalam algoritma-algoritma klasifikasi, seperti Support Vector Machines (SVM), di mana hyperplane digunakan untuk memaksimalkan batasan antara kelas-kelas data yang berbeda.

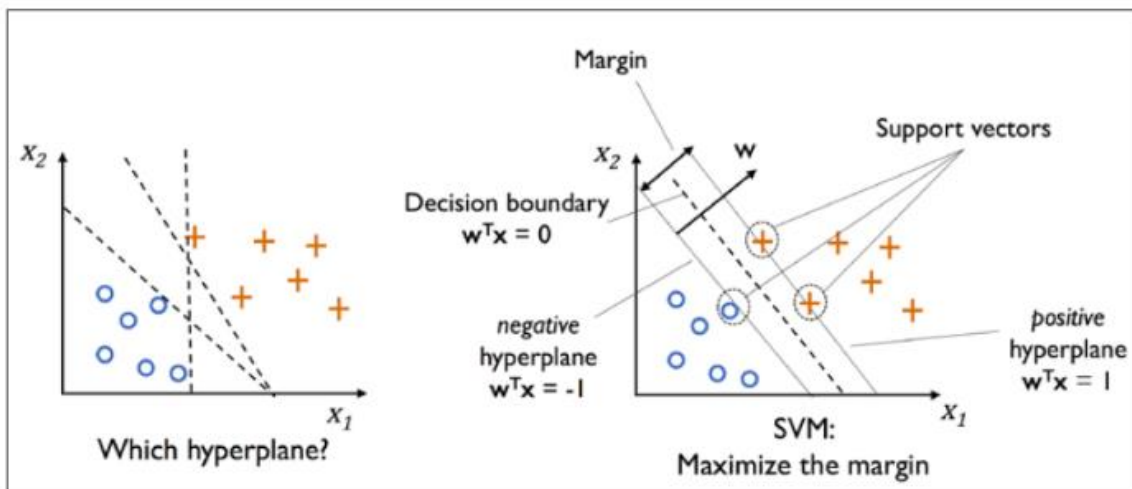
Dengan demikian, hyperplane bukan hanya suatu entitas matematis yang memisahkan fitur-fitur, tetapi juga merupakan konsep fundamental yang memainkan peran penting dalam analisis data multidimensional dan pengambilan keputusan berbasis model.[13]

Support Vector Machine (SVM) adalah algoritma pembelajaran mesin yang digunakan untuk klasifikasi dan regresi. Dalam konteks klasifikasi linier, proses SVM dimulai dengan inialisasi bobot (weights) w dan bias b , yang kemudian digunakan untuk menghitung nilai output z dengan menggunakan fungsi $z = w \cdot x + b$, di mana x adalah vektor fitur data. Fungsi keputusan $f(x)$ kemudian ditentukan berdasarkan nilai z , di mana hasil klasifikasi positif atau negatif bergantung pada apakah z lebih besar atau sama dengan nol.

SVM berfokus pada optimalisasi margin, yang dihitung sebagai $\frac{2}{\|w\|}$. Tujuan dari SVM adalah memaksimalkan margin ini dengan mengoptimalkan w dan b dengan meminimalkan $\frac{1}{2} \|w\|^2$, sambil mematuhi Batasan Batasan $y_i (w \cdot x_i + b) \geq 1$, di mana y_i adalah label kelas data.

Dalam konteks penggunaan kernel SVM, data pertama kali diubah ke dimensi ruang fitur yang lebih tinggi menggunakan kernel function. Sebagai contoh, kernel Radial Basis Function (RBF) sering digunakan dan dihitung dengan $K(x, x') = \exp\left(-\frac{\|x-x'\|^2}{2\sigma^2}\right)$. Proses ini memungkinkan SVM menangani masalah klasifikasi yang tidak linier. Hasil akhir fungsi keputusan pada SVM dengan kernel diperoleh melalui perhitungan $z = \sum_{i=1}^n \alpha_i y_i K(x, x') + b$, dengan α_i sebagai bobot Lagrange dan N adalah jumlah sampel. Fungsi keputusan $f(x)$ dihasilkan dari menentukan tanda (positif/negatif) dari z .

