

IMPLEMENTASI ALGORITMA NEURAL NETWORK UNTUK PREDIKSI PENYAKIT DIABETES



NAMA : RIZKYANDY WIBOWO

NIM : 191011402578

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS PAMULANG
KOTA TANGERANG SELATAN
2023/2024**

IMPLEMENTASI ALGORITMA NEURAL NETWORK UNTUK PREDIKSI PENYAKIT DIABETES

SKRIPSI

**Diajukan Untuk Melengkapi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer**



Oleh:

RIZKYANDY WIBOWO

191011400278

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS PAMULANG
2024**

LEMBAR PERNYATAN

Yang bertanda tangan dibawah ini:

Nama : RIZKYANDY WIBOWO
NIM : 191011400278
Program Studi : Teknik Informatika
Fakultas : Ilmu Komputer
Jenjang Pendidikan : Strata 1

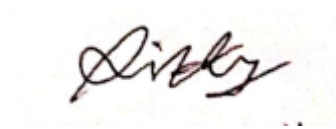
Menyatakan bahwa skripsi yang saya buat dengan judul:

IMPLEMENTASI ALGORITMA NEURAL NETWORK UNTUK PREDIKSI
PENYAKIT DIABETES

1. Merupakan hasil karya tulis ilmiah sendiri, bukan merupakan karya yang pernah diajukan untuk memperoleh gelar akademik oleh pihak lain, dan bukan merupakan hasil plagiat.
2. Saya ijin untuk dikelola oleh Universitas Pamulang sesuai dengan norma hukum dan etika yang berlaku.

Pernyataan ini saya buat dengan penuh tanggung jawab dan saya bersedia menerima konsekuensi apapun sesuai aturan yang berlaku apabila di kemudian hari pernyataan ini tidak benar.

Tangerang Selatan, November 2024



(Rizkyandy Wibowo)

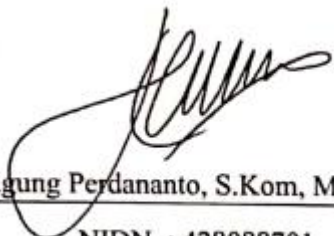
LEMBAR PERSETUJUAN

NIM : 191011400278
Nama : RIZKYANDY WIBOWO
Program Studi : TEKNIK INFORMATIKA
Fakultas : ILMU KOMPUTER
Jenjang Pendidikan : STRATA 1
Judul Skripsi : IMPLEMENTASI ALGORITMA NEURAL
NETWORK UNTUK PREDIKSI
PENYAKIT DIABETES

Skripsi ini telah diperiksa dan disetujui oleh pembimbing untuk persyaratan sidang skripsi

Tangerang Selatan, November 2024


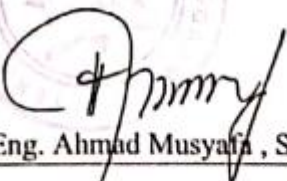
Pembimbing



Agung Perdananto, S.Kom, M.Kom.

NIDN : 428088701

Mengetahui,
Ketua Program Studi



Dr.Eng. Ahmad Musyafa, S.Kom, M.Kom.

NIDN : 0425018609

KATA PENGANTAR

Puji syukur Alhamdulillah ke hadirat Allah SWT yang telah melimpahkan segala rahmat dan karunia-Nya. Akhirnya, penyusunan skripsi yang berjudul "IMPLEMENTASI ALGORITMA *NEURAL NETWORK* UNTUK PREDIKSI PENYAKIT DIABETES" dapat diselesaikan. Skripsi ini merupakan salah satu persyaratan untuk menyelesaikan program studi strata satu (S1) pada program studi Teknik Informatika di Universitas Pamulang.

Dengan segala keterbatasan, penulis menyadari bahwa skripsi ini tidak akan terwujud tanpa bantuan, bimbingan, dan kerjasama dari berbagai pihak. Penulis juga menyadari bahwa skripsi ini masih jauh dari kata sempurna. Oleh karena itu, kritik dan saran akan senantiasa penulis terima dengan lapang hati. Dengan segala kerendahan hati, penulis menyampaikan ucapan terima kasih dan penghargaan atas keberhasilan dan keselamatan selama menempuh pendidikan kepada:

DAFTAR ISI

IMPLEMENTASI ALGORITMA NEURAL NETWORK UNTUK PREDIKSI PENYAKIT DIABETES	1
IMPLEMENTASI ALGORITMA NEURAL NETWORK UNTUK PREDIKSI PENYAKIT DIABETES	2
LEMBAR PERNYATAN.....	iii
LEMBAR PERSETUJUAN.....	iv
KATA PENGANTAR.....	v
DAFTAR ISI.....	vi
DAFTAR GAMBAR.....	x
DAFTAR TABEL	xi
ABSTRACT.....	xii
ABSTRAK	xiii
BAB I.....	1
1.1 Latar Belakang	1
1.2 Batasan Masalah.....	3
1.3 Rumusan Masalah	4
1.4 Tujuan Penelitian.....	4
1.5 Manfaat Penelitian.....	4
1.6 Metodologi Penelitian	5
1.7 Sistematika Penulisan.....	6
BAB II	7
2.1 Tinjauan Pustaka	7
2.2 Landasan Teori.....	10
2.2.1 <i>Data Mining</i>	10
2.2.2 <i>Machine Learning</i>	12
2.2.3 <i>Python</i>	12

2.2.4 <i>Library Python</i>	14
2.2.5 Javascript	16
2.2.6 Neural Network	16
2.2.7 Pengujian Model.....	19
2.2.8 Framework.....	21
2.2.9 Aplikasi Pendukung.....	21
2.2.10 Kerangka Pemikiran	22
2.3 Tinjauan Objek.....	23
2.3.1 Diabetes Melitius.....	23
BAB III	24
3.1 Metode.....	24
3.2 <i>Flowchart</i> Sistem	24
3.3 Analisa Kebutuhan	25
3.3.1 Kebutuhan Fungsional.....	25
3.3.2 Analisa Kebutuhan Non-Fungsional	27
3.3.3 Pengumpulan Data(Dataset)	28
3.3.4 Pengolahan Data Awal	29
3.4 Pearancangan Penelitian.....	30
3.4.1 Membangun Model.....	30
3.4.2 Pengujian Model.....	31
3.4.3 Evaluasi & Validasi Hasil.....	32
3.4.4 Pembuatan Database.....	32
3.4.5 Pembuatan Server Backend	33
3.4.6 Pembuatan Frontend UI.....	33
3.5 <i>Sequence</i> Diagaram	33
3.5.1 <i>Sequence</i> Diagram <i>Record</i> /Prediksi	35
3.5.2 <i>Sequence</i> Diagram <i>History Login</i>	36

3.5.3 <i>Sequence Diagram History Record/Prediksi</i>	36
3.6 Desain User Interface (UI/UX)	37
3.6.1 Halaman Awal:	37
3.6.2 Halaman Login	38
3.6.3 Halaman Dashboard	38
3.6.4 Halaman Add Record atau Record Page	39
3.6.5 Halaman History Record	40
3.6.6 Halaman History Login	41
3.6.7 Halaman Feedback	42
3.6.8 Halaman Healty Recipe	43
3.6.9 Halaman About.....	43
BAB IV	45
4.1 Hasil	45
4.1.1 Persiapan Data	45
4.2 Pembahasan.....	45
4.2.1 Pengolahan Data	45
4.2.2 Eksplorasi Data.....	45
4.2.3 Membangun Model.....	50
4.2.4 Pengujian Model.....	51
4.3 Implementasi Sistem	52
4.4 Membangun Web App	53
4.4.1 Integrasi Model <i>Neural Network</i>	53
4.4.2 Penanganan Input dan Prediksi.....	54
4.4.3 Pengujian Aplikasi.....	54
4.4.4 Analisis dan Evaluasi.....	77
BAB V	79
KESIMPULAN DAN SARAN	79

5.1	Kesimpulan.....	79
5.2	Saran.....	79
DAFTAR PUSTAKA		81

DAFTAR GAMBAR

Gambar 3. 1 Rumus <i>Neural Network</i>	24
Gambar 3. 2 <i>Flowchart</i> Sistem	25
Gambar 3. 3 Rancangan Penelitian	30
Gambar 3. 4 Desain Database	32
Gambar 3. 5 Sequence Diagram Login.....	34
Gambar 3. 6 Sequence Diagram Record/Prediksi.....	35
Gambar 3. 7 Sequence Diagram History Login.....	36
Gambar 3. 8 Sequence Diagram History Record/Prediksi.....	36
Gambar 3. 9 Halaman Awal.....	37
Gambar 3. 10 Halaman Login.....	38
Gambar 3. 11 Halaman Dashboard.....	39
Gambar 3. 12 Halaman Add Record atau Record Page.....	40
Gambar 3. 13 Halaman History Record.....	41
Gambar 3. 14 Halaman History Login.....	41
Gambar 3. 15 Halaman Feedback.....	42
Gambar 3. 16 Halaman Blog	43
Gambar 3. 17 Halaman About	44
 Gambar 4. 1 Tabel Login/User	 52
Gambar 4. 2 Tabel <i>Record</i> /Prediksi.....	53
Gambar 4. 3 Tabel User Session.....	53
Gambar 4. 4 Tabel Feedback	53
Gambar 4. 5 Diagram Pemrosesan.....	54
Gambar 4. 6 Whitebox Testing Login	56
Gambar 4. 7 Whitebox Testing Add Record/Prediksi	59
Gambar 4. 8 Whitebox Testing Forgot /Change Password.....	60
Gambar 4. 9 Whitebox Testing History Login	62
Gambar 4. 10 Whitebox Testing History Record	63
Gambar 4. 11 SQL Injection.....	75
Gambar 4. 12 Owasp-Zap	76
Gambar 4. 13 Burp-Suite	77

DAFTAR TABEL

Tabel 3. 1 Analisa Kebutuhan Non-Fungsional.....	28
Tabel 3. 2 Atribut dan Deskripsi.....	29
Tabel 3. 3 Atribut dan Tipe Data	29
Tabel 3. 4 Evaluasi & Validasi Hasil.....	32
Tabel 4. 1 Persiapan Data	45
Tabel 4. 2 Tabel Tipe Data	46
Tabel 4. 3 Tabel Class.....	46
Tabel 4. 4 Atribut Pregnancies.....	47
Tabel 4. 5 Atribut Glucose	47
Tabel 4. 6 Atribut Blood Pressure	47
Tabel 4. 7 Atribut Skin Thickness	48
Tabel 4. 8 Atribut Insulin.....	48
Tabel 4. 9 Atribut BMI	48
Tabel 4. 10 Atribut Diabetes Pedigree Function.....	49
Tabel 4. 11 Atribut Age	49
Tabel 4. 12 Statistik Data.....	50
Tabel 4. 13 Dataset Dan Jumlah Data.....	50
Tabel 4. 14 Split Data	50
Tabel 4. 15 Training Data	51
Tabel 4. 16 Hasil dari model ANN	51
Tabel 4. 17 Black-box Testing Login	66
Tabel 4. 18 Black-box Testing Add Record/Prediksi	69
Tabel 4. 19 Black-box Testing Forgot /Change Password	72
Tabel 4. 20 Black-box Testing History Login	73
Tabel 4. 21 Black-box Testing History Record	75

ABSTRACT

Early detection of diabetes mellitus is crucial to prevent serious complications. Machine learning technology has been widely used in various fields, including medicine, to enhance the accuracy and efficiency of disease diagnosis. This thesis aims to develop a diabetes prediction model using a neural network algorithm. The dataset used is the Pima Indians diabetes dataset, which includes various health features such as age, body mass index, and blood glucose levels. The neural network model is built with multiple layers and trained using the backpropagation algorithm.

The results of the study show that the developed model has an accuracy rate of 78.9% in detecting diabetes. This success demonstrates that neural networks can be effectively used as an auxiliary tool in diabetes diagnosis, although further testing with more diverse and real-world data is needed. These findings are expected to serve as a foundation for the development of more advanced and applicable health prediction systems in the future (SA, JJV, & H, 2020).

ABSTRAK

Deteksi dini diabetes mellitus sangat penting untuk mencegah komplikasi serius. Teknologi machine learning telah digunakan secara luas dalam berbagai bidang, termasuk kedokteran, untuk meningkatkan akurasi dan efisiensi diagnosis penyakit. Skripsi ini bertujuan untuk mengembangkan model prediksi diabetes menggunakan algoritma neural network. Dataset yang digunakan adalah dataset diabetes dari Pima Indians yang mencakup berbagai fitur kesehatan seperti usia, indeks massa tubuh, dan kadar glukosa darah. Model neural network dibangun dengan beberapa lapisan dan dilatih menggunakan algoritma backpropagation.

Hasil penelitian menunjukkan bahwa model yang dikembangkan memiliki tingkat akurasi sebesar 78.9% dalam mendeteksi diabetes. Keberhasilan ini menunjukkan bahwa neural network dapat digunakan sebagai alat bantu yang efektif dalam diagnosis diabetes, meskipun masih diperlukan pengujian lebih lanjut dengan data yang lebih beragam dan nyata. Temuan ini diharapkan dapat menjadi dasar bagi pengembangan sistem prediksi kesehatan yang lebih canggih dan aplikatif di masa mendatang (SA, JJV, & H, 2020).

BAB I

PENDAHULUAN

1.1 Latar Belakang

Diabetes merupakan suatu penyakit tidak menular yang cukup serius di mana pankreas tidak dapat memproduksi insulin secara maksimal. Diabetes dapat menyerang siapa saja tanpa mengenal usia baik lansia, orang dewasa, maupun anak-anak yang ditandai dengan meningkatnya kadar gula (glukosa) darah dalam tubuh manusia. Diabetes dapat disebabkan oleh banyak faktor seperti tekanan darah tinggi, kadar gula berlebih, berat badan, Riwayat keturunan diabetes, usia, jumlah kehamilan seseorang, ketebalan lipatan kulit, jumlah kadar insulin dalam tubuh, kurangnya aktivitas fisik dan pola hidup, serta diet tidak sehat. Faktor-faktor tersebut merupakan variabel yang digunakan dalam penelitian ini untuk membuat sistem cerdas yang dapat memprediksi penyakit diabetes (Cahyani & dkk, 2022)

Strategi klasifikasi digunakan secara luas di bidang medis untuk mengklasifikasikan data ke dalam kelas yang berbeda menurut beberapa kendala yang secara komparatif merupakan pengklasifikasi individu. Diabetes mellitus (DM), menurut definisi World Health Organization (WHO), adalah penyakit degeneratif kronis yang disebabkan oleh produksi insulin yang tidak mencukupi di pankreas atau oleh ketidakmampuan tubuh untuk secara efektif menggunakan insulin yang diproduksi, mengambil hyperglycemia (peningkatan glukosa darah) sebagai indikator utama. Karena gejalanya yang mirip dengan kondisi sakit biasa, banyak orang yang tidak menyadari bahwa mereka mengidap penyakit diabetes dan bahkan sudah mengarah pada komplikasi. Untuk memastikan bahwa seseorang apakah mengidap diabetes atau tidak maka perlu diagnosis dokter melalui cek darah. Bagi orang awam, setidaknya harus mengenal beberapa gejala yang biasanya mengiringi penyakit diabetes ini seperti, sering buang air kecil, mudah merasa haus, mudah merasa lapar, turunnya berat badan secara drastis, kulit kering, penyembuhan luka relatif lama, dan adanya gangguan penglihatan. Hampir setengah dari semua penderita diabetes memiliki faktor keturunan, yang merupakan salah satu ciri terpenting DM. Salah satu machine learning yang dapat membantu dalam mendeteksi resiko diabetes adalah neural

network. Neural Network adalah sebuah cabang dari kecerdasan buatan (artificial intelligence) yang cara kerjanya meniru cara kerja syaraf-syaraf otak manusia. Dengan cara ini, Neural Network memberikan program komputer sebuah kemampuan untuk bisa mengenali pola dan menyelesaikan berbagai masalah (Apriliah & dkk, 2021).

Decision Tree adalah algoritma dalam pembelajaran mesin yang digunakan untuk pengambilan keputusan dan prediksi. Pohon (*tree*) adalah sebuah struktur data yang terdiri dari simpul (*node*) dan rusuk (*edge*). Simpul pada sebuah pohon dibedakan menjadi tiga, yaitu simpul akar (*root node*), simpul percabangan/ internal (*branch/ internal node*) dan simpul daun (*leaf node*) (Eska, 2016). Kekurangan algoritma *Decision Tree* cenderung mempelajari pola yang sangat spesifik pada data pelatihan, yang dapat mengakibatkan overfitting pada dataset tersebut. Ini berarti model akan berkinerja buruk pada data baru yang belum pernah dilihat sebelumnya. *Decision Tree* sensitif terhadap perubahan kecil dalam data pelatihan, sedikit perubahan dalam data dapat menghasilkan struktur pohon yang berbeda, yang pada gilirannya dapat menghasilkan hasil yang berbeda.

Naive Bayes merupakan salah satu algoritma metode pengklasifikasian suatu probabilitas dan statistik yang diperoleh Thomas Bayes seorang ilmuwan Inggris dengan cara melakukan prediksi peluang di masa depan berdasarkan pengalaman pada masa sebelumnya (Bustami, 2014). Kelemahan pada algoritma ini yaitu sensitivitas terhadap data latih, jika jumlah data latih sangat terbatas atau terdapat *outlier* yang signifikan, dapat membuat estimasi probabilitas dapat menjadi tidak akurat. Algoritma *Naive Bayes* cenderung hanya memerhatikan hubungan statistik antara fitur-fitur dalam data latih, tanpa memperhatikan konteks atau makna di balik data tersebut. Ini dapat mengurangi kemampuan algoritma untuk memahami aspek kontekstual dari data.

Support Vector Machine (SVM) adalah sistem pembelajaran yang menggunakan ruang hipotesis berupa fungsi-fungsi linier dalam sebuah ruang fitur (*feature space*) berdimensi tinggi, dilatih dengan algoritma pembelajaran yang didasarkan pada teori optimasi dengan mengimplementasikan *learning bias* yang berasal dari teori pembelajaran statistik (Munawarah, Soesanto, & Faisal, 2016). Kekurangan dari metode SVM adalah sulit diaplikasikan untuk dataset dengan jumlah dimensi yang sangat besar (Aulia, Aprianti, Supriyanto, & Rozikin, 2022).

K-Nearest Neighbor (KNN) merupakan metode klasifikasi terhadap sekumpulan data berdasarkan pembelajaran data yang sudah terklasifikasikan sebelumnya (Admojo &

Ahsanawati, 2020). Kekurangan pada metode ini yaitu KNN harus menyimpan semua data *training* dalam memori dan melakukan perhitungan jarak dengan semua data tersebut, algoritma ini mungkin menjadi kurang efisien pada dataset yang sangat besar.

Neural Network adalah sebuah model komputasi yang terinspirasi oleh struktur dan fungsi jaringan saraf biologis dalam otak manusia, *neural network* mengandung elemen pemrosesan dan pembobotan yang saling terhubung. Setiap lapisan dalam jaringan berisi oleh kelompok elemen pemrosesan (Hadianto, Novitasari, & Rahmawati, 2019), kelebihan dari metode ini yaitu mampu mengenali dan memahami pola-pola yang sangat kompleks dan abstrak dalam data, melalui lapisan-lapisan dan hubungan antara unit-unit pemrosesan, *neural network* dapat memahami representasi data dalam berbagai tingkat abstraksi dan hirarki. Ini berarti mereka dapat menangkap fitur-fitur yang semakin kompleks dan informasi yang semakin berguna pada setiap lapisan

Deteksi dini diabetes sangat penting untuk mencegah komplikasi serius. Salah satu metode yang dapat digunakan untuk mendeteksi diabetes secara akurat adalah neural network, sebuah teknik dalam machine learning yang memiliki kemampuan untuk menganalisis dan memprediksi data dengan tingkat akurasi yang tinggi. Neural network telah berhasil diterapkan di berbagai bidang, termasuk kesehatan, untuk deteksi penyakit, analisis gambar medis, dan prediksi hasil pengobatan. Selain itu, pengembangan aplikasi web untuk implementasi model ini akan memudahkan tenaga medis dalam mengakses dan menggunakan alat prediksi secara real-time.

1.2 Batasan Masalah

Untuk memperjelas arah permasalahan yang akan dibahas dari rumusan masalah diatas, maka batasan masalah pada penelitian ini hanya membahas sebagai berikut:

- a) Pada penelitian ini data yang digunakan merupakan data dari penelitian sebelumnya, yaitu dataset yang diambil pada situs website (<https://www.kaggle.com/datasets/iammustafatz/diabetes-prediction-dataset>).
- b) Implementasi algoritma neural network untuk prediksi diabetes.
- c) Pengembangan aplikasi web untuk implementasi model neural network.
- d) Pengujian model berdasarkan metrik akurasi, presisi, dan recall.

1.3 Rumusan Masalah

- a) Bagaimana proses pelatihan dan pengujian model neural network menggunakan dataset diabetes?
- b) Bagaimana mengembangkan aplikasi web untuk implementasi model neural network dalam prediksi diabetes?
- c) Seberapa efektif model neural network dalam memprediksi diabetes dibandingkan dengan metode tradisional?

1.4 Tujuan Penelitian

- a) Mengembangkan model neural network yang dapat mendeteksi diabetes dengan akurasi tinggi.
- b) Melakukan pelatihan dan pengujian model menggunakan dataset yang relevan.
- c) Mengembangkan aplikasi web yang memungkinkan akses dan penggunaan model neural network secara real-time oleh tenaga medis.
- d) Mengevaluasi kinerja model neural network dalam mendeteksi diabetes dibandingkan dengan metode tradisional.

1.5 Manfaat Penelitian

Adap beberapa manfaat penelitian yang dibuat oleh peneliti antara lain :

a. Peningkatan Akurasi Deteksi Diabetes:

Penelitian ini bertujuan untuk mengembangkan model neural network yang mampu mendeteksi diabetes dengan tingkat akurasi yang tinggi. Dengan akurasi yang lebih baik, diharapkan proses diagnosa dini dapat dilakukan dengan lebih tepat, sehingga dapat membantu tenaga medis dalam mengambil keputusan klinis yang lebih baik dan cepat.

b. Pengembangan Aplikasi Web untuk Tenaga Medis:

Aplikasi web yang dikembangkan dalam penelitian ini memudahkan tenaga medis untuk mengakses dan menggunakan model prediksi diabetes secara real-time. Aplikasi ini dirancang untuk user-friendly, sehingga dapat dioperasikan dengan mudah oleh tenaga

medis, yang pada akhirnya diharapkan dapat meningkatkan efisiensi dalam proses diagnosa dan perawatan pasien.

c. Kontribusi terhadap Pengembangan Teknologi di Bidang Kesehatan:

Hasil dari penelitian ini diharapkan dapat memberikan kontribusi signifikan terhadap penerapan teknologi machine learning dalam bidang kesehatan, khususnya dalam deteksi penyakit. Penelitian ini juga dapat menjadi referensi bagi penelitian-penelitian selanjutnya yang ingin mengembangkan atau mengaplikasikan teknologi serupa untuk berbagai jenis penyakit lainnya.

1.6 Metodologi Penelitian

Metodologi yang digunakan pada penelitian ini melalui beberapa tahapan ,yaitu :

a) Pengembangan Model Neural Network:

Model neural network dikembangkan menggunakan framework machine learning seperti TensorFlow atau Keras. Tahapan pengembangan meliputi pemilihan arsitektur jaringan, konfigurasi layer, dan penyesuaian hyperparameter untuk mengoptimalkan performa model.

b) Pelatihan dan Pengujian Model:

Model neural network dilatih menggunakan dataset diabetes yang telah dibagi menjadi data pelatihan dan data pengujian. Proses pelatihan melibatkan pengaturan parameter training dan pemantauan performa model. Setelah itu, model diuji dengan data pengujian untuk mengevaluasi akurasi, presisi, recall, dan F1-score.

c) Pengembangan dan Implementasi Aplikasi Web:

Aplikasi web dikembangkan untuk mengintegrasikan model neural network. Tahapan pengembangan mencakup desain frontend yang user-friendly, pengembangan backend untuk mendukung operasi model, dan integrasi model neural network ke dalam aplikasi web. Aplikasi ini kemudian diuji coba untuk memastikan fungsionalitas dan keandalannya, serta dievaluasi berdasarkan umpan balik dari pengguna dan hasil prediksi.

1.7 Sistematika Penulisan

Untuk memudahkan dalam proses penyusunan tugas akhir, penulis menyusun ke dalam lima bab. Setiap bab tersebut secara keseluruhan saling berkaitan satu sama lain, dimana diawali dengan bab pendahuluan dan diakhiri dengan bab penutup yang berupa kesimpulan dan saran. Maka dibuat suatu sistematika penulisan sebagai berikut :

BAB I PENDAHULUAN

Bab ini berisi penjelasan secara sistematis mengenai landasan topik penelitian yang meliputi latar belakang, identifikasi masalah, rumusan masalah dan batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan yang terakhir sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Bab ini menguraikan tentang penelitian sebelumnya yang pernah dilakukan dan landasan teori penelitian mengenai data *mining*, *Machine Learning*, dan metode Algoritma *Neural Network* yang berkaitan langsung dengan penelitian ini.

