

**APLIKASI PENDETEKSI PENYAKIT DIABETES  
MENGUNAKAN MACHINE LEARNING DENGAN  
ALGORITMA K-NEAREST NEIGHBOR(K-NN) DI  
PUSKESMAS KENANGA**

**PROYEK AKHIR**

Laporan akhir ini dibuat dan diajukan untuk memenuhi salah satu syarat kelulusan Sarjana Terapan/Diploma IV Politeknik Manufaktur Negeri Bangka Belitung



Disusun Oleh :

Imam Rayhan Ekadinata

NIRM : 1061810

**POLITEKNIK MANUFAKTUR NEGERI  
BANGKA BELITUNG  
TAHUN 2023**

## LEMBAR PENGESAHAN

### JUDUL PROYEK AKHIR

Aplikasi Pendeteksi Penyakit Diabetes Menggunakan Machine Learning Dengan Algoritma K-Nearest Neighbor(K-NN) di Puskesmas Kenanga

Oleh :

Imam Rayhan Ekadinata / 106810

Laporan akhir ini telah disetujui dan disahkan sebagai salah satu syarat kelulusan Program Sarjana Terapan/Diploma III Politeknik Manufaktur Negeri Bangka

Belitung  
Menyetujui,

Pembimbing 1



Ahmat Josi, S.Kom, M.Kom

NIP: 198908202019031015

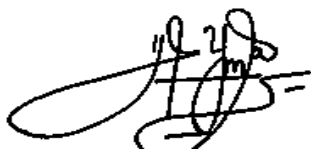
Pembimbing 2



Linda Fujiyanti, S.T, M.T.I.

NIP: 198109262014042001

Penguji 1



Yang Agita Rindri, S.Kom., M.Eng.

NIP: 198609282022032003

Penguji 2



Sari Mubaroh, M.Pd

NIP:198501122019032015

## PERNYATAAN BUKAN PLAGIAT

Yang bertanda tangan di bawah ini :

Nama Mahasiswa : Imam Rayhan Ekadinata

NIRM : 1061810

Dengan Judul : Aplikasi Pendeteksi Penyakit Diabetes Menggunakan Machine Learning Dengan Algoritma K-Nearest Neighbor(K-NN) di Puskesmas Kenanga

Menyatakan bahwa laporan akhir ini adalah hasil kerja kami sendiri dan bukan merupakan plagiat. Pernyataan ini kami buat dengan sebenarnya dan bila ternyata dikemudian hari ternyata melanggar pernyataan ini, kami bersedia menerima sanksi yang berlaku.

Sungailiat, 16 Januari 2023

Nama Mahasiswa

Tanda Tangan

Imam Rayhan Ekadinata



## ABSTRAK

Kesejahteraan penting dalam hidup. Jika kesejahteraan terganggu maka akan terganggu. Salah satu Penyakit luar biasa yang mengganggu manusia adalah Diabetes. Berdasarkan data Worldwide Diabetes Alliance, Indonesia berada di peringkat ke-7 dunia dengan lebih dari 10 juta orang menderita Diabetes Mellitus. Organisasi Kesejahteraan Dunia memperkirakan penderita di Indonesia sampai dengan 21 juta orang pada tahun 2030. Pengecekan diabetes di puskesmas kenanga masih dilakukan secara manual sehingga dibuatlah deteksi diabetes berbasis machine learning agar dapat membantu mempermudah pekerjaan khususnya dalam deteksi diabetes itu sendiri.

Perhitungan K-NN / Nearest Neighbor adalah perhitungan yang dapat dimanfaatkan untuk mengklasifikasikan berdasarkan bagian terbesar dari nilai tetangga yang paling dekat. Maksud dari perhitungan ini adalah untuk mengklasifikasikan benda-benda yang tidak terpakai berdasarkan kualitas dan pengujian dari penyusunan informasi. Dengan perhitungan K-Nearest Neighbor akan digunakan untuk meramalkan Diabetes Melitus berdasarkan informasi yang sesuai dengan pengguna. Hasil ramalannya akan berada dalam bingkai kelas positif dan negatif.

Kerangka kerja ini dicoba 3 kali. Jumlah keseluruhan informasi yang digunakan adalah 768 data. Dari pengujian yang dilakukan, diketahui bahwa presisi paling tinggi terdapat pada pengujian pertama, yaitu dengan presisi sebesar 84%. Pada pengujian awal ini, perbandingan penyiapan informasi dan pengujian informasi digunakan 90% dengan 10% (691 training dan 77 uji).