Penting untuk dicatat bahwa proses lebih lanjut, seperti optimasi parameter seperti σ untuk kernel RBF, melibatkan teknik matematika dan optimasi yang kompleks. Dalam praktiknya, implementasi SVM sering memanfaatkan perangkat lunak atau library khusus untuk mengatasi perhitungan yang rumit ini.



Gambar 7. Klasifikasi data menggunakan Support Vector Machine (SVM)

Garis H1, H2, dan hyperplane memisahkan kedua kelas, seperti yang ditunjukkan pada gambar 3. Persamaan 3 menghubungkan data kelas A (H1) dengan dot product variabel dan konstanta pada setiap notasi, dan W adalah nilai yang tegak lurus dengan.

$$X \cdot w + b \leq -1$$

$$w \cdot X_i + b \text{ lebih dari } 1$$

Hyperplane bersinggungan dengan data kelas B (H2) saat ini dalam Persamaan 4.

$$w \cdot X + b = 0$$

Selain itu, hyperplane garis di antara hyperplane kelas A dan kelas B disebut Persamaan 5. Data yang berhubungan dengan vektor pendukung H1 di kelas A dan H3 di kelas B disebut sebagai vektor pendukung.

Untuk menemukan titik minimal, yaitu dengan $\frac{1}{\|w\|}$, diperlukan penentuan margin. Persamaan untuk mencari titik minimal adalah sebagai berikut:

$$\min_w \tau(w) = \frac{1}{2} \|w\|^2$$

Dengan memperhatikan nilai constrain:

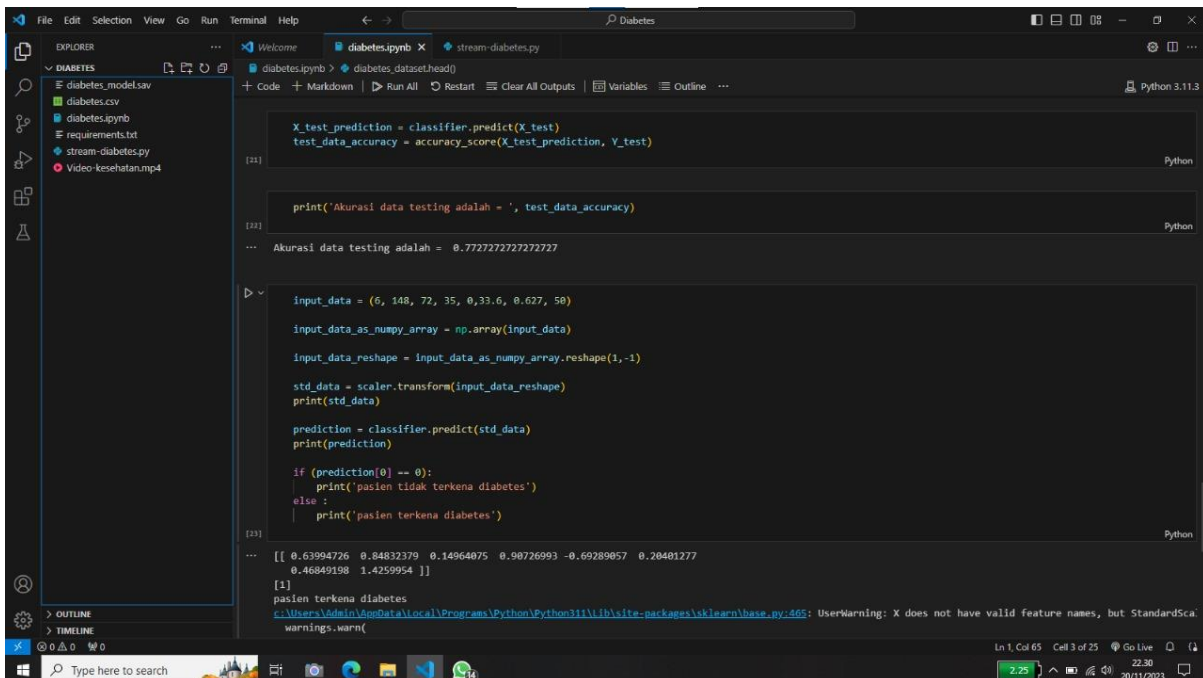
$$y_i (X_i \cdot w + b) - 1 \geq 0, \quad \forall_i$$