BAB III METODE PENELITIAN

Bab ini menjelaskan secara sistematis, bagaimana proses penelitian dilakukan. Penjelasan pada bab ini meliputi Analisa kebutuhan sistem dan perancangan penelitian.

BAB IV IMPLEMENTASI DAN PENGUJIAN

Bab ini menjelaskan hasil pengujian yang dilakukan serta analisis dari data yang diperoleh dari hasil pengujian yang dilakukan.

BAB V KESIMPULAN DAN SARAN

Bab ini menjelaskan hasil akhir dari penelitian ini dan saran untuk penelitian yang akan datang.

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian yang dilakukan oleh Kamrul hasan dkk. pada tahun 2020 dengan judul penelitian “*Diabetes Prediction Using Ensembling of Different Machine Learning Classifiers*” metode yang digunakan pada penelitian ini adalah *k-nearest neighbors*. Sedangkan masalah yang dihadapi pada penelitian ini adalah meningkatkan prediksi diabetes di mana bobot diperkirakan dari Kurva Area Di Bawah ROC (AUC) model ML yang sesuai. Adapun tujuan penelitian ini adalah dapat menghasilkan kinerja yang lebih baik dalam prediksi diabetes. Adapun hasil dari penelitian ini didapatkan hasil tingkat akurasi sebesar 95% (Hasan, Alam, Hussein, & Hasan, 2020).

Pada penelitian yang dibuat oleh Mitushi soni pada tahun 2020 dengan judul penelitian “*Diabetes Prediction using Machine Learning Techniques*” metode yang dipakai pada penelitian ini adalah *Random Forest*. Sedangkan masalah yang dihadapi pada penelitian ini adalah mencari model *machine learning* manakah yang paling akurat untuk system deteksi diabetes. Adapun tujuan penelitian ini adalah untuk menyelidiki model yang dapat memprediksi diabetes dengan akurasi yang lebih baik. Pendekatan yang diusulkan menggunakan berbagai metode pembelajaran klasifikasi dan ansambel yang menggunakan pengklasifikasi SVM, Knn, Random Forest, Decision Tree, Logistic Regression dan Gradient Boosting. Dan akurasi klasifikasi telah tercapai sebesar 77% (Soni & Varma, 2020).

Penelitian yang dikerjakan oleh P. Nagaraj dan P. Deepalakshmi pada tahun 2021 dengan judul “*Diabetes Prediction Using Enhanced SVM and Deep Neural Network Learning Techniques: An Algorithmic Approach for Early Screening of Diabetes*”. metode yang digunakan pada penelitian ini adalah SVM dan *Deep Neural network*. Sedangkan masalah yang dialami pada penelitian tersebut adalah meningkatkan efektifitas yang lebih baik dengan *mengupgrade vector deep learning*. untuk mencapai akurasi tinggi dan waktu pemrosesan yang lebih singkat dengan kumpulan data yang sangat besar. Kemudian tujuan dari penelitian ini adalah untuk mencapai akurasi tinggi dan waktu pemrosesan yang lebih

singkat dengan kumpulan data yang sangat besar. Adapun hasil yang didapat dari eksperimen adalah akurasi klasifikasi maksimum yang diperoleh melalui ESVM dan DNN sebesar 98,45% (Nagaraj & Deepalakshmi, 2021)

Penelitaian yang dikerjakan oleh Arianna Dagliati dkk pada tahun 2018 yang berjudul “*Machine Learning Methods to Predict Diabetes Complications*” metode yang digunakan pada penelitian ini adalah *Random Forest* dan *Logistic Regression*. sedangkan masalah yang dihadapi pada penelitian berikut adalah memprediksi komplikasi diabetes dengan menggunakan metode *Random forest* dan *Logistic Regression* untuk menangani ketidak seimbangan pada class. Mengenai tujuan dari penelitian ini adalah menggabungkan data mining dan machine learning untuk memprediksi komplikasi diabetes tipe 2 dengan lebih efektif. kemudian hasil yang diperoleh dari penelitian ini adalah untuk model LR retinopati didapatkan prediksi sebesar 99% sedangkan untuk LR neuropati sebesar 95% (Dagliati, Marini, & dkk, 2018)

Penelitian yang dilakukan oleh Radhanath Patra dan Bonomali Khuntia pada tahun 2021 dengan judul “*Analysis and Prediction Of Pima Indian Diabetes Dataset Using SDKNN Classifier Technique*”. metode yang digunakan dalam penelitian ini adalah *standard deviation K-nearest neighbour*. Sedangkan masalah yang dialami pada penelitian tersebut adalah menggunakan rumus perhitungan jarak terbaru untuk menemukan nearest neighbour pada KNN. Kemudian tujuan dari penelitian ini adalah untuk mencapai akurasi yang lebih tinggi dengan menggunakan metode KNN untuk memprediksi diabetes. Mengenai hasil yang didapat dari penelitian ini adalah akurasi klasifikasi rata-rata memberikan hasil sebesar 83,2% sebuah peningkatan besar dibandingkan Teknik konvensional lainnya (Patra & Khuntia, 2021).

Penelitian yang dilakukan oleh Madhusmita Rout dan Amandeep Kaur pada tahun 2020 dengan judul “*Evaluating Machine Learning Methods for Predicting Diabetes among Female Patients in Bangladesh*”. lalu metode yang digunakan dalam penelitian ini adalah *Naïve Bayes*. Sedangkan masalah yang dialami pada penelitian tersebut adalah mengatasi perbedaan dataset antara negara yang berbeda. Kemudian tujuan dari penelitian ini adalah untuk mendeteksi diabetes pada pasien menggunakan Machine Learning. adapun hasil yang diperoleh dari penelitian tersebut adalah nilai akurasi tertinggi sebesar 81,2% dengan menggunakan metode KNN (Kaur & Rout, 2020)

Peneliti	Tahun	Topik Penelitian	Metode	Hasil Penelitian
Kamrul hasan dkk	2020	<i>Diabetes Prediction Using Ensembling of Different Machine Learning Classifiers</i>	<i>K-Nearest Neighbour, Decision Tree, Random Forest, Xgboost dan Naive Bayes</i>	hasil dari penelitian ini didapatkan hasil tingkat akurasi sebesar 95% dari model Xgboost
Mitushi soni	2020	<i>Diabetes Prediction using Machine Learning Techniques</i>	<i>Support Vector Machine, K-Nearest Neighbour, Decision Tree, Logistic Regression, Random Forest dan Gradient Boost</i>	Hasil perbandingan tertinggi dari model penelitian ini adalah Forest dengan akurasi sebesar 77%
P. Nagaraj dan P. Deepalakshmi	2021	<i>Diabetes Prediction Using Enhanced SVM and Deep Neural Network Learning Techniques: An Algorithmic Approach for Early Screening of Diabetes</i>	<i>Decision Tree, Neural Networks, Logistic Regression, Naive Bayes dan Support Vector Machine</i>	Hasil dari penelitian ini diperoleh nilai akurasi klasifikasi maksimum yang diperoleh melalui ESVM dan DNN sebesar 98,45%
Arianna Dagliati dkk	2018	<i>Machine Learning Methods to Predict Diabetes Complications</i>	<i>Support Vector Machine, Random Forest, Naive Bayes dan Logistic Regression</i>	hasil yang diperoleh dari penelitian ini adalah untuk model LR retinopati didapatkan prediksi sebesar 99% sedangkan untuk LR neuropati sebesar 95%

Radhanath Patra dan Bonomali khuntia	2021	<i>Analysis and Prediction Of Pima Indian Diabetes Dataset Using SDKNN Classifier Technique</i>	<i>K-Nearest Neighbour</i>	hasil yang didapat dari penelitian ini adalah akurasi klasifikasi rata-rata memberikan hasil sebesar 83,2% sebuah peningkatan besar dibandingkan Teknik konvensional lainnya
Madhusmita Rout dan Amandeep Kaur	2020	<i>Evaluating Machine Learning Methods for Predicting Diabetes among Female Patients in Bangladesh</i>	<i>K-Nearest Neighbour, Decision Tree, Random Forest dan Naive Bayes</i>	hasil yang diperoleh dari penelitian tersebut adalah nilai akurasi tertinggi sebesar 81,2% dengan menggunakan metode KNN

2.2 Landasan Teori

2.2.1 Data Mining

Data mining adalah analisa terhadap data untuk menemukan hubungan yang jelas serta menyimpulkannya yang belum diketahui sebelumnya dengan cara terkini dipahami dan berguna bagi pemilik data tersebut. Data mining adalah metoda yang digunakan untuk mengekstraksi informasi prediktif tersembunyi pada *database*, ini adalah teknologi yang sangat potensial bagi perusahaan yang sangat potensial bagi perusahaan dalam memberdayakan data warehouse (Noviyanto, 2020).

Data Mining adalah suatu istilah yang digunakan untuk menguraikan penemuan pengetahuan didalam database, dapat diartikan bahwa *Data Mining* digunakan untuk ekstraksi dari informasi penting yang tersembunyi dari database yang besar. Dalam beberapa tahun belakangan ini, kemajuan dalam beberapa bidang ilmu pengetahuan seperti *science*, *business* dan lain-lain, dampaknya adalah peningkatan koleksi database sehingga kumpulan data yang demikian banyak dapat didayagunakan untuk pengambilan keputusan. (Jollyta, Ramdhan, & Zarlis, 2020)

2.2.1.1 Tahap-tahap Data Mining :

Ada Tujuh Tahapan dari *data mining* sebagai berikut (Romadhon & Kodar, 2020);

- a. Pembersihan Data, Proses untuk membuang data yang tidak valid dan yang tidak sesuai untuk digunakan.

- b. Integrasi Data, Merupakan proses penggabungan sumber-sumber data.
- c. Seleksi Data, Proses pengambilan data-data yang sesuai untuk analisis.
- d. Transformasi Data, Pengubahan data menjadi bentuk yang tepat sesuai data mining.
- e. Proses Mining, Proses awal pengkajian metode.
- f. Evaluasi Data, Proses untuk mengidentifikasi pola yang menarik untuk mewakili pengetahuan dari data yang tersembunyi.
- g. Presentasi pengetahuan, merupakan proses presentasi pengetahuan dan Teknik visualisasi dapat membantu memberitahukan hasil dari data mining.

2.2.1.2 Metode Data Mining

Metode data mining merupakan suatu proses utama yang digunakan saat metode diterapkan untuk menemukan pengetahuan berharga dari data, ada beberapa teknik dan sifat analisa yang dapat digolongkan dalam *data mining* (Muslim, et al., 2019) yaitu;

a. Classification

Klasifikasi merupakan teknik yang digunakan untuk menemukan model agar dapat menjelaskan atau membedakan konsep atau kelas data, dengan tujuan untuk dapat memperkirakan kelas dari suatu objek yang labelnya tidak diketahui. Metode klasifikasi yang sering digunakan yaitu, *Support Vector Machine, Multilayer Perceptron, Naive bayes, ID3, Ensemble Methode, dll.*

b. Association

Association rule mining adalah teknik data mining yang berfokus untuk menemukan aturan kesamaan dalam suatu kejadian. Contoh aturan asosiasi yang sering dijumpai adalah proses pembelian barang dagangan pada pusat perbelanjaan. Metode asosiasi yang umum digunakan adalah *FP-Growth, Coefficient of Correlation, Chi Square, A Priori, dll.*

c. Clustering

Clustering merupakan teknik dengan cara mengelompokkan data secara otomatis tanpa diberitahukan label kelasnya. *Clustering* dapat digunakan untuk memberikan label pada kelas data yang belum diketahui, karena *clustering* sering digolongkan sebagai metode *unsupervised learning*. Metode *clustering* yang sering digunakan yaitu, *K-Medoids, K-Means, Fuzzy C-Means, Self-Organizing Map (SOM), dll.*

d. Prediction

memperkirakan suatu nilai di masa mendatang, misalnya memprediksi stok barang tiga tahun ke depan. Yang termasuk fungsi ini antara lain metode Neural Network, Decision Tree, dan k –Nearest Neighbor.

e. *Regression*

Digunakan untuk mencari pola dan menentukan nilai numerik. Metode yang sering digunakan adalah linear regression, logistic regression, support vector regression, dll.

f. *Description*

memberi gambaran secara ringkas terhadap sejumlah data yang berskala besar dan memiliki banyak jenis. Termasuk di dalamnya metode *Decision Tree*, *Exploratory Data Analysis* dan *Neural Network*.

2.2.2 Machine Learning

Machine Learning merupakan salah satu cabang ilmu Kecerdasan Buatan, khususnya yang mempelajari tentang bagaimana computer mampu belajar dari data untuk meningkatkan kecerdasannya. *Machine Learning* memiliki fokus pada pengembangan sebuah sistem yang mampu belajar sendiri untuk memutuskan sesuatu, tanpa harus berulang kali di program oleh manusia. Dengan metode tersebut, mesin tidak hanya menemukan aturan untuk perilaku optimal dalam pengambilan keputusan, namun dapat juga beradaptasi dengan perubahan yang terjadi. (Wahyono, 2018).

Machine Learning (ML) atau Mesin Pembelajaran adalah cabang dari AI yang fokus belajar dari data (*learn from data*), yaitu fokus pada pengembangan sistem yang mampu belajar secara “mandiri” tanpa harus berulang kali diprogram manusia. ML membutuhkan Data yang valid sebagai bahan belajar (ketika proses *training*) sebelum digunakan ketika *testing* untuk hasil *output* yang optimal (Cholissodin, Sutrisno, Soebroto, Hasanah, & Febiola, 2020).

2.2.3 Python

Python merupakan bahasa pemrograman tingkat tinggi (*haight level language*) yang dikembangkan Oleh Guido van Rossum pada tahun 1989 dan diperkenalkan untuk pertama kalinya pada tahun 1991. *Python* lahir atas dasar keinginan untuk mempermudah seorang programmer dalam menyelesaikan tugas-tugasnya dengan cepat. *Python* dirancang untuk

memberikan kemudahan yang sangat luar biasa kepada programmer baik dari segi efisiensi waktu, maupun kemudahan dalam pengembangan program khususnya dalam hal kompatibiliras dengan sistem (Wadi, 2015)

Python adalah bahasa pemrograman tingkat tinggi yang sering digunakan dalam berbagai bidang, termasuk pengembangan perangkat lunak, analisis data, kecerdasan buatan, pemrosesan bahasa alami, pengembangan web, dan sebagainya. *Python* memiliki sintaks yang mudah dipahami dan digunakan, serta berbagai *library* dan *framework* yang kuat yang membuatnya populer di kalangan pengembang. beberapa fitur dan kegunaan *Python* Menurut (Nelli, 2015) sebagai berikut :

- a. Sintaks yang Mudah: *Python* dirancang dengan sintaks yang mudah dibaca dan dipahami, mirip dengan bahasa Inggris. Ini membuatnya menjadi bahasa yang cocok untuk pemula dan mengurangi waktu yang dibutuhkan untuk mengembangkan kode.
- b. *Platform-Independent*: *Python* dapat dijalankan di berbagai platform, termasuk *Windows*, *macOS*, *Linux*, dan banyak lagi. Ini memungkinkan pengembang untuk membuat aplikasi yang dapat berjalan di berbagai lingkungan.
- c. Kaya *Library*: *Python* memiliki ekosistem yang kaya dengan banyak *library* dan modul yang tersedia, seperti *NumPy* untuk komputasi numerik, *pandas* untuk analisis data, *TensorFlow* dan *PyTorch* untuk kecerdasan buatan, *Django* dan *Flask* untuk pengembangan web, dan banyak lagi. *Library-library* ini membantu memperluas kemampuan *Python* dan mempercepat proses pengembangan.
- d. Analisis Data: *Python* sangat populer dalam analisis data dan ilmu data. Dengan *library* seperti *pandas*, *NumPy*, dan *matplotlib*, *Python* memungkinkan pengguna untuk membaca, memanipulasi, dan menganalisis data dengan mudah. Ini menjadikannya alat yang kuat untuk melakukan eksplorasi data, visualisasi, dan pembuatan model prediktif.
- e. Kecerdasan Buatan: *Python* adalah pilihan yang populer untuk pengembangan kecerdasan buatan. *Library* seperti *TensorFlow*, *Keras*, *PyTorch*, dan *scikit-learn* menyediakan alat yang kuat untuk melatih dan menerapkan model pembelajaran mesin dan jaringan saraf.
- f. Pengembangan Web: *Python* digunakan secara luas dalam pengembangan web. *Framework* populer seperti *Django* dan *Flask* menyediakan alat untuk membangun aplikasi web yang efisien dan aman. *Python* juga dapat digunakan untuk mengembangkan API, *backend*, dan bahkan *frontend* menggunakan *framework* seperti *Django REST framework* dan *Flask-RESTful*.

- g. Automasi Tugas: *Python* dapat digunakan untuk otomatisasi tugas sehari-hari. Dengan *Python*, Anda dapat menulis skrip untuk mengotomatisasi tugas rutin seperti pengolahan file, pengambilan data dari web, pemrosesan teks, dan sebagainya.

Python adalah bahasa yang serbaguna dan kuat, dengan komunitas yang besar dan aktif. Kelebihan-kelebihan di atas menjadikan *Python* pilihan yang baik untuk berbagai kebutuhan pemrograman, dari pengembangan perangkat lunak hingga analisis data dan kecerdasan buatan.

2.2.4 Library Python

2.2.4.1 Pandas

Panda bertujuan untuk menjadi blok bangunan tingkat tinggi mendasar untuk melakukan praktik, analisis data dunia nyata dengan *Python*. Selain itu, ia memiliki tujuan yang lebih luas untuk menjadi yang paling kuat dan fleksibel alat analisis / manipulasi data sumber terbuka tersedia dalam bahasa apa pun. (PyData, 2023).

2.2.4.2 Matplotlib

Matplotlib adalah *library plot Python* 2D yang menghasilkan kualitas publikasi angka dalam berbagai format *hardcopy* dan lingkungan interaktif di seluruh *Platform*, *Matplotlib* adalah *library komprehensif* untuk membuat statis, animasi, dan visualisasi interaktif dengan *Python*. *Matplotlib* membuat segalanya menjadi mudah hal-hal mudah dan sulit mungkin (Hunter, Dale, Firing, & Droettboom, 2023). *Matplotlib* adalah *library Python* yang fokus pada visualisasi data seperti membuat plot grafik. *Matplotlib* pertama kali diciptakan oleh John D. Hunter dan sekarang telah dikelola oleh tim *developer* yang besar. Awalnya *matplotlib* dirancang untuk menghasilkan plot grafik yang sesuai pada publikasi jurnal atau artikel ilmiah. *Matplotlib* dapat digunakan dalam skrip *Ipython shell-server* aplikasi web, dan beberapa *tollkit graphical user interface* (GUI) lainnya (Rohman, 2019)

2.2.4.3 NumPy

NumPy adalah proyek *open source* yang memungkinkan komputasi numerik dengan *Python*. Itu dibuat pada tahun 2005 membangun karya awal *library Numerik* dan *Numarray*. *NumPy* adalah paket dasar untuk komputasi ilmiah dengan *Python*. Ini adalah *Python library* yang menyediakan objek *array* multidimensi, berbagai turunan objek (seperti *array* dan matriks bertopeng), dan bermacam-macam rutinitas untuk operasi cepat pada *array*, termasuk matematika, logis, manipulasi bentuk, menyortir, memilih, I / O, transformasi *Fourier diskrit*,

aljabar linier dasar, operasi statistik dasar, simulasi acak dan banyak lagi. (R, Harris, Millman, & Vander, 2020).

2.2.4.4 Seaborn

Seaborn adalah sebuah pustaka *Python* yang digunakan untuk visualisasi data berbasis statistik. Pustaka ini dibangun di atas pustaka *matplotlib* dan dirancang khusus untuk membuat visualisasi data yang lebih menarik dan informatif dengan lebih sedikit kode, menurut (Waskom, 2021) *Seaborn* adalah *library* visualisasi data *Python* berdasarkan *matplotlib*. Ini menyediakan antarmuka tingkat tinggi untuk menggambar grafik statistik yang menarik dan informatif.

Seaborn menyediakan berbagai jenis plot yang telah diatur dengan baik dan memiliki gaya yang estetik secara default. Hal ini memungkinkan pengguna untuk dengan mudah membuat plot seperti scatter plot, line plot, bar plot, histogram, heatmap, dan banyak jenis plot lainnya.

2.2.4.5 Scikit-Learn

Scikit-Learn adalah modul *Python* yang mengintegrasikan banyak algoritma *machine learning*. Pustaka ini awalnya dikembangkan oleh Cournapeu pada tahun 2007, namun rilis pertama yang sebenarnya terjadi pada tahun 2010 (Nelli, Python Data Analytics, 2015). Terdapat beberapa pustaka *Python* yang menyediakan implementasi yang solid dari berbagai algoritma *machine learning*. Salah satu yang paling terkenal adalah *Scikit-Learn*, sebuah paket yang menyediakan versi efisien dari sejumlah besar algoritma umum. *Scikit-Learn* ditandai dengan API yang bersih, seragam, dan efisien, serta dokumentasi *online* yang sangat berguna dan lengkap. Keuntungan dari keseragaman ini adalah setelah Anda memahami penggunaan dasar dan sintaksis *Scikit-Learn* untuk satu jenis model, beralih ke model atau algoritma baru akan sangat mudah (VanderPlas, 2017).

2.2.4.6 Keras

Keras adalah pustaka pembelajaran mesin (*machine learning*) yang populer dan kuat yang awalnya ditulis dalam bahasa pemrograman *Python*. Pustaka ini dirancang untuk memudahkan pembuatan, pelatihan, dan evaluasi model jaringan saraf tiruan (*neural networks*) serta model pembelajaran mesin lainnya.

Keras dapat menyederhanakan *script TensorFlow* dan memberikan solusi yang jelas untuk permasalahan dalam mengembangkan model *deep learning* (Syarovy & Sutiarso,

2023). Keras banyak digunakan oleh berbagai organisasi ilmiah di seluruh dunia, termasuk CERN, NASA, dan NIH (Gulli & Pal, 2017).

2.2.4.7 Tensorflow

TensorFlow adalah platform open source end-to-end diperuntukan machine learning. Yang memiliki circle ekosistem alat, pustaka, dan sumber daya komunitas yang sangat fleksibel dan komprehensif yang memungkinkan seorang ahli ataupun pemula untuk mengembangkan dan membangun aplikasi dengan menerapkan model machine learning di platform TensorFlow. Peneliti mendorong machine learning mutakhir dan pengembang dengan mudah membangun dan menerapkan aplikasi yang didukung machine learning. TensorFlow selalu menyediakan jalur langsung ke produksi. Baik di server, perangkat edge, atau web, TensorFlow memungkinkan Anda melatih dan menerapkan model dengan mudah, apapun bahasa yang Anda gunakan (tensorflow, 2021).

2.2.5 Javascript

JavaScript adalah sebuah bahasa script dinamis yang dapat dipakai untuk membangun interaktifitas pada halamanhalaman HTML statis. Ini dilakukan dengan menamakan blok-blok kode JavaScript di hamper semua tempat pada halaman web(Siahaan & Rismon ,2020)

2.2.6 MySQL

MySQL merupakan database engine atau server database yang mendukung bahasa database SQL sebagai bahasa interaktif dalam mengelola data. MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL atau DBMS yang multithread dan multi-user (Fitri, 2020). Kegunaan bahasa SQL yaitu membangun basis data, menjalankan query terhadap basis data, melakukan penambahan, pengurangan, perubahan terhadap data yang ada (Pratama et al., 2018).