Kata kunci : Diabetes, Prediksi, K-NN, Machine Learning

## **ABSTRACT**

*Prosperity is important in life. If welfare is disturbed, it will be disturbed. One of the irresistible illnesses that disturbs human exercises is Diabetes. Based on information from the Around the world Diabetes Organization together, Indonesia is positioned 7th within the world with more than 10 million individuals enduring from Diabetes Mellitus. The World Welfare Organization gauges that there will be up to 21 million sufferers in Indonesia in 2030. Diabetes checks at Ylanganga health centers are still done manually, so machine learning-based diabetes detection was created to help make work easier, especially in detecting diabetes itself.*

*The K-NN / Nearest Neighbor calculation is a calculation that can be used to classify based on the largest portion of the closest neighbor values. The purpose of this calculation is to classify unused objects based on quality and testing of the preparation of information. With the K-Nearest Neighbor calculation it'll be utilized to anticipate Diabetes Mellitus based on data fitting to the client. The expectation results will be within the positive and negative course outlines.*

*This system was attempted 3 times. The total amount of information used is 768 data. From the tests carried out, it is known that the highest precision was found in the first test, namely with a precision of 84%. In this initial test, the ratio of information preparation and information testing used was 90% to 10% (691 training and 77 tests).*

*Keywords : Diabetes, Predict, K-NN, Machine Learning*

## KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah yang telah melimpahkan rahmat serta hidayah-Nya, tidak lupa shalawat serta salam selalu penulis panjatkan kepada junjungan kita Nabi Muhammad SAW, yang telah menuntun umatnya sehingga penulis dapat menyelesaikan proyek akhir dengan baik.

Salah satu syarat utama untuk menyelesaikan program sarjana terapan pada Program Studi Rekayasa Perangkat Lunak Politeknik Manufaktur Negeri Bangka Belitung adalah tugas akhir yang berjudul "Aplikasi Deteksi Diabetes Menggunakan Machine Learning With the K-Nearest Neighbor (K- NN) Algoritma di Puskesmas Kenanga." Bantuan berbagai pihak sangat diperlukan untuk keberhasilan penyelesaian proyek akhir ini. Oleh karena itu, penulis ingin menyampaikan rasa hormat dan terima kasih kepada:

1. Bapak I Made Andik Setiawan, M.Eng., Ph.D. selaku Direktur Poliklinik Manufaktur Negeri Bangka Belitung.
2. Bapak Ahmat Josi, S.Kom, M.Kom. selaku ketua Program Studi Teknologi Rekayasa Perangkat Lunak Politeknik Manufaktur Negeri Bangka Belitung dan Dosen Pembimbing 1 yang telah membantu penulis dengan saran dan waktunya.
3. Ibu Linda Fujiyanti, S.T, M.T.I. selaku pembimbing 2 yang juga telah membantu penulis dengan saran dan waktunya.

Penulis menyadari sepenuhnya masih terdapat kekurangan, maka dari itu kritik dan saran serta masukan dari berbagai pihak akan penulis terima dengan lapang. Semoga proyek akhir yang sederhana ini dapat bermanfaat bagi pihak yang membutuhkan.

Sungailiat, 16 Januari 2023

Imam Rayhan Ekadinata

## DAFTAR ISI

LEMBAR PENGESAHAN .....	ii
PERNYATAAN BUKAN PLAGIAT .....	iii
ABSTRAK.....	iv
<i>ABSTRACT</i> .....	v
KATA PENGANTAR.....	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xi
DAFTAR LAMPIRAN .....	xii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah .....	2
1.4 Tujuan .....	3
1.5 Manfaat.....	3
BAB II LANDASAN TEORI.....	5
2.1 Kajian Pustaka .....	5
2.1 Machine Learning.....	8
2.2 Diabetes Mellitus .....	9
2.3 K-Nearest Neighbor(K-NN).....	10
2.4 Euclidean Distance .....	13
2.5 Normalisasi Data .....	13
2.6 Confusion Matrix.....	14
2.7 Flowchart .....	15
2.8 Python .....	15
2.9 Library Pada Python .....	16
2.10 Dataset .....	17
2.11 Jupyter Notebook.....	19