Tidak mungkin untuk menggunakan SVM linear karena beberapa sampel data saat ini tidak memiliki data yang terpisah secara linear. Jika dipaksakan, akan memberikan hasil klasifikasi yang buruk dan tidak optimal. Karena itu, SVM linear harus diubah menjadi SVM non-linear dengan metode kernel. Metode ini berbeda dari metode klasifikasi umum karena mengurangi aspek awal untuk menyederhanakan proses komputasi dan meningkatkan kinerja. [14]

3.6 Modelling

Source code yang digunakan difokuskan pada pengembangan aplikasi web menggunakan Streamlit dan pemanfaatan model yang telah melalui tahap pelatihan sebelumnya. Sebelumnya, proses pelatihan model Support Vector Machine (SVM) dilakukan dengan menggunakan dataset yang relevan untuk memastikan kualitas dan akurasi model. Hasil dari pelatihan tersebut, yang disebut sebagai `diabetes_model`, disimpan dalam sebuah file bernama `'diabetes_model.sav'`. File ini memuat informasi dan parameter-parameter yang telah dipelajari oleh model selama tahap pelatihan.

Dengan adanya model yang telah tersimpan, aplikasi web ini dapat dengan mudah mengaksesnya menggunakan library `pandas` untuk membaca file `'diabetes_model.sav'` dan menggambarkan langkah-langkah proses prediksi penyakit diabetes berdasarkan nilai-nilai yang dimasukkan oleh pengguna. Oleh karena itu, source code ini menghadirkan antarmuka yang memudahkan pengguna untuk melakukan prediksi dengan model SVM yang telah terlatih tersebut, serta menampilkan hasil prediksi dengan jelas melalui pesan yang diberikan oleh aplikasi. Penting untuk dicatat bahwa keberhasilan dan keandalan aplikasi ini sangat bergantung pada kualitas dan representativitas dataset yang digunakan dalam tahap pelatihan model SVM sebelumnya.



Gambar 8. Modeling

3.7 Akurasi

Akurasi dari data testing sebesar 77% dalam konteks aplikasi pendeteksi diabetes mengindikasikan sejauh mana model mampu memberikan prediksi yang benar terhadap data yang belum pernah dilihatnya sebelumnya. Secara umum, akurasi dihitung sebagai persentase dari jumlah prediksi yang benar terhadap total data testing.

Dalam kasus ini, akurasi sebesar 77% berarti bahwa dari total data yang digunakan untuk pengujian, model berhasil memprediksi dengan benar pada 77% kasus. Sisanya, sekitar 23%, mungkin merupakan prediksi yang tidak tepat.

Namun, perlu diingat bahwa akurasi sendiri mungkin tidak memberikan gambaran lengkap tentang kinerja model, terutama jika kelas-kelas dalam dataset tidak seimbang. Oleh karena itu, bergantung pada kebutuhan spesifik aplikasi, sebaiknya juga mempertimbangkan metrik evaluasi lain seperti presisi, recall, atau F1-score, terutama jika hasil positif dan negatif memiliki dampak yang berbeda secara signifikan.

Akurasi adalah metrik evaluasi yang mengukur sejauh mana model dapat memberikan prediksi yang benar terhadap data yang belum pernah dilihatnya sebelumnya. Dalam konteks aplikasi pendeteksi diabetes, hasil akurasi sebesar 77% mengindikasikan seberapa baik model dapat mengklasifikasikan data testing.

Secara umum, akurasi dihitung sebagai persentase dari jumlah prediksi yang benar terhadap total data testing. Dalam rumus, akurasi dapat dinyatakan sebagai berikut:

$$\text{Akurasi} = \frac{\text{Jumlah Prediksi Benar}}{\text{Total Data Testing}} \times 100\%$$

Dalam kasus ini, dengan akurasi sebesar 77%, dapat diartikan bahwa dari seluruh data yang digunakan untuk pengujian, model berhasil memprediksi dengan benar pada 77% kasus. Sisanya, sekitar 23%, mungkin merupakan prediksi yang tidak tepat. Meskipun akurasi memberikan gambaran umum tentang kinerja model, perlu diingat bahwa metrik lain seperti presisi, recall, dan F1-score juga memberikan wawasan yang berharga, terutama jika ada ketidakseimbangan antara kelas-kelas yang diuji.