2.2.7 Neural Network

Neural Network adalah model matematika yang terinspirasi dari cara kerja jaringan saraf dalam otak manusia. Ini adalah salah satu komponen inti dalam *machine learning* dan *artificial intelligence*. *Neural Network* merupakan salah satu metode dalam data mining yang memiliki keunggulan tingkat akurasi yang lebih baik atau optimal dibandingkan dengan metode lainnya (Handayani, Riandini, & Situmorang, 2022)

Neural Network adalah sebuah metode yang terinspirasi oleh struktur jaringan syaraf otak manusia, yang dirancang untuk meniru cara otak manusia melakukan pemrosesan dan penyimpanan informasi (Ramdhan, 2019). *Neural Network* terdiri dari *node* yang

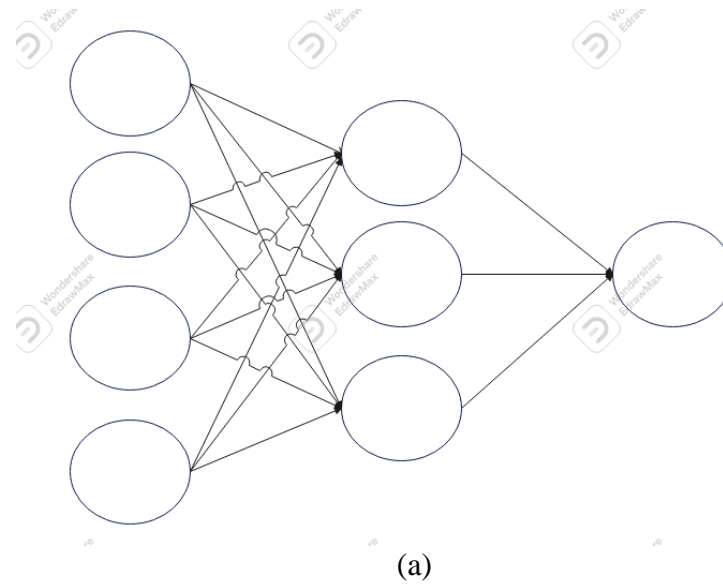
menggabungkan inputnya, yang bisa berupa variabel dari database atau output dari node lainnya. *Node-node* ini dapat dikelompokkan menjadi tiga lapisan yang sederhana, yaitu lapisan input, lapisan output, dan lapisan tersembunyi (Bhakti, 2019). Berikut adalah gambar jaringan syaraf sederhana dengan fungsi aktivasi F.

2.2.7.1 Arsitektur Jaringan

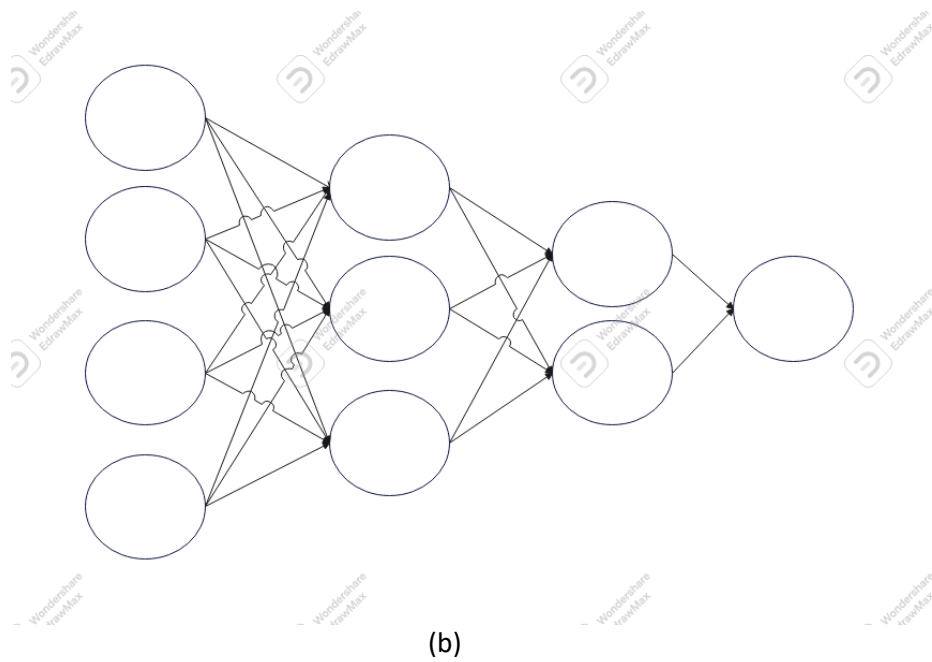
Menurut (Ataline Jeanethe Maya Hukubun, 2022) struktur Neural Network Neural Network terdiri dari 3 lapisan, yaitu :

- a. Lapisan input atau masukan (buffer) memiliki beberapa neuron input (perceptron), jumlahnya tergantung dataset yang dibutuhkan dalam proses pembelajaran. Neuron lapisan input tidak memiliki fungsi transfer, tapi ada faktor skala di setiap input untuk menormalkan sinyalnya dengan mentransmisikan sinyal yang dihitung ke lapisan tersembunyi lalu ke lapisan output.
- b. Lapisan tersembunyi berfungsi untuk menghubungkan lapisan input dan output, dengan melewati bobot yang dihitung ke lapisan keluaran. Kesalahan disajikan ke lapisan input melalui propagasi balik (umpan balik). Lapisan tersembunyi juga terdiri dari beberapa neuron tersembunyi.
- c. Lapisan output atau keluaran (buffer) berisi beberapa neuron keluaran (atau satu neuron), yang mewakili sebuah kelas dari kumpulan data keluaran.

Neural network bisa berisi satu atau lebih lapisan tersembunyi, seperti yang ditunjukkan pada gambar 1. Semua neuron di lapisan input terhubung ke neuron lapisan tersembunyi. Pemrosesan komputasi data di lapisan tersembunyi sangat kompleks dan tidak kita ketahui.



Gambar 1 Struktur jaringan syaraf tiruan : (a) berisi satu lapisan masukan, satu lapisan tersembunyi, dan satu lapisan keluaran. (b) satu lapisan masukan, dua lapisan tersembunyi, dan satu lapisan keluaran.



2.2.7.2 Artificial Neural Network

Artificial Neural Network Artificial (ANN) atau Jaringan Syaraf Tiruan merupakan sebuah teknik atau pendekatan pengolahan informasi yang terinspirasi oleh cara kerja sistem saraf biologis, khususnya pada sel otak manusia dalam memproses informasi. Elemen kunci dari teknik ini adalah struktur sistem pengolahan informasi yang bersifat unik dan beragam untuk tiap aplikasi. Neural Network terdiri dari sejumlah besar elemen pemrosesan informasi (neuron) yang saling terhubung dan bekerja bersama-sama untuk menyelesaikan sebuah masalah tertentu, yang pada umumnya adalah masalah klasifikasi ataupun prediksi (Widiputra, 2016).

2.2.8 Pengujian Model

2.2.8.1 F1-Score

F1-score adalah salah satu metrik evaluasi yang umum digunakan dalam bidang klasifikasi dalam pembelajaran mesin untuk memberikan gambaran yang lebih komprehensif tentang kinerja model dalam memprediksi kelas yang benar .

F1-score digunakan untuk mengukur kombinasi hasil *precision* dan *recall*, sehingga menjadi satu nilai pengukuran (Istighfarizky, et al., 2022). *F-1 score* membantu mengetahui kemampuan model dalam mengenali *false negative* dan *false positive* (Abdurrohman, Dini, & Muharram, 2018).

$$F1 = 2 \times \frac{precision \times recall}{precision + recall}$$

2.2.8.2 Recall

Recall adalah adalah metrik evaluasi yang digunakan dalam bidang klasifikasi untuk mengukur sejauh mana model dapat mengidentifikasi semua instance positif yang sebenarnya. *recall* menunjukkan berapa persen data kategori positif yang terklasifikasikan dengan benar oleh *system* (Istighfarizky, et al., 2022), Dalam kata lain, *recall* mengukur kemampuan model untuk menemukan atau "mengingat" semua contoh yang benar dari kelas positif. Persamaan *recall* perbandingan antara *true positive* terhadap total contoh yang benar-benar *positive* (Nasution & Hayaty, 2019):

$$Recall = \frac{TP}{TP+FN}$$

Dimana :

- a) TP yaitu *True Positive*
- b) FP yaitu *False Positive*
- c) FN yaitu *False Negative*

2.2.8.3 Akurasi

Akurasi menggambarkan seberapa akurat system dapat mengklasifikasikan data secara benar. Dengan kata lain, nilai akurasi merupakan perbandingan antara data yang terklasifikasi benar dengan keseluruhan data (Istighfarizky, et al., 2022)

$$\text{Akurasi} = \frac{TP+TN}{TP+TN+FP+FN}$$

2.2.8.4 Confussion Matrix

Confusion matrix adalah alat yang digunakan untuk menganalisis kemampuan model klasifikasi dalam mengenali berbagai *tuple* data yang berbeda. (Nasution & Hayaty, 2019). *Confusion Matrix* adalah tabel dengan 4 kombinasi berbeda dari nilai prediksi dan nilai aktual. Ada empat istilah yang merupakan representasi hasil proses klasifikasi pada confusion matrix yaitu *True Positif*, *True Negatif*, *False Positif*, dan *False Negatif* (University, 2023).

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Gambar 2. 1 Confusion Matrix

Dimana :

- a) TP (*True Positive*) Hasil memprediksi positif dan itu benar
- b) TN (*True Negative*) Hasil memprediksi negatif dan itu benar
- c) FP (*False Positive*) Hasil memprediksi positif dan itu salah
- d) FN (*False Negative*) Hasil memprediksi negatif dan itu salah

2.2.9 Framework

2.2.9.1 React Js

ReactJS adalah library JavaScript populer yang dibuat oleh Facebook yang digunakan dalam proses pengembangan aplikasi mobile dan web. untuk memfasilitasi pembuatan daripada komponen antarmuka yang interaktif, stateful, serta mudah untuk digunakan ulang. ReactJS sangatlah cocok digunakan untuk rendering antarmuka yang kompleks dengan performa tinggi (Kumar & Singh, 2016).

2.2.9.2 Express Js

Express.js adalah satu web framework paling populer di dunia Node.js. Dokumentasinya yang lengkap dan penggunaannya yang cukup mudah, dapat membuat kita mengembangkan berbagai produk seperti aplikasi web ataupun RESTful API. Express.js pun dapat digunakan menjadi pijakan untuk membangun web framework yang lebih kompleks seperti, Sails.js, MEAN (MongoDB, Express.js, Angular.js, Node.js) dan MERN (MongoDB, Express.js, React.js, Node.js). Express.js dibuat oleh TJ Holowaychuk dan sekarang dikelola oleh komunitas (Muhammad Arslan, 2016).

2.2.10 Aplikasi Pendukung

2.2.10.1 Browser Google Chrome

Google Chrome adalah peramban web lintas platform yang dikembangkan oleh Google. Peramban ini pertama kali dirilis pada tahun 2008 untuk Microsoft Windows, kemudian diporting ke Android, iOS, Linux, dan macOS yang menjadikannya sebagai peramban bawaan dalam sistem operasi. [15] Peramban ini juga merupakan komponen utama Chrome OS, yang berfungsi sebagai platform untuk aplikasi web (Wikipedia, 2024).

2.2.10.2 Visual Studio Code

Visual Studio Code, juga biasa disebut VS Code, [12] adalah editor kode sumber yang dibuat oleh Microsoft dengan Electron Framework, untuk Windows, Linux dan macOS. [13] Fitur-fiturnya mencakup dukungan untuk debugging, penyorotan sintaksis, penyelesaian kode cerdas, cuplikan, pemfaktoran ulang kode, dan Git yang tertanam. Pengguna dapat mengubah tema, pintasan keyboard, preferensi, dan pemasangan ekstensi yang menambah fungsionalitas.

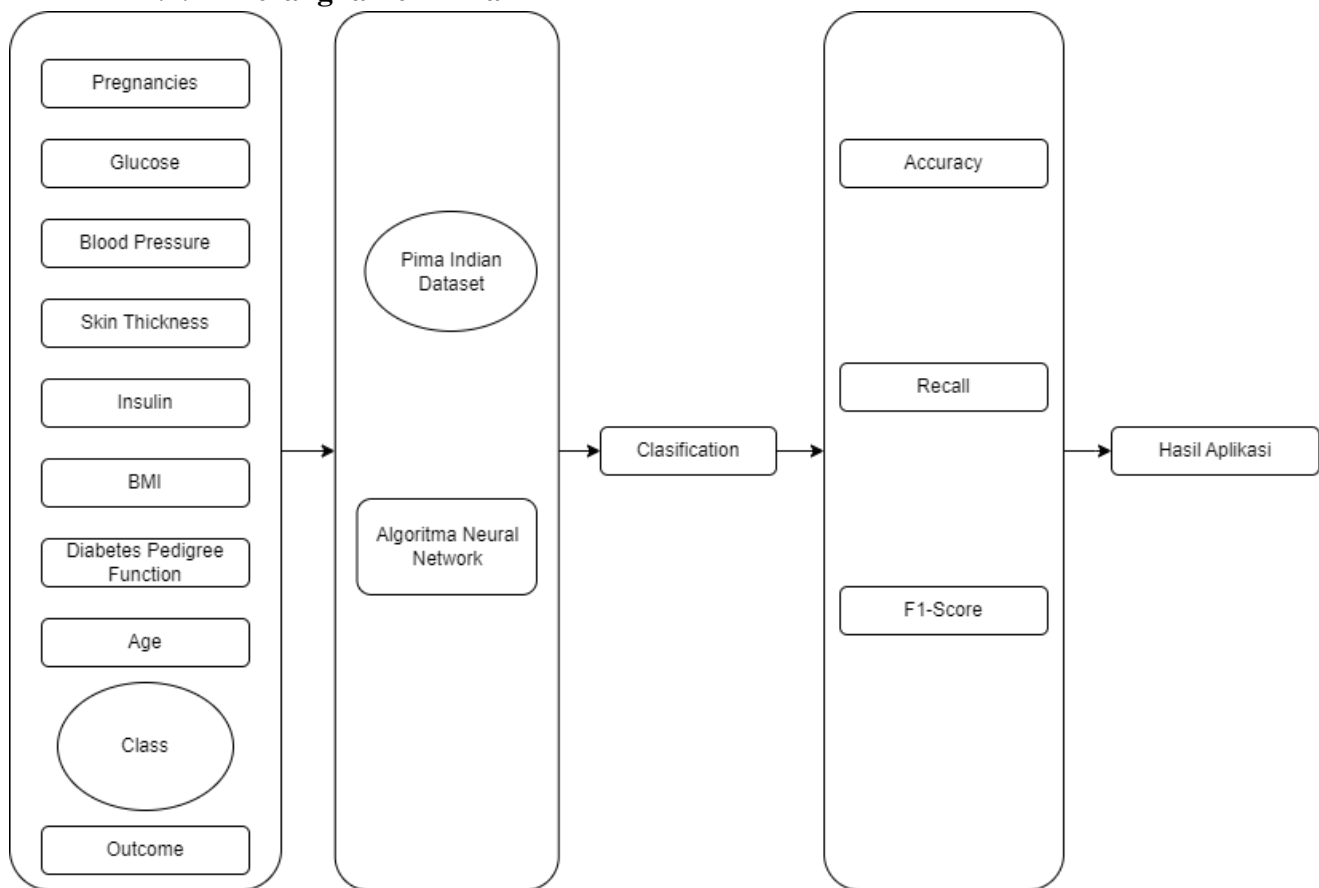
Dalam Survei Pengembang Stack Overflow 2023, Visual Studio Code menduduki peringkat alat lingkungan pengembang paling populer di antara 86.544 responden, dengan 73,71% melaporkan bahwa mereka menggunakannya. Hal ini meningkatkan

penggunaannya di kalangan mereka yang belajar coding dibandingkan dengan mereka yang sedang mengembangkan profesi(Wikipedia,2023).

2.2.10.3 Jupyter Notebook

Jupyter Notebook biasa juga di sebut *jupyter* adalah dokumen yang dapat dibagikan yang menggabungkan kode komputer, bahasa sederhana deskripsi, data, visualisasi kaya seperti model 3D, bagan, grafik, dan angka, dan kontrol interaktif. Buku catatan, bersama dengan editor (seperti *JupyterLab*), menyediakan lingkungan interaktif yang cepat untuk pembuatan prototipe dan menjelaskan kode, mengeksplorasi dan memvisualisasikan data, dan berbagi ide dengan Lain (Collonov, Dafna, Ivanov, & Eric, 2023).

2.2.11 Kerangka Pemikiran



Kerangka Pemikiran

Dalam kerangka pemiikiran diatas dijelaskan bahwa penelitian ini ditunjukan untuk mengetahui seberapa tinggi tingkat akurasi *Neural Network* dalam memprediksi orang yang beresiko terkena Diabetes melitius dan menghasilkan nilai akurasi , *F1-score*, dan *Recall* serta pengimplementasian model yang telah dibuat kedalam sebuah aplikasi.

2.3 Tinjauan Objek

2.3.1 Diabetes Melitius

Menurut Kemenkes RI (2020), menjelaskan bahwa diabetes mellitus (DM) adalah penyakit kronis atau menahun berupa gangguan metabolik yang ditandai dengan peningkatan kadar glukosa darah diatas normal. Diabetes mellitus adalah penyakit kronis yang kompleks yang membutuhkan perawatan medis berkelanjutan dengan strategi pengurangan risiko multifaktor di luar kendali glikemik (American Diabetes Association, 2018).

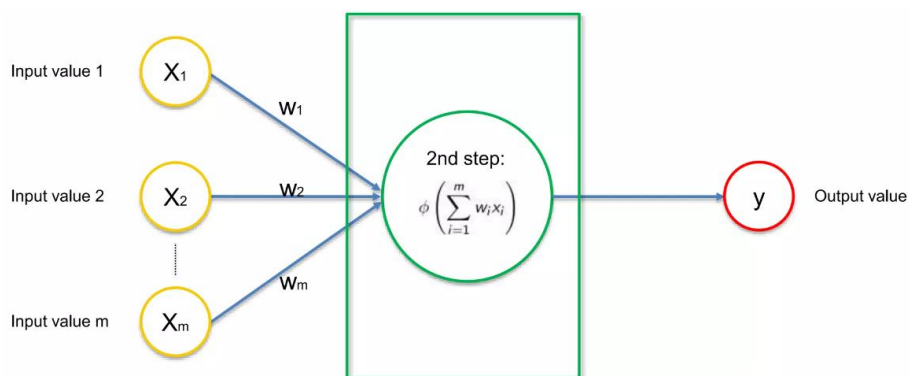
Menurut P2PTM Kemenkes RI (2020), diabetes mellitus merupakan suatu penyakit menahun yang ditandai oleh kadar glukosa darah yang melebihi nilai normal. Dimana nilai normal gula darah sewaktu (GDS) / tanpa puasa adalah < 200 mg/dl sedangkan gula darah puasa (GDP) < 126 mg/dl. Diabetes mellitus disebabkan oleh kekurangan hormon insulin yang dihasilkan oleh pankreas untuk menurunkan kadar gula darah.

BAB III

METODE PENELITIAN

3.1 Metode

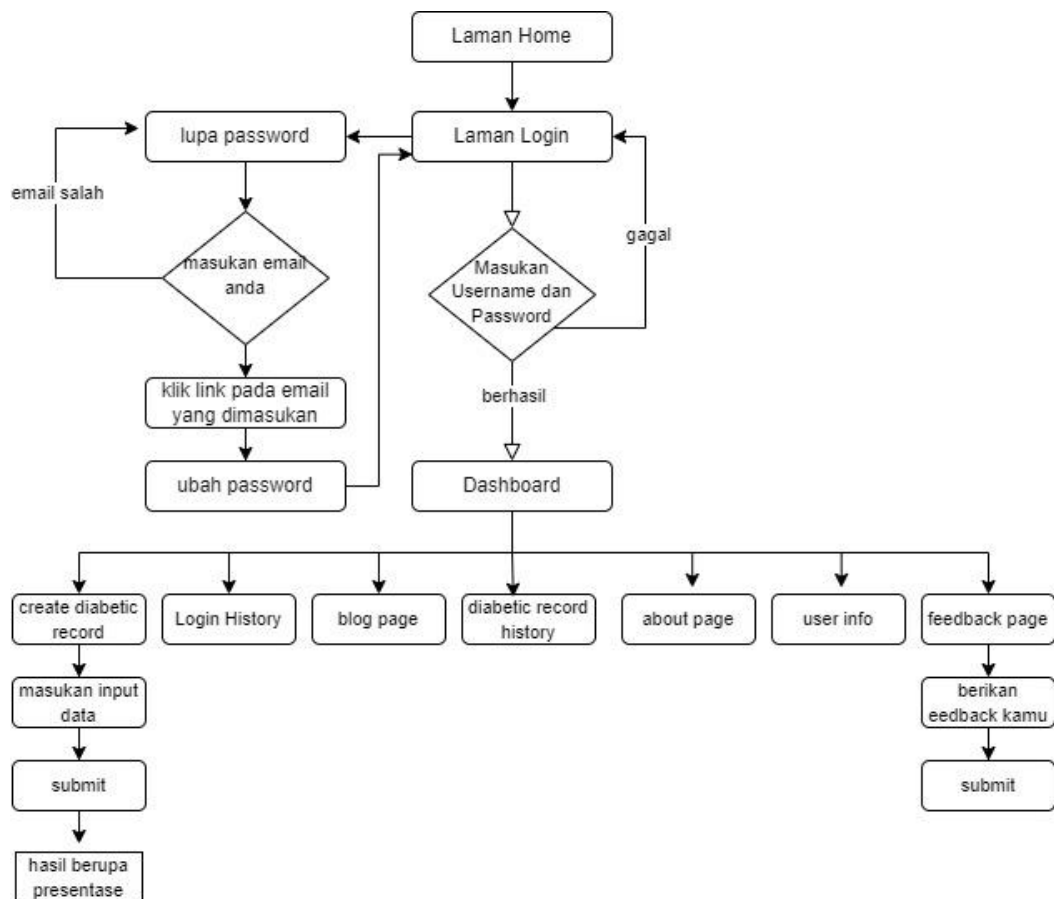
Pada penelitian ini saya menggunakan metode *Artificial Neural Network* Metode neural network prediksi, atau *Artificial Neural Network (ANN)*, adalah sebuah model yang meniru jaringan saraf biologis untuk mempelajari pola dan membuat prediksi. Metode ini bekerja dengan cara yang mirip seperti neuron biologis yang saling berinteraksi untuk mengidentifikasi pola, menganalisis opsi, dan sampai pada kesimpulan. *ANN* sangat berguna dalam menangani masalah yang kompleks dan nonlinear, karena kemampuannya untuk belajar dari data dan menyesuaikan diri dengan perubahan pola. rumus metode *Artificial Neural Network* :



Gambar 3. 1 Rumus Neural Network

3.2 Flowchart Sistem

Flowchart adalah diagram yang digunakan untuk menggambarkan alur kerja atau proses dalam sebuah aplikasi. Diagram ini menggunakan simbol-simbol grafis untuk merepresentasikan berbagai langkah atau tahapan yang terjadi dalam sistem, serta bagaimana aliran data atau proses berpindah dari satu langkah ke langkah lainnya.



Gambar 3. 2 Flowchart Sistem

3.3 Analisa Kebutuhan

Tahap analisis kebutuhan merupakan langkah penting dalam pengembangan sistem yang bertujuan untuk memahami dan mendefinisikan kebutuhan dari sistem yang akan dikembangkan. Analisis ini membantu memastikan bahwa sistem yang dibangun sesuai dengan harapan pengguna dan memenuhi tujuan utama penelitian. Kebutuhan sistem dapat dikategorikan menjadi dua jenis utama: kebutuhan fungsional dan kebutuhan non-fungsional.

3.3.1 Kebutuhan Fungsional

Kebutuhan fungsional adalah kebutuhan yang terkait langsung dengan fungsi dan fitur yang harus dimiliki oleh sistem agar dapat berjalan sesuai dengan tujuannya. Berikut adalah beberapa kebutuhan fungsional yang diidentifikasi untuk sistem deteksi diabetes menggunakan machine learning:

3.3.1.1 Penerimaan Input Data:

Sistem harus dapat menerima input data dari dataset yang telah ditentukan. Dataset ini akan digunakan sebagai basis untuk pelatihan dan pengujian model prediksi. Data yang diterima harus mencakup semua variabel yang relevan seperti usia, jenis kelamin, BMI, kadar glukosa, tekanan darah, insulin, dan faktor risiko lainnya.

3.3.1.2 Pengolahan Data Awal:

Sistem harus mampu melakukan pengolahan data awal yang mencakup pembersihan data, normalisasi data, dan pembagian dataset menjadi data pelatihan dan data pengujian. Pengolahan data awal ini penting untuk memastikan bahwa data yang digunakan dalam pelatihan model adalah data yang berkualitas dan representatif.

3.3.1.3 Pembangunan Model Prediksi:

Sistem harus mampu membangun model prediksi diabetes menggunakan algoritma Neural Network. Model ini akan dilatih menggunakan data pelatihan yang telah diproses sebelumnya dan harus mampu mengenali pola-pola dalam data yang menunjukkan adanya risiko diabetes.

3.3.1.4 Pengujian dan Evaluasi Model:

Sistem harus dapat menguji dan mengevaluasi model yang telah dibuat. Pengujian dilakukan dengan menggunakan data pengujian untuk mengukur kinerja model dalam memprediksi diabetes. Evaluasi kinerja model dilakukan dengan menggunakan metrik evaluasi seperti akurasi, presisi, recall, dan F1-score.

3.3.1.5 Antarmuka Pengguna (UI):

Sistem harus memiliki antarmuka pengguna (UI) yang memudahkan interaksi pengguna. Antarmuka ini harus intuitif dan user-friendly sehingga pengguna, termasuk mereka yang tidak memiliki latar belakang teknis, dapat dengan mudah mengoperasikan sistem dan memahami hasil prediksi yang diberikan.

