

**DETEKSI PENYAKIT DIABETES MENGGUNAKAN ALGORITMA  
*DEEP NEURAL NETWORK* DAN ALGORITMA *SUPPORT VECTOR*  
*MACHINE* BERBASIS *WEBSITE***

**SKRIPSI**

**KEVIN WIJAYA**

**201401144**



**PROGRAM STUDI S-1 ILMU KOMPUTER  
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI  
UNIVERSITAS SUMATERA UTARA**

**MEDAN**

**2024**

**DETEKSI PENYAKIT DIABETES MENGGUNAKAN ALGORITMA  
*DEEP NEURAL NETWORK* DAN ALGORITMA *SUPPORT VECTOR*  
*MACHINE* BERBASIS *WEBSITE***

**SKRIPSI**

Diajukan untuk melengkapi tugas dan memenuhi syarat memperoleh ijazah

Sarjana Komputer

**KEVIN WIJAYA**

**201401144**



**PROGRAM STUDI S-1 ILMU KOMPUTER  
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI  
UNIVERSITAS SUMATERA UTARA**

**MEDAN**

**2024**

**PERSETUJUAN**

Judul : DETEKSI PENYAKIT DIABETES MENGGUNAKAN  
ALGORITMA *DEEP NEURAL NETWORK* DAN  
ALGORITMA *SUPPORT VECTOR MACHINE*  
BERBASIS *WEBSITE*

Kategori : SKRIPSI

Nama : KEVIN WIJAYA

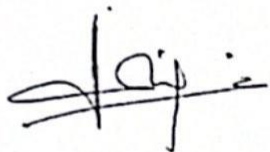
Nomor Induk Mahasiswa : 201401144

Program Studi : SARJANA (S1) ILMU KOMPUTER

Fakultas : ILMU KOMPUTER DAN TEKNOLOGI INFORMASI  
USU

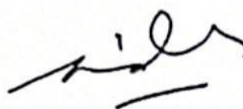
Medan, 24 Juni 2024

Komisi Pembimbing:  
Pembimbing II



Dr. Ir. Elviawaty Muisa Zamzami  
S.T., M.T., MM., IPU  
NIP 197007162005012002

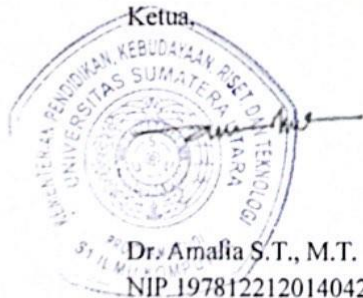
Pembimbing I



Anandhini Medianty Nababan S.KOM.,  
M.T.  
NIP 199304132021022001

Diketahui/disetujui oleh  
Program Studi S1 Ilmu Komputer

Ketua,



Dr. Amalia S.T., M.T.  
NIP 197812212014042001

**Universitas Sumatera Utara**

**PERNYATAAN**

DETEKSI PENYAKIT DIABETES MENGGUNAKAN ALGORITMA *DEEP NEURAL NETWORK* DAN ALGORITMA *SUPPORT VECTOR MACHINE* BERBASIS *WEBSITE*

**SKRIPSI**

Saya mengakui bahwa skripsi ini adalah hasil karya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing telah disebutkan sumbernya.

Medan, 24 Juni 2024



Kevin Wijaya

NIM. 201401144

## PENGHARGAAN

Puji syukur ke hadirat Tuhan Yang Maha Esa, karena berkat, rahmat, dan pertolongan- Nya, sehingga penulis dapat menyelesaikan penyusunan skripsi ini sebagai syarat untuk memperoleh gelar Sarjana Komputer, pada Program Studi S-1 Ilmu Komputer, Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Sumatera Utara.

Penulis ingin menyampaikan rasa hormat dan terima kasih yang sebesar-besarnya kepada pihak yang telah membantu penulis dalam bentuk doa, bimbingan, semangat, dan ilmu selama pembuatan dan penyelesaian skripsi ini.

1. Orang tua penulis yang istimewa, Ayah dan Ibu yang telah memberikan dukungan, semangat dan doa kepada penulis.
2. Bapak Dr. Muryanto Amin, S.Sos., M.Si. selaku Rektor Universitas Sumatera Utara.
3. Ibu Dr. Maya Silvi Lydia, B.Sc., M.Sc. selaku Dekan Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.
4. Bapak Dr. Mohammad Andri Budiman, S.T., M.Comp.Sc., M.E.M. selaku Wakil Dekan 1 Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara yang telah memberikan banyak motivasi dan semangat.
5. Ibu Dr. Amalia, S.T., M.T. selaku Kepala Program Studi S-1 Ilmu Komputer, Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Sumatera Utara.
6. Ibu Sri Melvani Hardi, S.Kom., M.Kom. selaku Sekretaris Program Studi S-1 Ilmu Komputer Universitas Sumatera Utara.
7. Pak Heriyance , selaku Dosen Pembimbing Akademik, yang telah memberikan arahan, nasihat, dan dukungan kepada penulis sebagai mahasiswa dalam menempuh berbagai tahapan akademik.
8. Ibu Anandhini Medianty Nababan S.KOM., M.T. selaku dosen pembimbing I yang telah memberikan bimbingan, kritik, saran, serta dukungan semangat kepada penulis dalam menyelesaikan skripsi ini.
9. Ibu Dr. Ir. Elviawaty Muisa Zamzami S.T., M.T., MM., IPU selaku dosen

pembimbing II yang telah memberikan bimbingan, arahan, dukungan serta masukan kepada penulis.

10. Seluruh dosen, staf pengajar, dan pegawai di Program Studi S-1 Ilmu Komputer, Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Sumatera Utara yang telah membantu penulis dalam proses penyusunan skripsi.
11. Teman-teman mahasiswa/i program studi Ilmu Komputer.
12. Dan semua pihak yang terlibat, baik secara langsung maupun tidak langsung, yang tidak dapat disebutkan satu per satu.

Semoga Tuhan Yang Maha Esa memberikan berkah kepada semua pihak yang telah mendukung dalam pengerjaan tugas akhir ini. Semoga tugas akhir ini bermanfaat bagi pribadi, keluarga, masyarakat, dan negara, serta kemajuan dunia pendidikan.

Medan, 24 Juni 2023

Penulis



Kevin Wijaya

# DETEKSI PENYAKIT DIABETES MENGGUNAKAN ALGORITMA DEEP NEURAL NETWORK DAN ALGORITMA SUPPORT VECTOR MACHINE BERBASIS WEBSITE

## ABSTRAK

Penyakit diabetes adalah penyakit kronis yang prevalensinya terus meningkat di seluruh dunia. Diagnosis penyakit ini dapat dilakukan melalui pemeriksaan gula darah dan tes Hemoglobin Terглиkasi (HbA1C), serta tes Tes Penapisan Komplikasi. Beberapa penelitian sebelumnya telah berhasil membuat sistem deteksi penyakit diabetes menggunakan algoritma *Random Forest*, *Naive Bayes*, dan sebagainya, tetapi penelitian ini menggunakan algoritma *Deep Neural Network* (DNN) dan algoritma *Support Vector Machine* (SVM) dengan menggunakan *dataset* publik yang tersedia pada Kaggle. Pendekatan ini melibatkan pra-pemrosesan data, teknik *oversampling* data, pembagian data menjadi *training-validation-testing* set, normalisasi data, dan penggunaan algoritma *Deep Neural Network* (DNN) dan algoritma *Support Vector Machine* (SVM) dalam pengembangan model *machine learning*. Arsitektur DNN dan SVM yang digunakan terdiri dari lapisan *input*, diikuti oleh 3 *dense layer* dengan masing-masing 32, 16, dan 8 neuron, serta menggunakan fungsi aktivasi ReLU (*Rectified Linear Units*). Terdapat juga 3 lapisan *dropout* dengan tingkat 0.4 di antara setiap *dense layer*, dan lapisan *output* dengan fungsi aktivasi *sigmoid*. Model ini dikompilasi dengan *optimizer* Adam dengan kecepatan pembelajaran 0.001, menggunakan fungsi *loss squared hinge*, dan metrik akurasi. Pelatihan model dilakukan untuk masing-masing *dataset* asli, *dataset* dengan algoritma DNN, algoritma SVM, dan algoritma DNN-SVM. Hasilnya menunjukkan akurasi terbaik pada *dataset* yang diterapkan algoritma DNN-SVM dengan akurasi pelatihan sebesar 97,12% dan *loss* pelatihan sebesar 0,4279, dengan akurasi validasi sebesar 97,12% dan *loss* validasi 0,4256, serta akurasi pengujian sebesar 97,12% dan *loss* pengujian sebesar 0,4256. Model ini kemudian diintegrasikan ke dalam *website* untuk memudahkan pengguna mendeteksi apakah mereka menderita penyakit diabetes berdasarkan data yang dimasukkan.

**Kata kunci** : Penyakit Diabetes, *Deep Neural Network*, *Support Vector Machine*, Website, *Machine Learning*

## DETECTION OF DIABETES USING WEBSITE-BASED DEEP NEURAL NETWORK AND SUPPORT VECTOR MACHINE ALGORITHMS

### **ABSTRACT**

*Diabetes disease is a chronic disease whose prevalence continues to increase throughout the world. Diagnosis of this disease can be done through blood sugar checks and Glycated Hemoglobin (HbA1C) tests, as well as Complication Screening Tests. Several previous studies have succeeded in creating a diabetes detection system using algorithms Random Forest, Naive Bayes, and so on, but this research uses algorithms Deep Neural Network (DNN) and algorithms Support Vector Machine (SVM) by using dataset publicly available on Kaggle. This approach involves data pre-processing, techniques oversampling data, division of data into training-validation-testing sets, data normalization, and use of algorithms Deep Neural Network (DNN) and algorithms Support Vector Machine (SVM) in model development machine learning. The DNN and SVM architecture used consists of layers input, followed by 3 dense layer with 32, 16, and 8 neurons respectively, and using the ReLU activation function (Rectified Linear Units). There are also 3 layers dropout with a rate of 0.4 between each dense layer, and layers output with activation function sigmoid. This model is compiled with optimizer Adam with a learning rate of 0.001, using functions loss squared hinge, and accuracy metrics. Model training is performed for each dataset original, dataset with DNN algorithm, SVM algorithm, and DNN-SVM algorithm. The results show the best accuracy in dataset The DNN-SVM algorithm was applied with a training accuracy of 97,12% and loss training was 0,4279, with validation accuracy of 97,12% and loss validation of 0,4256, and testing accuracy of 97,12% and loss testing of 0,4256. This model is then integrated into website to make it easier for users to detect whether they suffer from diabetes based on the data entered.*

**Keywords :** *Diabetes Disease, Deep Neural Network, Support Vector Machine, Website, Machine Learning*



## DAFTAR ISI

<b>PERSETUJUAN .....</b>	<b>iii</b>
<b>PERNYATAAN.....</b>	<b>iv</b>
<b>PENGHARGAAN.....</b>	<b>v</b>
<b>ABSTRAK .....</b>	<b>vii</b>
<b>ABSTRACT .....</b>	<b>viii</b>
<b>DAFTAR ISI.....</b>	<b>ix</b>
<b>DAFTAR GAMBAR.....</b>	<b>xii</b>
<b>DAFTAR TABEL .....</b>	<b>xiv</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
<b>1.1. Latar Belakang.....</b>	<b>1</b>
<b>1.2. Rumusan Masalah.....</b>	<b>3</b>
<b>1.3. Batasan Masalah .....</b>	<b>3</b>
<b>1.4. Tujuan Penelitian .....</b>	<b>3</b>
<b>1.5. Manfaat Penelitian .....</b>	<b>4</b>
<b>1.6. Penelitian yang Relevan.....</b>	<b>4</b>
<b>1.7. Metodologi Penelitian .....</b>	<b>5</b>
<b>1.8. Sistematika Penulisan .....</b>	<b>7</b>
<b>BAB II LANDASAN TEORI .....</b>	<b>9</b>
<b>2.1 Penyakit Diabetes.....</b>	<b>9</b>
<b>2.2 <i>Deep Neural Network</i>.....</b>	<b>10</b>
2.2.1 <i>Dropout Layer</i> .....	12
2.2.2 <i>Optimizer</i> .....	13
2.2.3 <i>Loss Function</i> .....	13
2.2.4 <i>Epoch</i> .....	13
<b>2.3 <i>Support Vector Machine</i> .....</b>	<b>14</b>

2.4	Python .....	15
2.5	PHP.....	15
2.6	Website .....	16
<b>BAB III ANALISIS DAN PERANCANGAN SISTEM .....</b>		<b>17</b>
3.1.	<b>Gambaran Umum Sistem .....</b>	<b>17</b>
3.2.	<b>Analisis Sistem.....</b>	<b>20</b>
3.2.1.	Analisis Masalah .....	20
3.2.2.	Analisis Kebutuhan .....	20
3.3.	<b>Flowchart .....</b>	<b>21</b>
3.3.1.	<i>Flowchart Deep Neural Network .....</i>	<i>22</i>
3.3.2.	<i>Flowchart Support Vector Machine .....</i>	<i>23</i>
3.3.3.	<i>Flowchart Deep Neural Network + Support Vector Machine .....</i>	<i>24</i>
3.4.	<b>Pengambilan Dataset.....</b>	<b>25</b>
3.5.	<b>Pra-pemrosesan Data.....</b>	<b>25</b>
3.5.1.	<i>Label Encoding .....</i>	<i>25</i>
3.5.2.	<i>Pemeriksaan Missing Value Data .....</i>	<i>26</i>
3.5.3.	<i>Data Split Variabel Independen dan Variabel Dependen.....</i>	<i>26</i>
3.5.4.	<i>Train-test Split.....</i>	<i>26</i>
3.5.5.	<i>Normalisasi Data .....</i>	<i>27</i>
3.6.	<b>Pengembangan Model.....</b>	<b>27</b>
3.7.	<b>Pelatihan Model.....</b>	<b>27</b>
3.8.	<b>Evaluasi Model .....</b>	<b>28</b>
3.9.	<b>Pemodelan Sistem .....</b>	<b>28</b>
3.9.1.	<i>Use Case Diagram .....</i>	<i>29</i>
3.9.2.	<i>Activity Diagram .....</i>	<i>30</i>
3.9.3.	Halaman Beranda .....	31
3.9.4.	Halaman Prediksi Diabetes .....	31
3.9.5.	Halaman Hasil Prediksi .....	32
<b>BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM .....</b>		<b>33</b>
4.1.	<b>Implementasi Sistem .....</b>	<b>33</b>
4.1.1.	Perangkat Keras.....	33
4.1.2.	Perangkat Lunak.....	33

<b>4.2. Implementasi Tahap Pengambilan Dataset .....</b>	<b>34</b>
<b>4.3. Implementasi Tahap Pra-pemrosesan Data .....</b>	<b>36</b>
4.3.1. <i>Label Encoding .....</i>	36
4.3.2. <i>Pemeriksaan Missing Value .....</i>	36
4.3.3. <i>Data Split Variabel Independen dan Variabel Dependen.....</i>	37
4.3.4. <i>Train-test Split.....</i>	37
4.3.5. <i>Normalisasi Data .....</i>	38
<b>4.4. Implementasi Pembangunan Model.....</b>	<b>38</b>
4.4.1. <i>Model Support Vector Machine .....</i>	38
4.4.2. <i>Model Deep Neural Network .....</i>	38
4.4.3. <i>Model Deep Neural Network + Support Vector Machine .....</i>	39
<b>4.5. Implementasi Pelatihan Model .....</b>	<b>40</b>
4.5.1. <i>Pelatihan Model Support Vector Machine.....</i>	40
4.5.2. <i>Pelatihan Model Deep Neural Network.....</i>	41
4.5.3. <i>Pelatihan Model Deep Neural Network + Support Vector Machine .....</i>	41
<b>4.6. Implementasi Evaluasi Model.....</b>	<b>42</b>
4.6.1. <i>Evaluasi Model Support Vector Machine.....</i>	42
4.6.2. <i>Evaluasi Model Deep Neural Network.....</i>	42
4.6.3. <i>Evaluasi Model Deep Neural Network + Support Vector Machine .....</i>	43
<b>4.7. Implementasi Model ke Sistem Website .....</b>	<b>44</b>
<b>BAB V PENUTUP.....</b>	<b>48</b>
<b>5.1. Kesimpulan .....</b>	<b>48</b>
<b>5.2. Saran .....</b>	<b>49</b>
<b>DAFTAR PUSTAKA.....</b>	<b>50</b>