Untuk lebih merinci sistem perhitungan Support Vector Machine (SVM) dalam mencapai hasil akurasi tersebut, perlu dipahami bahwa SVM bekerja dengan menemukan hyperplane yang memaksimalkan margin antara kelas-kelas. Proses pelatihan melibatkan optimasi parameter, seperti bobot dan bias, yang memungkinkan SVM membuat keputusan terbaik dalam menentukan batas keputusan antar kelas. SVM juga dapat menggunakan kernel untuk menangani masalah klasifikasi non-linier dengan mentransformasi data ke dalam ruang fitur yang lebih tinggi.

Namun, untuk mendapatkan pemahaman yang lebih mendalam tentang kinerja SVM dalam aplikasi ini, perlu meninjau parameter khusus yang digunakan (seperti jenis kernel, parameter C, dll.), serta melakukan analisis lebih lanjut terhadap metrik evaluasi lainnya guna memahami aspek-aspek tertentu dari kinerja model.

```

X_test_prediction = classifier.predict(X_test)
test_data_accuracy = accuracy_score(X_test_prediction, Y_test)

print('Akurasi data testing adalah = ', test_data_accuracy)

```

... Akurasi data testing adalah = 0.7727272727272727

Gambar 9. Akurasi Data

3.8 Performa Aplikasi

Aplikasi ini menggunakan Streamlit sebagai platform untuk mengimplementasikan antarmuka pengguna yang memungkinkan pengguna memprediksi kemungkinan diabetes berdasarkan model yang telah dilatih menggunakan algoritma Support Vector Machine (SVM). Dalam proses inisialisasi, aplikasi membaca model SVM yang telah disimpan sebelumnya dalam file 'diabetes_model.sav'. Meskipun proses loading model dapat mempengaruhi performa aplikasi, khususnya jika modelnya sangat besar, namun hal ini umumnya tidak memerlukan waktu yang signifikan.

Antarmuka pengguna menyediakan kolom input di mana pengguna diminta untuk memasukkan nilai-nilai seperti jumlah kehamilan, kadar glukosa, tekanan darah, ketebalan kulit, insulin, indeks massa tubuh (BMI), fungsi keturunan diabetes, dan usia. Proses prediksi terjadi ketika pengguna menekan tombol "test prediksi diabetes". Tombol ini dirancang untuk memberikan kontrol pengguna atas kapan proses prediksi dilakukan, mengoptimalkan interaksi pengguna.

Proses prediksi sendiri dilakukan oleh model SVM berdasarkan nilai-nilai yang dimasukkan. Hasil prediksi kemudian ditampilkan dalam antarmuka pengguna dengan memberikan informasi apakah pasien diperkirakan terkena diabetes atau tidak. Tampilan hasil prediksi dibuat informatif dengan warna latar belakang yang berbeda,

di mana warna hijau menunjukkan prediksi yang benar (sesuai dengan keadaan sebenarnya), sedangkan warna merah menandakan prediksi yang salah.

Secara keseluruhan, performa aplikasi ini dapat dianggap baik karena sifatnya yang sederhana dan responsif. Untuk evaluasi performa yang lebih mendalam, diperlukan metrik khusus seperti waktu respons, tingkat akurasi, dan parameter evaluasi lainnya.

Hasil performa aplikasi diadetek yang dicantumkan melibatkan tiga metrik evaluasi utama: presisi, recall, dan F1-score. Berikut adalah uraian lebih terperinci untuk masing-masing metrik tersebut:

1. **Presisi (Precision):**

- Presisi mengukur seberapa banyak dari hasil positif yang diberikan oleh model yang sebenarnya relevan. Dalam konteks aplikasi ini, presisi mencapai 77%. Artinya, dari semua hasil positif yang diidentifikasi oleh aplikasi, 77% di antaranya benar-benar relevan atau akurat.