3.3.1.6 Pengelolaan Koneksi Frontend dan Backend:

Sistem harus dapat mengelola koneksi antara frontend dan backend dengan baik. Frontend, yang dibangun menggunakan React.js, harus dapat berkomunikasi dengan

backend yang menggunakan Express.js. Backend ini kemudian akan berinteraksi dengan server machine learning yang menggunakan Flask untuk melakukan prediksi.

3.3.2 Analisa Kebutuhan Non-Fungsional

Kebutuhan non-fungsional adalah kebutuhan yang berkaitan dengan kualitas sistem dan bagaimana sistem tersebut beroperasi. Berikut adalah beberapa kebutuhan non-fungsional yang diidentifikasi untuk sistem ini:

a) Waktu Respons yang Cepat:

Sistem harus memiliki waktu respons yang cepat dalam memproses data. Waktu respons yang cepat penting agar pengguna tidak mengalami keterlambatan dalam mendapatkan hasil prediksi.

b) Kemudahan Penggunaan:

Sistem harus mudah digunakan dan dioperasikan oleh pengguna awam. Ini berarti bahwa antarmuka pengguna harus dirancang sedemikian rupa sehingga tidak memerlukan pelatihan khusus untuk menggunakannya. Instruksi yang jelas dan navigasi yang mudah akan membantu pengguna dalam berinteraksi dengan sistem.

Software	Spesifikasi
Sistem Operasi	Windows 10 atau setara
Aplikasi Simulator	Pemrograman dan Browser
Library Python	Pandas, Pycaret, Scikit-learn, Numpy, Tensorflow, Matplotlib, Seaborn, Keras, Tailwindcss
Library Javascript	Cors, Axios, Nodemailer, JSONwebtoken, Nodemon, Crypto, Bcrypt, MySQL2,

Framework Aplikasi	Vite React Js,Node Js, Flask, Express Js
Database	MySQL

Tabel 3. 1Analisa Kebutuhan Non-Fungsional

3.3.3 Pengumpulan Data(Dataset)

Pada penelitian Dataset yang digunakan adalah dataset yang tersedia secara public di situs Kaggle.com dan telah banyak digunakan oleh peneliti untuk mengevaluasi kinerja deteksi kemungkinan diabetes , yang memiliki atribut kolom sebagai berikut :

No	Nama Atribut	Keterangan
1	Pregnancies	Jumlah kehamilan yang pernah dialami oleh pasien.
2	Glucose	Kadar glukosa (gula) dalam darah pasien, yang diukur dalam miligram per desiliter (mg/dL).
3	BloodPressure	Tekanan darah sistolik pasien (tekanan saat jantung berkontraksi) dalam milimeter raksa (mm Hg).
4	SkinThickness	Ketebalan lipatan kulit pada trisep (lengan atas) pasien dalam milimeter.
5	Insulin	Kadar insulin dalam darah pasien.
6	BMI	Indeks massa tubuh pasien,. Ini adalah indikator umum untuk mengukur apakah seseorang memiliki berat badan yang sehat atau tidak.
7	DiabetesPedigreeFunction	Fungsi yang menggambarkan seberapa besar riwayat diabetes ada dalam keluarga pasien. Ini sering

		digunakan untuk memperkirakan risiko genetik diabetes.
8	Age	Usia pasien dalam satuan tahun.
9	Outcome	Hasil dari tes diabetes , yang mana hasilnya adalah angka 0 dan 1 jika angkanya 0 maka pasien tersebut bebas diabetes dan sebaliknya.

Tabel 3. 2 Atribut dan Deskripsi

3.3.4 Pengolahan Data Awal

Dalam dataset ini mencakup sejumlah fitur atau atribut yang digunakan untuk memprediksi apakah seseorang mengidap diabetes (Outcome). Dataset ini digunakan untuk mengembangkan model prediksi diabetes berdasarkan fitur-fitur ini.

Yang kemudian, data ini akan dibagi menjadi dua bagian: satu untuk melatih model (training data) dan yang lainnya untuk menguji model (testing data). Tujuan akhirnya adalah untuk membangun model yang dapat memprediksi apakah seseorang memiliki risiko diabetes berdasarkan atribut-atribut ini.

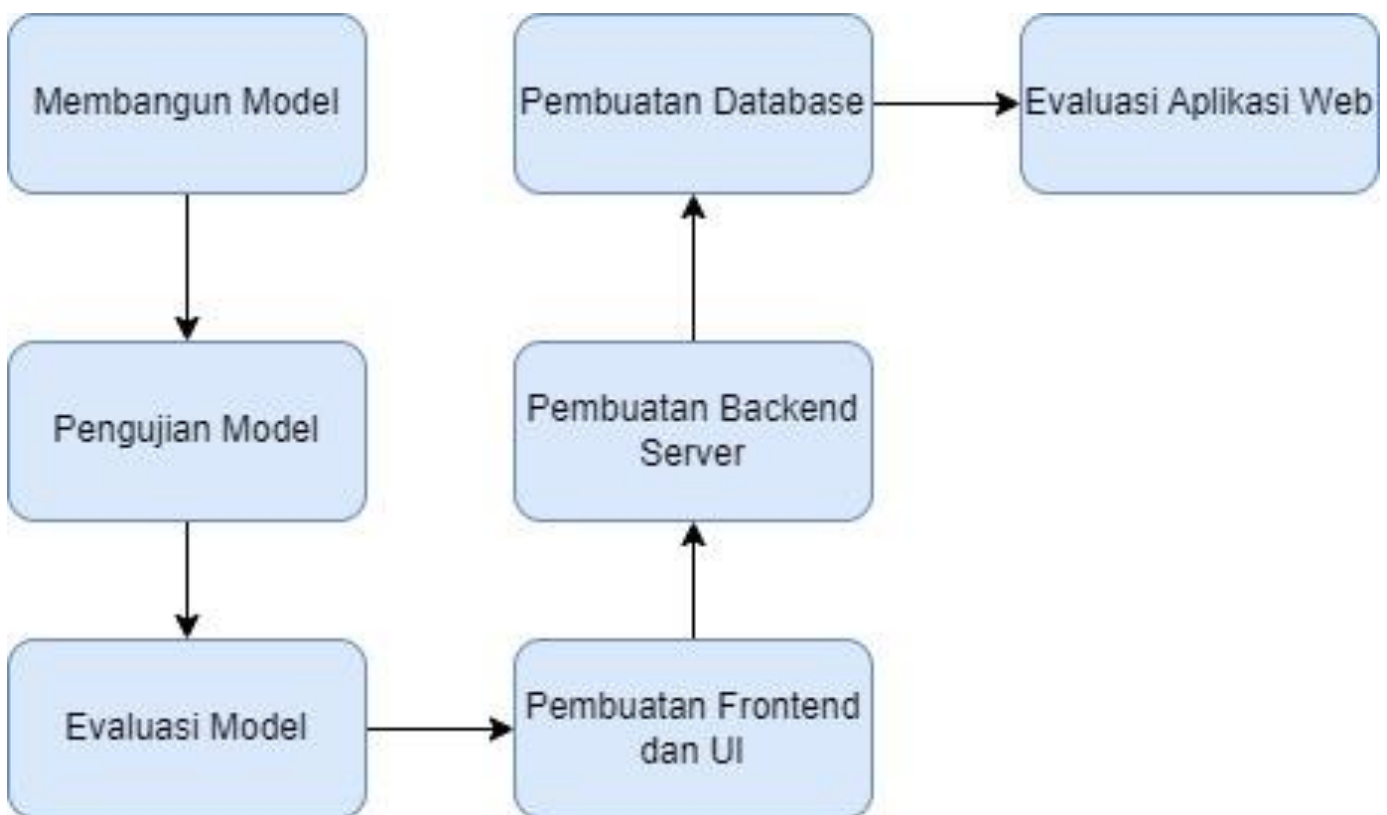
No.	Nama Atribut	Tipe Data
1	Pregnancies	Int
2	Glucose	Int
3	BloodPressure	Int
4	SkinThickness	Int
5	Insulin	Int
6	BMI	Float
7	DiabetesPedigreeFunction	Float
8	Age	Int
9	Outcome	Int

Tabel 3. 3 Atribut dan Tipe Data

3.4 Pearancangan Penelitian

Pada penelitian ini menggunakan metode Neural Network dalam mendeteksi siapa yang lebih beresiko terkena diabetes, yang dimana data yang digunakan merupakan dataset yang tersedia secara public di situs web (Kaggle .com)

Tahapan tahapan yang perlu dilakukan dalam penelitian ini adalah : karena data yang diambil sudah bersih dan semua datanya sudah sesuai kita bisa langsung Membangun model, lalu pengujian model dan terakhir evaluasi & validasi hasil Pembuatan Aplikasi Web dan Evaluasi Aplikasi seperti gambar berikut ini :



Gambar 3. 3 Rancangan Penelitian

Berdasarkan gambar diatas , dapat dijelaskan bahwa tahapan dalam penelitian ini adalah sebagai berikut :

3.4.1 Membangun Model

Selanjutnya yaitu membangun model dengan algoritma Neural Network. Ini diimplementasikan dengan menggunakan library scikit-learn, Keras, Tensorflow,

Seaborn, Matplotlib. Scikit-learn atau Sklearn adalah library berbasis Python untuk membangun model pembelajaran mesin.

3.4.2 Pengujian Model

Pengujian model merupakan langkah penting dalam mengevaluasi performa dan efektivitas model yang telah dibangun dalam pembelajaran mesin atau statistik. Tujuan dari pengujian model adalah untuk memberikan pemahaman yang jelas mengenai kemampuan model dalam melakukan prediksi atau klasifikasi data yang belum pernah dilihat sebelumnya dengan akurasi dan keandalan yang tinggi. Dalam pengujian model, terdapat beragam metrik dan teknik yang digunakan, termasuk:

1. Akurasi (*Accuracy*): Metrik ini mengukur sejauh mana model mampu mengklasifikasikan data dengan benar. Akurasi dihitung dengan membandingkan prediksi model dengan nilai sebenarnya dari data pengujian.
2. Recall (*Recall*): Juga dikenal sebagai sensitivitas, metrik ini menunjukkan sejauh mana model dapat mengidentifikasi data positif dengan baik. Ini mengukur proporsi data positif yang berhasil diklasifikasikan dengan benar dari keseluruhan data positif yang ada.
3. F1-Skor (*F1-Score*): F1-Skor merupakan ukuran yang menggabungkan presisi (*precision*) dan Recall. Metrik ini memberikan keseimbangan antara presisi dan Recall, dan seringkali digunakan ketika terdapat ketidakseimbangan antara kelas-kelas dalam dataset.
4. Matriks Confusion (*Confusion Matrix*): Matriks ini digunakan untuk menggambarkan klasifikasi yang benar dan salah yang dilakukan oleh model. Matriks ini terdiri dari empat istilah: True Positive (*TP*), True Negative (*TN*), False Positive (*FP*), dan False Negative (*FN*).

Dengan menggunakan berbagai metrik ini, pengujian model memungkinkan para peneliti dan praktisi untuk mendapatkan pemahaman yang lebih dalam tentang seberapa baik model dapat digunakan untuk memecahkan masalah yang ada.

3.4.3 Evaluasi & Validasi Hasil

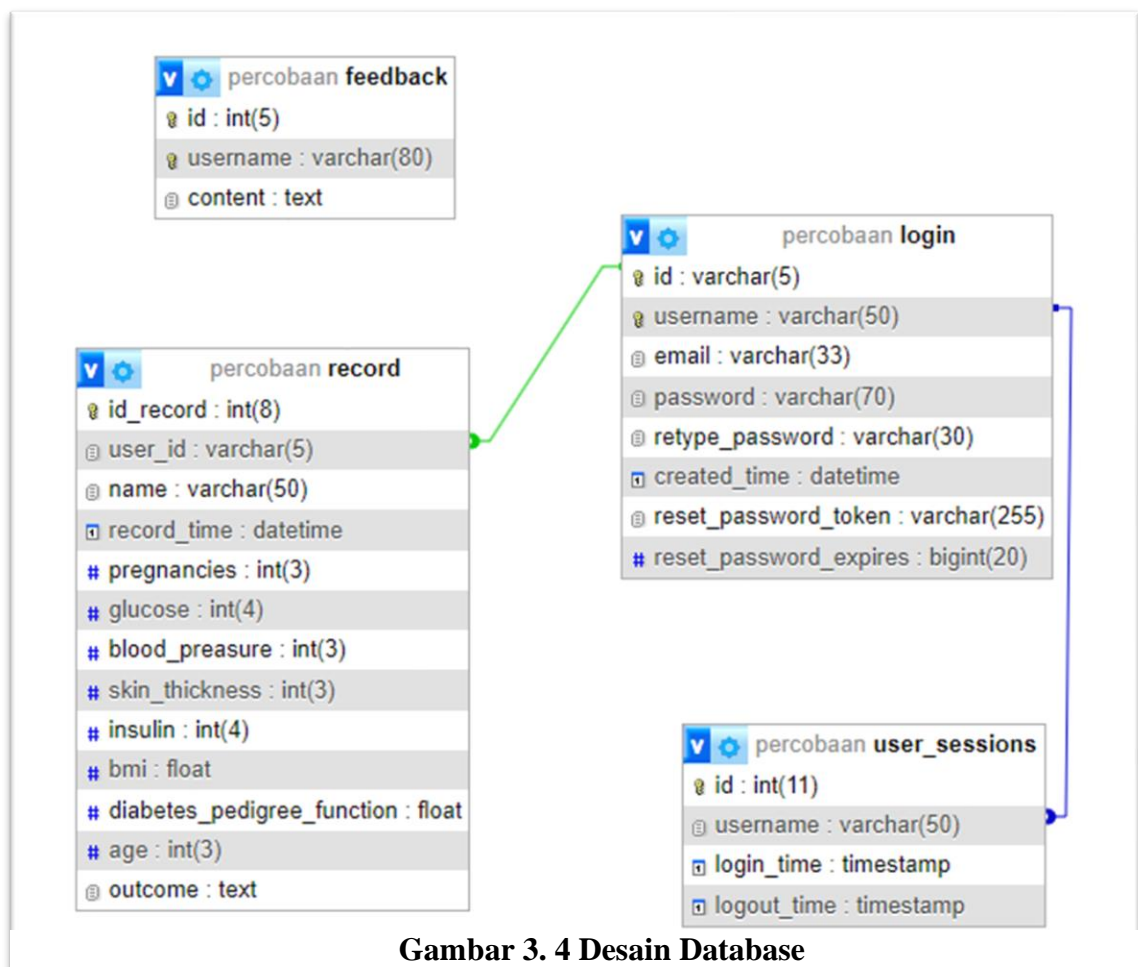
Evaluasi hasil dalam penelitian ini adalah proses pengujian model machine learning untuk deteksi diabetes yang telah dibuat dengan menggunakan pola data baru yaitu data uji 154 baris data 8 kolom. Dimana pada saat pembuatan proses model machine learning menggunakan data latih sebanyak 614 baris data 8 kolom.

<i>Dataset</i>	<i>Jumlah</i>
Data Latih	614 baris data 8 kolom
Data Uji	154 baris data 8 kolom

Tabel 3. 4 Evaluasi & Validasi Hasil

3.4.4 Pembuatan Database

Tahap ini melibatkan pembuatan dan konfigurasi database yang akan digunakan untuk menyimpan data pengguna, hasil prediksi, dan informasi lainnya yang relevan. MySQL digunakan sebagai sistem manajemen basis data, dan berikut desain *database*-nya



Gambar 3. 4 Desain Database

3.4.5 Pembuatan Server Backend

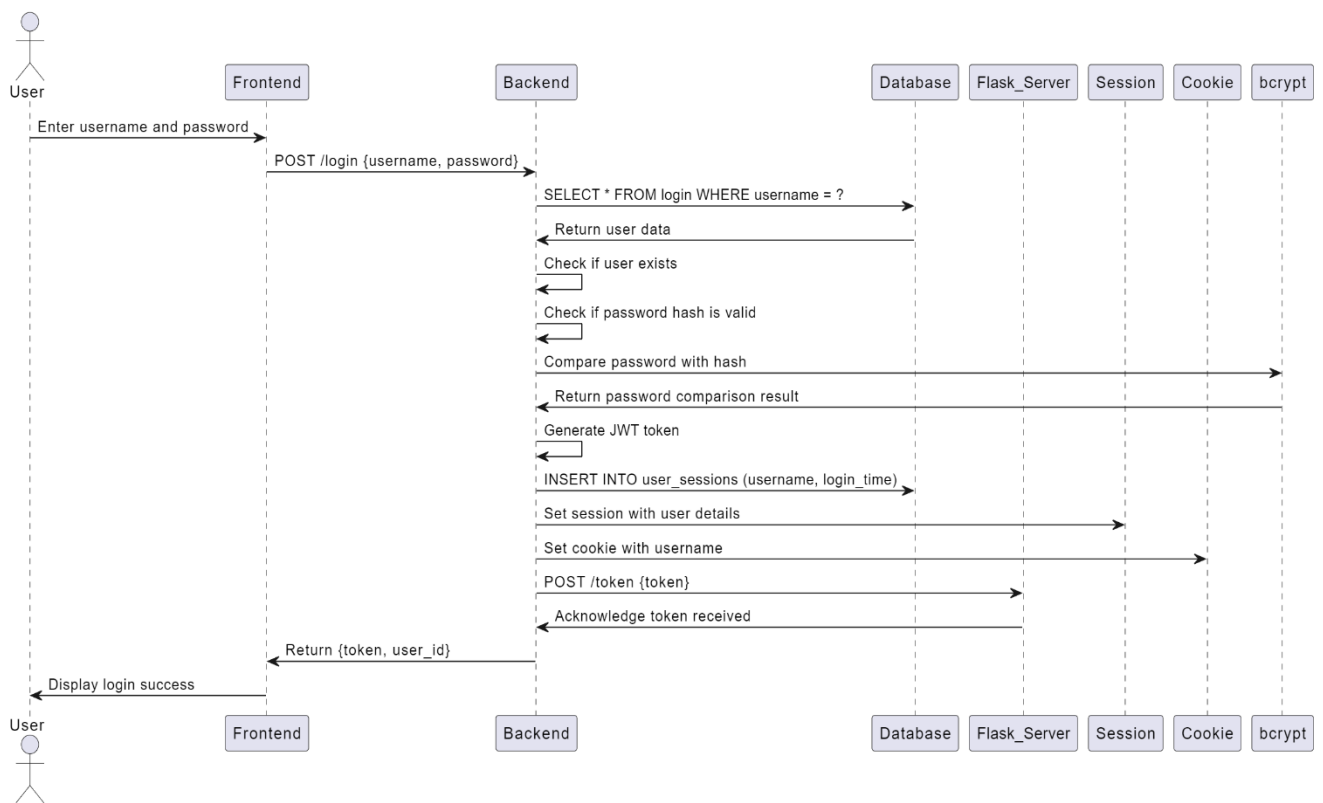
Backend server dikembangkan menggunakan *Express.js* dan *Flask* untuk menangani permintaan dari *frontend* dan berinteraksi dengan *database*. *Backend* ini juga bertanggung jawab untuk mengirimkan data medis ke server *machine learning* dan menerima data yang akan prediksi menggunakan *Flask framework* untuk memuat model dan melakukan perhitungan .

3.4.6 Pembuatan Frontend UI

Frontend dikembangkan menggunakan *React.js* untuk menyediakan antarmuka pengguna yang interaktif dan responsif. Antarmuka ini memungkinkan pengguna untuk memasukkan data medis, melihat hasil prediksi, dan mengakses fitur-fitur lainnya seperti riwayat *login*, *feedback*, dan resep sehat.

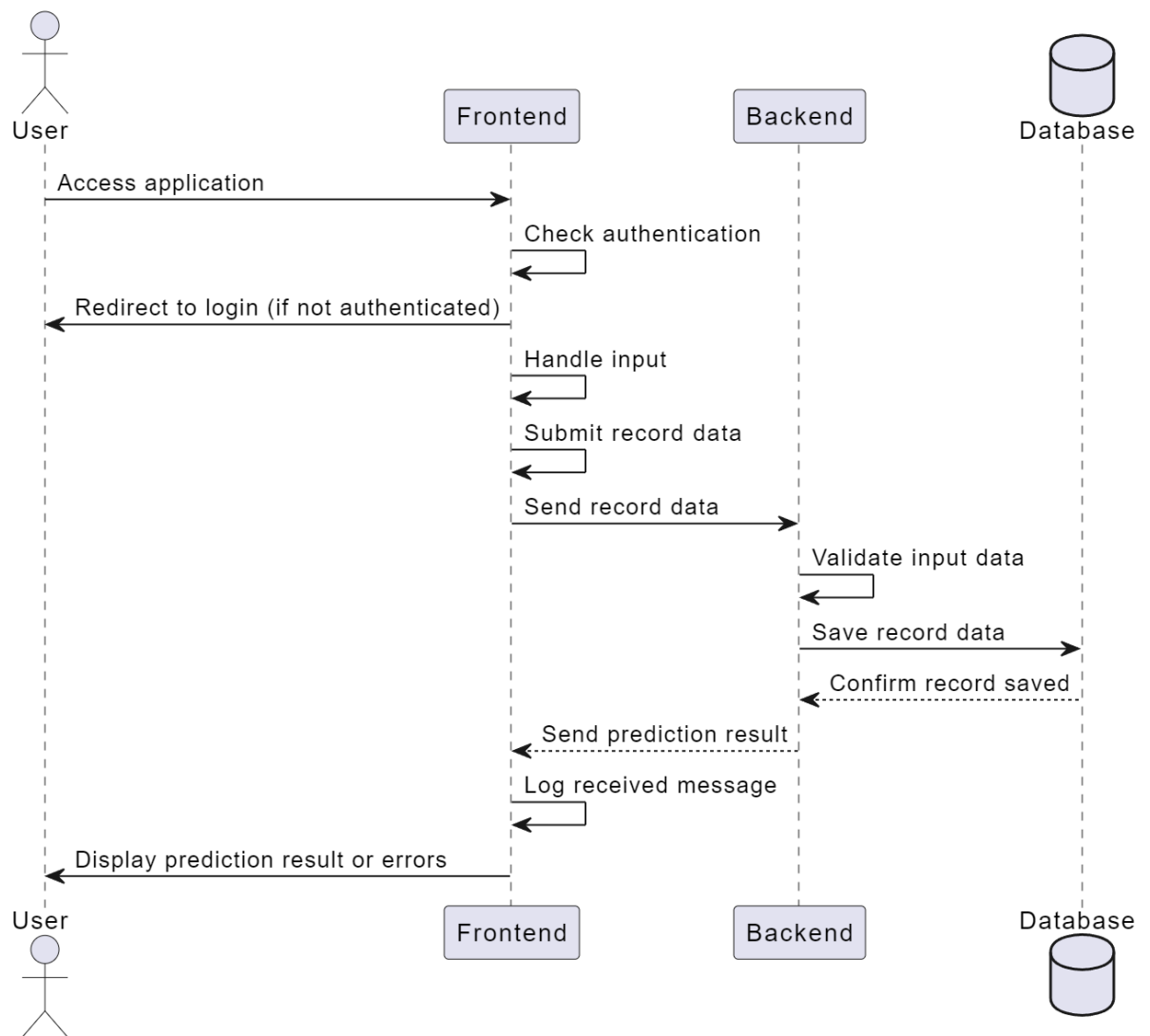
3.5 Sequence Diagram

Sequence diagram adalah alat pemodelan perangkat lunak yang digunakan untuk menggambarkan bagaimana objek-objek dalam suatu sistem berinteraksi satu sama lain berdasarkan urutan waktu. Diagram ini memperlihatkan urutan pesan yang dikirim antar objek untuk menyelesaikan suatu fungsi atau proses dalam aplikasi, seperti *login*, *logout*, atau pengambilan *database*. Dengan memvisualisasikan alur kerja dan skenario penggunaan, *sequence* diagram sangat berguna untuk memahami, mendokumentasikan, dan memfasilitasi komunikasi antar tim *Developer* dalam proses pengembangan perangkat lunak. Penggunaan *sequence* diagram membantu memastikan bahwa semua pihak memiliki pemahaman yang sama mengenai bagaimana sistem seharusnya bekerja, yang pada gilirannya meningkatkan efisiensi dan kualitas pengembangan sistem.