## DAFTAR GAMBAR

<b>Gambar 2.1.</b> Arsitektur <i>Deep Neural Network</i> .....	11
<b>Gambar 2.2.</b> Fungsi DNN .....	11
<b>Gambar 2.3.</b> Ilustrasi <i>Dropout Layer</i> .....	12
<b>Gambar 2.4.</b> Fungsi SVM.....	15
<b>Gambar 3.1.</b> Flowchart Rancangan Umum Sistem .....	18
<b>Gambar 3.2.</b> Diagram Umum Sistem .....	19
<b>Gambar 3.3.</b> <i>Flowchart Deep Neural Network</i> .....	22
<b>Gambar 3.4.</b> <i>Flowchart Support Vector Machine</i> .....	23
<b>Gambar 3.5.</b> <i>Flowchart Deep Neural Network + Support Vector Machine</i> .....	24
<b>Gambar 3.6.</b> <i>Use Case Diagram</i> Sistem Pendeteksi Penyakit Diabetes .....	29
<b>Gambar 3.7.</b> <i>Activity Diagram</i> Sistem Pendeteksi Penyakit Diabetes .....	30
<b>Gambar 3.8.</b> Rancangan Halaman Beranda.....	31
<b>Gambar 3.9.</b> Rancangan Halaman Prediksi Diabetes.....	32
<b>Gambar 3.10.</b> Rancangan Halaman Hasil Prediksi .....	32
<b>Gambar 4.1.</b> Hasil Pemeriksaan <i>Missing Value</i> .....	37
<b>Gambar 4.2.</b> Rasion <i>Train-test Split</i> .....	37
<b>Gambar 4.3.</b> Penerapan Model <i>Support Vector Machine</i> (SVM) .....	38
<b>Gambar 4.4.</b> Arsitektur Model <i>Deep Neural Network</i> (DNN) .....	39
<b>Gambar 4.5.</b> Arsitektur Model (DNN+SVM) .....	40
<b>Gambar 4.6.</b> Pelatihan Model <i>Support Vector Machine</i> (SVM) .....	40
<b>Gambar 4.7.</b> Pelatihan Model <i>Deep Neural Network</i> (DNN) .....	41
<b>Gambar 4.8.</b> Pelatihan Model (DNN+SVM).....	41
<b>Gambar 4.9.</b> Evaluasi Model <i>Support Vector Machine</i> (SVM) .....	42
<b>Gambar 4.10.</b> Hasil Pelatihan Model <i>Deep Neural Network</i> (DNN) .....	43
<b>Gambar 4.11.</b> Evaluasi Model <i>Deep Neural Network</i> (DNN).....	43
<b>Gambar 4.12.</b> Hasil Pelatihan Model (DNN+SVM) .....	44
<b>Gambar 4.13.</b> Evaluasi Model (DNN+SVM).....	44

<b>Gambar 4.14.</b> Proses Menyimpan dan Ekspor Model.....	45
<b>Gambar 4.15.</b> Halaman Beranda Website .....	45
<b>Gambar 4.16.</b> Tombol Prediksi Pada Halaman Beranda <i>Website</i> .....	46
<b>Gambar 4.17.</b> Halaman Prediksi Diabetes <i>Website</i> .....	46
<b>Gambar 4.18.</b> Halaman Hasil Prediksi Diabetes .....	47

## DAFTAR TABEL

<b>Tabel 3.1.</b> Contoh Teknik <i>Label Encoding</i> .....	26
<b>Tabel 4.1.</b> Hasil Pengambilan <i>Dataset</i> (1).....	34
<b>Tabel 4.2.</b> Hasil Pengambilan <i>Dataset</i> (2).....	35
<b>Tabel 4.3.</b> Hasil Deskripsi Isi <i>Dataset</i> .....	35
<b>Tabel 4.4.</b> Hasil Teknik <i>Label Encoding</i> pada Fitur Data Gender .....	36

## **BAB 1**

### **PENDAHULUAN**

#### **1.1. Latar Belakang**

Kesehatan tubuh adalah hal paling krusial dalam hidup seseorang. Ketika kesehatan tubuh terganggu, aktivitas sehari-hari akan terganggu pula. Ada banyak kondisi kesehatan yang dapat mengancam manusia dan bahkan berpotensi menjadi penyebab kematian. Salah satu contohnya adalah Diabetes Melitus.

Diabetes merupakan suatu penyakit kronis yang terkait dengan peningkatan kadar glukosa darah, yang seiring waktu dapat menyebabkan kerusakan serius pada organ tubuh seperti jantung, pembuluh darah, mata, ginjal, dan saraf. Diabetes tipe 2 adalah yang paling umum, terjadi pada orang dewasa, di mana tubuh menjadi resisten terhadap insulin atau menghasilkan insulin dalam jumlah yang tidak mencukupi. Prevalensi diabetes tipe 2 telah meningkat secara signifikan dalam tiga dekade terakhir, bahkan di negara-negara dengan berbagai tingkat pendapatan. Diabetes tipe 1, sebelumnya dikenal sebagai diabetes remaja atau insulin-dependen, adalah kondisi kronis di mana pankreas menghasilkan sedikit atau tidak sama sekali insulin. Bagi penderita diabetes, akses terhadap pengobatan yang terjangkau, termasuk insulin, sangat penting untuk mempertahankan kesehatan dan kelangsungan hidup mereka. (WHO, 2023).

Menurut data dari Infodatin Diabetes Melitus 2020 yang diterbitkan oleh Pusat Data dan Informasi Kementerian Kesehatan Republik Indonesia, International Diabetes Federation (IDF) juga mengungkapkan bahwa sekitar 463 juta orang berusia 20-79 tahun menderita diabetes pada tahun 2019. Proyeksi menunjukkan angka tersebut diperkirakan akan terus meningkat menjadi sekitar 578 juta pada tahun 2030 dan mencapai 700 juta pada tahun 2045. Indonesia termasuk dalam sepuluh besar negara dengan jumlah penduduk yang menderita diabetes, dengan sekitar 10,7 juta orang, menempatkannya pada peringkat ke-7. Prediksi juga menyatakan bahwa jumlah penderita diabetes di Indonesia diperkirakan akan terus meningkat, mencapai sekitar 13,7 juta pada tahun 2030, dan 16,6 juta pada tahun 2045. (Kementrian Kesehatan Republik Indonesia, 2020).

Banyak orang yang baru mengetahui bahwa mereka menderita diabetes setelah mengalami komplikasi. Namun, situasi tersebut dapat dihindari dengan melakukan diagnosis lebih awal, sehingga pengobatan untuk diabetes dapat dimulai dengan cepat dan risiko terjadinya komplikasi penyakit lain dapat diminimalkan. Penerapan teknik *data mining* dapat menjadi pendukung dalam upaya ini dengan menghasilkan informasi yang berharga dari data mengenai diabetes. (Isbandiyo, 2016). Dalam mengembangkan teknologi untuk menganalisis data diabetes menjadi prediksi, teknik *data mining* menjadi krusial. *Data mining* merupakan proses pengolahan, eksplorasi, dan pemodelan data besar untuk menghasilkan pemahaman. Salah satu teknik *data mining* yang relevan dalam prediksi adalah klasifikasi, yang bertugas memprediksi keluaran variabel/kelas yang nilainya bersifat kategorial atau polinomial. Dengan menggunakan teknik ini, sebuah sistem dapat dibangun untuk memprediksi penyakit tersebut. (Hovi Sohibul Wafa et al, 2022).

Penelitian ini menyarankan penggunaan algoritma *Deep Neural Network* (DNN), yang merupakan jenis jaringan saraf tiruan yang terinspirasi dari struktur dan fungsi otak manusia. Keunggulan utama DNN terletak pada kemampuannya dalam mengenali pola-pola kompleks dan menganalisis data secara mendalam. Selain itu, penelitian ini juga merekomendasikan penerapan algoritma *Support Vector Machine* (SVM), yang merupakan salah satu metode klasifikasi yang dikenal memiliki tingkat keberhasilan tinggi dalam melakukan prediksi dan pengklasifikasian potensi pada data. (Hovi Sohibul Wafa, Asep Id Hadiana, & Fajri Rakhmat Umbara, 2022).

Penelitian sebelumnya telah menunjukkan bahwa Algoritma DNN memiliki kemampuan untuk mengolah dan menganalisis data medis yang kompleks, sementara SVM terbukti efektif dalam mengklasifikasikan data pada konteks kesehatan. Gabungan kedua algoritma ini diharapkan dapat meningkatkan akurasi dalam mendeteksi penyakit diabetes, memberikan dukungan untuk pengambilan keputusan klinis yang lebih tepat, dan mendorong respons medis yang lebih dini.

Pengembangan sebuah aplikasi sistem pendeteksi penyakit diabetes berbasis website memberikan dimensi aksesibilitas yang sangat penting dalam upaya deteksi dan pencegahan. Dengan website sebagai platform layanan kesehatan, informasi dan hasil deteksi dapat disampaikan dengan cepat, memungkinkan individu untuk lebih aktif dalam memantau kesehatan mereka. Pendekatan ini dapat menjadi solusi yang



efektif, terutama di tengah masyarakat yang semakin terhubung secara digital.

Dengan penelitian sebelumnya sebagai landasan, pengembangan aplikasi sistem pendeteksi penyakit diabetes berbasis website menggunakan algoritma DNN dan SVM diharapkan dapat memberikan solusi yang lebih efisien, praktis, dan sesuai dengan kebutuhan masyarakat modern. Langkah ini tidak hanya ditujukan untuk meningkatkan kualitas pelayanan kesehatan, tetapi juga untuk berkontribusi pada upaya pencegahan dan pengelolaan penyakit diabetes secara global.

## 1.2. Rumusan Masalah

Berdasarkan uraian latar belakang di atas, rumusan masalah yang akan menjadi fokus utama dalam penelitian ini adalah melihat hasil akurasi deteksi diabetes dengan menggunakan algoritma DNN, dan mencoba melihat kontribusi algoritma SVM dalam mengoptimasi hasil deteksi, dan seberapa efektif integrasi kedua algoritma tersebut dalam meningkatkan performa keseluruhan sistem. Dan mencakup evaluasi aspek penggunaan aplikasi berbasis *website*, termasuk seberapa mudahnya aksesibilitas informasi hasil deteksi bagi pengguna serta tingkat kepuasan pengguna terhadap penggunaan sistem ini

## 1.3. Batasan Masalah

Adapun batasan masalah yang akan dikaji dalam penelitian ini adalah sebagai berikut:

1. Pendeteksi dari Penyakit Diabetes berbasis web dengan menggunakan Bahasa Pemrograman Python dan HTML untuk pembuatan sistemnya.
2. Metode yang digunakan untuk memprediksi penyakit Diabetes menggunakan algoritma *Deep Neural Network* dan algoritma *Support Vector Machine*.
3. Hasil deteksi penyakit diabetes berupa klasifikasi hanya sebatas pada klasifikasi dua kelas (*binary classification*), yaitu terkena diabetes dan tidak terkena diabetes.
4. Data yang digunakan merupakan dataset publik yang berasal dari <https://www.kaggle.com/>.

#### 1.4. Tujuan Penelitian

Adapun tujuan dari penelitian yang dilakukan pada skripsi ini adalah mengembangkan sistem deteksi penyakit diabetes dengan memanfaatkan algoritma *Deep Neural Network* (DNN) dan *Support Vector Machine* (SVM). Penelitian akan difokuskan pada optimalisasi parameter dan struktur model DNN dan SVM, serta integrasi kedua algoritma ke dalam suatu website dengan antarmuka pengguna yang user-friendly. Selain itu, penelitian ini bertujuan untuk melakukan validasi dan evaluasi performa model, termasuk pengujian menyeluruh dan perbandingan kinerja antara DNN dan SVM dalam konteks deteksi diabetes.

#### 1.5. Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah untuk memberikan kontribusi yang berarti dalam pengembangan teknologi deteksi penyakit terhadap gejala-gejala yang dimiliki oleh pengguna, serta membantu meningkatkan kesadaran masyarakat secara luas tentang diabetes dan pentingnya pencegahan penyakit diabetes. Aplikasi berbasis website ini juga dapat diakses dengan mudah oleh pengguna dari berbagai tempat dan perangkat atau device.

#### 1.6. Penelitian yang Relevan

Terdapat beberapa penelitian yang relevan terhadap penelitian yang akan dilakukan adalah sebagai berikut:

1. Berdasarkan penelitian yang dilakukan (Fitriyani, 2021), adalah Dalam penelitian ini, algoritma *machine learning* yang digunakan adalah *Naïve Bayes*. *Naïve Bayes* dipilih karena algoritma ini sederhana dan mudah digunakan. Hasil dari eksperimen menunjukkan bahwa model *Naïve Bayes* yang menggunakan *Greedy Forward Selection* (NB+GFS) memiliki akurasi terbaik, yaitu sebesar 91.73%. Sementara itu, model *Naïve Bayes* tanpa menggunakan seleksi fitur mendapatkan akurasi sebesar 87.69%. Dari hasil eksperimen ini, dapat disimpulkan bahwa algoritma *Naïve Bayes* dan *Greedy Forward Selection* mampu memprediksi diabetes dengan sangat baik.
2. Berdasarkan Penelitian yang dilakukan (Apriliyah et al., 2021), adalah kontribusi utama penelitian tersebut diadakan untuk mengetahui algoritma

terbaik untuk prediksi resiko diabetes. Percobaan dilakukan pada dataset Diabetes Hospital in Sylhet. Hasil percobaan menentukan kecukupan sistem yang dirancang dengan akurasi yang dicapai sebesar 97,88%. Algoritma *Random Forest* telah bekerja dengan akurasi terbaik.

3. Berdasarkan Penelitian yang dilakukan (Maulidah et al., 2021), adalah Berdasarkan hasil penelitian, ditemukan bahwa algoritma *Support Vector Machine* (SVM) memiliki tingkat akurasi yang lebih tinggi dibandingkan dengan algoritma *Naive Bayes*. Akurasi model menggunakan algoritma SVM adalah 78,04%, sedangkan algoritma *Naive Bayes* memiliki akurasi sebesar 76,98%. Terdapat selisih akurasi sebesar 1,06%. Dengan demikian, dapat disimpulkan bahwa penerapan algoritma *Support Vector Machine* menghasilkan tingkat akurasi diagnosis penyakit diabetes yang lebih baik dibandingkan dengan penggunaan algoritma *Naive Bayes*.
4. Berdasarkan Penelitian yang dilakukan (Sunanto & Falah, 2022), Dari hasil penelitian yang dilakukan terhadap sistem prediksi penyakit diabetes menggunakan metode algoritma *Decision Tree* C4.5, dapat disimpulkan bahwa penggunaan algoritma *Decision Tree* C4.5 merupakan metode yang efektif untuk memprediksi penyakit diabetes secara awal. Hasil prediksi penyakit menggunakan algoritma *Decision Tree* C4.5 menunjukkan performa yang baik, seperti terlihat dari perhitungan *confusion matrix* yang menghasilkan akurasi sebesar 95,51% dan tingkat kesalahan klasifikasi sebesar 4,49%.

### 1.7. Metodologi Penelitian

Adapun metodologi penelitian yang akan dilakukan dalam penelitian ini adalah sebagai berikut:

#### 1. Studi Pustaka

Tahapan awal yang akan dilaksanakan pada penelitian ini adalah mengumpulkan dan mempelajari berbagai informasi terbaru dan relevan mengenai penerapan algoritma *Deep Neural Network* (DNN) dan algoritma *Support Vector Machine* (SVM) dalam mendeteksi penyakit diabetes. Proses studi literatur ini mencakup pemahaman mendalam tentang dasar konsep dan prinsip keduanya, serta pemahaman terkini dalam pengaplikasian kedua

algoritma tersebut dalam ranah medis dan aplikasi kesehatan berbasis *website*.

## 2. Analisis dan Perancangan Sistem

Pada tahapan analisis dan perancangan sistem dilakukan analisis kebutuhan untuk mengidentifikasi kebutuhan fungsional dan non-fungsional dan memodelkan permasalahan dengan memanfaatkan diagram alir (*flowchart*). Selanjutnya dilakukan perancangan antarmuka pengguna berbasis *website*, dan implementasi integrasi algoritma DNN dan SVM ke dalam sistem.

## 3. Pencarian *Dataset*

Pencarian *dataset* dilakukan dengan mencari *dataset* terkait penyakit diabetes yang tersedia secara publik. Tujuan dari pencarian *dataset* ini adalah untuk keperluan data yang dilatih kepada model *machine learning* yang dibangun.

## 4. Pra-pemrosesan Data

*Dataset* yang telah didapatkan dari penyedia *dataset* publik perlu dilakukan pembersihan data melalui tahap pra-pemrosesan data. Hal ini dilakukan untuk mencegah *dataset* yang mungkin berantakan atau kotor, serta memastikan *dataset* yang digunakan sudah bersih dan benar-benar sudah siap untuk melatih model *machine learning*.

## 5. Membangun Model *Machine Learning*

Membuat arsitektur model *machine learning* dengan bahasa Python untuk membangun *Deep Neural Network* (DNN) dan *Support Vector Machine* (SVM) serta melatih model *machine learning* tersebut menggunakan data yang telah dilakukan pra-pemrosesan data sebelumnya.

## 6. Evaluasi dan Pengujian Model *Machine Learning*

Setelah model *machine learning* selesai dibangun, dilakukan tahap evaluasi model *machine learning* dengan metrik yang telah ditentukan untuk melihat performa *machine learning* tersebut dalam mempelajari pola data. Kemudian pengujian model *machine learning* dilakukan dengan cara memasukkan data baru untuk mengetahui apakah *machine learning* tersebut berhasil melakukan pendeteksian secara akurat.

## 7. Implementasi Sistem

Pada tahap ini, analisis, perancangan sistem, serta model *machine learning* yang telah mendapatkan hasil evaluasi yang cukup ideal, diimplementasikan ke

dalam sebuah sistem dengan menggunakan bahasa pemrograman Python sesuai dengandiagram alir yang telah dirancang.

#### 8. Pengujian Sistem

Pengujian Sistem Pada tahap ini, dilakukan uji coba sistem dan semua fitur yang telah dibangun untuk mengetahui kinerja dan hasil sistem tersebut telah sesuai dalam melakukan pendeteksian penyakit diabetes berdasarkan data yang dimasukkan oleh pengguna sistem.

#### 9. Dokumentasi Sistem

Pada tahap ini, penelitian ini didokumentasikan ke dalam bentuk penyusunan laporan akhir, mulai dari tahap analisis, hingga ke tahap pengujian sistem dalam format penulisan penelitian skripsi.