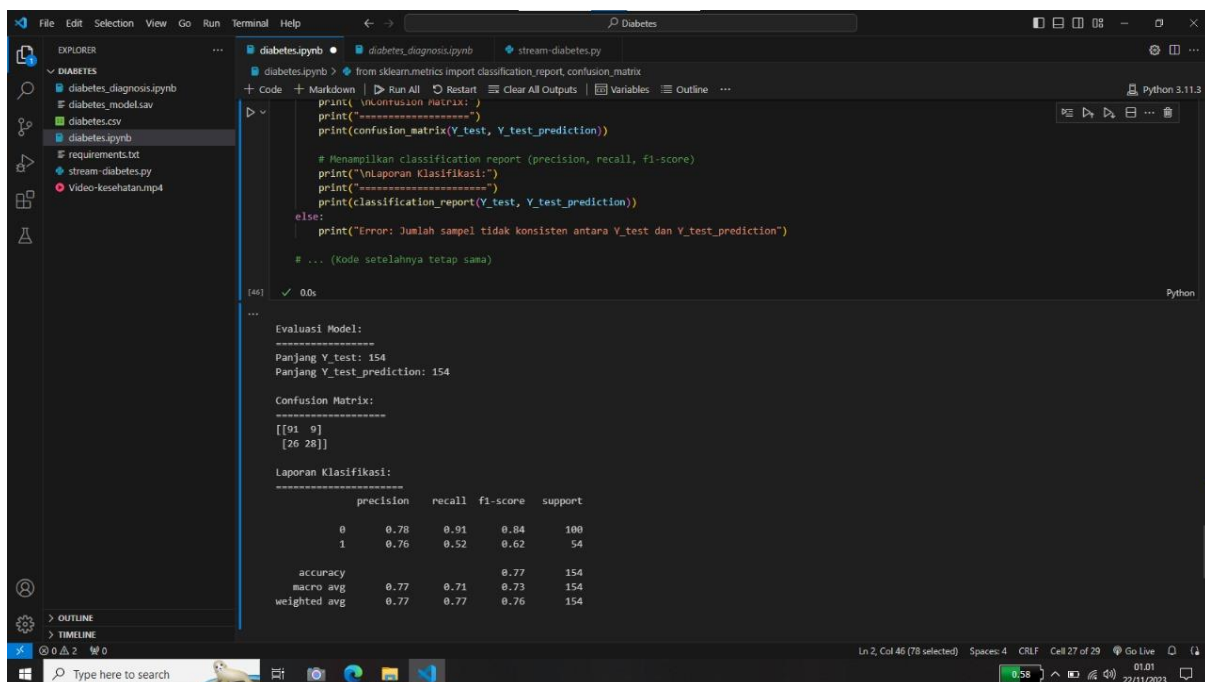
2. **Recall (Sensitivitas atau True Positive Rate):**

- Recall mengukur seberapa banyak dari total kasus positif yang diidentifikasi oleh model. Dalam hasil performa ini, recall mencapai 91%. Ini berarti bahwa aplikasi berhasil mengidentifikasi 91% dari keseluruhan kasus positif yang sebenarnya.

3. **F1-score:**

- F1-score adalah metrik yang menyatukan presisi dan recall untuk memberikan gambaran holistik tentang kinerja model. F1-score mencapai 91% pada aplikasi ini. F1-score yang tinggi menunjukkan keseimbangan yang baik antara presisi dan recall, menunjukkan bahwa model memiliki kemampuan baik dalam mengidentifikasi kelas positif dan menghindari kesalahan klasifikasi.

Dengan angka-angka ini, dapat disimpulkan bahwa aplikasi memiliki kinerja yang kuat dalam melakukan klasifikasi atau deteksi, dengan kemampuan tinggi dalam mengenali kasus positif dan memberikan hasil yang akurat. Namun, tetap penting untuk mempertimbangkan konteks spesifik dari aplikasi ini, dan evaluasi lebih lanjut dapat dilakukan untuk memahami kecocokan model dengan kebutuhan dan tujuan penggunaannya.