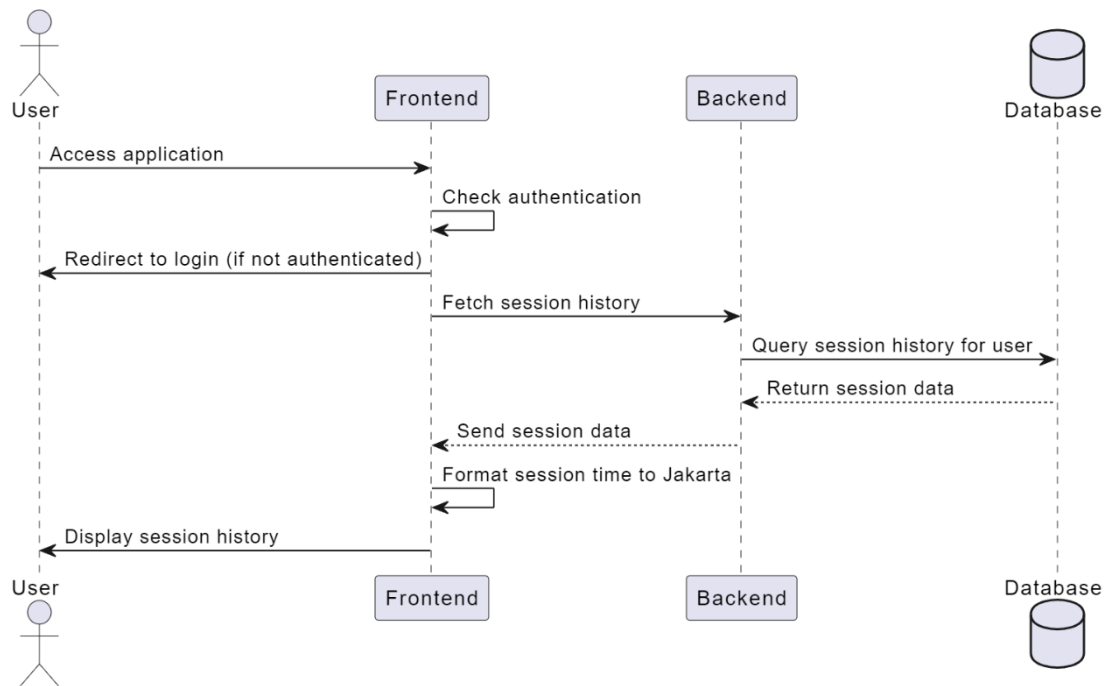
Gambar 3. 5 Sequence Diagram Login

3.5.1 Sequence Diagram Record/Prediksi



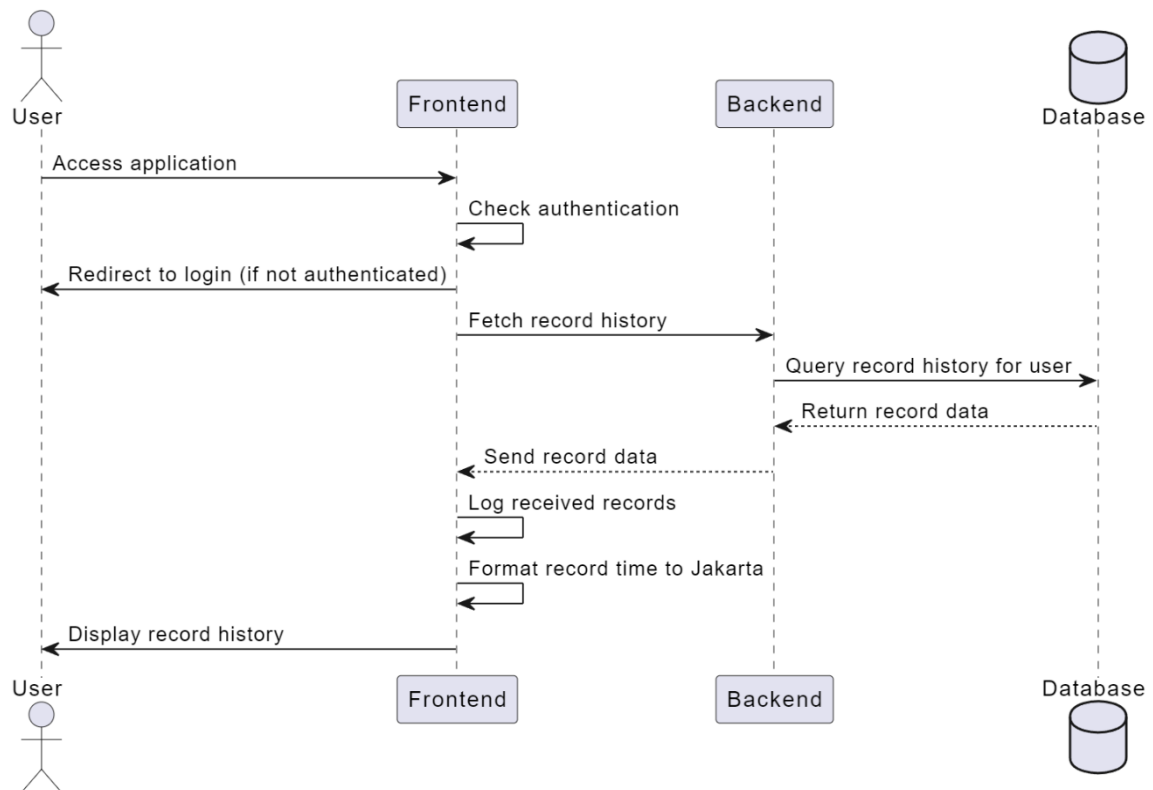
Gambar 3. 6 Sequence Diagram Record/Prediksi

3.5.2 Sequence Diagram History Login



Gambar 3. 7 Sequence Diagram History Login

3.5.3 Sequence Diagram History Record/Prediksi



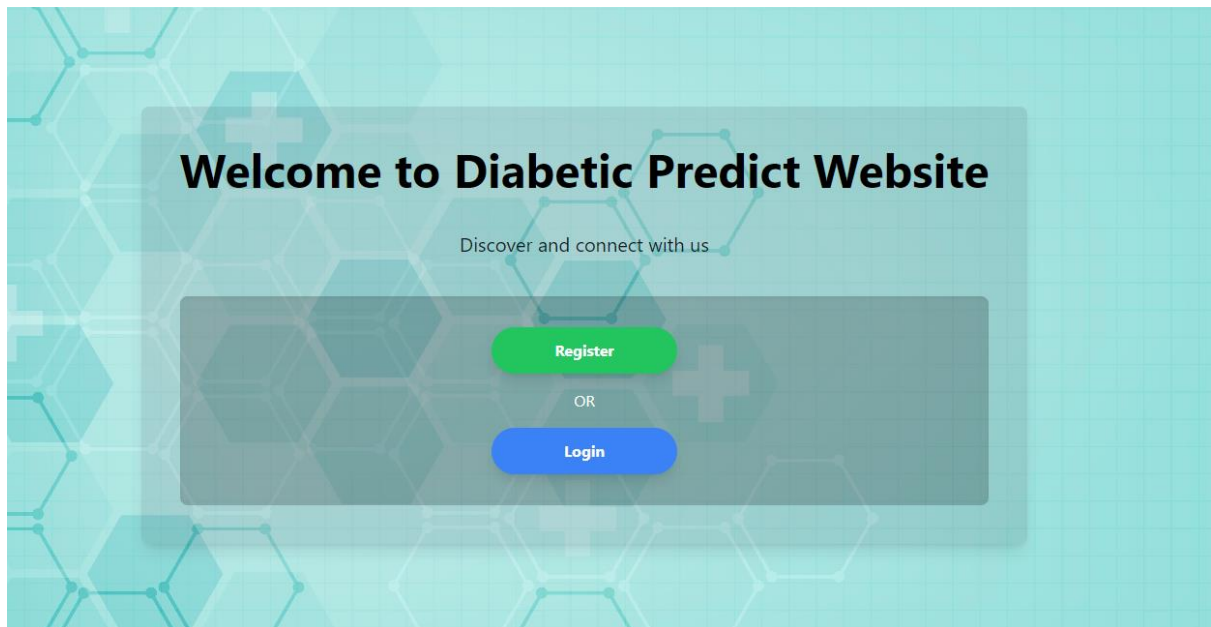
Gambar 3. 8 Sequence Diagram History Record/Prediksi

3.6 Desain User Interface (UI/UX)

Desain *User Interface (UI/UX)* aplikasi web prediksi diabetes dibuat dengan mempertimbangkan kebutuhan pengguna serta memberikan pengalaman pengguna yang sederhana dan informatif. Beberapa elemen UI yang disertakan dalam aplikasi web ini antara lain:

3.6.1 Halaman Awal:

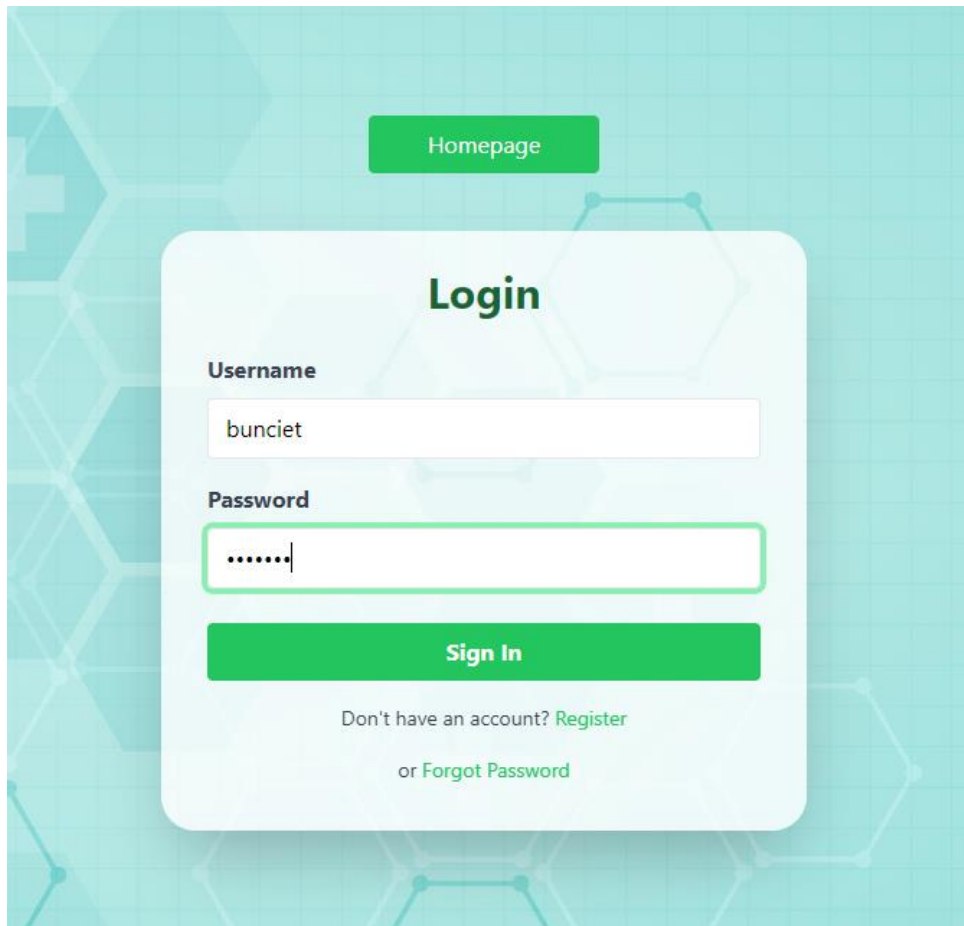
Halaman utama berisi Menu Registrasi dan Login yang mana jika user ingin menggunakan Aplikasi ini Harus memiliki akun terlebih dahulu kemudian melakukan Login untuk masuk ke Halaman Dashboard Page yang bersisi Menu yang lainnya antara lain:



Gambar 3. 9 Halaman Awal

3.6.2 Halaman Login

Halaman Login adalah halaman awal yang akan diakses oleh pengguna sebelum mereka dapat mengakses fitur-fitur lain dalam aplikasi web. Halaman ini dirancang untuk memastikan bahwa hanya pengguna yang sah yang dapat masuk dan menggunakan sistem.`

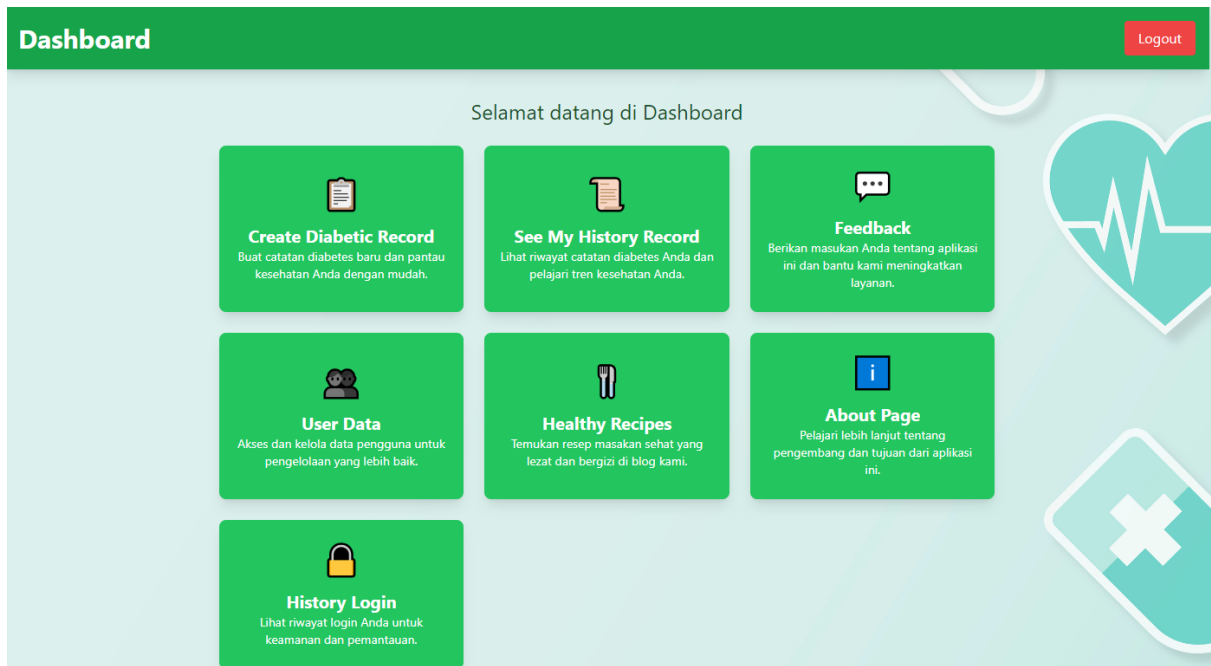


The image shows a web login interface. At the top, there is a green button labeled "Homepage". Below it is a white rounded rectangle containing the title "Login" in bold. Under the title are two input fields: "Username" with the text "bunciet" and "Password" with masked characters ".....". Below these fields is a green "Sign In" button. At the bottom of the form, there is a link "Don't have an account? Register" and a link "or Forgot Password". The background is a teal color with a faint geometric pattern.

Gambar 3. 10 Halaman Login

3.6.3 Halaman Dashboard

Halaman Dashboard adalah pusat kontrol utama dalam aplikasi web di mana pengguna dapat mengakses berbagai fitur dan fungsi yang disediakan oleh sistem. Halaman ini dirancang untuk memberikan pengguna akses yang cepat dan mudah ke berbagai menu utama dan informasi penting.



Gambar 3. 11 Halaman Dashboard

3.6.4 Halaman Add Record atau Record Page

Halaman Record Prediksi adalah halaman yang dirancang untuk memungkinkan pengguna memasukkan data medis, melakukan prediksi diabetes, dan menyimpan hasil prediksi ke dalam database berdasarkan user ID. Halaman ini memainkan peran penting dalam pengumpulan data dan analisis prediksi. Dan Field yang harus diisi antara lain :

- Jumlah Kehamilan
- Glukosa Darah
- Tekanan Darah
- Ketebalan Kulit
- Insulin
- Indeks Massa Tubuh (BMI)
- Fungsi Pewarisan Diabetes
- Usia

Glucose

148

Blood Pressure

86

Skin Thickness

25

Insulin

12

BMI

34,3

Diabetes Pedigree Function

0,345

Age

25

Prediction Result: 75.6743311882019%

Calculate

Back to Dashboard

Gambar 3. 12 Halaman Add Record atau Record Page

Tombol "Prediksi" untuk memulai proses prediksi yang dilakukan oleh model yang telah dibuat yang nanti akan menampilkan hasil pada bagian hasil prediksi.

3.6.5 Halaman History Record

Halaman History Record adalah halaman yang memberikan pengguna akses untuk melihat hasil rekaman prediksi diabetes yang pernah mereka lakukan sebelumnya. Halaman ini dirancang untuk membantu pengguna melacak riwayat prediksi dan memahami perkembangan kondisi kesehatan mereka berdasarkan data yang telah dimasukkan dan juga berisi tanggal dan waktu serta nama orang yang diprediksi agar si pengguna mengetahui data prediksi milik siapa dan kapan ia pernah melakukan prediksi ini.

History Records											Back to Dashboard
Records Table											
NO	NAME	PREGNANCIES	GLUCOSE	BLOOD PRESSURE	SKIN THICKNESS	INSULIN	BMI	DIABETES PEDIGREE FUNCTION	AGE	OUTCOME	RECORD TIME
1	Moe	2	148	80	25	10	31.7	0.512	35	71.4893639087677	Kamis, 27/06/2024, 06.34.31
2	John Doe	3	130	72	35	0	33.6	0.627	50	63.142091035842896	Kamis, 27/06/2024, 06.34.44
3	Test	3	120	80	20	100	25	0.5	30	79.19003367424011	Rabu, 03/07/2024, 07.29.32
4	Indonesia	3	148	72	25	12	33.4	0.325	25	69.42925453186035	Rabu, 03/07/2024, 11.11.28
5	web	3	149	69	25	12	33.4	0.325	25	67.44630336761475	Rabu, 03/07/2024, 11.37.31

Gambar 3. 13 Halaman History Record

3.6.6 Halaman History Login

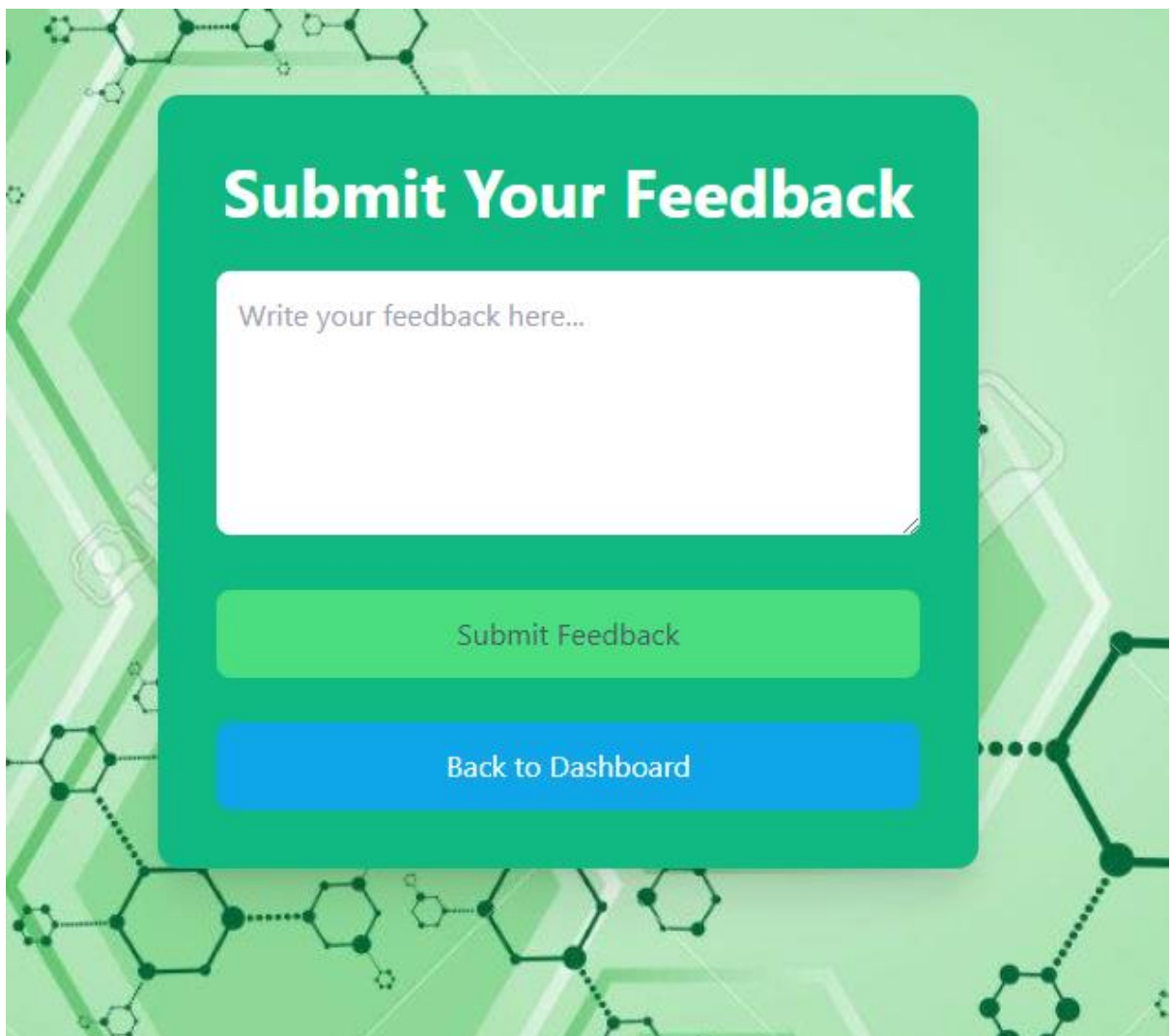
Halaman History Login adalah halaman yang memberikan pengguna akses untuk melihat riwayat login mereka. Halaman ini dirancang untuk meningkatkan keamanan akun pengguna dengan memungkinkan mereka memonitor aktivitas login. Pengguna dapat melihat waktu dan tanggal setiap kali mereka login ke sistem. Fitur ini membantu pengguna mendeteksi aktivitas yang mencurigakan dan mengambil tindakan cepat, seperti mengubah password, jika mereka menduga akun mereka telah diretas. Jika pengguna melihat aktivitas login yang tidak dikenali, mereka dapat dengan mudah mengakses link "Lupa Password" pada halaman login untuk mengganti kata sandi mereka dan mengamankan akun.

Your History Logins				Back to Dashboard
Login Sessions				
NO	DAY	LOGIN TIME	LOGOUT TIME	
1	Rabu	10/07/2024, 20.44.19	10/07/2024, 20.48.25	
2	Rabu	10/07/2024, 20.50.29	N/A	
3	Jumat	12/07/2024, 20.03.41	12/07/2024, 20.04.17	
4	Selasa	16/07/2024, 16.21.43	16/07/2024, 16.22.53	
5	Selasa	16/07/2024, 17.37.14	16/07/2024, 17.38.15	
6	Selasa	16/07/2024, 19.49.58	16/07/2024, 19.58.19	

Gambar 3. 14 Halaman History Login

3.6.7 Halaman Feedback

Halaman Feedback adalah halaman yang memungkinkan pengguna untuk memberikan penilaian dan berbagi pengalaman mereka setelah menggunakan aplikasi web ini. Halaman ini berfungsi sebagai sarana komunikasi antara pengguna dan pengembang, di mana pengguna dapat memberikan saran untuk peningkatan aplikasi sesuai dengan kebutuhan mereka. Selain itu, halaman ini juga berfungsi sebagai platform untuk melaporkan error atau bug yang ditemukan selama penggunaan aplikasi. Dengan feedback yang konstruktif, pengembang dapat terus memperbaiki dan mengembangkan aplikasi untuk memberikan layanan yang lebih baik kepada pengguna.

The image shows a web form titled "Submit Your Feedback" in white text on a green background. Below the title is a large white text input area with the placeholder text "Write your feedback here...". Underneath the input area are two buttons: a green button labeled "Submit Feedback" and a blue button labeled "Back to Dashboard". The entire form is set against a green background with a subtle pattern of molecular structures and circuit lines.

Gambar 3. 15 Halaman Feedback

3.6.8 Halaman Healthy Recipe

Halaman Healthy Recipe adalah halaman di mana pengguna dapat menemukan berbagai resep masakan sehat yang dirancang untuk membantu mereka menghindari diabetes. Resep-resep yang ditampilkan fokus pada makanan yang rendah gula dan kaya nutrisi. Setiap kali pengguna mengklik menu resep yang tersedia, mereka akan diarahkan ke halaman resep yang diambil dari API data web masakan sehat seperti <https://spoonacular.com/food-api>. Halaman ini memberikan variasi resep yang diperbarui secara berkala, memastikan pengguna selalu memiliki pilihan baru dan menarik untuk dicoba. Dengan menyediakan informasi tentang makanan sehat, halaman ini membantu pengguna dalam membuat keputusan yang lebih baik terkait pola makan mereka.