### 1.8. Sistematika Penulisan

Berikut merupakan sistematika penulisan skripsi ini, yang terbagi ke dalam beberapabagian utama, sebagaimana dijelaskan berikut ini.

#### **BAB I      PENDAHULUAN**

Bab ini membahas latar belakang dari penelitian yang berjudul “Deteksi Penyakit Diabetes Menggunakan Algoritma *Deep Neural Network* (DNN) dan Algoritma *Support Vector Machine* (SVM) Berbasis *Website*”. Bab ini mencakup rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, penelitian yang relevan, metodologi penelitian, dan sistematika penulisan.

#### **BAB II      LANDASAN TEORI**

Bab ini membahas topik-topik terkait penyakit *celiac*, teori metode *machine learning*, *Deep Neural Network* (DNN), *Support Vector Machine* (SVM), serta penerapan *machine learning* pada aplikasi berbasis website .

#### **BAB III    ANALISIS DAN PERANCANGAN**

Bab ini membahas tentang analisis masalah dan perancangan model sistem yang dibangun untuk mengatasi masalah tersebut.

#### **BAB IV    IMPLEMENTASI DAN PENGUJIAN**

Bab ini membahas pembuatan sistem sebagai hasil implementasi

dari analisis dan perancangan sebelumnya. Bab ini juga membahas pengujian sistem serta hasil analisis pengujian yang dilakukan.

## **BAB V KESIMPULAN DAN SARAN**

Bab terakhir ini membahas kesimpulan akhir dari penelitian yang telah dilakukan, serta saran-saran hasil penelitian yang telah diperoleh sehingga dapat membantu pengembangan penelitian lebih lanjut di masa yang akan datang.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Penyakit Diabetes**

Penyakit Diabetes adalah suatu kondisi metabolik yang bersifat kronis, ditandai dengan peningkatan kadar glukosa (gula darah) yang melebihi batas normal, yang disebabkan oleh gangguan dalam sekresi insulin atau gangguan efek biologisnya. Diabetes memiliki potensi untuk menimbulkan komplikasi di berbagai bagian tubuh dan secara keseluruhan dapat meningkatkan risiko kematian dini. Kemungkinan komplikasi meliputi gagal ginjal, amputasi kaki, kehilangan penglihatan, dan kerusakan saraf. Orang dewasa yang menderita diabetes juga memiliki risiko dua hingga tiga kali lipat lebih tinggi untuk mengalami serangan jantung dan stroke. Pada kehamilan, diabetes yang tidak terkontrol dapat meningkatkan risiko kematian pada janin serta komplikasi lainnya. Jumlah penderita diabetes terus meningkat dari tahun ke tahun, baik dari segi jumlah kasus maupun prevalensinya. (Erlin et al., 2022).

Menurut IDF, diperkirakan bahwa sekitar 483 juta orang berusia 20-79 tahun di seluruh dunia menderita diabetes pada tahun 2019, yang setara dengan prevalensi sekitar 9.3% dari total penduduk dalam kelompok usia tersebut. Berdasarkan perbedaan jenis kelamin, prevalensi diabetes pada tahun 2019 adalah sekitar 9% pada perempuan dan 9.65% pada laki-laki. Diperkirakan bahwa prevalensi diabetes akan meningkat seiring bertambahnya usia penduduk, mencapai sekitar 19.9% atau sekitar 112 juta orang pada kelompok usia 65-79 tahun. Proyeksi ini menunjukkan bahwa jumlah penderita diabetes diperkirakan akan terus meningkat, mencapai sekitar 578 juta pada tahun 2030 dan 700 juta pada tahun 2045. (Hovi, Asep, & Fajri, 2022).

Dengan melihat tingginya angka kasus diabetes, langkah awal yang diperlukan adalah melakukan prediksi untuk penanganan dini penyakit tersebut. Prediksi mengenai diabetes dapat dilakukan dengan mengumpulkan berbagai data pasien yang menderita diabetes yang tersimpan dalam basis data, kemudian diproses menggunakan pola tertentu sehingga hasilnya dapat digunakan untuk

diagnosis awal diabetes. (Baiq & Intan, 2021)

## **2.2 Deep Neural Network**

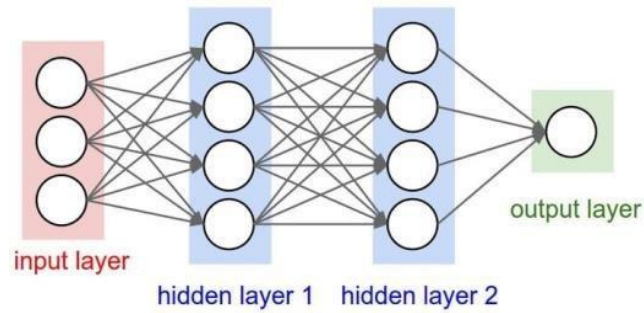
Deep Neural Network adalah jaringan syaraf tiruan yang struktur jaringan terdiri dari beberapa unit syaraf yang saling terhubung. Terdiri dari 3 lapisan yaitu lapisan masukan, lapisan tersembunyi dan lapisan keluaran. Bentuk jaringan syaraf tersebut menentukan respon/keluaran sesuai dengan unit syaraf yang terhubung. Struktur tersebut dapat disebut dengan perceptron. Jika jumlah lapisan tersembunyi lebih dari atau sama dengan dua, maka disebut dengan Multi Layer Perceptron (MLP) atau Deep Learning Neural Network (DNN)(Addo et al., 2018).

Deep Neural Network (DNN) merupakan suatu sistem proses informasi yang memiliki karakteristik performa tertentu dalam jaringan saraf biologis secara mendalam. Dalam penelitian ini jaringan syaraf sebagai suatu objek atau model time series, dengan beberapa definisi:

1. Pemrosesan informasi terjadi pada banyak elemen sederhana yang disebut neuron.
2. Sinyal dialirkan di antara neuron di atas jaringan terhubung.
3. Masing-masing jaringan terhubung memiliki weight yang dikalikan dengan sinyal yang ditransmisi.
4. Masing-masing neuron menggunakan fungsi aktivasi pada net masukan untuk menentukan sinyal output(Errissya, Rudolf, & Yovi, 2020).

Algoritma DNN telah luas digunakan dalam berbagai aplikasi, termasuk klasifikasi, pemrosesan bahasa, dan pemahaman citra. DNN mencoba untuk mereplikasi jaringan saraf yang ada di otak manusia, sehingga sistem tersebut memiliki kemampuan untuk memahami dan membuat keputusan dengan cara yang serupa dengan manusia. Algoritma ini bekerja dengan cara memetakan input ke output melalui serangkaian lapisan.





**Gambar 2.1** Arsitektur *Deep Neural Network*

DNN memiliki 3 lapisan layer , diantaranya :

1. Input layer adalah layer awal yang terdiri dari beberapa neuron, pada penelitian ini sesuai dengan banyaknya parameter pada dataset. Input layer adalah tempat masuknya data kedalam system yang selanjutnya diproses oleh neuron yang ada di layer selanjutnya.
2. Hidden layer adalah layer tersembunyi yang terdiri dari beberapa neuron yang menerima data dari layer sebelumnya, di mana data diproses untuk menghasilkan output neuron melalui activation function.
3. Output layer adalah layer akhir yang terdiri dari neuron untuk menerima data dari layer sebelumnya, yang akan menghasilkan output system. DNN membaca input dan menghitung jumlah input yang dibobotkan dan ditambahkan dengan bias. Perhitungan ini dicontohkan dalam bentuk fungsi 1. sebagai berikut

$$dot_j = \sum_i^3 w_{ji}x_i + b_j$$

$$h_j = \sigma(dot_j) = \max(0, dot_j)$$

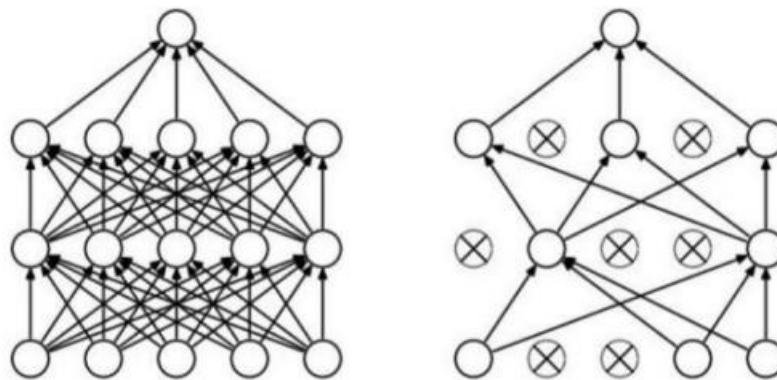
**Gambar 2.2** Fungsi DNN

Dimana  $i$  adalah node pada input layer,  $j$  adalah node pada hidden layer,  $n$  adalah jumlah neuron pada masing-masing lapisan,  $w$  adalah nilai weight pada masing-masing koneksi antar neuron,  $x$  adalah nilai input dari masing-masing neuron atau nilai terusan dari neuron sebelumnya,  $b$  adalah bias pada masing-masing neuron, sedangkan  $h$  adalah output dari node pada hidden layer. Dalam hal ini, input akan diteruskan dengan gambar 2 menuju layer-layer selanjutnya yaitu hidden layer hingga output layer yang selanjutnya hasil prediksi output akan

dibandingkan dengan target sehingga akan menghasilkan nilai error dari suatu output. Nilai error tersebutlah yang digunakan untuk mengukur seberapa akurat performa dari algoritma Deep Neural Network yang dibangun dalam melakukan prediksi terhadap target(Hilman et al., 2023).

### 2.2.1 Dropout Layer

Dropout Layer adalah bagian dari jaringan saraf tiruan yang bertugas untuk secara acak mengabaikan atau "meng-drop" sejumlah unit atau neuron selama proses pelatihan. Tujuan utama dari penggunaan layer dropout adalah untuk mengurangi kompleksitas model dan mencegah overfitting dengan memaksa jaringan saraf untuk belajar representasi yang lebih umum dan tidak tergantung pada keberadaan setiap neuron secara spesifik. (Dian & Chanifah, 2023).



**Gambar 2.3** Sebelum memakai Dropout(kiri) dan setelah memakai Dropout(kanan)

Dropout Layer merupakan metode regularisasi yang diterapkan pada jaringan saraf tiruan. Seperti yang ditunjukkan pada Gambar 3, Dropout Layer bekerja dengan memilih neuron secara acak dan mengabaikannya selama proses pelatihan. Hal ini dapat meningkatkan kecepatan pelatihan karena jumlah neuron yang harus diproses lebih sedikit. Selain itu, Dropout Layer juga bertujuan untuk mengurangi overfitting. Teknik ini dapat diterapkan baik pada lapisan ekstraksi maupun pada lapisan terhubung sepenuhnya dalam jaringan saraf. (Muhammad et al., 2021).

### 2.2.2 *Optimizer*

Optimizer merupakan sebuah algoritma yang bertujuan untuk mengurangi kesalahan atau perbedaan antara output yang dihasilkan oleh jaringan dan target yang diinginkan. Salah satu optimizer yang umum digunakan adalah Adam, yang menggabungkan teknik-teknik dari optimizer RMSProp dan momentum. Adam memiliki keunggulan dalam efisiensi komputasi, penggunaan memori yang hemat, dan dapat diterapkan dalam berbagai bidang pembelajaran mesin untuk mengoptimalkan fungsi-fungsi yang tidak konveks. Optimizer merupakan salah satu parameter penting yang digunakan untuk meminimalkan kesalahan sehingga dapat menghasilkan prediksi yang akurat. (Jihan et al., 2023).

### 2.2.3 *Loss Function*

Loss function adalah sebuah formula matematika yang digunakan untuk menghitung nilai kerugian. Nilai kerugian ini akan digunakan dalam proses backpropagation untuk mengevaluasi parameter seperti bobot dan bias sehingga jaringan saraf menjadi lebih optimal. Loss function dapat dibagi menjadi dua jenis, yaitu untuk klasifikasi dan regresi. Perbedaannya terletak pada jenis output yang diprediksi, di mana loss function untuk klasifikasi digunakan untuk memprediksi nilai kategori, sementara loss function untuk regresi digunakan untuk memprediksi nilai yang bersifat kontinu. Loss function berperan dalam menentukan seberapa baik model dapat memperkirakan target yang diinginkan. Sebuah loss function dianggap baik jika menghasilkan kesalahan yang kecil. (Umami & Noveri, 2023).

### 2.2.4 *Epoch*

Epoch merupakan satu putaran algoritma machine learning yang "belajar" dari kumpulan data pelatihan yang ingin diproses, yang juga dikenal sebagai mini-batch. Dengan menggunakan mini-batch, mesin tidak mempelajari seluruh dataset, tetapi mempelajari sejumlah batch data yang telah ditentukan. Satu epoch menandakan bahwa algoritma machine learning telah mempelajari keseluruhan data pelatihan. Dalam konteks jaringan saraf tiruan, proses pembelajaran yang berulang-ulang bertujuan untuk mencapai konvergensi nilai bobot. Jumlah epoch yang tepat biasanya tidak diketahui secara pasti dan tergantung pada ukuran dataset

yang digunakan. Pada tugas klasifikasi, epoch (iterasi) digunakan untuk melakukan proses perulangan klasifikasi gambar guna mencari nilai loss dan akurasi yang diinginkan. Akurasi yang dihasilkan dapat bervariasi tergantung pada jumlah epoch yang digunakan. Secara umum, penggunaan lebih banyak epoch dapat meningkatkan akurasi yang dicari, tetapi terlalu banyak epoch juga dapat menyebabkan akurasi stabil atau bahkan menurun. Oleh karena itu, penting untuk menggunakan jumlah epoch yang sesuai untuk memastikan akurasi maksimal dalam pembelajaran mesin. (Muhammad, Harianto, & Fathurrahman, 2022).

### **2.3 *Support Vector Machine***

Support Vector Machine (SVM) adalah sistem pembelajaran yang memanfaatkan ruang hipotesis yang terdiri dari fungsi linier dalam fitur yang memiliki dimensi tinggi. SVM dilatih menggunakan algoritma pembelajaran yang berasal dari teori optimasi (Puspitasari, Ratnawati, & Widodo, 2018). Support Vector Machine (SVM) diperkenalkan oleh Vapnik, Boser dan Guyon pada tahun 1992 (Fridayanthie, 2015) sebagai rangkaian dari beberapa konsep-konsep unggulan dalam bidang pattern recognition (Susilowati, Sabariah, & Gozali, 2015).

Akurasi dari model yang dihasilkan melalui proses pemodelan dengan SVM sangat dipengaruhi oleh jenis fungsi kernel dan parameter yang digunakan (Parapat & Furqon, 2018). Berdasarkan karakteristiknya, SVM dibagi menjadi dua metode, yaitu SVM Linier dan SVM Non-Linier. SVM Linier digunakan ketika data dapat dipisahkan secara linier, yang berarti kedua kelas dapat dipisahkan oleh hyperplane dengan margin yang fleksibel. Sedangkan SVM Non-Linier menggunakan teknik kernel trick untuk memproyeksikan data ke dalam ruang dimensi yang lebih tinggi agar dapat dipisahkan secara linier di sana (Rachman & Purnami, 2012).

$$h(x) = \begin{cases} +1, & \text{if } w \cdot x + b \geq 0 \\ -1, & \text{if } w \cdot x + b < 0 \end{cases}$$

**Gambar 2.4.** Fungsi SVM

Jika hasil perhitungan keputusan lebih dari sama dengan 0 maka nilai sign  $h(x)$  adalah +1, maka termasuk kelas positif dan jika hasil perhitungan keputusan kurang dari 0 maka nilai sign  $h(x)$  adalah -1, maka termasuk kelas negatif (Rian, Agung, & Ira, 2020).

## 2.4 Python

Python merupakan bahasa pemrograman yang dikembangkan oleh Guido van Rossum. Bahasa pemrograman ini sangat ramah bagi pemula yang ingin mempelajari pemrograman karena sintaksnya mudah dipahami dan mirip dengan bahasa manusia pada umumnya. Python mendukung beberapa paradigma pemrograman, termasuk pemrograman prosedural, pemrograman berorientasi objek, dan pemrograman fungsional. Selain itu, Python memiliki beragam library yang mendukung pengembangan proyek machine learning, seperti SKLearn, NumPy, dan Pandas (Widiyanto et al., 2022).

Python dianggap sebagai bahasa yang menggabungkan kemampuan, kapabilitas, dan sintaksis kode yang sangat jelas. Selain itu, Python dilengkapi dengan pustaka standar yang luas dan komprehensif, memungkinkannya untuk mendukung berbagai paradigma pemrograman. Selain itu, Python juga dikenal sebagai bahasa pemrograman yang dinamis dengan manajemen memori otomatis. Python digunakan sebagai bahasa skrip yang dapat berjalan di berbagai platform sistem operasi (Putra, Aditya, & Evanita, 2021).

## 2.5 PHP

PHP merupakan singkatan dari Hypertext Preprocessor, yang merupakan bahasa pemrograman sisi server (server-side programming). Peran utama PHP dalam pembuatan website adalah untuk melakukan manipulasi data pada basis data. Dengan menggunakan PHP, data website dapat dimasukkan ke dalam basis data, diedit, dihapus, dan kemudian ditampilkan di website sesuai dengan aturan yang

telah ditetapkan (Ahmat, 2017).