Gambar 10. Performa Aplikasi

3.9 . Tampilan Aplikasi

Pada tampilan aplikasi terdapat 8 kolom antara lain kolom input prenanais, input nilai glucose, ininput nilai blood pressur, input nilai skinthicnees, input nilai insulin, input nilai bmi, input nilai diabetes pedigree function, dan input nilai age.

Gambar 11. Tampilan Aplikasi

4. KESIMPULAN

Pembuatan Aplikasi DIATECT untuk prediksi penyakit diabetes menggunakan SVM berbasis web, dapat disimpulkan bahwa aplikasi ini berhasil menyederhanakan proses prediksi risiko diabetes dengan tingkat akurasi mencapai 77% dan performa aplikasi yang baik dengan nilai presisi mencapai 77%, recall mencapai 91%, dan F1-score mencapai 91. Aplikasi yang sedang dikembangkan ini memiliki tujuan khusus untuk menentukan apakah pengguna terjangkit diabetes atau tidak berdasarkan input yang dimasukkan. Aplikasi ini menggunakan Streamlit sebagai platform untuk mengimplementasikan antarmuka pengguna yang memungkinkan pengguna memprediksi kemungkinan diabetes berdasarkan model yang telah dilatih menggunakan algoritma Support Vector Machine (SVM). Fokus utama penelitian ini terletak pada pengembangan suatu sistem berbasis web yang menggunakan algoritma Support Vector Machine (SVM) untuk mendeteksi gejala diabetes pada pengguna berdasarkan data yang mereka masukkan. Dengan memanfaatkan model SVM yang telah dilatih, hasil web Aplikasi DIATECT fokus pada memberikan informasi singkat apakah pengguna teridentifikasi sebagai individu yang memiliki risiko diabetes atau tidak. Proses pengembangan aplikasi mencakup langkah-langkah efisien, seperti pengumpulan data latih, preprocessing, dan pelatihan model SVM. Hasil web yang dihasilkan menyajikan informasi prediksi dengan jelas dan mudah dipahami, memberikan pengguna pemahaman instan tentang status risiko diabetes mereka. Meskipun fokus utama aplikasi adalah pada prediksi diabetes, kesederhanaan hasil web tidak mengorbankan ketepatan atau keakuratan. Aplikasi ini tetap memberikan nilai tambah dengan memberikan informasi yang dapat membantu pengguna mengambil langkah-langkah preventif yang tepat sesuai dengan hasil prediksi. Dengan demikian, Aplikasi DIATECT berhasil menyederhanakan pengalaman pengguna dalam menerima informasi prediksi diabetes secara langsung melalui hasil web, memungkinkan pengguna untuk segera mengambil tindakan yang sesuai. Ini menegaskan bahwa teknologi SVM berbasis web dapat secara efektif mendukung upaya pencegahan penyakit diabetes dengan memberikan informasi yang tepat pada waktu yang tepat.

UCAPAN TERIMAKASIH

Penulis dengan penuh rasa tulus ingin menyampaikan penghargaan yang mendalam kepada semua pihak yang telah memberikan kontribusi yang tak ternilai dalam rangka penyelesaian penelitian ini. Terima kasih yang tak terhingga kami sampaikan kepada yang dengan murah hati memberikan dukungan finansial yang memungkinkan jalannya penelitian ini. Selanjutnya, ucapan terima kasih kami peruntukkan kepada setiap responden yang dengan sukarela dan antusias berpartisipasi dalam setiap tahapan penelitian ini, memberikan sudut pandang yang berharga.

Lebih lanjut, kami juga ingin mengungkapkan apresiasi yang sangat kepada orang tua kami yang telah memberikan dukungan moral dan motivasi selama penelitian, terimakasih kami dan menyampaikan penghargaan kepada Ketua Prodi Teknik Elektro di Universitas Islam Nusantara, yang telah memberikan bimbingan dan arahan

yang berharga dalam senantiasanya memberikan bimbingan, masukan berharga, dan dukungan teknis yang tidak ternilai selama seluruh proses penelitian. Serta kami sangat berterima kasih kepada teman-teman yang telah bertindak sebagai pilar dukungan sosial dan akademis, memberikan wawasan, saran, dan semangat yang telah membantu kami mencapai hasil penelitian yang signifikan.