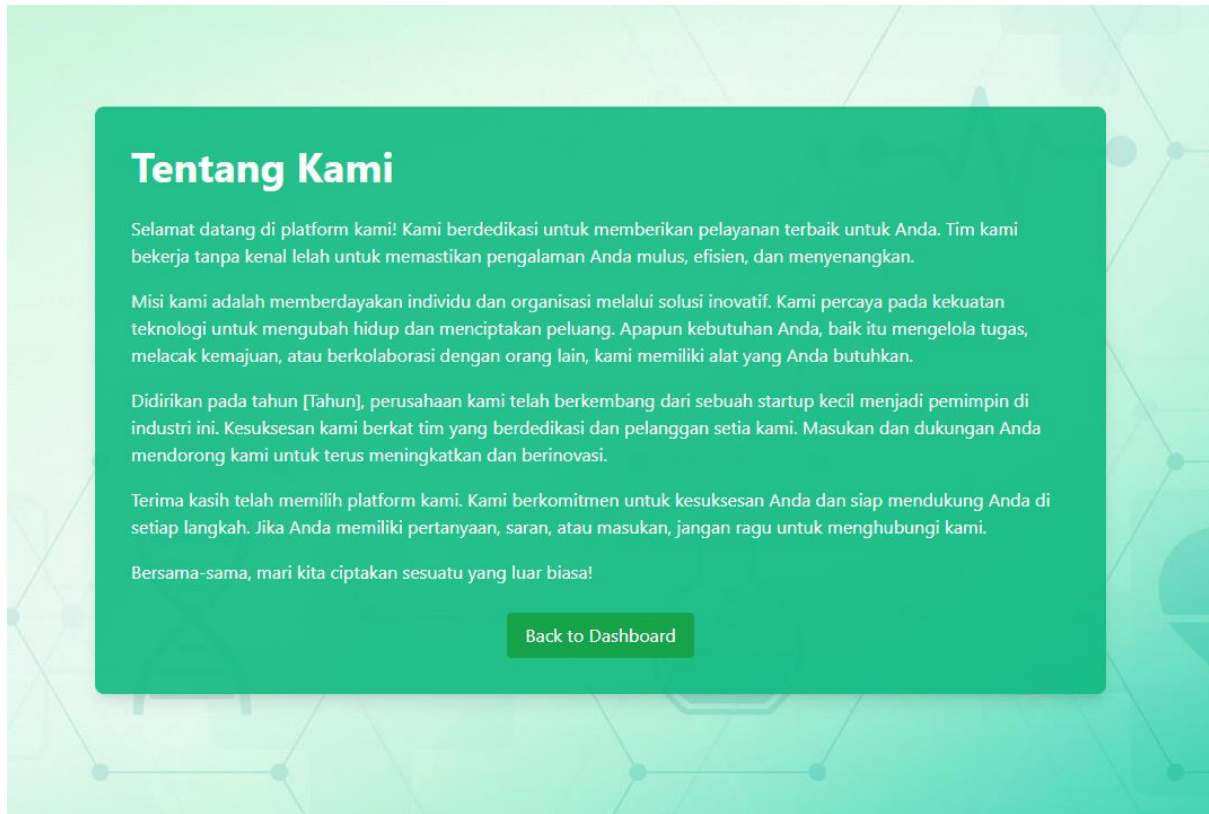


Gambar 3. 16 Halaman Blog

3.6.9 Halaman About

Halaman About adalah halaman yang memberikan informasi detail tentang aplikasi web ini, tujuan pengembangannya, dan tim yang bekerja di baliknya. Halaman ini bertujuan

untuk memberikan transparansi kepada pengguna mengenai misi dan visi aplikasi, serta teknologi yang digunakan dalam pembuatannya.



Gambar 3. 17 Halaman About

BAB IV

IMPLEMENTASI DAN HASIL PENGUJIAN

4.1 Hasil

Tujuan dari penelitian ini adalah untuk menerapkan Algoritma *Neural Network* untuk mendeteksi Diabetes Melitus. Yang dilakukan menggunakan *PIMA Indian Dataset*.

4.1.1 Persiapan Data

Pada penelitian ini memakai *PIMA Indian Dataset* yang tersedia secara publik pada Website [kaggle.com](https://www.kaggle.com). Gambaran dari Datasetnya seperti pada tabel dibawah ini :

No	Nama	Record	Feature	Class
1	<i>PIMA Indian Dataset</i>	768	8	1

Tabel 4. 1 Persiapan Data

Dataset ini telah banyak digunakan oleh peneliti sebelumnya yaitu (Md Kamrul Hasan, Md Ashraful Alam, Dola Das, Eklas Hossain, Mahmudul Hasan, 2020)

4.2 Pembahasan

4.2.1 Pengolahan Data

Pada tahap ini dilakukan pengolahan data dengan cara melihat apakah ada data non-numerik jika ada nantinya akan diubah menjadi format yang dapat dipahami oleh model atau algoritma *Neural Network* biasanya diubah menjadi numerik.

4.2.2 Eksplorasi Data

Pada tahap ini akan dilakukan pengecekan data yaitu berupa pencarian isi, jumlah dan tipe data pada setiap atribut, mencari apa terdapat nilai kosong dan mencari data yang duplikat.

4.2.2.1 Analisa Data

Pada fase penelitian ini dilakukan pembersihan data yang terdapat dalam kumpulan data untuk memastikan kualitas dan kevalidan data yang akan digunakan dengan membersihkan data yang duplikat, outlier, dan missing value.

No.	Nama Atribut	Tipe Data
1	Pregnancies	Int
2	Glucose	Int
3	BloodPressure	Int
4	SkinThickness	Int
5	Insulin	Int
6	BMI	Float
7	DiabetesPedigreeFunction	Float
8	Age	Int
9	Outcome	Int

Tabel 4. 2 Tabel Tipe Data

Pada dataset ini, semua kolom sudah bertipe integer dan float sehingga dapat langsung dilakukan pemrosesan.

Kemudian dilakukan pengecekan terhadap sampel data yaitu *Class* atau label.

a. Class atau Label

Class atau label pada dataset ini adalah *class* yang akan berfungsi sebagai output dari model mesin *learning* yang dibuat, hasil analisa pada kolom *class* dapat dilihat pada tabel dibawah ini.

Keterangan	Nilai
Jumlah Baris	768 Baris Data
Keunikan setiap data	2 Baris Data, 1 = <i>orang tersebut terkena diabetes</i> 0 = <i>orang tersebut bebas diabetes</i>
Frekuensi kemunculan	500
Data yang sering muncul	0

Tabel 4. 3Tabel Class

b. *Feature* atau Atribut

Pengecekan data dari 8 sampel atribut atau feature (*Pregnancies, Glucose Blood Pressure, Skin Thickness, Insulin, BMI, Diabetes Pedigree Function, Age*).

1. *Pregnancies*

Pada kolom *Pregnancies* terdapat 768 baris data yang dimana data tersebut memiliki nilai rata-rata 3.845052 dengan nilai standar deviasi sebesar 3.369578 dengan nilai minimal 0 dan nilai maksimal 17.

Keterangan	Nilai
Jumlah Baris	768 Baris Data
Rata-rata	3.845052
Standar Deviasi	3.369578

Tabel 4. 4 Atribut Pregnancies

2. *Glucose*

Pada kolom *Pregnancies* terdapat 768 baris data yang dimana data tersebut memiliki nilai rata-rata 120.894531 dengan nilai standar deviasi sebesar 31.972618 dengan nilai minimal 0 dan nilai maksimal 199.

Keterangan	Nilai
Jumlah Baris	768 Baris Data
Rata-rata	120.894531
Standar Deviasi	31.972618

Tabel 4. 5 Atribut Glucose

3. *Blood Pressure*

Pada kolom *Pregnancies* terdapat 768 baris data yang dimana data tersebut memiliki nilai rata-rata 69.105469 dengan nilai standar deviasi sebesar 19.355807 dengan nilai minimal 0 dan nilai maksimal 122.

Keterangan	Nilai
Jumlah Baris	768 Baris Data
Rata-rata	69.105469
Standar Deviasi	19.355807

Tabel 4. 6Atribut Blood Pressure

4. *Skin Thickness*

Pada kolom *Pregnancies* terdapat 768 baris data yang dimana data tersebut memiliki nilai rata-rata 20.536458 dengan nilai standar deviasi sebesar 15.952218 dengan nilai minimal 0 dan nilai maksimal 99.

Keterangan	Nilai
Jumlah Baris	768 Baris Data
Rata-rata	20.536458
Standar Deviasi	15.952218

Tabel 4. 7 Atribut Skin Thickness

5. *Insulin*

Pada kolom *Pregnancies* terdapat 768 baris data yang dimana data tersebut memiliki nilai rata-rata 79.799479 dengan nilai standar deviasi sebesar 115.244002 dengan nilai minimal 0 dan nilai maksimal 846.

Keterangan	Nilai
Jumlah Baris	768 Baris Data
Rata-rata	79.799479
Standar Deviasi	115.244002

Tabel 4. 8 Atribut Insulin

6. *BMI*

Pada kolom *Pregnancies* terdapat 768 baris data yang dimana data tersebut memiliki nilai rata-rata 31.992578 dengan nilai standar deviasi sebesar 7.884160 dengan nilai minimal 0 dan nilai maksimal 67.1.

Keterangan	Nilai
Jumlah Baris	768 Baris Data
Rata-rata	31.992578
Standar Deviasi	7.884160

Tabel 4. 9 Atribut BMI

7. *Diabetes Pedigree Function*

Pada kolom *Pregnancies* terdapat 768 baris data yang dimana data tersebut memiliki nilai rata-rata 0.471876 dengan nilai standar deviasi sebesar 0.331329 dengan nilai minimal 0 dan nilai maksimal 2.420.

Keterangan	Nilai
Jumlah Baris	768 Baris Data
Rata-rata	0.471876
Standar Deviasi	0.331329

Tabel 4. 10 Atribut Diabetes Pedigree Function

8. Age

Pada kolom *Pregnancies* terdapat 768 baris data yang dimana data tersebut memiliki nilai rata-rata 33.240885 dengan nilai standar deviasi sebesar 11.760232 dengan nilai minimal 0 dan nilai maksimal 81.

Keterangan	Nilai
Jumlah Baris	768 Baris Data
Rata-rata	33.240885
Standar Deviasi	11.760232

Tabel 4. 11 Atribut Age

4.2.2.2 Statistik Data

Pada proses ini dilakukan proses penggalan informasi terkait nilai-nilai statistik pada dataset, khususnya pada kolom-kolom yg bertipe numerik.

	Pregna ncies	Glucos e	BloodPr essure	SkinThi ckness	Insulin	BMI	DiabetesPedigr eeFunction	Age	Outco me
co un t	768.00 0000	768.00 0000	768.000 000	768.00 0000	768.00 0000	768.00 0000	768.000000	768.00 0000	768.00 0000
me an	3.8450 52	120.89 4531	69.1054 69	20.536 458	79.799 479	31.992 578	0.471876	33.240 885	0.3489 58
std	3.3695 78	31.972 618	19.3558 07	15.952 218	115.24 4002	7.8841 60	0.331329	11.760 232	0.4769 51
mi n	0.0000 00	0.0000 00	0.00000 0	0.0000 00	0.0000 00	0.0000 00	0.078000	21.000 000	0.0000 00
25 %	1.0000 00	99.000 000	62.0000 00	0.0000 00	0.0000 00	27.300 000	0.243750	24.000 000	0.0000 00
50 %	3.0000 00	117.00 0000	72.0000 00	23.000 000	30.500 000	32.000 000	0.372500	29.000 000	0.0000 00

75 %	6.0000 00	140.25 0000	80.0000 00	32.000 000	127.25 0000	36.600 000	0.626250	41.000 000	1.0000 00
max	17.000 000	199.00 0000	122.000 000	99.000 000	846.00 0000	67.100 000	2.420000	81.000 000	1.0000 00

Tabel 4. 12 Statistik Data

4.2.2.3 Visualisasi Data

Pada tahap ini akan dilakukan eksplorasi data menggunakan grafik atau diagram untuk memberikan gambaran terhadap sebuah data pada tiap kolom.

a. *Class* atau label

Dari total 768 data *Class*, terdapat sebanyak 5540 (63%) data yang bukan merupakan malware dan sebanyak 1719 (37%) data yang merupakan *malware*.

4.2.3 Membangun Model

Membangun model merupakan langkah krusial dalam analisis data, di mana kita menciptakan suatu representasi matematis yang dapat menggambarkan hubungan antara variabel dalam *Dataset*. Pada penelitian ini saya menggunakan metode Neural Network untuk membangun model *machine learning* ini.

Dataset	Jumlah
<i>PIMA Indian Dataset</i>	768 baris dan 9 kolom

Tabel 4. 13 Dataset Dan Jumlah Data

a. Split Data

Pada tahap ini, data akan dibagi menjadi data latih dan data uji menggunakan `train_test_split`. Adapun besaran dari data latih yaitu 80% dan data uji yaitu 20%. Seperti pada tabel berikut:

Dataset	Jumlah
<i>Data Train</i>	614 baris dan 8 kolom
<i>Data Test</i>	154 baris dan 8 kolom

Tabel 4. 14 Split Data

b. Training Model

Model yang digunakan yaitu klasifikasi, yang bertujuan untuk mendeteksi penderita diabetes melitus berdasarkan berbagai fitur atau atribut yang ada dalam dataset.

Dataset	Nilai
<i>epochs</i>	50
<i>Learning Rate</i>	0.002
<i>Layer</i>	1 layer input dengan 8 unit neuron 1 hidden layer dengan 30 unit neuron 1 hidden layer dengan 10 unit neuron 1 output layer dengan 1 neuron

Tabel 4. 15 Training Data

Berdasarkan gambar dibawah ini, terdapat 8 neuron pada layer input, 30 neuron dan 10 neuron pada 2 hidden layer, dan 2 neuron pada layer output. Cara kerja ketiga layer tersebut adalah input layer menerima masukan nilai awal dari data yang akan diproses, nantinya dilewatkan ke hidden layer lalu akan keluar melalui output layer dalam bentuk bobot pula.

4.2.4 Pengujian Model

Pada tahap ini dilakukan pengujian performa model dengan menggunakan metrik evaluasi yang digunakan dalam metode *Neural Network* (NN). Metrik evaluasi ini akan memberikan pemahaman tentang sejauh mana model dapat mendeteksi *malware* pada sistem android dengan akurat. Beberapa metrik evaluasi yang digunakan dalam pengujian model ini meliputi akurasi, *recall*, *F1-Score*, dan *Confusion matrix*.

Model	Hasil
Akurasi	76.3%
<i>Recall</i>	76.1
<i>F1-Score</i>	76.1

Tabel 4. 16 Hasil dari model ANN

Setelah kedua model tersebut dievaluasi dan dilakukan validasi hasil kemudian kita lakukan penyimpanan kedua model tersebut kedalam format H5. Format file H5 (HDF5) adalah format penyimpanan yang efisien dan dapat digunakan untuk menyimpan berbagai jenis data, termasuk model neural network. Pustaka seperti Keras dapat menyimpan model mereka dalam format H5, yang kemudian dapat diimpor kembali untuk digunakan atau disaring lebih lanjut.

4.3 Implementasi Sistem

Implementasi sistem adalah proses menerapkan dan menjalankan sistem baru atau yang telah diperbarui di lingkungan nyata. Tujuan utamanya adalah memastikan bahwa sistem tersebut berfungsi dengan baik dan memenuhi kebutuhan pengguna. Proses ini mencakup pemasangan perangkat lunak dan perangkat keras, pengujian untuk memastikan semuanya berjalan lancar, pelatihan pengguna agar mereka dapat menggunakan sistem dengan efektif, dan pemeliharaan serta dukungan berkelanjutan untuk memastikan sistem tetap beroperasi dengan baik. Berikut ini adalah Struktur Tabel yang telah dibangun :

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
<input type="checkbox"/>	1 id_record	int(8)			Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hapus Lainnya
<input type="checkbox"/>	2 user_id	varchar(5)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	3 name	varchar(50)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	4 record_time	datetime			Tidak	current_timestamp()			Ubah Hapus Lainnya
<input type="checkbox"/>	5 pregnancies	int(3)			Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	6 glucose	int(4)			Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	7 blood_pressure	int(3)			Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	8 skin_thickness	int(3)			Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	9 insulin	int(4)			Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	10 bmi	float			Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	11 diabetes_pedigree_function	float			Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	12 age	int(3)			Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	13 outcome	text	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya

Gambar 4. 1 Tabel Login/User

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
<input type="checkbox"/>	1 id	varchar(5)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	2 username	varchar(50)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	3 email	varchar(33)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	4 password	varchar(70)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	5 retype_password	varchar(30)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	6 created_time	datetime			Tidak	current_timestamp()			Ubah Hapus Lainnya
<input type="checkbox"/>	7 reset_password_token	varchar(255)	utf8mb4_general_ci		Ya	NULL			Ubah Hapus Lainnya
<input type="checkbox"/>	8 reset_password_expires	bigint(20)			Ya	NULL			Ubah Hapus Lainnya

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
<input type="checkbox"/>	1 id	int(11)			Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hapus Lainnya
<input type="checkbox"/>	2 username	varchar(50)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	3 login_time	timestamp			Tidak	current_timestamp()			Ubah Hapus Lainnya
<input type="checkbox"/>	4 logout_time	timestamp			Ya	NULL			Ubah Hapus Lainnya

Gambar 4. 3 Tabel User Session

Gambar 4. 2 Tabel Record/Prediksi

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
<input type="checkbox"/>	1 id	int(5)			Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hapus Lainnya
<input type="checkbox"/>	2 username	varchar(80)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	3 content	text	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya

Gambar 4. 4 Tabel Feedback

4.4 Membangun Web App

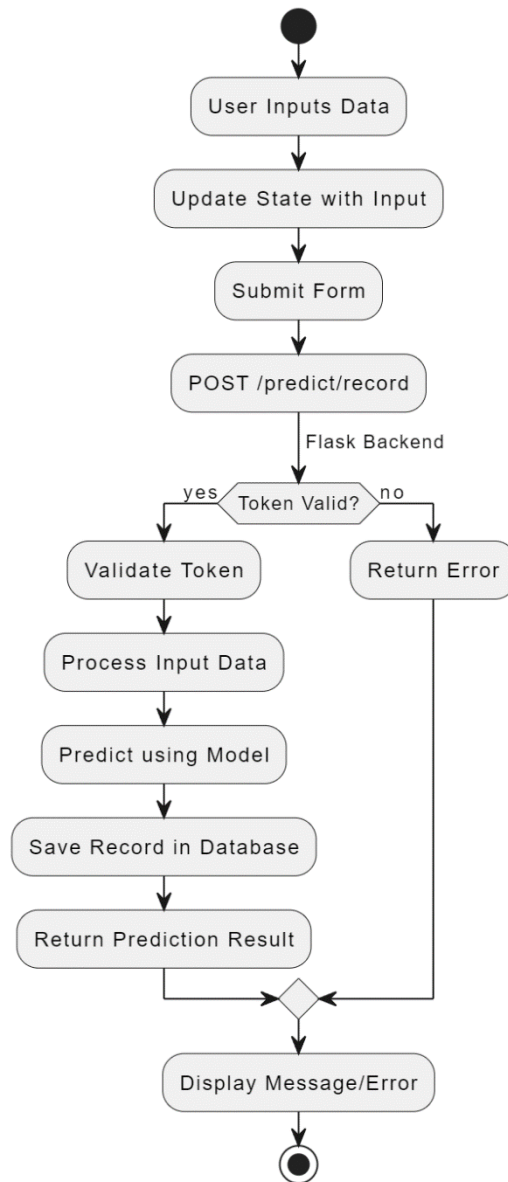
Pada tahap ini penulis akan mengimplementasikan model yang telah dibuat ke dalam sebuah web app sederhana yang dapat memprediksi seseorang terkena diabetes melitus dengan memasukkan beberapa nilai pada variabel seperti *Pregnancies*, *BMI*, *Bloodpressure*, *Skin thickness*, *DiabetesPedigreeFunction*, *Glucose*, dan *Insulin*. dan user dapat langsung mengetahui hasilnya berupa presentase dari hasil kalkulasi yang dilakukan oleh model Neural Network serta Web App ini memiliki beberapa menu yang lainnya.

4.4.1 Integrasi Model Neural Network

Model *Neural Network* yang telah dilatih sebelumnya diintegrasikan ke dalam aplikasi web menggunakan Framework Express dan React Js lalu menghubungkannya dengan Framework Flask yang memuat model yang telah dibuat beserta bobotnya. Hal ini memungkinkan aplikasi untuk melakukan prediksi berdasarkan input pengguna.

4.4.2 Penanganan Input dan Prediksi

Aplikasi web dibangun untuk menangani input dari pengguna dan menggunakan model Neural Network untuk menghasilkan prediksi. Alur kerja ini diimplementasikan secara bersih dan efisien.



Gambar 4. 5 Diagram Pemrosesan

4.4.3 Pengujian Aplikasi

Pada tahap ini, aplikasi web prediksi diabetes diuji coba untuk memastikan kinerjanya sesuai dengan yang diharapkan. Pengujian mencakup uji fungsionalitas, responsivitas, dan keakuratan prediksi dan juga kegunaan menu yang lainnya.

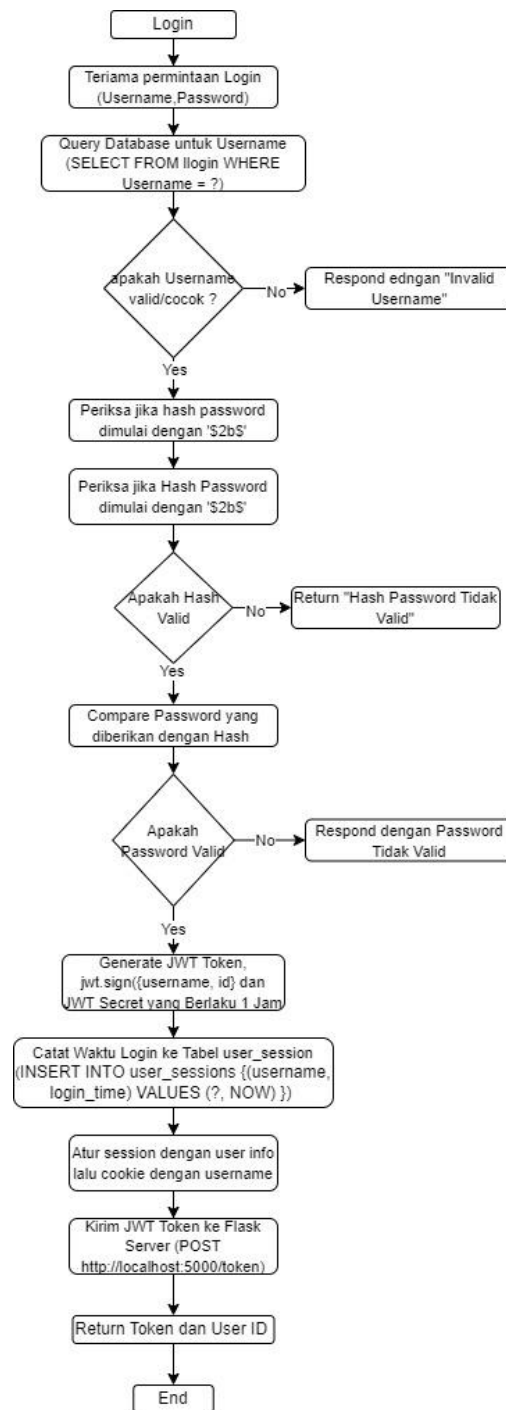
4.4.3.1 Pengujian *Whitebox Testing*

White-box testing, juga dikenal sebagai *glass-box testing* atau *clear-box testing*, adalah metode pengujian di mana penguji memiliki pengetahuan tentang struktur internal atau kode sumber dari aplikasi yang diuji. Penguji dapat melihat dan menggunakan pengetahuan tentang kode untuk merancang kasus uji yang lebih spesifik dan mendalam. Misalnya, penguji akan memeriksa bagaimana fungsi-fungsi dipanggil, alur logika, dan jalur eksekusi dalam program. dan berikut alur dari *whitebox testing* :

a) Whitebox Testing Login

Proses login dimulai dengan menerima permintaan yang berisi username dan password. Server kemudian melakukan query ke database untuk memeriksa apakah username yang diberikan ada. Jika username tidak valid, server akan merespons dengan "Invalid Username". Jika valid, server memeriksa apakah hash password memiliki prefix yang benar ('\$2b\$'). Jika hash password valid, server membandingkan password yang diberikan dengan hash yang disimpan di database. Jika password tidak valid, server akan merespons dengan "Password Tidak Valid". Jika password valid, server menghasilkan token JWT dan mencatat waktu login ke tabel *user_session* di database, kemudian mengatur

session dan user info, mengirim token JWT ke server Flask, dan mengembalikan token serta user ID ke pengguna.