## **2.6 Website**

Website adalah kumpulan halaman informasi yang dapat diakses melalui internet. Setiap individu dari berbagai lokasi dan kapan pun dapat mengaksesnya selama terhubung ke jaringan internet. Secara teknis, sebuah website terdiri dari sejumlah halaman yang tergabung dalam sebuah domain atau subdomain tertentu (Yusuf et al., 2021).

## BAB III

### ANALISIS DAN PERANCANGAN SISTEM

#### 3.1. Gambaran Umum Sistem

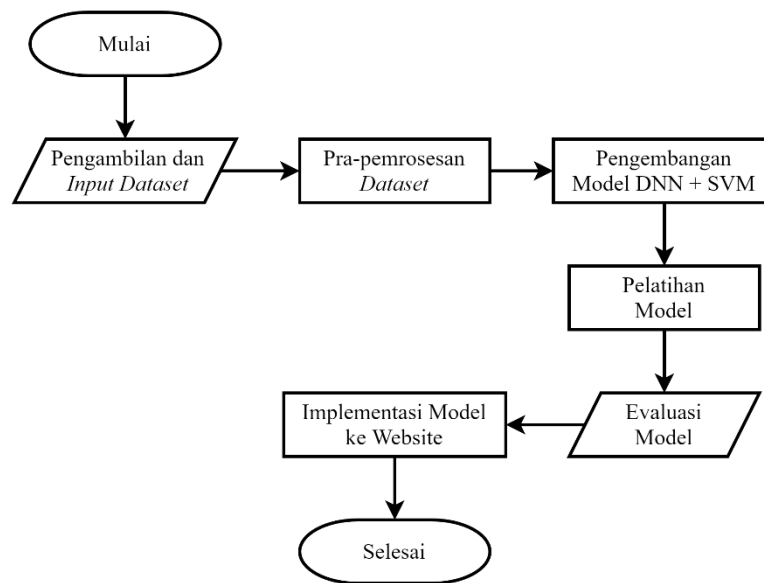
Pada penelitian ini dibangun sebuah sistem untuk mendeteksi seseorang apakah terkena penyakit diabetes atau tidak menggunakan aplikasi berbasis *website*. Perancangan sistem dilakukan menggunakan metode *deep learning* dengan algoritma *Deep Neural Network* dan *Support Vector Machine* (DNN + SVM). Di mana algoritma *Deep Neural Network* (DNN) berperan dalam ekstraksi fitur, mempelajari, dan mengenali pola-pola data yang rumit, kemudian diteruskan ke *Support Vector Machine* (SVM) yang berfungsi sebagai lapisan *output* klasifikasi akhir. Dengan kemampuan *Support Vector Machine* (SVM) untuk menemukan *hyperplane* yang optimal, klasifikasi dapat dilakukan dengan baik dan memberikan akurasi yang lebih tinggi.

Pengembangan sistem diawali dengan pencarian dan pengambilan *dataset* medis pasien yang memiliki penyakit diabetes maupun yang tidak memiliki penyakit diabetes. *Dataset* tersebut diperoleh dari penyedia *dataset* publik, Kaggle, yaitu *Early Classification of Diabetes* (<https://www.kaggle.com/datasets/andrewmvd/early-diabetes-classification>). *Dataset* tersebut kemudian perlu dilakukan beberapa tahap pra-pemrosesan seperti, *label encoding*, *missing value data*, *data splitting*, *train-test split*, dan normalisasi data, sebelum data tersebut di-*input* sebagai data latih model.

Model yang dikembangkan adalah model *deep learning* dengan arsitektur *Deep Neural Network* (DNN) yang memiliki *input layer*, *hidden layer*, serta *output layer* yang dioptimisasi menggunakan algoritma *Support Vector Machine* (SVM). Selanjutnya akan dilakukan pelatihan model tersebut dengan data latih (*train data*) untuk melihat performa model dalam mempelajari pola data yang diberikan. Kemudian, tahap evaluasi model dilakukan untuk menguji model dalam memprediksi data uji (*test data*). Evaluasi model tersebut menggunakan metrik akurasi *training*, akurasi alidasi, *loss function training*, dan *loss function validasi* untuk dapat melihat performa model hasil pelatihan dalam memprediksi data.

Terakhir, model tersebut disimpan menjadi sebuah *file* untuk tahap implementasi model ke *website*. Website akan menerima data *input* dari pengguna berupa data medis yang terkait dengan penyakit diabetes. Data tersebut lalu diproses terlebih dahulu, dan akan masuk ke dalam prediksi model untuk mendapatkan *output* berupa terkena penyakit diabetes atau tidak.

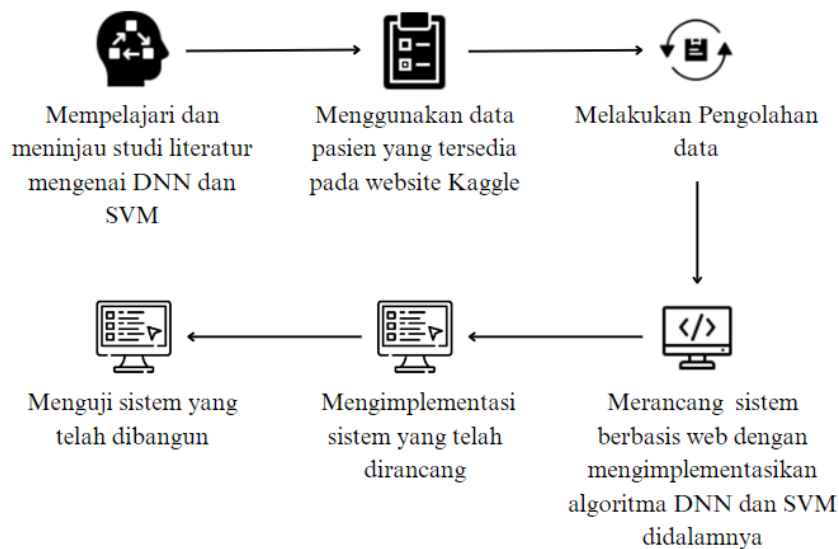
Alur atau urutan metode pengembangan sistem secara umum dapat dilihat pada *flowchart* Gambar 3.1. berikut ini.



**Gambar 3.1.** *Flowchart* Rancangan Umum Sistem

Sedangkan alur dan proses sistem yang dibangun sebagai gambaran diagram umum sistem yang digunakan oleh pengguna pada penelitian ini dapat dilihat pada Gambar 3.2.





**Gambar 3.2.** Diagram Umum Sistem

Secara umum, diagram sistem dapat diuraikan sebagai berikut :

1. Mempelajari dan meninjau studi literatur mengenai *Deep Neural Network* dan *Support Vector Machine* dari berbagai referensi seperti *internet*, jurnal *online*, artikel, dan lain sebagainya.
2. Data pasien yang digunakan pada sistem ini merupakan data publik yang telah tersedia pada website Kaggle
3. Dari data yang telah didapatkan dan dikumpulkan, kemudian data tersebut diolah untuk dilakukan pendeteksian sesuai dengan metode atau algoritma yang digunakan.
4. Setelah melakukan pengolahan data, langkah berikutnya yaitu melakukan perancangan sistem berbasis web dengan mengimplementasikan algoritma *Deep Neural Network* dan *Support Vector Machine*.
5. Setelah perancangan sistem selesai, selanjutnya adalah implementasi sistem untuk melakukan pendeteksian pada pasien yang menderita penyakit diabetes
6. Setelah Implementasi selesai, kemudian dilakukan pengujian sistem yang telah dibangun untuk memastikan sistem sesuai dengan yang diharapkan.

### 3.2. Analisis Sistem

Tahapan analisis sistem merupakan salah satu tahapan yang bertujuan untuk mengidentifikasi kebutuhan yang dapat dimanfaatkan agar sistem dapat bekerja secara optimal sesuai dengan yang diharapkan. Adapun terdapat beberapa jenis analisis sistem yang dapat dilakukan adalah analisis kebutuhan sistem yang terdiri dari kebutuhan fungsional dan non-fungsional, serta analisis masalah.

#### 3.2.1. Analisis Masalah

Analisis masalah merupakan tahapan yang dilakukan untuk mengidentifikasi permasalahan yang dibahas dan dikaji pada penelitian sehingga penelitian ini. Fokus permasalahan yang akan dikaji pada penelitian ini adalah pengujian kemampuan algoritma DNN dan SVM dalam mendeteksi penyakit diabetes.

#### 3.2.2. Analisis Kebutuhan

Sasaran utama analisis kebutuhan adalah memastikan bahwa sistem dapat memenuhi harapan pengguna. Untuk itu, analisis kebutuhan digunakan untuk mengenali data dan proses yang diperlukan dalam perancangan sistem. Analisis ini mencakup persyaratan fungsional dan non-fungsional yang harus dipenuhi dalam desain sistem agar sistem yang sedang dibangun dapat mencapai tujuannya. Pada penelitian ini diuraikan analisis kebutuhan menjadi dua kelompok yakni:

##### 1. Kebutuhan Fungsional

Analisis kebutuhan fungsional melibatkan penjelasan mengenai prosedur yang harus dijalankan oleh sistem untuk memenuhi persyaratan. Persyaratan fungsional yang diperlukan dalam sistem ini mencakup:

- a. Sistem harus dapat memproses data yang dimasukkan dengan menggunakan algoritma DNN dan SVM untuk mendeteksi kemungkinan diabetes.
- b. Memungkinkan pengguna untuk menginput data gejala awal yang dimiliki oleh pengguna.
- c. Sistem menampilkan hasil diagnosis dan rekomendasi tindak lanjut

kepada pengguna.

## 2. Kebutuhan Non-Fungsional

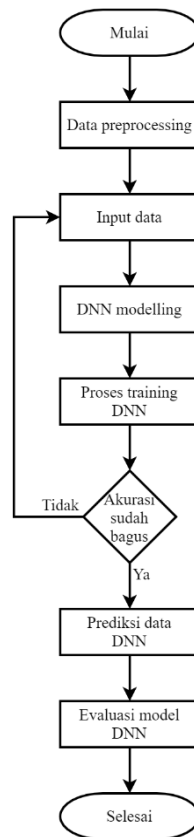
Kebutuhan non-fungsional mengacu pada fitur, karakteristik, atau batasan yang terkait dengan fungsi atau pelayanan yang disediakan oleh sistem. Berikut adalah persyaratan fungsional yang diperlukan untuk sistem ini:

- a. Antarmuka sistem ini dirancang agar mudah dipahami, sehingga dapat digunakan dengan terukur.
- b. Sistem harus dapat diakses oleh pengguna dari berbagai lokasi dan perangkat.
- c. Batasan gejala penyakit yang dapat di deteksi adalah umur, jenis kelamin, poliuria (sering buang air kecil), polidipsia (sering merasa haus), penurunan berat badan, kelelahan berlebihan, polifagia (rasa lapar berlebih), infeksi jamur pada daerah genital, penglihatan kabur, gatal-gatal, mudah marah, kejadian penyembuhan luka yang lama, otot terasa lemah, otot terasa kaku, alopesia (rambut sering rontok), dan obesitas.
- d. Sistem harus memberikan hasil diagnosis dalam waktu yang cepat.

### 3.3. *Flowchart*

*Flowchart* adalah diagram Yang menggunakan simbol-simbol grafis untuk menjelaskan tahapan dalam suatu proses atau alur kerja. Simbol-simbol tersebut dihubungkan dengan panah untuk menunjukkan urutan dan arah aliran. Flowchart membantu memvisualisasikan proses secara sederhana, memudahkan pemahaman, dan digunakan untuk menganalisis atau mendesain sistem.

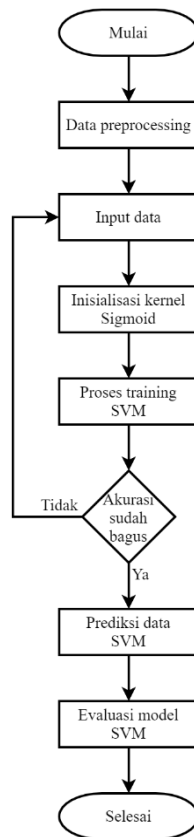
### 3.3.1. Flowchart *Deep Neural Network* (DNN)



**Gambar 3.3.** *Flowchart Deep Neural Network*

*Flowchart* algoritma *Deep Neural Network* dimulai dengan tahap *preprocessing data*. Memasukkan data yang telah ada untuk melakukan pemodelan *Deep Neural Network*. Proses berlanjut dengan menghitung tingkat akurasi dari model yang telah di bangun dan jika akurasi yang di dapatkan kurang bagus maka akan kembali ke tahap awal. Deteksi penyakit diabetes menggunakan hasil model akurasi yang sudah bagus, dan hasil deteksi di evaluasi untuk mengukur kinerja sistem sebelum proses selesai.

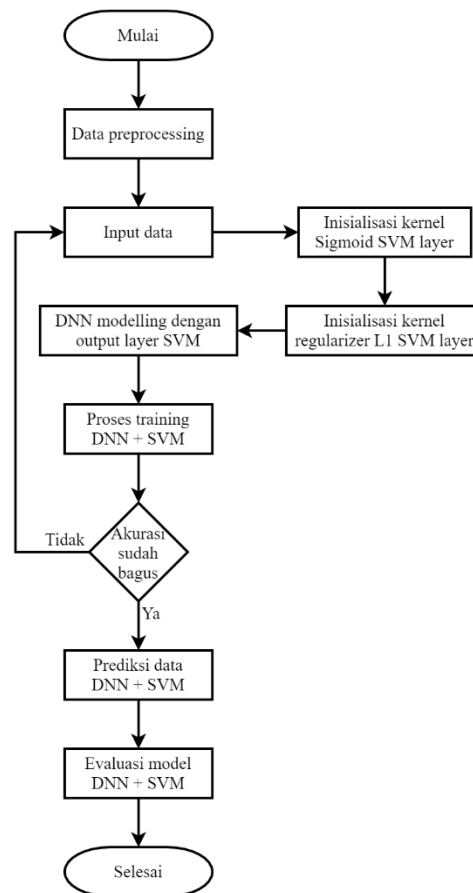
### 3.3.2. Flowchart *Support Vector Machine*



**Gambar 3.4.** *Flowchart Support Vector Machine*

*Flowchart* algoritma *Support Vector Machine* dimulai dengan tahap *preprocessing data*. Memasukkan data untuk inisialisasi kernel *Sigmoid*. Dilakukan proses pelatihan model SVM untuk mengetahui akurasi yang di dapatkan dan jika akurasi yang didapatkan tidak bagus maka akan kembali ke tahap awal. Deteksi penyakit diabetes menggunakan hasil model akurasi yang sudah bagus, dan hasil deteksi di evaluasi untuk mengukur kinerja sistem sebelum proses selesai.

### 3.3.3. Flowchart *Deep Neural Network + Support Vector Machine* (DNN+SVM)



**Gambar 3.5.** Flowchart *Deep Neural Network + Support Vector Machine* (DNN+SVM)

Flowchart algoritma *Deep Neural Network + Support Vector Machine* dimulai dengan tahap *preprocessing data*. Memasukkan data untuk melakukan Inisialisasi kernel *Sigmoid* pada layer SVM. Lalu inisialisasi kernel *Regularizer* pada layer SVM. Selanjutnya dibangun model DNN dengan output layer SVM. Proses berlanjut pada pelatihan model yang telah dibangun untuk melihat akurasi yang didapatkan dan jika tidak bagus maka akan kembali ke tahap awal. Deteksi penyakit diabetes menggunakan hasil model akurasi yang sudah bagus, dan hasil deteksi di evaluasi untuk mengukur kinerja sistem sebelum proses selesai.

### 3.4. Pengambilan *Dataset*

Langkah awal dalam penelitian ini adalah tahap pencaharian dan pengambilan *dataset* yang digunakan, yaitu dari penyedia *dataset* publik, Kaggle, yang berjudul *Early Classification of Diabetes* (<https://www.kaggle.com/datasets/andrewmvd/early-diabetes-classification>). *Dataset* tersebut juga dapat dilihat di *Conference Paper* pada tahun 2020 yang berjudul, “*Likelihood Prediction of Diabetes at Early Stage Using Data Mining Techniques*” ([https://doi.org/10.1007/978-981-13-8798-2\\_12](https://doi.org/10.1007/978-981-13-8798-2_12)). Berdasarkan penelitian tersebut, *dataset* diambil di *Sylhet Diabetes Hospital*, Bangladesh melalui kuesioner langsung ke orang yang memiliki penyakit diabetes, atau orang yang tidak memiliki penyakit diabetes, tetapi memiliki beberapa gejala penyakit diabetes.

Adapun beberapa parameter yang menjadi penentu penyakit diabetes pada *dataset* ini, yaitu umur, jenis kelamin, poliuria (sering buang air kecil), polidipsia (sering merasa haus), penurunan berat badan, kelelahan berlebihan, polifagia (rasa lapar berlebih), infeksi jamur pada daerah genital, penglihatan kabur, gatal-gatal, mudah marah, kejadian penyembuhan luka yang lama, otot terasa lemah, otot terasa kaku, alopesia (rambut sering rontok), obesitas, dan status diabetes. Secara keseluruhan. *Dataset* ini berisi sebanyak 520 baris data dan 17 kolom fitur data.

### 3.5. Pra-pemrosesan Data

Langkah selanjutnya adalah pra-pemrosesan *dataset* sehingga data yang akan digunakan untuk proses pelatihan model dapat lebih optimal dipelajari oleh model tersebut. Tahap ini meliputi *label encoding*, pemeriksaan *missing value data*, *data split* menjadi variabel independen dan variabel dependen, *train-test-split* data, dan normalisasi data.