2.12 Visual Studio Code .....	19
BAB III METODE PELAKSANAAN .....	20
3.1 Alur Penelitian.....	20
3.2 Flowchart Proses.....	22
3.3 Use Case Diagram .....	23
3.4 Activity Diagram.....	24
3.5. Analisis Kebutuhan.....	24
3.5.1 Analisis Kebutuhan Fungsional .....	24
3.5.2 Analisis Kebutuhan Non Fungsional.....	25
3.6 Analisis Prediksi Dengan Metode K-Nearest Neighbor(K-NN) .....	25
3.7 K Nearest-Neighbor.....	27
3.8 Normalisasi Data .....	28
3.9 Euclidean Distance atau Jarak.....	31
3.10 Menghitung Nilai Tetangga atau K.....	32
3.11 Evaluasi .....	34
3.12 Analisis Kebutuhan Sistem .....	37
3.12.1 Perancangan Antarmuka Halaman Beranda.....	37
3.12.2 Perancangan Antarmuka Halaman Informasi .....	38
3.12.3 Perancangan Antarmuka Halaman Diagnosis .....	38
3.12.4 Perancangan Antarmuka Halaman Hasil Prediksi .....	40
3.12.5 Perancangan Antarmuka Halaman Tentang .....	41
BAB IV PEMBAHASAN .....	42
4.1 Deskripsi.....	42
4.2 Implementasi Model .....	42
4.2.1 Pembuatan Library.....	42
4.2.2 Pemuatan Data .....	44
4.2.3 Pendeskripsian Data.....	44
4.2.4 Pengecekan Data.....	45
4.2.5 Ubah Data .....	45
4.2.6 Pemisahan Variabel Hasil.....	46
4.2.7 Normalisasi Data .....	47
4.2.8 Split Data .....	48
4.2.9 Modeling KNN .....	49
4.2.10 Evaluasi .....	49



4.3 Implementasi Pembuatan Sistem .....	51
4.3.1 Penyimpanan Pemodelan .....	51
4.3.2 Penginstalan Kebutuhan Interface.....	52
4.3.3 Pembuatan Python.....	53
4.3.4 Halaman Beranda.....	55
4.3.5 Halaman Informasi.....	56
4.3.6 Halaman Prediksi.....	57
4.3.7 Halaman Tentang .....	58
4.3.8 Halaman Hasil Prediksi .....	58
BAB V PENUTUP .....	59
5.1 Kesimpulan.....	59
5.2 Saran .....	59
DAFTAR PUSTAKA.....	60
LAMPIRAN .....	62
Data Pribadi.....	62
Pendidikan .....	62

## DAFTAR TABEL

Tabel 2.1 Penelitian Pembeding.....	6
Tabel 2.2 Penelitian Pembeding (lanjutan) .....	8
Tabel 2.3 Gejala Penyakit .....	9
Tabel 2.4 Kinerja Klasifikasi .....	14
Tabel 2.5 Fitur Python.....	16
Tabel 2.6 Variabel.....	18
Tabel 3.1 Perangkat lunak membangun sistem .....	25
Tabel 3.2 Perangkat keras membangun sistem .....	25
Tabel 3.3 Model Data Training .....	26
Tabel 3.4 Model Data Testing .....	28
Tabel 3.6 Nilai minimal dan maksimal data uji.....	28
Tabel 3.7 Model Data Latih dan Data Uji .....	29
Tabel 3.8 Model data uji setelah normalisasi data.....	30
Tabel 3.9 Model data latih setelah normalisasi data .....	30
Tabel 3.10 Hasil dari perhitungan <i>Euclidean Distance</i> .....	31
Tabel 3.11 Hasil Ujian K = 5 .....	32
3.12 Tabel Jumlah Kelas dan Hasil Prediksi .....	33
3.13 Tabel Contoh hasil perbandingan prediksi dengan aktual .....	34
3.14 Tabel <i>Confusion Matrix</i> .....	35
3.15 Tabel <i>Confusion Matrix</i> dengan perhitungan Excel .....	35

## DAFTAR GAMBAR

Gambar 3.3 Use Case Diagram.....	23
Gambar 3.4 Activity Diagram .....	24
Gambar 3.6 Rancangan antarmuka halaman proses diagnosis .....	39
Gambar 4.1 Penginstallan Numpy.....	43
Gambar 4.2 Import library Python yang dibutuhkan .....	43
Gambar 4.3 Memasukan dataset pada model python .....	44
Gambar 4.4 Menampilkan informasi dari dataset .....	44
Gambar 4.5 Mencari nilai yang kosong pada data.....	45
Gambar 4.6 Mengubah data 0 menjadi nilai rata-rata .....	45
Gambar 4.7 Memisahkan data menjadi x dan y.....	46
Gambar 4.8 Normalisasi Data .....	47
Gambar 4.9 Pembagian data uji dan data latih.....	48
Gambar 4.10 Pembuatan KNN .....	49
Gambar 4.11 Menampilkan tabel Confusion Matrix.....	50
Gambar 4.12 Label Confusion Matrix .....	50
Gambar 4.13 Hasil Klasifikasi .....	51
Gambar 4.14 Penyimpanan Model .....	52
Gambar 4.15 Penginstallan virtualenv .....	52
Gambar 4.16 Penginstallan Flask .....	53
Gambar 4.20 Halaman Informasi .....	56
Gambar 4.23 Halaman Hasil Prediksi .....	58