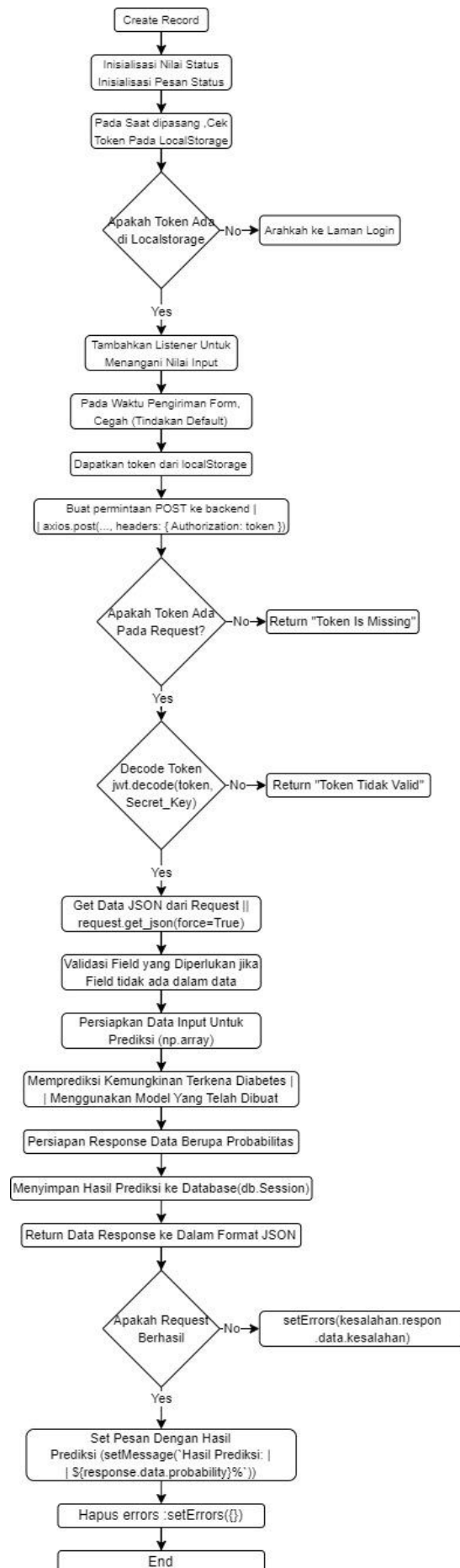


Gambar 4. 6 Whitebox Testing Login

b) *Whitebox Testing Add Record/Prediksi*

Proses penambahan record dimulai dengan inialisasi nilai status dan pesan status. Saat halaman diakses, sistem memeriksa apakah token tersedia di localStorage. Jika token tidak ada, pengguna diarahkan ke halaman login. Jika token ada, listener ditambahkan

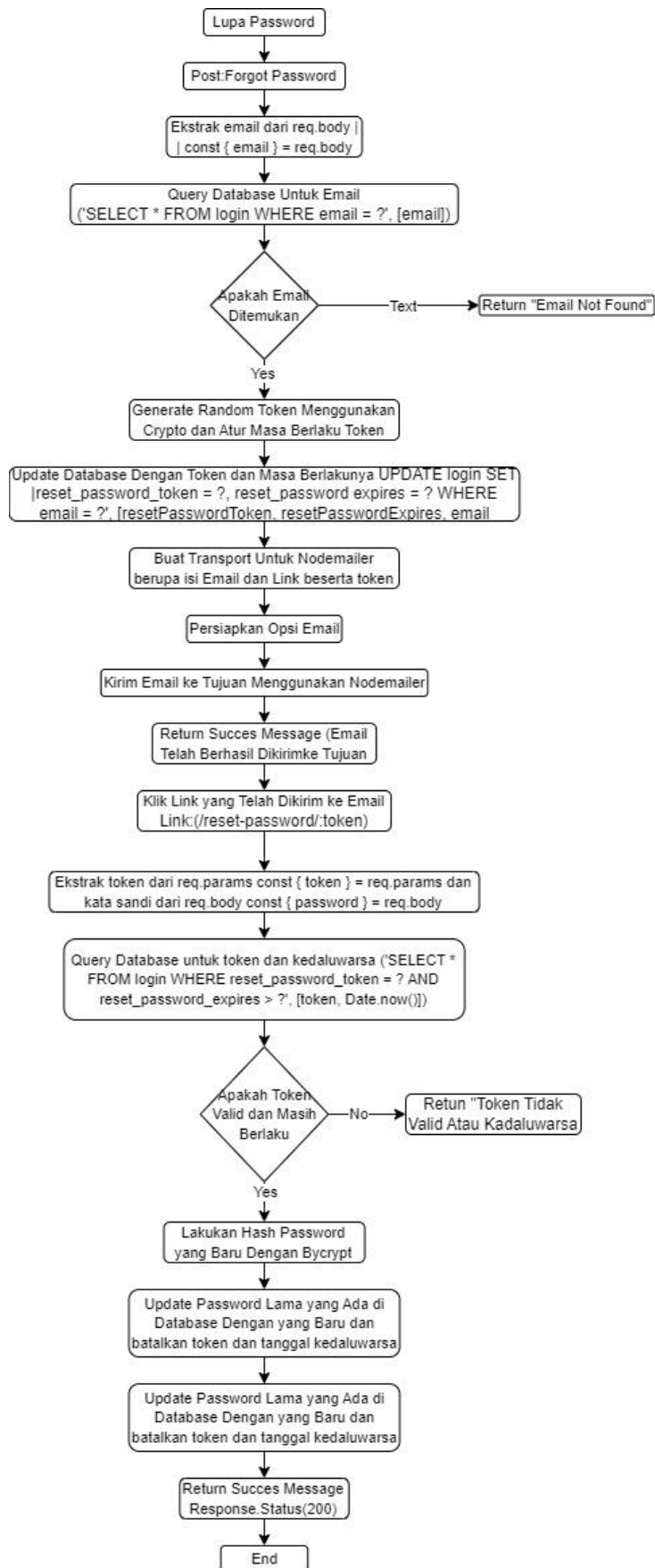
untuk menangani nilai input. Saat form dikirimkan, tindakan default dicegah dan token diperoleh dari `localStorage`. Permintaan POST dikirim ke backend dengan menyertakan token dalam header `Authorization`. Backend memeriksa keberadaan token pada permintaan; jika token tidak ada, respons "Token Is Missing" dikirimkan. Jika token ada, token didekode menggunakan `secret key`. Jika token tidak valid, respons "Token Tidak Valid" dikirimkan. Jika token valid, data JSON diperoleh dari permintaan dan field yang diperlukan divalidasi. Data input dipersiapkan untuk prediksi dan model prediksi digunakan untuk memprediksi kemungkinan terkena diabetes. Hasil prediksi disimpan ke database dan respons data dalam format JSON dikembalikan. Jika permintaan berhasil, pesan hasil prediksi diatur; jika gagal, kesalahan disimpan dalam `setErrors`. Proses diakhiri dengan menghapus errors jika tidak ada kesalahan.



Gambar 4. 7 Whitebox Testing Add Record/Prediksi

c) Whitebox Testing Forgot /Change Password

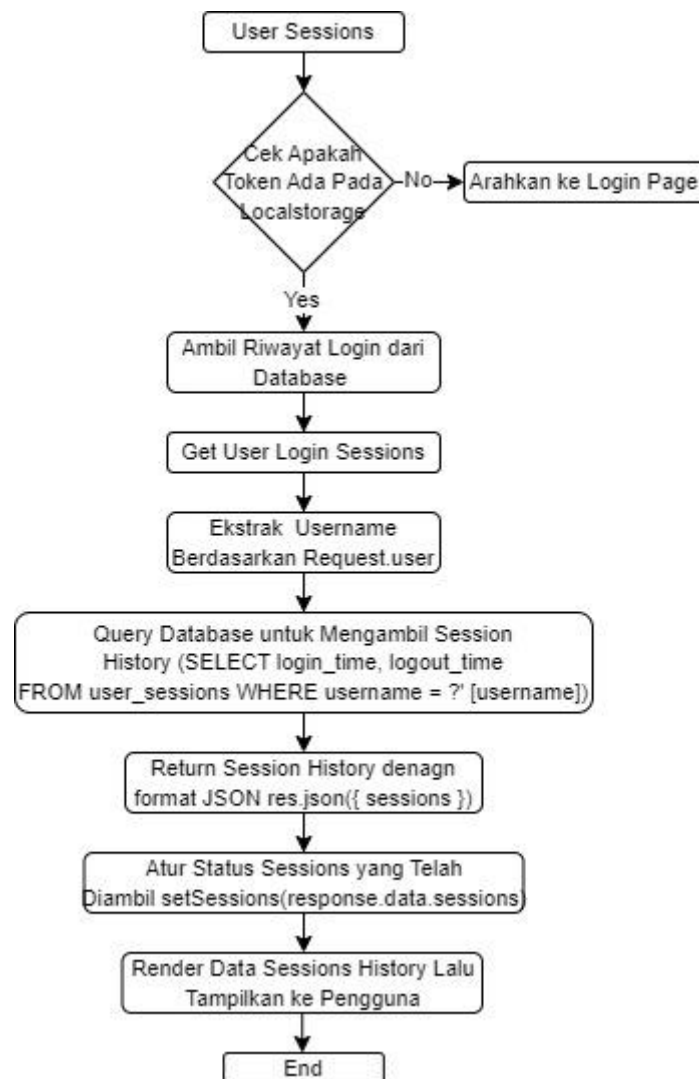
Proses forgot password dimulai ketika pengguna mengirimkan permintaan forgot password dengan email. Sistem akan mengekstrak email dari permintaan dan memeriksa keberadaan email di database. Jika email tidak ditemukan, sistem mengirimkan respons "Email Not Found". Jika email ditemukan, sistem akan menghasilkan token acak menggunakan crypto dan mengatur masa berlakunya token. Token dan masa berlakunya disimpan di database. Sistem akan mempersiapkan opsi email dan mengirimkan email ke alamat yang diberikan dengan menggunakan nodemailer. Jika email berhasil dikirim, sistem mengirimkan pesan sukses. Pengguna kemudian akan mengklik link yang dikirimkan ke email mereka untuk memulai proses reset password. Sistem akan mengekstrak token dari URL dan memeriksa validitas serta masa berlakunya token di database. Jika token tidak valid atau sudah kadaluarsa, sistem mengirimkan respon "Token Tidak Valid atau Kadaluarsa". Jika token valid, sistem akan melakukan hash password baru dengan bcrypt dan memperbarui password di database serta menghapus token reset. Sistem kemudian mengirimkan pesan sukses sebagai Respon.



Gambar 4. 8 Whitebox Testing Forgot /Change Password

d) Whitebox Testing History Login

Diagram alur kerja "User Sessions" ini menggambarkan proses whitebox testing untuk pengelolaan sesi login pengguna pada sebuah aplikasi. Proses dimulai dengan pengecekan apakah token pengguna ada pada local storage. Jika token tidak ditemukan, pengguna akan diarahkan ke halaman login. Jika token ditemukan, sistem akan mengambil riwayat login pengguna dari database. Username diekstraksi dari request.user dan digunakan untuk mengambil riwayat sesi dari tabel user_sessions di database, termasuk waktu login dan logout. Data riwayat sesi kemudian dikembalikan dalam format JSON dan status sesi diatur berdasarkan data yang diambil. Akhirnya, data riwayat sesi dirender dan ditampilkan kepada pengguna. Testing mencakup verifikasi setiap langkah proses ini untuk memastikan bahwa sesi pengguna dikelola dengan benar dan sesuai dengan yang diharapkan.

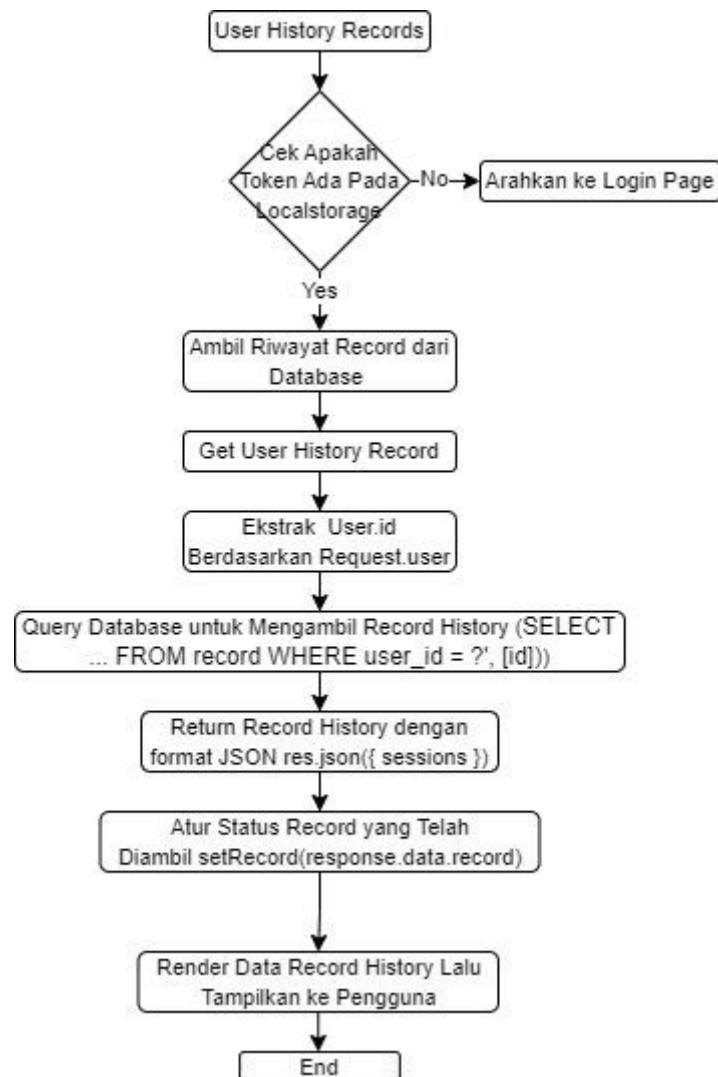


Gambar 4. 9 Whitebox Testing History Login

e) *Whitebox Testing History Record*

Diagram alur kerja "User History Records" ini menggambarkan proses whitebox testing untuk pengelolaan riwayat record pengguna pada sebuah aplikasi. Proses dimulai dengan pengecekan apakah token pengguna ada pada local storage. Jika token tidak ditemukan, pengguna akan diarahkan ke halaman login. Jika token ditemukan, sistem akan mengambil riwayat record pengguna dari database. User ID diekstraksi dari request.user dan digunakan untuk mengambil riwayat record dari tabel record di database. Data riwayat record kemudian dikembalikan dalam format JSON dan status record diatur berdasarkan data yang diambil. Akhirnya, data riwayat record dirender dan ditampilkan kepada pengguna. Testing mencakup verifikasi setiap langkah

proses ini untuk memastikan bahwa riwayat record pengguna dikelola dengan benar dan sesuai dengan yang diharapkan.



Gambar 4. 10 Whitebox Testing History Record

4.4.3.2 Pengujian *Black-box Testing*

Black-box testing adalah metode pengujian di mana penguji tidak memiliki pengetahuan tentang struktur internal atau kode sumber dari aplikasi yang diuji. Penguji hanya berfokus pada *input* yang diberikan dan *output* yang dihasilkan. Pengujian ini bertujuan untuk memverifikasi bahwa aplikasi berfungsi sesuai dengan spesifikasi dan kebutuhan yang telah ditentukan. Penguji tidak perlu tahu bagaimana aplikasi bekerja di dalam, cukup mengetahui apakah aplikasi menghasilkan output yang diharapkan atau tidak. dan berikut *black-box testing*nya:

a) *Black-box Testing Login*

No	Langkah	Deskripsi	Kondisi	Hasil Pengujian
1	Start	Mulai pengujian	-	-
2	Registrasi Pengguna	Pengguna mengisi formulir registrasi	-	-
3	Apakah username tersedia?	Cek ketersediaan username	Username tidak tersedia	Return kegagalan registrasi
	-	-	Username tersedia	Lanjut ke langkah berikutnya
4	Simpan pengguna ke database	Simpan data pengguna baru ke database	-	Pengguna tersimpan, lanjut ke login
5	Login Pengguna	Pengguna memasukkan kredensial login	-	-
6	Apakah kredensial benar	Cek Validasi kredensial	Kredensial salah	Return kegagalan login
	-	-	Kredensial benar	Lanjut ke langkah berikutnya

7	Buat token JWT	Buat token JWT untuk sesi login	-	Token dibuat
8	Simpan sesi login ke database	Simpan informasi sesi login ke database	-	Sesi login tersimpan
9	Kembalikan token JWT dan info pengguna	Return token dan informasi pengguna	-	Token dan info pengguna dikembalikan
10	Validasi Token	Validasi token yang diterima	-	-
11	Apakah token valid?	Cek validitas token	Token tidak valid	Return kegagalan validasi
		-	Token valid	Lanjut ke langkah berikutnya
12	Decode token JWT	Decode token untuk mendapatkan info pengguna	-	Info pengguna diperoleh
13	Logout Pengguna	Pengguna melakukan logout	-	-

14	Simpan waktu logout ke DB	Simpan waktu logout ke database	-	Waktu logout tersimpan
15	Selesai	Pengujian selesai	-	-

Tabel 4. 17 Black-box Testing Login

b) *Black-box Testing Add Record/Prediksi*

No	Langkah	Deskripsi	Kondisi	Hasil Pengujian
1	Inisialisasi Frontend & Backend	Mulai pengujian	-	-
2	Frontend: Inisialisasi nilai state	Inisialisasi nilai state di frontend	-	State values diinisialisasi
3	Frontend: Inisialisasi pesan state	Inisialisasi state pesan di frontend	-	State message dan errors diinisialisasi
4	Frontend: Periksa token di localStorage	Cek token di localStorage saat komponen dimuat	Token tidak ada	Arahkan ke halaman login
	-	-	Token ada	Lanjut ke langkah berikutnya

5	Frontend: Tambah event listener	Tambahkan event listener untuk menangani input form	-	Event listener ditambahkan
6	Frontend: Mencegah aksi default pada form	Mencegah aksi default pada submit form	-	Aksi default dicegah
7	Frontend: Ambil token dari localStorage	Ambil token dari localStorage	-	Token diambil
8	Frontend: Kirim POST request ke backend	Membuat permintaan POST ke backend dengan token	-	Permintaan POST dibuat
9	Backend: Apakah token ada di request?	Cek apakah token ada dalam permintaan	Token tidak ada	Kembalikan pesan 'Token is missing!'
		-	Token ada	Lanjut ke langkah berikutnya
10	Backend: Decode token	Dekode token dengan kunci rahasia	Token tidak valid	Kembalikan pesan 'Token is invalid!'
	-	-	Token valid	Lanjut ke langkah berikutnya

11	Backend: Ambil data JSON dari request	Ambil data JSON dari permintaan	-	Data JSON diambil
12	Backend: Validasi field yang diperlukan	Validasi field yang diperlukan	Field tidak lengkap	Return pesan kesalahan validasi
		-	Field lengkap	Lanjut ke langkah berikutnya
13	Backend: Siapkan data untuk prediksi	Siapkan data input untuk prediksi	-	Data input disiapkan
14	Backend: Prediksi probabilitas diabetes	Prediksi probabilitas diabetes menggunakan model	-	Prediksi dibuat
15	Backend: Siapkan data respons	Siapkan data respons	-	Data respons disiapkan
16	Backend: Simpan hasil prediksi ke database	Simpan hasil prediksi ke database	-	Hasil prediksi disimpan ke database
17	Backend: Kembalikan data respons	-	-	Data JSON dikembalikan

	sebagai JSON			
18	Frontend: Apakah request berhasil?	Cek apakah permintaan berhasil	Permintaan gagal	Set state errors dengan pesan kesalahan
	-	-	Permintaan berhasil	Lanjut ke langkah berikutnya
19	Frontend: Set pesan dengan hasil prediksi	Set pesan dengan hasil prediksi	-	Pesan di-set dengan hasil prediksi
20	Frontend: Hapus errors	Bersihkan state errors	-	State errors dibersihkan

Tabel 4. 18 Black-box Testing Add Record/Prediksi

c) Black-box Testing Forgot /Change Password

No	Langkah	Deskripsi	Kondisi	Hasil Pengujian
1	Mulai (Forgot Password & Reset Password)	Mulai pengujian	-	-
2	Endpoint: POST /forgot-password	Pengguna mengirim permintaan lupa kata sandi	-	-

3	Backend: Ekstrak email dari req.body	Ekstraksi email dari tubuh permintaan	-	Email diekstrak
4	Backend: Query database untuk email	Cek ketersediaan email di database	Email tidak ditemukan	Kembalikan pesan 'Email not found'
	-	-	Email ditemukan	Lanjut ke langkah berikutnya
5	Generate token acak	Buat token acak untuk reset password	-	Token acak dibuat
6	Atur waktu kedaluwarsa token	Setel waktu kedaluwarsa token	-	Waktu kedaluwarsa token disetel
7	Perbarui database dengan token dan waktu kedaluwarsa	Simpan token dan waktu kedaluwarsa ke database	-	Database diperbarui
8	Buat transport untuk nodemailer	Buat transport email menggunakan nodemailer	-	Transport nodemailer dibuat

9	Siapkan opsi email	Siapkan konfigurasi email	-	Opsi email disiapkan
10	Kirim email dengan nodemailer	Kirim email reset password	-	Email berhasil dikirim
11	Kembalikan pesan sukses	Return pesan sukses	-	Pesan sukses dikembalikan
12	Endpoint: POST /reset-password/	Pengguna mengirim permintaan reset password	-	-
13	Ekstrak token dari req.params dan password dari req.body	Ekstraksi token dan password dari permintaan	-	Token dan password diekstrak
14	Query database untuk token dan waktu kedaluwarsa	Cek token dan waktu kedaluwarsa di database	Token tidak valid/kedaluwarsa	Kembalikan pesan 'Token is invalid or expired'
			Token valid	Lanjut ke langkah berikutnya

15	Hash password baru	Buat hash untuk password baru	-	Password baru di-hash
16	Perbarui database dengan password baru	Simpan password baru ke database	-	Database diperbarui dengan password baru
17	Kembalikan pesan sukses	Return pesan sukses	-	Pesan sukses dikembalikan
18	Selesai	Pengujian selesai	-	-

Tabel 4. 19 Black-box Testing Forgot /Change Password

d) Black-box Testing History Login

No	Langkah	Deskripsi	Kondisi	Hasil Pengujian
1	Start	Mulai pengujian	-	-
2	Periksa token di localStorage	Cek apakah token ada di localStorage	Token tidak ada	Arahkan ke halaman login
			Token ada	Lanjut ke langkah berikutnya
3	Ambil riwayat sesi dari backend	Ambil data riwayat sesi dari backend	-	Data sesi berhasil diambil

4	Endpoint GET /history/sessions	Backend menerima permintaan GET riwayat sesi	-	Permintaan GET diterima
5	Ekstrak username dari req.user	Ekstraksi username dari req.user	-	Username diekstrak
6	Query database untuk riwayat sesi	Cek riwayat sesi di database	-	Riwayat sesi diambil dari database
7	Kembalikan riwayat sesi sebagai JSON	Return data riwayat sesi dalam format JSON	-	Riwayat sesi dikembalikan sebagai JSON
8	Set state dengan sesi yang diambil	Set state di frontend dengan data sesi yang diambil	-	State frontend diperbarui
9	Render riwayat sesi	Render data riwayat sesi di frontend	-	Riwayat sesi dirender

Tabel 4. 20 Black-box Testing History Login

e) Black-box Testing History Record

No	Langkah	Deskripsi	Kondisi	Hasil Pengujian
1	Start	Mulai pengujian	-	-

2	Periksa token di localStorage	Cek apakah token ada di localStorage	Token tidak ada	Arahkan ke halaman login
			Token ada	Lanjut ke langkah berikutnya
3	Ambil riwayat catatan dari backend	Ambil data riwayat catatan dari backend	-	Data catatan berhasil diambil
4	Endpoint GET /history/record	Backend menerima permintaan GET riwayat catatan	-	Permintaan GET diterima
5	Ekstrak user ID dari req.user	Ekstraksi user ID dari req.user	-	User ID diekstrak
6	Query database untuk riwayat catatan	Cek riwayat catatan di database	-	Riwayat catatan diambil dari database
7	Kembalikan riwayat catatan sebagai JSON	Return data riwayat catatan dalam format JSON	-	Riwayat catatan dikembalikan sebagai JSON
8	Set state dengan catatan yang diambil	Set state di frontend dengan data catatan yang diambil	-	State frontend diperbarui

9	Render riwayat catatan	Render data riwayat catatan di frontend	-	Riwayat catatan dirender
---	------------------------	---	---	--------------------------