#### 3.5.1. *Label Encoding*

Model *deep learning* umumnya membutuhkan *input* dalam representasi numerik, sehingga dibutuhkan teknik untuk mengonversi variabel non-numerik menjadi representasi numerik. *Label encoding* adalah teknik yang digunakan untuk mengubah data kategori menjadi nilai numerik. Data kategori adalah data yang memiliki nilai-nilai diskrit dan umumnya tidak berurutan, seperti warna dan jenis kelamin. Misalnya seperti contoh di bawah ini.

**Tabel 3.1.** Contoh Teknik *Label Encoding*

Jenis kelamin	Laki-laki	Perempuan
<i>Encoded</i>	0	1

### 3.5.2. Pemeriksaan *Missing Value Data*

Penanganan nilai yang hilang (*missing value handling*) dalam *machine learning* maupun *deep learning* merupakan tahap yang penting untuk memastikan integritas data dan akurasi model. Pendekatan umum yang dilakukan dapat dengan cara menghapus baris yang memiliki data hilang/kosong, atau berupa data *null*.

### 3.5.3. *Data Split Variabel Independen dan Variabel Dependen*

*Data split* dilakukan dengan cara memisahkan *dataset* menjadi dua bagian, yaitu variabel independen dan variabel dependen. Variabel independen adalah variabel data yang dapat berdiri sendiri dan tidak dipengaruhi oleh variabel dependen. Variabel dependen adalah variabel data yang diamati untuk mengevaluasi hasil dari perubahan variabel independen. Biasanya, variabel independen disimbolkan menjadi (X) sebagai fitur data *input*, sedangkan variabel dependen disimbolkan menjadi (y) sebagai fitur data *output* atau label/target data.

### 3.5.4. *Train-test Split*

Tahapan *train-test split* adalah tahap pembagian data lebih lanjut. Metode ini bertujuan untuk membagi *dataset* menjadi tiga bagian, data latih (*training data*) dan data uji (*test set*). Tujuannya adalah untuk mengembangkan model yang mampu menggeneralisasi dengan baik data yang belum pernah dilihat sebelumnya.

Data latih (*train set*) digunakan untuk melatih model dalam menemukan pola dan hubungan antar variabel independen (fitur data) dengan variabel dependen (target/label data). Sementara itu, data uji (*test set*) digunakan untuk menguji model, di mana data ini adalah data baru, data yang belum pernah dilihat oleh model tersebut selama proses pelatihan (*training*).



### 3.5.5. Normalisasi Data

Tahap normalisasi data dilakukan bertujuan untuk mengubah variabel-variabel atau fitur-fitur data ke dalam skala yang seragam tanpa mengubah distribusi aslinya. Dengan begitu, model dapat dengan mudah mengolah dan menginterpretasikan data tersebut, serta mengurangi kemungkinan bias yang mungkin timbul akibat variasi skala data antar variabel atau fitur data.

Dalam penelitian ini, penulis menerapkan teknik normalisasi Min-Max Scaling. Teknik ini mengubah nilai-nilai variabel atau fitur data sehingga berada dalam rentang antara 0 dan 1. Proses ini dilakukan dengan mengurangi nilai minimum dari setiap nilai variabel, kemudian membaginya dengan selisih nilai maksimum dan minimum, sebagaimana dijelaskan dalam Persamaan (1).

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (1)$$

### 3.6. Pengembangan Model

Tahap selanjutnya setelah pra-pemrosesan data adalah mengembangkan model *deep learning* yang digunakan dalam penelitian ini, yaitu model *Support Vector Machine* (SVM), model *Deep Neural Network* (DNN), dan model *Deep Neural Network* dengan *Support Vector Machine* (DNN + SVM). Dalam penelitian ini, penulis ingin membandingkan performa dari ketiga model tersebut. Model yang dikembangkan dibangun menggunakan pustaka atau *library* Python, yaitu *sci-kit learn*, *TensorFlow*, dan *Keras*.

Model SVM menggunakan *library* Python, yaitu *sci-kit learn*. Sedangkan model DNN dan DNN + SVM menggunakan *library* Python, yaitu *TensorFlow* dan *Keras*. Model DNN dan DNN + SVM terdiri dari lapisan *input*, lapisan tersembunyi (*hidden layer*), dan lapisan *output*. Untuk model DNN + SVM, lapisan *output* akan dioptimisasi menggunakan algoritma *Support Vector Machine* (SVM).

### 3.7. Pelatihan Model

Setelah model selesai dikembangkan, selanjutnya untuk model di-*compile* menggunakan *Adam optimizer*, metrik akurasi, serta *loss function binary cross-entropy* untuk model *Deep Neural Network* (DNN) dan *loss function squared hinge* untuk model *Deep Neural Network* dengan *Support Vector Machine* (DNN + SVM).

Kemudian, model dilatih menggunakan data latih (*train set*) dan dilakukan juga validasi model dengan menggunakan data uji (*test set*). Model *Support Vector Machine* (SVM) dilatih menggunakan *kernel sigmoid*, sedangkan model *Deep Neural Network* (DNN) dan *Deep Neural Network* dengan *Support Vector Machine* (DNN + SVM) dilatih sebanyak 200 *epoch* iterasi pelatihan.

### 3.8. Evaluasi Model

Langkah selanjutnya setelah pelatihan model menggunakan data latih adalah mengevaluasi performa model. Tujuannya adalah untuk melihat kemampuan model dalam mempelajari pola data yang diberikan. Evaluasi model dilakukan menggunakan data uji (*test set*) menggunakan metrik yang telah ditentukan.

Adapun metrik yang digunakan dalam mengevaluasi model yaitu, pada model *Support Vector Machine* (SVM) dievaluasi menggunakan metrik akurasi, *precision*, dan *recall*. Sedangkan model *Deep Neural Network* (DNN) dan *Deep Neural Network* dengan *Support Vector Machine* (DNN + SVM) dievaluasi menggunakan metrik akurasi dan *loss function*.

### 3.9. Pemodelan Sistem

Perancangan aplikasi sistem pendeteksi penyakit diabetes berbasis website dilakukan dengan membuat pemodelan sistem tersebut menggunakan Unified Modelling Language (UML). UML merupakan bahasa standar yang digunakan untuk mendesain, mendokumentasikan, dan memodelkan sistem perangkat lunak, baik yang kompleks maupun sederhana.

UML menyediakan serangkaian notasi grafis yang didefinisikan dengan baik untuk menggambarkan berbagai aspek dari sistem perangkat lunak, seperti

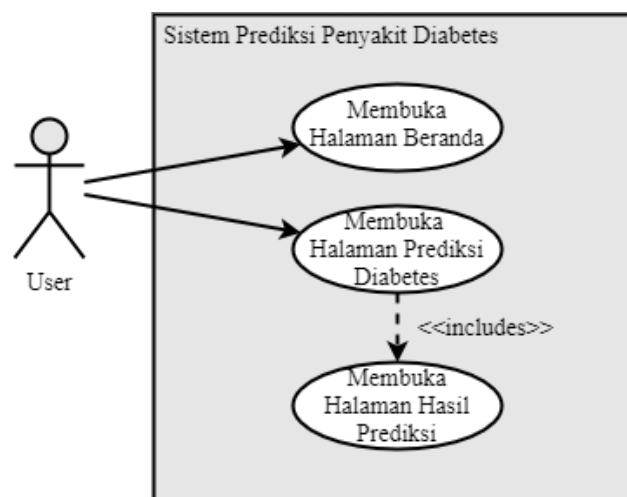
struktur sistem, perilaku, interaksi antar komponen, dan arsitektur. UML berperan untuk menyediakan cara standar bagi pengembang perangkat lunak untuk berkomunikasi dan memahami desain sistem secara efisien dengan spesifikasi yang jelas.

Pada penelitian ini, pemodelan sistem menggunakan *Use Case Diagram* dan *Activity Diagram* untuk membantu proses pengembangan alur sistem dengan tepat.

### 3.9.1. *Use Case Diagram*

*Use case diagram* merupakan salah satu jenis diagram UML untuk menggambarkan interaksi antara aktor (*user*) dengan sistem yang dibangun. Diagram ini membantu pemahaman fungsionalitas sistem dari sudut pandang pengguna. Dalam penelitian ini, *use case diagram* yang digunakan sistem dapat dilihat pada Gambar 3.2.

Pada Gambar 3.3. dapat dilihat bahwa aktor pengguna (*user*) memiliki akses untuk berinteraksi dengan sistem prediksi penyakit diabetes. Aktor pengguna (*user*) tersebut dapat melakukan aktivitas untuk mengakses halaman beranda, halaman prediksi diabetes, serta halaman hasil prediksi yang memiliki hubungan inklusif (*includes*) dan membutuhkan *case* halaman prediksi diabetes.



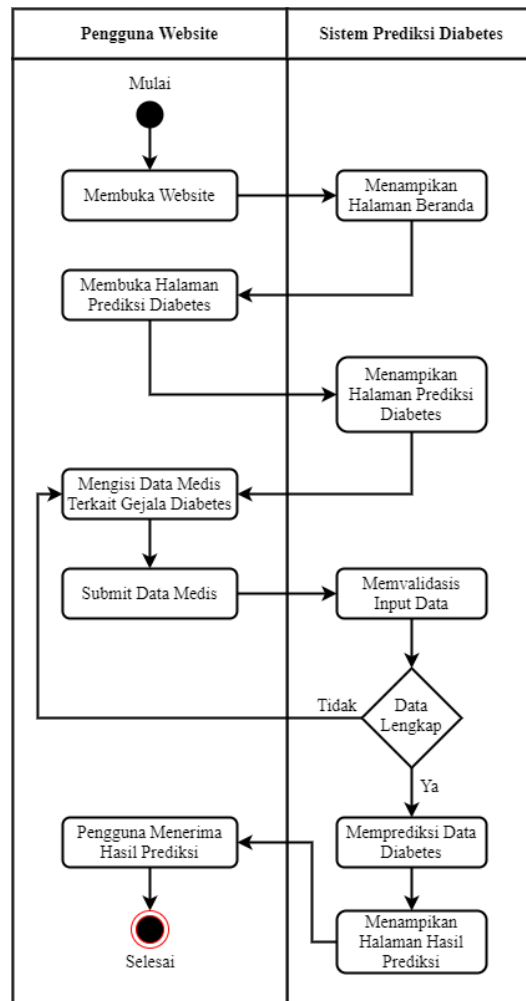
**Gambar 3.6.** *Use Case Diagram* Sistem Pendeteksi Penyakit Diabetes

### 3.9.2. Activity Diagram

*Activity diagram* adalah salah satu jenis diagram UML yang digunakan untuk menggambarkan alur kerja (*workflow*) atau proses bisnis dari sebuah sistem perangkat lunak. *Activity diagram* menggambarkan urutan langkah-langkah (aktivitas), keputusan (*decision*), garis aliran (*flow line*), dan kondisi lainnya yang mempengaruhi alur sistem tersebut. Fungsi dari *activity diagram* adalah untuk membantu dalam memodelkan berbagai jenis alur kerja (*workflow*) dengan cara visual dan dapat dipahami oleh berbagai pihak yang terlibat dalam pengembangan sistem. Dalam penelitian ini, *activity diagram* yang digunakan sistem dapat dilihat pada Gambar 3.4.

Pada Gambar 3.4. dapat dilihat terdapat pengguna *website* dan sistem prediksi diabetes sebagai bagian dari *activity diagram* tersebut. Pengguna *website* membuka *website* lalu, halaman beranda akan ditampilkan oleh sistem. Selanjutnya pengguna membuka halaman prediksi dan sistem akan menampilkannya agar pengguna dapat memasukkan dan mengirimkan data medis terkait gejala diabetes.

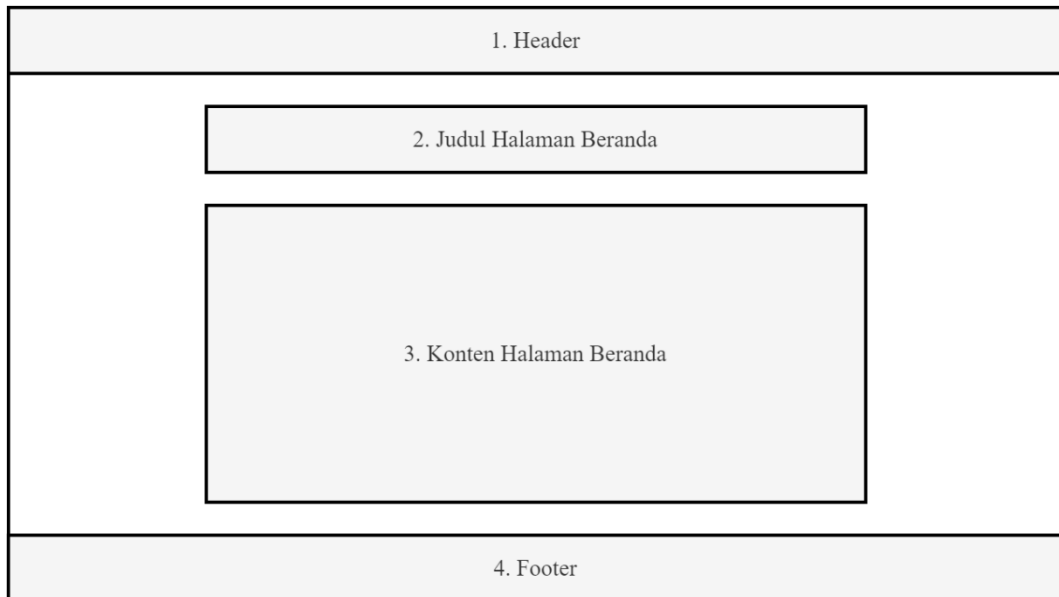
Selanjutnya sistem akan melakukan validasi kelengkapan *input* data menggunakan atribut *required*. Jika ada data yang tidak lengkap, maka sistem akan menampilkan pesan peringatan untuk mengisi data dengan lengkap, lalu mengarahkan ke tahap isi data medis kembali. Jika semua data sudah lengkap, sistem akan melanjutkannya ke prediksi data diabetes menggunakan model yang telah dilatih, dan akan menampilkan hasil prediksi kepada pengguna.



**Gambar 3.7.** Activity Diagram Sistem Pendeteksi Penyakit Diabetes

### 3.9.3. Halaman Beranda

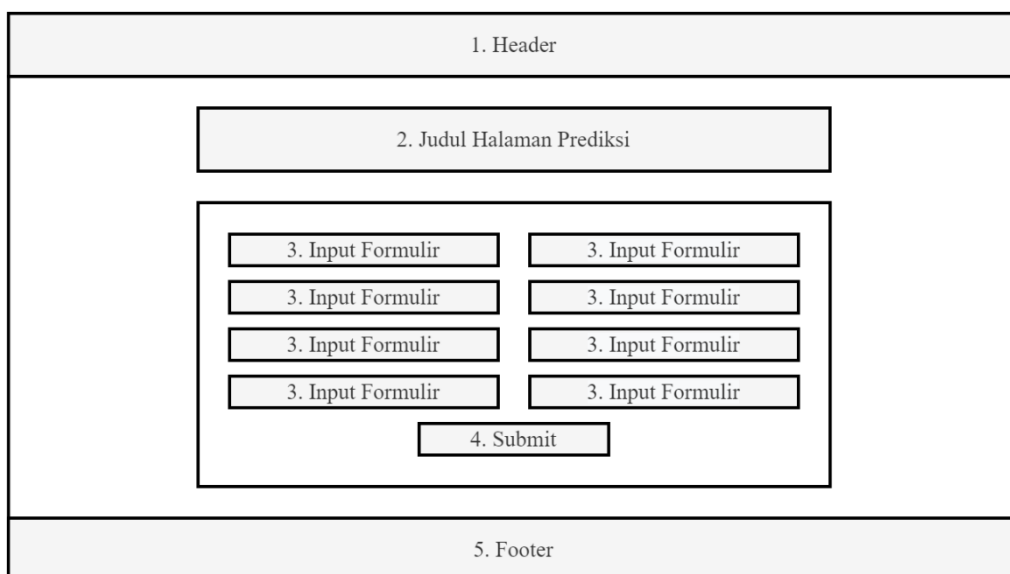
Halaman beranda *website* merupakan halaman yang pertama kali ditampilkan kepada pengguna ketika baru membuka *website*. Halaman ini berisi tampilan konten informasi *website* secara umum. Struktur laman beranda *website* mencakup bagian *header*, bagian judul halaman beranda, bagian konten atau isi *website*, serta bagian *footer*. Tata letak gambar perancangan ini dapat dilihat pada Gambar 3.5.