REFERENCES

- [1] F. Amira Mumtaz, R. Maulana, and A. S. Budi, "Sistem Pendeteksi Penyakit Diabetes Melitus berdasarkan Kondisi Urin dan Gas Buang Pernapasan menggunakan K-Nearest Neighbor berbasis Arduino," 2020. [Online]. Available: <http://j-ptiik.ub.ac.id>
- [2] Sulis Tina, "International Diabetes Federation (IDF)," 2020.
- [3] khoirudin Mohammad Burhan Hanif, "SISTEM APLIKASI PREDIKSI PENYAKIT DIABETES MENGGUNAKAN FITURE SELECTION KORELASI PEARSON DAN KLASIFIKASI NAIVE BAYES," *Pengembangan Rekayasa dan Teknologi*, vol. 16, Dec. 2020.
- [4] M. Ardiansyah, A. Sunyoto, and E. T. Luthfi, "Analisis Perbandingan Akurasi Algoritma Naïve Bayes Dan C4.5 untuk Klasifikasi Diabetes," *Edumatic: Jurnal Pendidikan Informatika*, vol. 5, no. 2, pp. 147–156, Dec. 2021, doi: 10.29408/edumatic.v5i2.3424.
- [5] "JEPIN (Jurnal Edukasi dan Penelitian Informatika) Peningkatan Kinerja Akurasi Prediksi Penyakit Diabetes Mellitus Menggunakan Metode Grid Search pada Algoritma Logistic Regression".
- [6] I. Sriani Masitha, N. Media, N. Wulandari, M. Amin Tohari, P. Kesehatan Masyarakat, and F. Kesehatan Masyarakat, "Seminar Nasional Pengabdian Masyarakat LPPM UMJ Website: http://jurnal.umj.ac.id/index.php/semnas_kat", [Online]. Available: <http://jurnal.umj.ac.id/index.php/semnas>
- [7] O. Maimon and L. Rokach, "Data Mining and Knowledge Discovery Handbook (Second Edition)."
- [8] F. R. Lumbanraja, F. Lufiana, Y. Heningtyas, and K. Muludi, "IMPLEMENTASI SUPPORT VECTOR MACHINE (SVM) UNTUK KLASIFIKASI PENDERITA DIABETES MELLITUS 1,*)."
- [9] N. Ketut *et al.*, "IMPLEMENTASI METODE SUPPORT VECTOR MACHINE PADA SISTEM PENGENALAN JEJAITAN," 2017.
- [10] H. Ichwanul Muslim Karo Karo, "Klasifikasi Penderita Diabetes Menggunakan Algoritma Machine Learning dan Z-Score," *Jurnal Teknologi Terpadu*, vol. 8, 2022.
- [11] "KLASIFIKASI PENDERITA PENYAKIT DIABETES BERBASIS DATA REKAM MEDIS MENGGUNAKAN SUPPORT VECTOR MACHINE (SVM) SKRIPSI Oleh: ARYA ABIMANYU NIM. 17650059 PROGRAM STUDI TEKNIK INFORMATIKA FAKULTAS SAINS DAN TEKNOLOGI UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG 2023."
- [12] M. D. Purbolaksono, M. Irvan Tantowi, A. Imam Hidayat, and A. Adiwijaya, "Perbandingan Support Vector Machine dan Modified Balanced Random Forest dalam Deteksi Pasien Penyakit Diabetes," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 5, no. 2, pp. 393–399, Apr. 2021, doi: 10.29207/resti.v5i2.3008.

- [13] F. Rosyking Lumbanraja, F. Lufiana, Y. Heningtyas, K. Muludi, J. Ilmu Komputer, and U. Lampung Jl Ir Sumantri Brojonegoro, "Penerapan Metode Support Vector Machine (SVM) dalam Klasifikasi Penderita Diabetes Mellitus," 2023.
- [14] F. Handayanna, "PENERAPAN METODE SUPPORT VECTOR MACHINE MENGGUNAKAN OPTIMASI GENETIC ALGORITHM UNTUK PREDIKSI PENYAKIT DIABETES."