Tabel 4. 21 Black-box Testing History Record

4.4.3.3 Pengeujian Keamanan/*Penetration Testing*

Penetration testing, atau pen-test, adalah metode pengujian keamanan di mana penguji mencoba mengeksploitasi kerentanan dalam sistem untuk menemukan dan memperbaiki celah keamanan sebelum penjahat siber menemukannya. Ini mirip dengan hacking, tetapi dilakukan dengan izin dan tujuan yang baik. Penetration testing membantu organisasi mengetahui seberapa rentan sistem mereka terhadap serangan dan meningkatkan keamanan sistem tersebut.

```

[20:10:07] [INFO] testing 'Oracle stacked queries (heavy query - comment)'
[20:10:08] [INFO] testing 'IBM DB2 stacked queries (heavy query - comment)'
[20:10:08] [INFO] testing 'SQLite > 2.0 stacked queries (heavy query - comment)'
[20:10:08] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'
[20:10:08] [INFO] testing 'MySQL >= 5.0.12 OR time-based blind (query SLEEP)'
[20:10:08] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (SLEEP)'
[20:10:08] [INFO] testing 'MySQL >= 5.0.12 OR time-based blind (SLEEP)'
[20:10:08] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (SLEEP - comment)'
[20:10:08] [INFO] testing 'MySQL >= 5.0.12 OR time-based blind (SLEEP - comment)'
[20:10:08] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP - comment)'
[20:10:08] [INFO] testing 'MySQL >= 5.0.12 OR time-based blind (query SLEEP - comment)'
[20:10:08] [INFO] testing 'MySQL < 5.0.12 AND time-based blind (BENCHMARK)'
[20:10:08] [INFO] testing 'MySQL > 5.0.12 AND time-based blind (heavy query)'
[20:10:08] [INFO] testing 'MySQL < 5.0.12 OR time-based blind (BENCHMARK)'
[20:10:09] [INFO] testing 'MySQL > 5.0.12 OR time-based blind (heavy query)'
[20:10:09] [INFO] testing 'MySQL >= 5.0.12 RLIKE time-based blind'
[20:10:09] [INFO] testing 'MySQL >= 5.0.12 RLIKE time-based blind (query SLEEP)'
[20:10:09] [INFO] testing 'MySQL AND time-based blind (ELT)'
[20:10:09] [INFO] testing 'MySQL OR time-based blind (ELT)'
[20:10:09] [INFO] testing 'PostgreSQL > 8.1 AND time-based blind'
[20:10:09] [INFO] testing 'PostgreSQL AND time-based blind (heavy query)'
[20:10:10] [INFO] testing 'PostgreSQL OR time-based blind (heavy query)'
[20:10:10] [INFO] testing 'Microsoft SQL Server/Sybase time-based blind (IF)'
[20:10:10] [INFO] testing 'Microsoft SQL Server/Sybase AND time-based blind (heavy query)'
[20:10:10] [INFO] testing 'Microsoft SQL Server/Sybase OR time-based blind (heavy query)'
[20:10:10] [INFO] testing 'Oracle AND time-based blind'
[20:10:11] [INFO] testing 'Oracle OR time-based blind'
[20:10:11] [INFO] testing 'Oracle AND time-based blind (heavy query)'
[20:10:11] [INFO] testing 'Oracle OR time-based blind (heavy query)'
[20:10:11] [INFO] testing 'IBM DB2 AND time-based blind (heavy query)'
[20:10:11] [INFO] testing 'IBM DB2 OR time-based blind (heavy query)'
[20:10:11] [INFO] testing 'SQLite > 2.0 AND time-based blind (heavy query)'
[20:10:11] [INFO] testing 'SQLite > 2.0 OR time-based blind (heavy query)'
[20:10:11] [INFO] testing 'Informix AND time-based blind (heavy query)'
[20:10:11] [INFO] testing 'Informix OR time-based blind (heavy query)'
[20:10:11] [INFO] testing 'MySQL >= 5.1 time-based blind (heavy query) - PROCEDURE ANALYSE (EXTRACTVALUE)'
[20:10:11] [INFO] testing 'MySQL >= 5.0.12 time-based blind - Parameter replace'
[20:10:11] [INFO] testing 'MySQL >= 5.0.12 time-based blind - Parameter replace (substraction)'
[20:10:11] [INFO] testing 'PostgreSQL > 8.1 time-based blind - Parameter replace'
[20:10:11] [INFO] testing 'Oracle time-based blind - Parameter replace (DBMS_LOCK.SLEEP)'
[20:10:11] [INFO] testing 'Oracle time-based blind - Parameter replace (DBMS_PIPE.RECEIVE_MESSAGE)'
[20:10:11] [INFO] testing 'MySQL >= 5.0.12 time-based blind - ORDER BY, GROUP BY clause'
[20:10:11] [INFO] testing 'PostgreSQL > 8.1 time-based blind - ORDER BY, GROUP BY clause'
[20:10:11] [INFO] testing 'Oracle time-based blind - ORDER BY, GROUP BY clause (DBMS_LOCK.SLEEP)'
[20:10:11] [INFO] testing 'Oracle time-based blind - ORDER BY, GROUP BY clause (DBMS_PIPE.RECEIVE_MESSAGE)'
[20:10:11] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[20:10:12] [INFO] testing 'Generic UNION query (random number) - 1 to 10 columns'
[20:10:12] [INFO] testing 'MySQL UNION query (NULL) - 1 to 10 columns'
[20:10:12] [INFO] testing 'MySQL UNION query (random number) - 1 to 10 columns'
[20:10:12] [WARNING] parameter 'Referer' does not seem to be injectable
[20:10:13] [CRITICAL] all tested parameters do not appear to be injectable. Try to increase values for '--level'/'--risk' options if you wish to perform more tests. If you suspect that there is some kind of protection mechanism involved (e.g. WAF) maybe you could try to use option '--tamper' (e.g. '--tamper=space2comment') and/or switch '--random-agent'

```

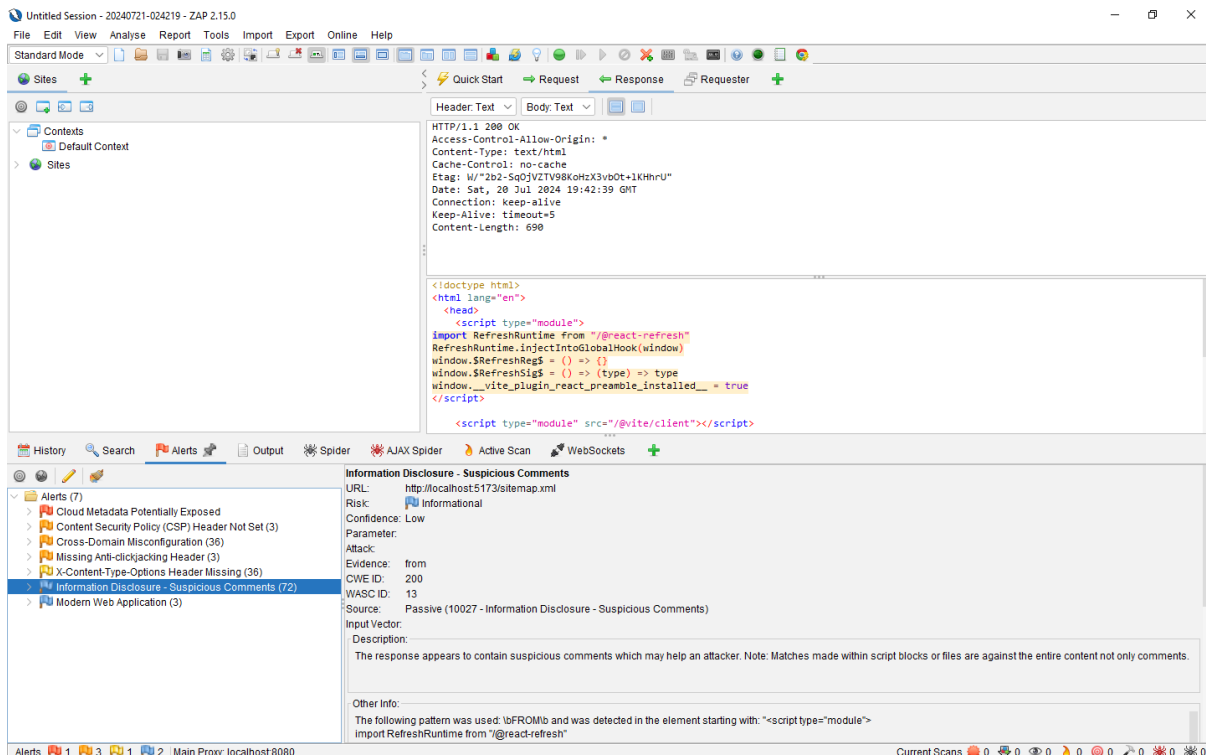
Gambar 4. 11 SQL Injection

a) Menggunakan SQL Injection

dari pengujian penetration test SQL Injection menggunakan alat seperti sqlmap. Dari hasil tersebut, alat ini mencoba berbagai teknik SQL Injection pada berbagai jenis database, termasuk Oracle, IBM DB2, MySQL, PostgreSQL, Microsoft SQL Server, dan lainnya. Teknik-teknik yang diuji meliputi time-based blind, error-based, UNION query, dan stacked queries. Setiap percobaan menghasilkan informasi mengenai apakah teknik

tersebut berhasil atau tidak dalam memanipulasi basis data target. Pada akhirnya, hasil menunjukkan bahwa parameter yang diuji, "Referer", tidak dapat di-inject dengan SQL Injection. Ini menunjukkan kemungkinan adanya mekanisme perlindungan seperti Web Application Firewall (WAF) yang aktif. Terdapat saran untuk meningkatkan parameter '--level' dan '--risk' untuk pengujian lebih mendalam, atau menggunakan opsi '--tamper' dengan berbagai teknik untuk mengelabui WAF, serta opsi '--random-agent' untuk mengganti agen pengguna.

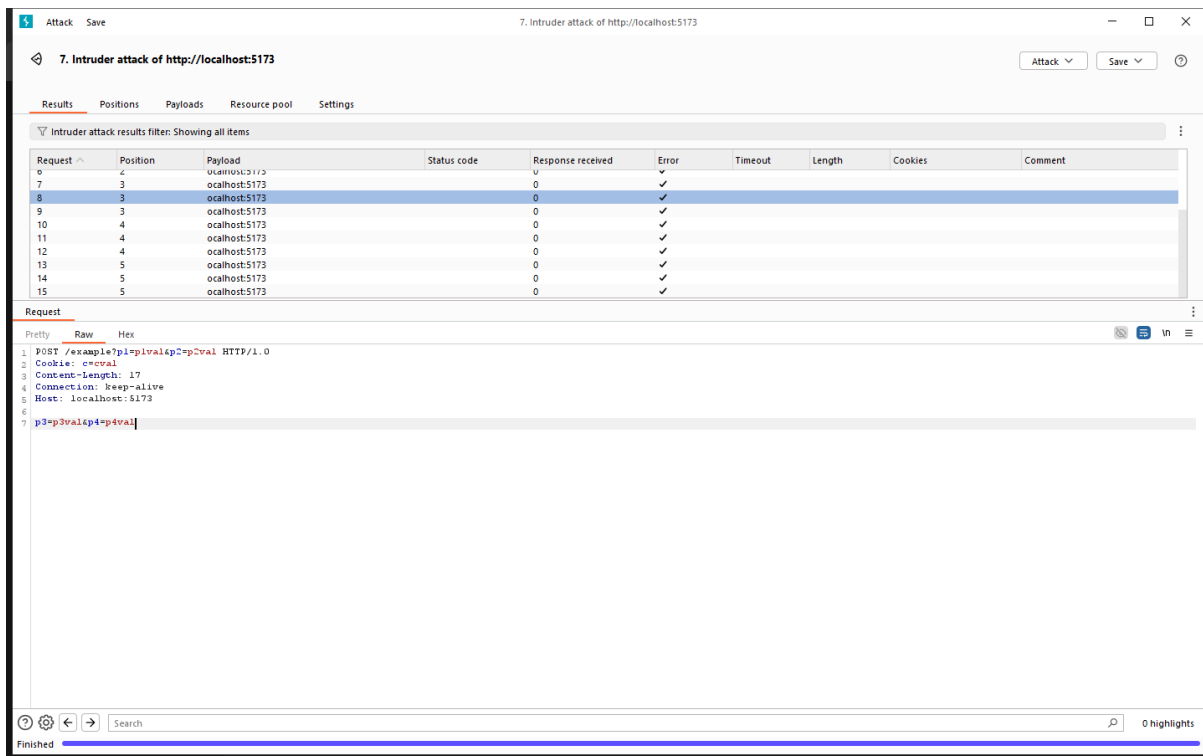
b) Menggunakan Owasp-Zap



Gambar 4. 12 Owasp-Zap

Hasil analisis keamanan menggunakan OWASP ZAP terhadap endpoint <http://localhost:5173/sitemap.xml> menunjukkan risiko rendah terkait Information Disclosure - Suspicious Comments. Temuan ini mengindikasikan bahwa respons dari server mengandung komentar mencurigakan yang dapat memberikan petunjuk kepada penyerang. Terdeteksi pola dalam elemen `<script>` yang menunjukkan penggunaan React Refresh, yang mungkin digunakan untuk pengembangan dan debugging. Disarankan untuk meninjau dan membersihkan semua komentar atau informasi sensitif dalam kode publik, serta memastikan tidak ada informasi debug yang tersedia yang dapat dieksploitasi. Audit keamanan berkala tetap diperlukan untuk menjaga keamanan aplikasi web.

c) Menggunakan *Burp-Suite*



Gambar 4. 13 Burp-Suite

Hasil serangan intruder Burp Suite terhadap `http://localhost:5173` menunjukkan semua permintaan dengan payload berbeda pada parameter p1, p2, p3, dan p4 menerima status kode 0, menandakan tidak ada respons dari server dan adanya masalah dalam pengiriman atau penerimaan data. Header permintaan menunjukkan pengaturan standar, dan body mengandung parameter `p3=p3val&p4=p4val`. Kesimpulannya, server mungkin tidak merespons atau ada mekanisme yang menghalangi permintaan, seperti firewall atau masalah konfigurasi server. Disarankan memeriksa log server dan memastikan tidak ada penghalang yang menghalangi permintaan.

4.4.4 Analisis dan Evaluasi

Hasil pengujian dievaluasi dan dianalisis untuk memastikan bahwa aplikasi web dapat memberikan prediksi diabetes dengan akurat dan memberikan pengalaman pengguna yang baik.

Pada tahap ini dilakukan evaluasi pada Web App dengan cara mengamati nilai prediksi dari pengidap diabetes dan yang tidak terkena diabetes dari dataset dengan cara melakukan input pada Web App dari dataset yang digunakan pada model *Neural network*.

Lalu disini didapatkan pola jika nilai prediksinya kurang dari 0.7 maka orang tersebut terkena diabetes lalu jika nilai prediksinya lebih dari 0.7 maka orang tersebut bebas diabetes setelah itu kita lakukan evaluasi pada bagian prediksinya dengan memberi aturan jika nilai prediksinya lebih dari 0.7 maka orang tersebut bebas diabetes.

Kemudian dilakukan lagi pengujian dan hasilnya cukup akurat untuk memprediksi diabetes dari dataset tadi.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan implementasi dan pengujian yang telah dilakukan sebelumnya, diperoleh hasil akurasi, f1-score, dan recall. Hasil-hasil tersebut membuktikan bahwa:

a) **Aplikasi Deteksi Diabetes Berbasis Neural Network Efektif:**

Aplikasi deteksi diabetes menggunakan algoritma neural network pada sistem ini terbukti efektif dalam mendiagnosis kondisi diabetes. Hasil akurasi yang tinggi menunjukkan potensi aplikasi untuk membantu pengguna meminimalisir risiko terkena penyakit tersebut.

b) **Identifikasi Cepat:**

Penerapan algoritma neural network pada aplikasi deteksi diabetes mempermudah identifikasi berbagai faktor dan gejala yang terkait dengan kondisi diabetes. Hal ini memungkinkan pengguna untuk lebih cepat mengenali potensi risiko diabetes, memungkinkan tindakan pencegahan yang lebih efektif.

c) **Pentingnya Penerapan Teknologi dalam Kesehatan:**

Keberhasilan aplikasi ini menegaskan pentingnya penerapan teknologi, khususnya algoritma neural network, dalam bidang kesehatan. Teknologi dapat menjadi alat yang efektif untuk mendukung diagnosis dan pencegahan penyakit.

5.2 Saran

Adapun saran yang ingin disampaikan oleh penulis untuk penelitian berikutnya antara lain :

a) **Uji Klinis dan Validasi Medis:**

Melibatkan lebih banyak uji klinis dan validasi medis akan meningkatkan kepercayaan serta penerimaan aplikasi ini dalam komunitas medis. Langkah ini juga dapat membuka peluang untuk integrasi yang lebih luas ke dalam praktik klinis sehari-hari.

b) Evaluasi Aspek Keamanan dan Privasi:

Penelitian yang mendalam mengenai aspek keamanan dan privasi dalam penggunaan aplikasi kesehatan sangatlah penting. Evaluasi yang menyeluruh terkait kebijakan privasi serta langkah-langkah keamanan harus diterapkan untuk memastikan perlindungan data pengguna.

c) Pengembangan Versi Aplikasi yang Lebih Mudah Diakses:

Agar manfaat aplikasi ini dapat dirasakan oleh lebih banyak orang, pengembangan versi yang lebih mudah diakses, seperti versi aplikasi seluler, perlu dipertimbangkan dengan serius.

d) Penelitian Lanjutan Mengenai Model Neural Network yang Optimal:

Penelitian lanjutan mengenai jenis dan konfigurasi model neural network yang paling optimal untuk deteksi diabetes dapat membantu meningkatkan kinerja dan akurasi aplikasi ini.

e) Kolaborasi dengan Tenaga Kesehatan dan Institusi Medis:

Kolaborasi dengan tenaga kesehatan dan institusi medis akan membuka peluang besar bagi penerapan aplikasi ini dalam lingkungan klinis yang lebih luas, sehingga dapat memberikan manfaat yang lebih signifikan dalam dunia kesehatan.

DAFTAR PUSTAKA

- Abdurrohman, H., Dini, R., & Muharram, A. P. (2018). Evaluasi Performa metode Deep Learning untuk Klasifikasi Citra Lesi Kulit The HAM10000. *Seminar Nasional Instrumentasi, Kontrol dan Otomasi (SNIKO)*.
- Admojo, F. T., & Ahsanawati. (2020). Klasifikasi Aroma Alkohol Menggunakan Metode KNN. *Indonesian Journal of Data and Science*, 34-38.
- Aprilia, W., & dkk. (2021). Prediksi Kemungkinan Diabetes pada Tahap Awal. *Sistemasi: Jurnal Sistem Informasi*, 163-171.
- Ataline Jeanethe Maya Hukubun. (2022).
- Aulia, A., Aprianti, B., Supriyanto, Y., & Rozikin, C. (2022). prediksi harga emas dengan menggunakan algoritma support vector regression (Svr) dan linear regression (lr). *jurnal ilmiah wahana pendidikan*, 84-88.
- Bhakti, H. D. (2019). Aplikasi Artificial Neural Network (ANN) untuk Memprediksi Masa Studi Mahasiswa Program Studi Teknik Informatika Universitas Muhammadiyah Gresik. *Jurnal Eksplora Informatika*, 88-95.
- Bustami. (2014). Penerapan Algoritma Naive Bayes Untuk Mengklasifikasi Data Nasabah Asuransi. *Jurnal Informatika* , 884-898.
- Cahyani, Q., & dkk. (2022). Prediksi Risiko Penyakit Diabetes menggunakan Algoritma Regresi Logistik. *Journal of Machine Learning and Artificial Intelligence*, 107-114.
- Collonov, F., Dafna, I., Ivanov, P., & Eric, C. (2023, July 4). *Project Jupyter's origins and governance*. Retrieved from Jupyter: <https://jupyter.org/>
- Dagliati, A., Marini, S., & dkk. (2018). Machine Learning Methods to Predict Diabetes. *Journal of diabetes science and technology*, 295-302.
- Eska, J. (2016). Penerapan Data Mining Untuk Prediksi Penjualan Wallpaper Menggunakan Algoritma C4.5. *JURTEKSI (Jurnal Teknologi dan Sistem Informasi)*, 9-13.
- Gulli, A., & Pal, S. (2017). *Deep Learning With Keras*. Brimingham: Packt Publishing.
- Hadianto, N., Novitasari, H. B., & Rahmawati, A. (2019). Klasifikasi Peminjaman Nasabah Bank Menggunakan Metode Neural Network. *PILAR Nusa Mandiri*, 163-170.

- Handayani, M., Riandini, M., & Situmorang, Z. (2022). Perbandingan Fungsi Optimasi Neural Network Dalam Klasifikasi Kelayakan Calon Suami. *Jurnal Informatika*, 78-84.
- Hasan, K., Alam, A., Hussein, D., & Hasan, M. (2020). Diabetes Prediction Using Ensembling of Different Machine Learning Classifiers. *IEEE Access* 8, 76516-76531.
- Hunter, J., Dale, D., Firing, E., & Droettboom, M. (2023, february 13). *Library matplotlib*. Retrieved from matplotlib: https://matplotlib.org/stable/users/release_notes.html
- Istighfarizky, F., ER, N. A., Widiartha, I. M., Astutia, L. G., Putra, I. G., & Suhartana, I. K. (2022). Klasifikasi Jurnal menggunakan Metode KNN dengan Mengimplementasikan Perbandingan Seleksi Fitur. *Jurnal Elektronik Ilmu Komputer Udayana*, 167-176.
- Kaur, A., & Rout, M. (2020). Evaluating Machine Learning Methods for Predicting Diabetes among Female Patients in Bangladesh. *International Conference on Intelligent Engineering and Management (ICIEM)*, 1-20.
- Munawarah, R., Soesanto, O., & Faisal, M. R. (2016). Penerapan Metode Support Vector Machine Pada Diagnosa Hepatitis. *Kumpulan Jurnal Ilmu Komputer (KLIK)*, 103-113.
- Nagaraj, P., & Deepalakshmi, P. (2021). Diabetes Prediction Using Diabetes Prediction Using Enhanced SVM and Deep Neural Network Learning Techniques: An Algorithmic Approach for Early Screening of Diabetes. *International Journal of Healthcare Information Systems and Informatics (IJHISI)*, 1-20.
- Nasution, M. R., & Hayaty, M. (2019). Perbandingan Akurasi dan Waktu Proses Algoritma K-NN dan SVM dalam Analisis Sentimen Twitter. *JURNAL INFORMATIKA*, 6(2), 212-218.
- Naz, H., & Ahuja, S. (2020). Deep learning approach for diabetes prediction using PIMA Indian dataset. *Journal of Diabetes & Metabolic Disorders*, 391-403.
- Nelli, F. (2015). *Data Analysis and Science Using Pandas, matplotlib, and the Python Programming Language*.
- Nelli, F. (2015). *Python Data Analytics*. New York: Apress Media.
- Noviyanto. (2020). Penerapan Data Mining dalam Mengelompokkan Jumlah Kematian. *Jurnal Informatika dan Komputer*, 183-188.
- Patra, R., & Khuntia, B. (2021). Analysis and Prediction Of Pima Indian Diabetes Dataset Using SDKNN Classifier Technique. *IOP Conference Series: Materials Science and Engineering*, 1-14.

- R, C., Harris, Millman, K. J., & Vander, J. W. (2020, sep). Array programming with NumPy. *Nature*, 585, 357-362. Retrieved from <https://doi.org/10.1038/s41586-020-2649-2>
- Ramadhan, N. A. (2019). Penerapan Metode Neural Network Untuk Prediksi Nilai Ujian Nasional (Study Kasus Di Smk Muhammadiyah Slawi). *Jurnal Ilmiah Indonesia*, 118-130.
- Rohman, y. A. (2019, 12 8). *Pengenalan NumPy, Pandas, Matplotlib*. Retrieved from medium.com: <https://medium.com/@yasirabd/pengenalan-numpy-pandas-matplotlib-b90bafd36c0>
- SA, A., JJV, N., & H, S. (2020). Deep convolutional neural network for diabetes mellitus prediction. *Neural Computing and Applications*, 1319-1327.
- Soni, M., & Varma, S. (2020). Diabetes Prediction using Machine Learning Techniques. *International Journal of Engineering Research & Technology (ijert)*, 921-925.
- Syarovy, M., & Sutiarso, A. P. (2023). Pemanfaatan Model Neural Network Dalam Generasi Baru Pertanian Presisi di Perkebunan Kelapa Sawit. *Warta PPKS*, 39-54.
- VanderPlas, J. (2017). *Python Data Science Handbook*. United States of America: O'Reilly Media, Inc.,.
- Wadi, H. (2015). *Pemrograman Python : untuk pelajar dan mahasiswa*. Bandung: TR Publisher.
- Waskom, M. L. (2021). seaborn: statistical data visualization. *Journal of Open Source Software*, 6, 3021. Retrieved from <https://doi.org/10.21105/joss.03021>
- Widiputra, H. D. (2016).