**Gambar 3.8.** Rancangan Halaman Beranda

#### 3.9.4. *Halaman Prediksi Diabetes*

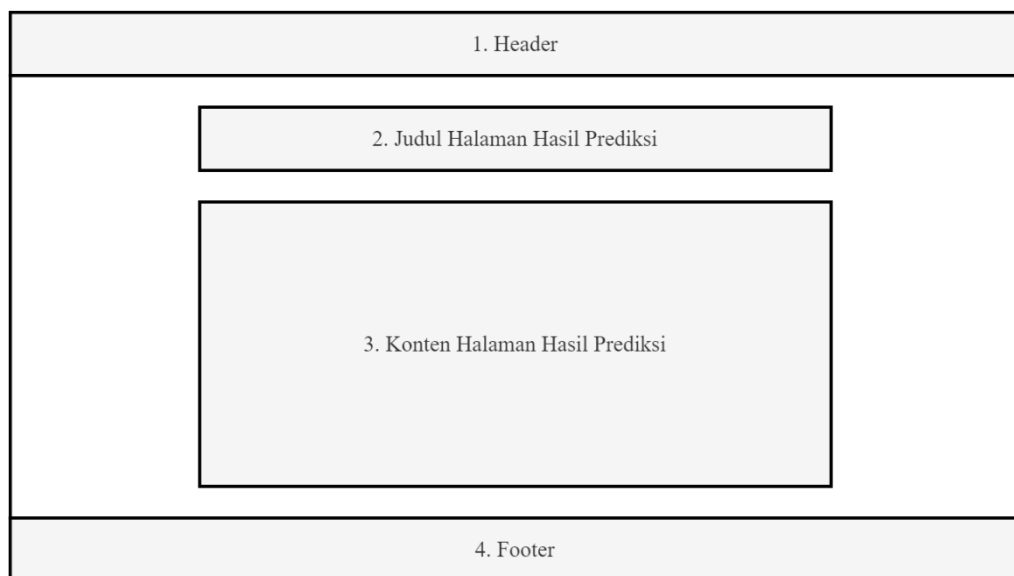
Halaman prediksi *website* merupakan halaman yang dapat diakses pengguna melalui tombol prediksi pada halaman beranda. Halaman ini berisi sebuah formulir dan input data medis pengguna yang berkaitan dengan penyakit diabetes. Struktur laman prediksi *website* mencakup bagian *header*, bagian judul halaman prediksi, bagian konten atau isi *website* berupa formulir dan tombol submit, serta bagian *footer*. Tata letak gambar perancangan ini dapat dilihat pada Gambar 3.6.



**Gambar 3.9.** Rancangan Halaman Prediksi Diabetes

### 3.9.5. Halaman Hasil Prediksi

Halaman hasil prediksi *website* merupakan halaman yang berisi hasil prediksi dan klasifikasi penyakit diabetes setelah pengguna menekan tombol submit pada formulir input data di halaman prediksi diabetes. Struktur laman beranda *website* mencakup bagian *header*, bagian judul halaman beranda, bagian konten atau isi *website* berupa hasil prediksi klasifikasi, serta bagian *footer*. Tata letak gambar perancangan ini dapat dilihat pada Gambar 3.7.



**Gambar 3.10.** Rancangan Halaman Hasil Prediksi

## BAB IV

### IMPLEMENTASI DAN PENGUJIAN SISTEM

#### 4.1 Implementasi Sistem

Dalam penelitian ini, sistem yang dikembangkan memerlukan komponen dan perangkat pendukung dalam mengumpulkan dan memproses *dataset*, melatih model *Deep Neural Network* (DNN) dan *Support Vector Machine* (SVM), serta menerapkannya pada sistem aplikasi *website*. Perangkat keras dan perangkat lunak yang digunakan memiliki spesifikasi sebagai berikut.

##### 4.1.1 Perangkat Keras

Spesifikasi perangkat keras yang dipakai pada penelitian ini adalah sebagai berikut:

1. Intel Core i7 8750H CPU @ 2.20GHz
2. NVIDIA GeForce GTX 1050 4 GB GDDR5
3. 16 GB DDR4 *Dual Channel* RAM

##### 4.1.2 Perangkat Lunak

Spesifikasi perangkat lunak yang dipakai pada penelitian ini adalah sebagai berikut:

1. Implementasi pengambilan *dataset*
  - A. Google Colaboratory
  - B. Kaggle.com API
2. Implementasi pra-pemrosesan *dataset*
  - A. Google Colaboratory
  - B. *Library* : Pandas, NumPy, scikit-learn
3. Implementasi membangun, melatih, dan mengevaluasi model
  - A. Google Colaboratory
  - B. *Library* : Pandas, scikit-learn, TensorFlow, Keras, Matplotlib
4. Implementasi model ke aplikasi berbasis *website*
  - A. Visual Studio Code
  - B. Bahasa : HTML, CSS, JavaScript, Python
  - C. *Library* : Bootstrap, Flask, Keras, os



#### 4.2 Implementasi Tahap Pengambilan Dataset

Dataset yang digunakan dalam penelitian ini adalah dataset *Early Classification of Diabetes* (<https://kaggle.com/datasets/andrewmvd/early-diabetes-classification>) yang diperoleh dari website Kaggle. Dataset ini memiliki 520 baris data dan 17 kolom fitur data. Beberapa fitur data yang terdapat dalam dataset ini adalah umur, jenis kelamin, poliuria (sering buang air kecil), polidipsia (sering merasa haus), penurunan berat badan, kelelahan berlebihan, polifagia (rasa lapar berlebih), infeksi jamur pada daerah genital, penglihatan kabur, gatal-gatal, mudah marah, kejadian penyembuhan luka yang lama, otot terasa lemah, otot terasa kaku, alopesia (rambut sering rontok), obesitas, dan status diabetes. Pada Tabel 4.1. dan Tabel 4.2. dapat dilihat isi dataset penyakit diabetes yang digunakan dalam penelitian ini.

**Tabel 4.1.** Hasil Pengambilan Dataset (1)

	Age	Gender	Poly- uria	Poly- dipsia	Sudden_ weight_ loss	Weak- ness	Poly- phagia	Genital_ thrush	Visual_ blurring
<b>0</b>	40	Male	0	1	0	1	0	0	0
<b>1</b>	58	Male	0	0	0	1	0	0	1
<b>2</b>	41	Male	1	0	0	1	1	0	0
<b>3</b>	45	Male	0	0	1	1	1	1	0
<b>4</b>	60	Male	1	1	1	1	1	0	1
...	...	...	...	...	...	...	...	...	...
<b>515</b>	39	Female	1	1	1	0	1	0	0
<b>516</b>	48	Female	1	1	1	1	1	0	0
<b>517</b>	58	Female	1	1	1	1	1	0	1
<b>518</b>	32	Female	0	0	0	1	0	0	1
<b>519</b>	42	Female	0	0	0	0	0	0	0

**Tabel 4.2.** Hasil Pengambilan *Dataset* (2)

	Itching	Irritability	Delayed_healing	Partial_paresis	Muscle_stiffness	alopecia	obesity	class
<b>0</b>	1	0	1	0	1	1	1	1
<b>1</b>	0	0	0	1	0	1	0	1
<b>2</b>	1	0	1	0	1	1	0	1
<b>3</b>	1	0	1	0	0	0	0	1
<b>4</b>	1	1	1	1	1	1	1	1
...	...	...	...	...	...	...	...	...
<b>515</b>	1	0	1	1	0	0	0	1
<b>516</b>	1	1	1	1	0	0	0	1
<b>517</b>	0	0	0	1	1	0	1	1
<b>518</b>	1	0	1	0	0	1	0	0
<b>519</b>	0	0	0	0	0	0	0	0

**Tabel 4.3.** Hasil Deskripsi Isi *Dataset*

#	Kolom	Non-Null Count	Tipe Data
<b>0</b>	Age	520 non-null	Int64
<b>1</b>	Gender	520 non-null	Object
<b>2</b>	Polyuria	520 non-null	Int64
<b>3</b>	Polydipsia	520 non-null	Int64
<b>4</b>	Sudden_weight_loss	520 non-null	Int64
<b>5</b>	Weakness	520 non-null	Int64
<b>6</b>	Polyphagia	520 non-null	Int64
<b>7</b>	Genital_thrush	520 non-null	Int64
<b>8</b>	Visual_blurring	520 non-null	Int64
<b>9</b>	Itching	520 non-null	Int64
<b>10</b>	Irritability	520 non-null	Int64
<b>11</b>	Delayed_healing	520 non-null	Int64
<b>12</b>	Partial_paresis	520 non-null	Int64
<b>13</b>	Muscle_stiffness	520 non-null	Int64
<b>14</b>	Alopecia	520 non-null	Int64
<b>15</b>	Obesity	520 non-null	Int64
<b>16</b>	Class	520 non-null	Int64

Berdasarkan Tabel 4.3. dapat dilihat hasil deskripsi isi *dataset* yang memiliki 17 fitur data atau kolom data dengan masing-masing sebanyak 520 data *non-null* dan tipe datanya adalah Int64, kecuali fitur data Gender yang berupa *Object* atau *String*.

### 4.3. Implementasi Tahap Pra-pemrosesan Data

Tahap Pra-pemrosesan *dataset* dilakukan dengan beberapa tahap, diawali dengan *label encoding* untuk fitur data Gender, kemudian pemeriksaan untuk penanganan data yang hilang atau kosong (*missing value*), lalu dilakukan *data split* menjadi variabel independen dan variabel dependen, *train-test split* data, dan terakhir adalah normalisasi data menggunakan metode Min-Max Scaling.

#### 4.3.1. Label Encoding

*Label encoding* adalah teknik pengubahan data kategori menjadi bentuk numerik dengan menggantikan setiap kategori dengan angka unik. Metode ini cocok digunakan untuk data yang tidak memiliki urutan atau hierarki antar kategorinya. Misalnya, seperti yang terdapat pada Tabel 4.4. menunjukkan fitur data Gender yang diubah dari “*Male*” menjadi 0 dan “*Female*” menjadi 1.

**Tabel 4.4.** Hasil Teknik *Label Encoding* pada Fitur Data Gender

Fitur Data	Kategori	Hasil <i>Label Encoding</i>
Gender	<i>Male</i>	0
	<i>Female</i>	1

#### 4.3.2. Pemeriksaan *Missing Value*

Tahap pemeriksaan *missing value* atau nilai yang hilang atau kosong adalah proses *identifikasi* dan penanganan data yang tidak lengkap dalam sebuah *dataset*. *Missing values* bisa terjadi karena berbagai alasan seperti kesalahan dalam pengumpulan data, kerusakan data, atau data yang tidak tersedia. Dalam penelitian ini, *dataset* yang digunakan tidak memiliki *missing value* atau data yang hilang, atau *null* seperti yang dapat dilihat pada Gambar 4.1.

```
[ ] 1 df.isnull().sum()
age      0
gender    0
polyuria  0
polydipsia 0
sudden_weight_loss 0
weakness  0
polyphagia 0
genital_thrush 0
visual_blurring 0
itching    0
irritability 0
delayed_healing 0
partial_paresis 0
muscle_stiffness 0
alopecia   0
obesity    0
class      0
dtype: int64
```

**Gambar 4.1.** Hasil Pemeriksaan *Missing Value*

#### 4.3.3. *Data Split Variabel Independen dan Variabel Dependen*

Selanjutnya, dilakukan tahap data splitting untuk memisahkan dataset menjadi variabel independen (X) dan variabel dependen (y). Variabel independen (X) sebagai fitur data yang terdiri dari umur, jenis kelamin, poliuria (sering buang air kecil), polidipsia (sering merasa haus), penurunan berat badan, kelelahan berlebihan, polifagia (rasa lapar berlebih), infeksi jamur pada daerah genital, penglihatan kabur, gatal-gatal, mudah marah, kejadian penyembuhan luka yang lama, otot terasa lemah, otot terasa kaku, alopesia (rambut sering rontok), dan obesitas. Sedangkan variabel dependen (y) sebagai target atau label data, yaitu status diabetes yang terdapat pada kolom terakhir di kolom ke-17.

#### 4.3.4. *Train-test Split*

Kemudian, tahap *train-test split* dilakukan dengan tujuan membagi *dataset* menjadi dua buah bagian, yaitu data latih (*training data*) dan data uji (*testing data*). Tahap *train-test split* ini menggunakan rasio 80-20, yaitu 80% untuk data latih dan 20% untuk data uji yang dapat dilihat pada Gambar 4.2.

<i>Train</i>	<i>Test</i>
80%	20%

**Gambar 4.2.** Rasio *Train-test Split*

#### 4.3.5. Normalisasi Data

Langkah terakhir dalam pra-pemrosesan *dataset* dalam penelitian ini, yaitu normalisasi data dengan menggunakan metode *Min-Max Scaling*. Metode ini bertujuan untuk mengubah data yang awalnya memiliki rentang yang berbeda-beda, menjadi data dengan rentang 0.0 hingga 1.0. *Min-Max Scaling* biasanya diterapkan pada data yang memiliki rentang terbatas.

### 4.4. Implementasi Pembangunan Model

Pada implementasi pengembangan model dilakukan menggunakan tiga buah algoritma, yaitu *Support Vector Machine* (SVM), *Deep Neural Network* (DNN), dan *Deep Neural Network + Support Vector Machine* (DNN + SVM). Model tersebut kemudian akan dievaluasi dan dibandingkan model mana yang memiliki metrik evaluasi yang terbaik.

#### 4.4.1. Model Support Vector Machine

Untuk pengembangan model *Support Vector Machine* (SVM) menggunakan *library* *sci-kit learn*. Algoritma SVM yang digunakan adalah jenis *Support Vector Machine Classifier* (SVC) untuk kasus klasifikasi dengan *kernel sigmoid*. Penerapannya dalam Python dapat dilihat pada Gambar 4.3.

```
[ ] 1 from sklearn import svm, metrics
    2
    3 model1 = svm.SVC(kernel='sigmoid')
```

**Gambar 4.3.** Penerapan Model *Support Vector Machine* (SVM)

#### 4.4.2. Model Deep Neural Network

Untuk pengembangan model *Deep Neural Network* (DNN) menggunakan *library* TensorFlow dan Keras. Model *Deep Neural Network* (DNN) yang dikembangkan terdiri dari sebuah *input layer* yang menerima 16 fitur data, 3 buah *Dense layer* dengan neuron sebanyak 32, 16, dan 8 masing-masing dengan fungsi aktivasi ReLU. Selain itu, juga menggunakan 3 buah *Dropout layer* sebesar 0.4 untuk mencegah *overfitting*, dan sebuah *output layer* sebesar 1 neuron dengan fungsi aktivasi *Sigmoid*. Arsitektur model DNN dapat dilihat pada Gambar 4.4.

```
[ ] 1 model2 = Sequential([
2     Dense(32, activation='relu', name='Dense_1', input_shape=(16,)),
3     Dropout(0.4, name='Dropout_1'),
4     Dense(16, activation='relu', name='Dense_2'),
5     Dropout(0.4, name='Dropout_2'),
6     Dense(8, activation='relu', name='Dense_3'),
7     Dropout(0.4, name='Dropout_3'),
8     Dense(1, activation='sigmoid', name='Dense_4')
9 ])
10
11 model2.summary()
```

Model: "sequential\_5"

Layer (type)	Output Shape	Param #
Dense_1 (Dense)	(None, 32)	544
Dropout_1 (Dropout)	(None, 32)	0
Dense_2 (Dense)	(None, 16)	528
Dropout_2 (Dropout)	(None, 16)	0
Dense_3 (Dense)	(None, 8)	136
Dropout_3 (Dropout)	(None, 8)	0
Dense_4 (Dense)	(None, 1)	9

---

Total params: 1217 (4.75 KB)  
 Trainable params: 1217 (4.75 KB)  
 Non-trainable params: 0 (0.00 Byte)

**Gambar 4.4.** Arsitektur Model *Deep Neural Network* (DNN)

#### 4.4.3. Model *Deep Neural Network* + *Support Vector Machine* (DNN + SVM)

Untuk pengembangan model *Deep Neural Network* + *Support Vector Machine* (DNN + SVM) menggunakan *library* TensorFlow dan Keras. Model ini hampir sama dengan arsitektur model DNN tanpa SVM yang terdiri dari sebuah *input layer* yang menerima 16 fitur data, 3 buah *Dense layer* dengan neuron sebanyak 32, 16, dan 8 masing-masing dengan fungsi aktivasi ReLU. Selain itu, juga menggunakan 3 buah *Dropout layer* sebesar 0.4 untuk mencegah *overfitting*, dan sebuah *output layer* sebesar 1 neuron dengan fungsi aktivasi *Sigmoid*. Namun, pada output layer menggunakan parameter *kernel regularizer* L1 sebesar 0.1. Arsitektur model DNN dapat dilihat pada Gambar 4.5.

```
[ ] 1 model3 = Sequential([
2     Dense(32, activation='relu', name='Dense_1', input_shape=(16,)),
3     Dropout(0.4, name='Dropout_1'),
4     Dense(16, activation='relu', name='Dense_2'),
5     Dropout(0.4, name='Dropout_2'),
6     Dense(8, activation='relu', name='Dense_3'),
7     Dropout(0.4, name='Dropout_3'),
8     Dense(1, kernel_regularizer=tf.keras.regularizers.l1(0.1), activation='sigmoid', name='Dense_4')
9 ])
10
11 model3.summary()
```

Model: "sequential\_6"

Layer (type)	Output Shape	Param #
Dense_1 (Dense)	(None, 32)	544
Dropout_1 (Dropout)	(None, 32)	0
Dense_2 (Dense)	(None, 16)	528
Dropout_2 (Dropout)	(None, 16)	0
Dense_3 (Dense)	(None, 8)	136
Dropout_3 (Dropout)	(None, 8)	0
Dense_4 (Dense)	(None, 1)	9

=====

Total params: 1217 (4.75 KB)  
Trainable params: 1217 (4.75 KB)  
Non-trainable params: 0 (0.00 Byte)

**Gambar 4.5.** Arsitektur Model *Deep Neural Network + Support Vector Machine* (DNN + SVM)

#### 4.5. Implementasi Pelatihan Model

Pada tahap implementasi pelatihan model, juga dilakukan pada ketiga model, yaitu *Support Vector Machine* (SVM), *Deep Neural Network* (DNN), dan *Deep Neural Network + Support Vector Machine* (DNN + SVM).

##### 4.5.1. Pelatihan Model *Support Vector Machine* (SVM)

Proses pelatihan model *Support Vector Machine* (SVM) menggunakan data latih (*training data*) pada variabel independen ( $X_{\text{train}}$ ) dan juga variabel dependen ( $y_{\text{train}}$ ) dengan SVC dan *kernel Sigmoid* yang dapat dilihat pada Gambar 4.6.

```
[ ] 1 model1.fit(X_train, y_train)
```

SVC

SVC(kernel='sigmoid')

**Gambar 4.6.** Pelatihan Model *Support Vector Machine* (SVM)

#### 4.5.2. Pelatihan *Model Deep Neural Network* (DNN)

Proses pelatihan model *Deep Neural Network* (DNN) terlebih dahulu dengan mengkompilasi (*compile*) model tersebut dengan *Adam optimizer*, metrik akurasi, dan *binary cross-entropy loss function*. Kemudian melakukan tahap pelatihan model dengan data latih (*training data*) dan data uji (*testing data*) untuk memvalidasi model. Proses pelatihan model dilakukan sebanyak 100 *epoch* yang dapat dilihat pada Gambar 4.7.

```
[ ] 1 model2.compile(  
2     optimizer='adam',  
3     loss='binary_crossentropy',  
4     metrics=['accuracy']  
5 )  
  
[ ] 1 h2 = model2.fit(  
2     X_train, y_train,  
3     validation_data=(X_test, y_test),  
4     epochs=100,  
5     verbose=2  
6 )
```

**Gambar 4.7.** Pelatihan Model *Deep Neural Network* (DNN)

#### 4.5.3. Pelatihan *Model Deep Neural Network + Support Vector Machine* (DNN + SVM)

Proses pelatihan model *Model Deep Neural Network + Support Vector Machine* (DNN + SVM) terlebih dahulu dengan mengkompilasi (*compile*) model tersebut dengan *Adam optimizer*, metrik akurasi, dan *squared hinge loss function*. Kemudian melakukan tahap pelatihan model dengan data latih (*training data*) dan data uji (*testing data*) untuk memvalidasi model. Proses pelatihan model dilakukan sebanyak 100 *epoch* yang dapat dilihat pada Gambar 4.8.

```
[ ] 1 model3.compile(  
2     optimizer='adam',  
3     loss='squared_hinge',  
4     metrics=['accuracy']  
5 )  
  
[ ] 1 h3 = model3.fit(  
2     X_train, y_train,  
3     validation_data=(X_test, y_test),  
4     epochs=100,  
5     verbose=2  
6 )
```

**Gambar 4.8.** Pelatihan Model *Deep Neural Network + Support Vector Machine* (DNN + SVM)



#### 4.6. Implementasi Evaluasi Model

Tahap akhir dari pengembangan model, yaitu evaluasi model yang sudah dilatih sebelumnya menggunakan data yang diberikan. Evaluasi model dilakukan pada ketiga model, yaitu *Support Vector Machine* (SVM), *Deep Neural Network* (DNN), dan *Deep Neural Network + Support Vector Machine* (DNN + SVM).

##### 4.6.1. Evaluasi Model *Support Vector Machine* (SVM)

Proses evaluasi model *Support Vector Machine* (SVM) menggunakan data uji (testing data) dan dievaluasi menggunakan metrik akurasi, *precision*, dan *recall*. Hasilnya didapatkan bahwa pada model SVM memperoleh akurasi sebesar 75,96% dengan metrik *precision* sebesar 0.8, serta metrik *recall* sebesar 0,8125 yang dapat dilihat pada Gambar 4.9.

```
[ ] 1 y_pred = model1.predict(X_test)

[ ] 1 print('Accuracy :', metrics.accuracy_score(y_test, y_pred))
    2 print('Precision :', metrics.precision_score(y_test, y_pred))
    3 print('Recall    :', metrics.recall_score(y_test, y_pred))

⇒ Accuracy : 0.7596153846153846
   Precision : 0.8
   Recall    : 0.8125
```

**Gambar 4.9.** Evaluasi Model *Support Vector Machine* (SVM)

##### 4.6.2. Evaluasi Model *Deep Neural Network* (DNN)

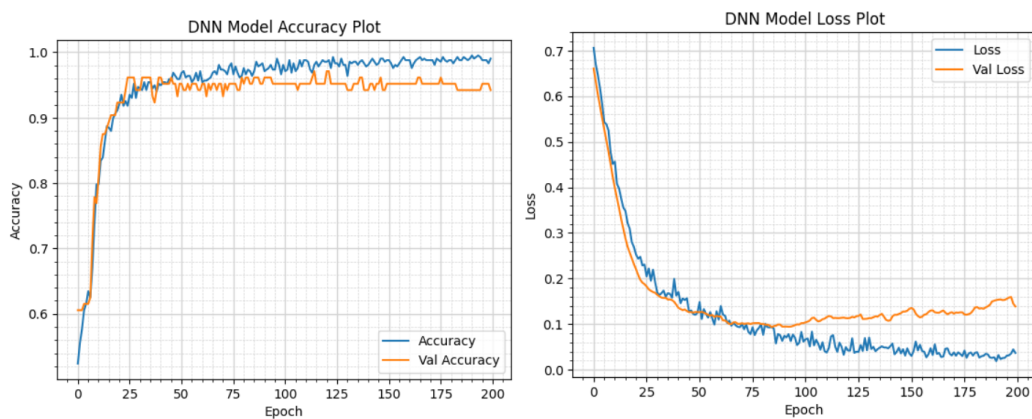
Proses evaluasi model *Deep Neural Network* (DNN) menggunakan metrik akurasi pelatihan, akurasi validasi, *loss function* pelatihan, dan *loss function* validasi. Pada model ini didapatkan tingkat akurasi pelatihan sebesar 99,04% dan akurasi validasi sebesar 94,23%, serta *loss function* pelatihan sebesar 0,1388 dan *loss function* validasi sebesar 0,0832. Hasil pelatihan model dapat dilihat pada Gambar 4.10. Sedangkan hasil evaluasi model berupa grafik akurasi pelatihan dan akurasi validasi, serta *loss function* pelatihan dan *loss function* validasi dapat dilihat pada Gambar 4.11.

```

Epoch 195/200
13/13 - 0s - loss: 0.0261 - accuracy: 0.9928 - val_loss: 0.1533 - val_accuracy: 0.9423 - 322ms/epoch - 25ms/step
Epoch 196/200
13/13 - 0s - loss: 0.0309 - accuracy: 0.9880 - val_loss: 0.1556 - val_accuracy: 0.9519 - 272ms/epoch - 21ms/step
Epoch 197/200
13/13 - 0s - loss: 0.0317 - accuracy: 0.9880 - val_loss: 0.1583 - val_accuracy: 0.9519 - 280ms/epoch - 22ms/step
Epoch 198/200
13/13 - 0s - loss: 0.0355 - accuracy: 0.9880 - val_loss: 0.1596 - val_accuracy: 0.9519 - 161ms/epoch - 12ms/step
Epoch 199/200
13/13 - 0s - loss: 0.0445 - accuracy: 0.9832 - val_loss: 0.1450 - val_accuracy: 0.9519 - 149ms/epoch - 11ms/step
Epoch 200/200
13/13 - 0s - loss: 0.0368 - accuracy: 0.9904 - val_loss: 0.1388 - val_accuracy: 0.9423 - 166ms/epoch - 13ms/step

```

**Gambar 4.10.** Hasil Pelatihan Model *Deep Neural Network* (DNN)



**Gambar 4.11.** Evaluasi Model *Deep Neural Network* (DNN)

#### 4.6.3. Evaluasi Model *Deep Neural Network* dan *Support Vector Machine* (DNN + SVM)

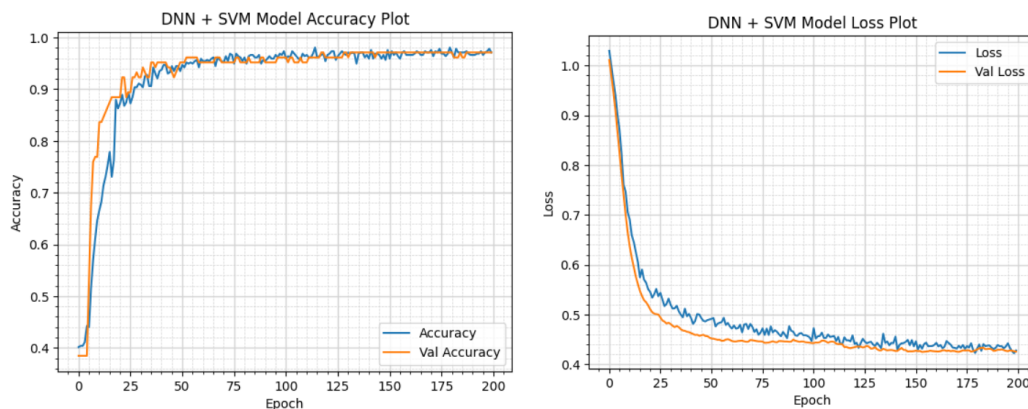
Proses evaluasi model *Deep Neural Network* + *Support Vector Machine* (DNN + SVM) menggunakan metrik akurasi pelatihan, akurasi validasi, *loss function* pelatihan, dan *loss function* validasi. Pada model ini didapatkan tingkat akurasi pelatihan sebesar 97,12% dan akurasi validasi sebesar 97,12%, serta *loss function* pelatihan sebesar 0,4379 dan *loss function* validasi sebesar 0,4256. Hasil pelatihan model dapat dilihat pada Gambar 4.12. Sedangkan hasil evaluasi model berupa grafik akurasi pelatihan dan akurasi validasi, serta *loss function* pelatihan dan *loss function* validasi dapat dilihat pada Gambar 4.13.

```

Epoch 195/200
13/13 - 0s - loss: 0.4361 - accuracy: 0.9736 - val_loss: 0.4273 - val_accuracy: 0.9712 - 85ms/epoch - 7ms/step
Epoch 196/200
13/13 - 0s - loss: 0.4424 - accuracy: 0.9663 - val_loss: 0.4266 - val_accuracy: 0.9712 - 92ms/epoch - 7ms/step
Epoch 197/200
13/13 - 0s - loss: 0.4330 - accuracy: 0.9712 - val_loss: 0.4281 - val_accuracy: 0.9712 - 79ms/epoch - 6ms/step
Epoch 198/200
13/13 - 0s - loss: 0.4269 - accuracy: 0.9736 - val_loss: 0.4275 - val_accuracy: 0.9712 - 92ms/epoch - 7ms/step
Epoch 199/200
13/13 - 0s - loss: 0.4224 - accuracy: 0.9784 - val_loss: 0.4264 - val_accuracy: 0.9712 - 90ms/epoch - 7ms/step
Epoch 200/200
13/13 - 0s - loss: 0.4279 - accuracy: 0.9712 - val_loss: 0.4256 - val_accuracy: 0.9712 - 97ms/epoch - 7ms/step

```

**Gambar 4.12.** Hasil Pelatihan Model *Deep Neural Network + Support Vector Machine* (DNN + SVM)



**Gambar 4.13.** Evaluasi Model *Deep Neural Network + Support Vector Machine* (DNN + SVM)

#### 4.7. Implementasi Model ke Sistem *Website*

Setelah model selesai dilatih dan dilakukan tahap evaluasi, selanjutnya adalah menyimpan dan mengekspor model yang telah dilatih tersebut untuk digunakan pada implementasi ke sistem *website*. Proses menyimpan dan mengekspor model yang digunakan pada sistem dapat dilihat pada Gambar 4.14.

Pada penelitian ini, model yang digunakan untuk implementasi model ke sistem *website* adalah model *Deep Neural Network + Support Vector Machine* (DNN + SVM). Model disimpan ke dalam sebuah *file* dengan format H5 (*Hierarchical Data Format 5 File*) yang berisi data *object array* multidimensional yang besar. Selain itu, model juga disimpan dalam format folder yang berisi aset, variabel, dan parameter model tersebut. Model dengan format H5 memiliki ukuran berkas 57,5 KB dan model dengan format folder 182 KB.

```
[ ] 1 model3.save('model_DNN_SVM')
    2 model3.save('model_DNN_SVM.h5')

/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py:3103: UserWarning: You are saving
saving_api.save_model(

[ ] 1 !zip -r model_DNN_SVM.zip model_DNN_SVM/

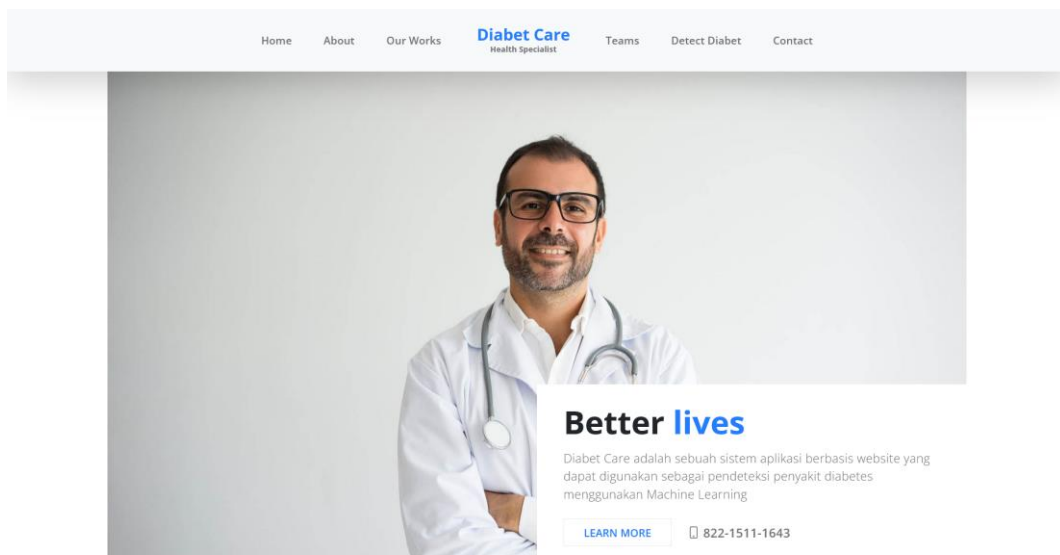
adding: model_DNN_SVM/ (stored 0%)
adding: model_DNN_SVM/assets/ (stored 0%)
adding: model_DNN_SVM/fingerprint.pb (stored 0%)
adding: model_DNN_SVM/variables/ (stored 0%)
adding: model_DNN_SVM/variables/variables.index (deflated 62%)
adding: model_DNN_SVM/variables/variables.data-00000-of-00001 (deflated 30%)
adding: model_DNN_SVM/saved_model.pb (deflated 88%)
adding: model_DNN_SVM/keras_metadata.pb (deflated 89%)

[ ] 1 !pip freeze > requirements.txt

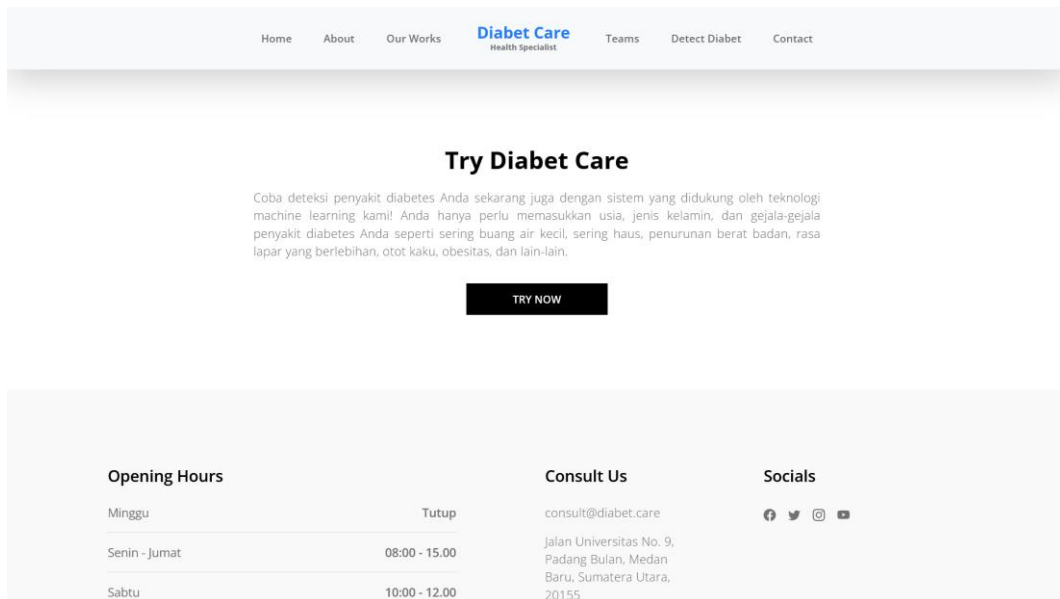
[ ] 1 files.download('model_DNN_SVM.zip')
    2 files.download('model_DNN_SVM.h5')
    3 files.download('requirements.txt')
```

**Gambar 4.14.** Proses Menyimpan dan Ekspor Model

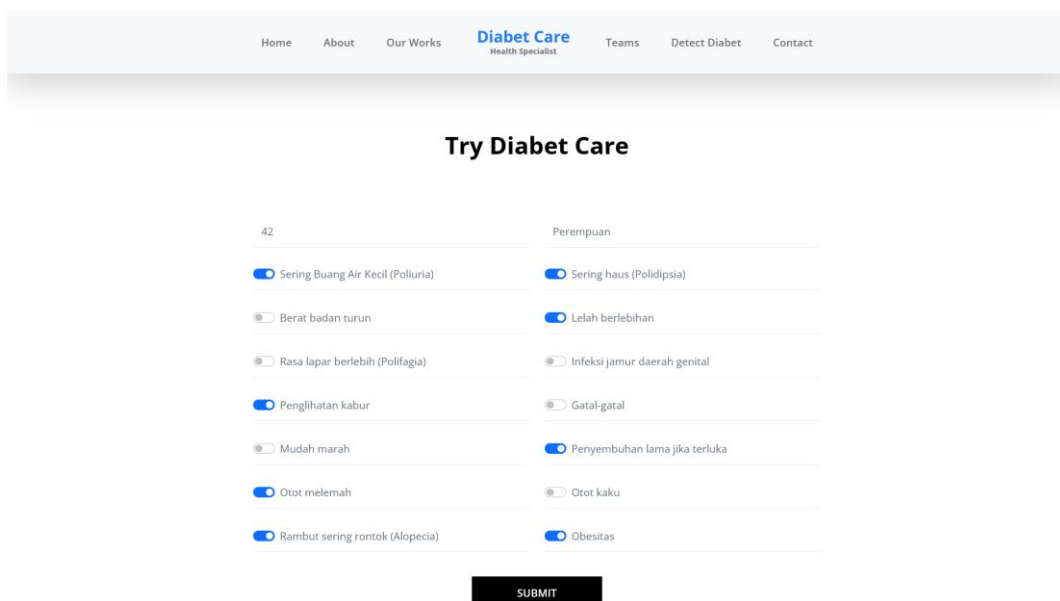
Implementasi model ke sistem *website* menggunakan *framework* Bootstrap sebagai tampilan *front-end* dan *framework* Flask Python sebagai *back-end*. Pada Gambar 4.15. dapat dilihat tampilan halaman beranda *website*. Pada Gambar 4.16. dapat dilihat tampilan tombol prediksi pada halaman beranda. Pada Gambar 4.17. dapat dilihat tampilan halaman prediksi diabetes. Pada Gambar 4.18. dapat dilihat tampilan halaman hasil prediksi diabetes.



**Gambar 4.15.** Halaman Beranda *Website*



**Gambar 4.16.** Tombol Prediksi pada Halaman Beranda *Website*



**Gambar 4.17.** Halaman Prediksi Diabetes *Website*

[Home](#) [About](#) [Our Works](#) [Diabet Care](#)  
Health Specialist [Teams](#) [Detect Diabet](#) [Contact](#)

## Try Diabet Care

Anda mungkin terkena diabetes! Segera periksa diri Anda ke dokter.

COBA LAGI

Usia	Pilih
<input type="checkbox"/> Sering Buang Air Kecil (Polyuria)	<input type="checkbox"/> Sering haus (Polydipsia)
<input type="checkbox"/> Berat badan turun	<input type="checkbox"/> Lelah berlebihan
<input type="checkbox"/> Rasa lapar berlebih (Polyfagia)	<input type="checkbox"/> Infeksi jamur daerah genital
<input type="checkbox"/> Penglihatan kabur	<input type="checkbox"/> Gatal-gatal
<input type="checkbox"/> Mudah marah	<input type="checkbox"/> Penyembuhan lama jika terluka
<input type="checkbox"/> Otot melemah	<input type="checkbox"/> Otot kaku

**Gambar 4.18.** Halaman Hasil Prediksi Diabetes

## BAB V

### PENUTUP

Pada bab ini dibahas kesimpulan dan saran dari sistem pendeteksi penyakit diabetes dengan metode *machine learning* menggunakan *Deep Neural Network* (DNN) dan *Support Vector Machine* (SVM) berbasis website.

#### 5.1. Kesimpulan

Berdasarkan hasil implementasi dan pengujian yang terdapat pada Bab 4, maka penulismenarik beberapa kesimpulan, yaitu:

1. Pada penelitian ini, model *Deep Neural Network* (DNN) yang dibangun dengan arsitektur sebuah *input layer*, tiga buah *dense layer* diikuti tiga buah *dropout layer* pada masing-masing *dense layer*, dan sebuah *output layer*, serta menerapkan algoritma *Support Vector Machine* (SVM) didapatkan akurasi sebesar 97.12% dalam evaluasi model, yang menunjukkan kemampuan yang baik dalam mengklasifikasikan penyakit diabetes. Selain itu, hasil pengujian dari loss function pada model tersebut juga cukup rendah 0.4256, yang menunjukkan bahwa model memiliki kemampuan yang baik dalam mengurangi kesalahan prediksi.
2. Berdasarkan hasil pengujian yang telah dilakukan pada penelitian ini, Model yang dibangun menggunakan algoritma *Support VectorMachine* (SVM) memiliki akurasi sebesar 75,96%, Model yang dibangun menggunakan algoritma *Deep Neural Network* (DNN) memiliki akurasi sebesar 99.04% dan akurasi evaluasi sebesar 94.23%, dan Model yang dibangun menggunakan algoritma *Deep Neural Network* dan *Support Vector Machine* (DNN + SVM) memiliki akurasi sebesar 97.12% dan akurasi evaluasi sebesar 97.12%.
3. Berdasarkan analisis terhadap hasil akurasi dan tingkat loss model machine learning tersebut, dapat disimpulkan bahwa model berhasil mengklasifikasikan data orang yang terkena diabetes dan orang yang tidak terkena diabetes menggunakan menggunakan algoritma *Deep Neural Network* dan *Support Vector Machine* (DNN + SVM). Model tersebut

menunjukkan tingkat akurasi yang cukup tinggi dalam evaluasi model, yang menunjukkan kemampuan yang baik dalam mengklasifikasikan penyakit diabetes. Selain itu, hasil dari loss function pada model tersebut juga cukup rendah, yang menunjukkan bahwa model memiliki kemampuan yang baik dalam mengurangi kesalahan prediksi. Terbukti dengan menggunakan arsitektur model DNN dan kombinasi algoritma SVM pada output layer dapat membantu meningkatkan akurasi validasi dan mencegah terjadinya *overfitting* pada proses *training*.

## 5.2. Saran

Berikut ini adalah beberapa saran dan masukan yang dapat diberikan oleh penulis dari hasil penelitian ini untuk pengembangan sistem selanjutnya, yaitu:

1. Melakukan penelitian lebih lanjut dengan pengumpulan data yang besar dan beragam untuk meningkatkan akurasi deteksi
2. Penelitian ini menggunakan data dari penyedia *dataset* publik Kaggle yang berisidata medis pasien penyakit diabetes. Sehingga sistem ini hanya dapat mendiagnosa gejala awal yang dimiliki oleh si pengguna. Akan lebih baik jika terdapat sebuah sistem yang memungkinkan pengguna mendeteksi penyakit diabetes berdasarkan data medis seperti nilai kadar gula yang dapat di cek secara mandiri. Sehingga pengguna tetap dapat menggunakan sistem tersebut tanpa harus memeriksa terlebih dahulu ke dokter dan mendapatkan hasil uji laboratorium.
3. Pengembangan lebih lanjut berupa sistem rekomendasi makanan yang dapat dikonsumsi ataupun yang tidak dapat dikonsumsi oleh pengguna berdasarkan komposisi atau kandungan gula yang terdapat pada makanan.
4. Pengembangan lebih lanjut berupa aplikasi android yang mobile-friendly untuk menjangkau pengguna smartphone dan meningkatkan kemudahan akses , dan juga dapat menambahkan fitur edukasi dan konsultasi dengan dokter.



## DAFTAR PUSTAKA

- Addo, P. M., Guegan, D., & Hassani, B. (2018). Credit risk analysis using machine and deep learning models. *Risks*, 6(2). <https://doi.org/10.3390/risks6020038>
- And, I., & Expert, D. (2022). Prediksi Penyakit Diabetes Menggunakan Algoritma Support Vector Machine (SVM) INFORMASI ARTIKEL ABSTRAK (Vol. 4, Nomor 1). <https://e-journal.unper.ac.id/index.php/informatics>
- Beghriche, T., Djerioui, M., Brik, Y., Attallah, B., & Belhaouari, S. B. (2021). An Efficient Prediction System for Diabetes Disease Based on Deep Neural Network. *Complexity*, 2021. <https://doi.org/10.1155/2021/6053824>
- Batista, G. E. A. P. A., Prati, R. C., & Monard, M. C. (2004). A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data. *SIGKDD Explor. Newsl.*, 6(1), 20–29. <https://doi.org/10.1145/1007730.1007735>
- Candra Permana, B. A., & Dewi Patwari, I. K. (2021). Komparasi Metode Klasifikasi Data Mining Decision Tree dan Naïve Bayes Untuk Prediksi Penyakit Diabetes. *Infotek : Jurnal Informatika dan Teknologi*, 4(1), 63–69. <https://doi.org/10.29408/jit.v4i1.2994>
- El-Attar, N. E., Moustafa, B. M., & Awad, W. A. (2022). Deep learning model to detect diabetes mellitus based on dna sequence. *Intelligent Automation and Soft Computing*, 31(1), 325–338. <https://doi.org/10.32604/IASC.2022.019970>
- Fajar, M., & Nugraha, Z. (2023). Deteksi Aritmia Menggunakan Algoritma Deep Neural Network (Dnn) Pada Sinyal Elektrokardiogram (Vol. 10, Nomor 5).
- Handoko, W. T., Supriyanto, E., Purwadi, D. I., Budiarmo, Z., & Listiyono, H. (2022). Klasifikasi Opini Pengguna Media Sosial Twitter Terhadap JNT Di Indonesia dengan Algoritma Decision Tree. Dalam *Jurnal Sains Komputer & Informatika (J-SAKTI)* (Vol. 6, Nomor 2).
- Hari Rachmawanto, E. (2022). Convolutional Neural Network (CNN) untuk Klasifikasi Citra Penyakit Diabetes Retinopathy. *SKANIKA: Sistem*

- Komputer dan Teknik Informatika, 5(2), 167–176.
- Goyal, S. (2022). Handling Class-Imbalance with KNN (Neighbourhood) Under-Sampling for Software Defect Prediction. *Artificial Intelligence Review*, 55(3), 2023–2064. <https://doi.org/10.1007/s10462-021-10044-w>
- Hendrawan, N. D., Saivul Affandi, A., & Fadhilrifat, R. F. (2023). Implementasi Support Vector Machine dalam Deteksi Diabetes Melalui Indikator Kesehatan. 664–671. <https://jurnal.usahidsolo.ac.id/>
- Josi, A., Akuntansi, K., Prabumulih, S., Patra No, J. L., Sukaraja, K., & Selatan, K. P. (2017). STMIK-MUSIRAWAS LUBUKLINGGAU 50 PENERAPAN METODE PROTOTIPING DALAM PEMBANGUNAN WEBSITE DESA (STUDI KASUS DESA SUGIHAN KECAMATAN RAMBANG). Dalam JTI (Vol. 9, Nomor 1).
- Jurnal, H., Aprilian Prastianing Huda, P., & Akbar Riadi, A. (2021). JURNAL MANAJEMEN INFORMATIKA KLASIFIKASI PENYAKIT TANAMAN PADA DAUN APEL DAN ANGGUR MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORKS. JUMIKA, 8(1).
- Kannadasan, K., Edla, D. R., & Kuppili, V. (2019). Type 2 diabetes data classification using stacked autoencoders in deep neural networks. *Clinical Epidemiology and Global Health*, 7(4), 530–535. <https://doi.org/10.1016/j.cegh.2018.12.004>
- Klasifikasi Kondisi Pernapasan Berdasarkan Sinyal Radar Menggunakan Convolutional Neural Network. (t.t.).
- Mohsen, H., El-Dahshan, E.-S. A., El-Horbaty, E.-S. M., & Salem, A.-B. M. (2018). Classification using deep learning neural networks for brain tumors. *Future Computing and Informatics Journal*, 3(1), 68–71. <https://doi.org/10.1016/j.fcij.2017.12.001>
- Monika Parapat, I., & Tanzil Furqon, M. (2018). Penerapan Metode Support Vector Machine (SVM) Pada Klasifikasi Penyimpangan Tumbuh Kembang Anak (Vol. 2, Nomor 10). <http://j-ptiik.ub.ac.id>
- Munarto, R., & Darma, A. (2021). Klasifikasi Gender dan Usia Berdasarkan Citra Wajah Manusia Menggunakan Convolutional Neural Network. *Setrum : Sistem Kendali-Tenaga-elektronika-telekomunikasi-komputer*, 10(2).

<https://doi.org/10.36055/setrum.v10i2.12991>

- Nora Marlim, Y., Suryati, L., & Agustina, N. (2022). Deteksi Dini Penyakit Diabetes Menggunakan Machine Learning dengan Algoritma Logistic Regression. Dalam Jurnal Nasional Teknik Elektro dan Teknologi Informasi | (Vol. 11, Nomor 2).
- Purbolaksono, M. D., Irvan Tantowi, M., Imam Hidayat, A., & Adiwijaya, A. (2021). Perbandingan Support Vector Machine dan Modified Balanced Random Forest dalam Deteksi Pasien Penyakit Diabetes. Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi), 5(2), 393–399. <https://doi.org/10.29207/resti.v5i2.3008>
- Puspitasari, A. M., Ratnawati, D. E., & Widodo, A. W. (2018). Klasifikasi Penyakit Gigi Dan Mulut Menggunakan Metode Support Vector Machine (Vol. 2, Nomor 2). <http://j-ptiik.ub.ac.id>
- Putra, A. (t.t.-a). Paralelisasi Klasifikasi Data Ekspresi Gen Kanker dengan Algoritma Deep Neural Network Menggunakan Stacked Sparse Autoencoder.
- Rasywir, E., Sinaga, R., & Pratama, Y. (2020a). JURNAL MEDIA INFORMATIKA BUDIDARMA Evaluasi Pembangunan Sistem Pakar Penyakit Tanaman Sawit dengan Metode Deep Neural Network (DNN). 4, 1206–1215. <https://doi.org/10.30865/mib.v4i4.2518>
- Sisodia, D., & Sisodia, D. S. (2018). Prediction of Diabetes using Classification Algorithms. Procedia Computer Science, 132, 1578–1585. <https://doi.org/10.1016/j.procs.2018.05.122>
- Sri Rahmadhani, U., & Lysbetti Marpaung, N. (2023a). Klasifikasi Jamur Berdasarkan Genus Dengan Menggunakan Metode CNN. 8(2).
- Sukma Hani, D., & Indah Ratnasari, C. (2023). JURNAL MEDIA INFORMATIKA BUDIDARMA Klasifikasi Masalah Pada Komunitas Marah-Marah di Twitter Menggunakan Long Short-Term Memory. 7, 1829–1837. <https://doi.org/10.30865/mib.v7i4.6755>
- Susilowati, E., Kania Sabariah, M., & Akbar Gozali, A. (t.t.). IMPLEMENTASI METODE SUPPORT VECTOR MACHINE UNTUK MELAKUKAN

# KLASIFIKASI KEMACETAN LALU LINTAS PADA TWITTER IMPLEMENTATION SUPPORT VECTOR MACHINE METHOD FOR TRAFFIC JAM CLASSIFICATION ON TWITTER.

- Syafudin, S., Agastya Nugraha, R., Handayani, K., Gata, W., & Linawati, S. (2021). Prediksi Status Pinjaman Bank dengan Deep Learning Neural Network (DNN). *Jurnal Teknik Komputer AMIK BSI*, 7(2). <https://doi.org/10.31294/jtk.v4i2>
- Şengül, G., Karakaya, M., Misra, S., Abayomi-Alli, O. O., & Damaševičius, R. (2022). Deep Learning Based Fall Detection Using Smartwatches for Healthcare Applications. *Biomedical Signal Processing and Control*, 71, 103242. <https://doi.org/10.1016/j.bspc.2021.103242>
- Tineges, R., Triayudi, A., & Sholihati, I. D. (2020). Analisis Sentimen Terhadap Layanan Indihome Berdasarkan Twitter Dengan Metode Klasifikasi Support Vector Machine (SVM). *JURNAL MEDIA INFORMATIKA BUDIDARMA*, 4(3), 650. <https://doi.org/10.30865/mib.v4i3.2181>
- Tugas, J., Fakultas Informatika, A., Afif, M., Fawwaz, A., Ramadhani, K. N., & Sthevanie, F. (2020). Klasifikasi Ras pada Kucing menggunakan Algoritma Convolutional Neural Network(CNN).
- Wasil, M. (2022). Pengaruh Epoch pada Akurasi menggunakan Convolutional Neural Network untuk Klasifikasi fashion dan Furniture. *Jurnal Informatika dan Teknologi*, 5(1), 53. <https://doi.org/10.29408/jit.v5i1.4393>
- Wei, G., Mu, W., Song, Y., & Dou, J. (2022). An Improved and Random Synthetic Minority Oversampling Technique for Imbalanced Data. *Knowledge-Based Systems*, 248, 108839. <https://doi.org/10.1016/j.knosys.2022.108839>
- Xin, L. K., & Rashid, N. binti A. (2021). Prediction of Depression Among Women Using Random Oversampling and Random Forest. *2021 International Conference of Women in Data Science at Taif University (WiDSTaif)*, 1–5. <https://doi.org/10.1109/WiDSTaif52235.2021.9430215>
- Yathiraju, N. (2022). Investigating the use of an Artificial Intelligence Model in an ERP Cloud-Based System. *International Journal of Electrical, Electronics and Engineering*