## DAFTAR LAMPIRAN

1. Data Pribadi .....	62
2. Pendidikan .....	62



# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Suatu penyakit dimana tubuh tidak dapat menyerap atau mengolah gula dalam darah disebut juga dengan penyakit diabetes melitus atau kencing manis. Kondisi normal tubuh kemudian memungkinkan produksi insulin secara otomatis, yang menjaga kadar gula darah di atas batas normal. Diabetes akan berkembang jika salah satu sel dalam tubuh tidak merespon insulin atau tubuh tidak memproduksi cukup hormon insulin. Keinginan untuk buang air kecil secara berlebihan, rasa haus yang terus menerus, dan rasa lapar yang ekstrim adalah gejala yang paling umum[1].

Menurut informasi terbaru yang diberikan oleh International Diabetes Federation (IDF), Indonesia menempati urutan ketujuh dari sepuluh negara dengan proporsi penderita diabetes tertinggi[2]. Di Indonesia, prevalensi diabetes mencapai 6,2%, setara dengan 10,8 juta kasus diabetes per tahun pada tahun 2020. Prevalensi diabetes biasanya lebih tinggi pada laki-laki dibandingkan perempuan[2]. Hal ini karena penduduk setempat yang ingin lebih sehat telah mengubah rutinitas dan cara hidup mereka.

Sebagai solusi dari penyakit diabetes itu sendiri, diperlukan tindakan pada banyak kasus diabetes di Indonesia dan di seluruh dunia. Ini mutlak membutuhkan keputusan medis yang dibuat berdasarkan pengalaman dan logika medis. Menggunakan beberapa data pasien diabetes yang telah disimpan dalam dataset untuk membuat pola menggunakan klasifikasi dari bahasa mesin sehingga hasil yang akan diperoleh tidak jauh berbeda dan dapat dihindari dengan tepat merupakan salah satu cara agar prediksi itu sendiri dapat dilakukan. Dari prediksi tepat yang dibuat dengan algoritma K-Nearest Neighbor (KNN), sebuah inovasi dan kontribusi untuk teknologi perencanaan masa depan dengan menggunakan algoritma yang tepat.

Selain menggunakan algoritma K-NN juga menggunakan metode Machine learning. Machine learning merupakan mesin yang dapat bekerja sama dengan cara melakukan pendekatan kepada manusia, yang berguna menyelesaikan akar permasalahan dengan cara yang terukur. Dari pembahasan ini diketahui bahwa algoritma KNN dapat digunakan untuk memprediksi diabetes melitus, berdasarkan set data yang tersedia, diharapkan algoritma KNN akan memberikan ketepatan dalam hal memprediksi diabetes melitus secara akurat[3].

Puskesmas Kenanga merupakan salah satu puskesmas atau bentuk layanan kesehatan yang berada di kabupaten bangka. Berdasarkan hasil survey atau kunjungan bahwa di puskesmas kenanga masih menggunakan cara manual sehingga

memakan waktu sedikit lebih lama untuk menunggu mengetahui hasil dan harus melakukan perbandingan uji terlebih dahulu antara data yang ada dan data yang diujikan membutuhkan waktu yang cukup lama, oleh karena itu dilakukan pembuatan aplikasi deteksi diabetes ini berbasis machine learning yang diharapkan dapat membantu mempermudah pekerjaan di puskesmas khususnya untuk mendeteksi penyakit diabetes melitus tersebut, dengan menerapkan machine learning dimana terdapat dataset sebagai model awal yang akan dicek dengan data uji untuk menentukan performa dan kebenaran dalam korelasi yang dibuat.

Aplikasi Pendeteksi Diabetes ini dengan menerapkan Machine Learning dengan menggunakan algoritma K-Nearest Neighbor ini dapat membantu yang selama ini prediksi atas deteksi yang manual dan melalui tahapan yang panjang dengan berapa langkah-langkah dapat diautomatisasi dengan mengandalkan sebuah dataset awal untuk membuat sebuah model dan melakukan data uji hingga dapat mengetahui prediksi deteksi penyakit diabetes ini lebih baik berdasar data-data yang telah tersedia. Atas dasar diatas, saya bermaksud mengangkat masalah tersebut kedalam judul proyek akhir saya yaitu “Prediksi Penyakit Diabetes Menggunakan Machine Learning Dengan Algoritma K- Nearest Neighbor(K-NN).

## **1.2 Rumusan Masalah**

Berdasarkan latar belakang masalah di atas dapat dirumuskan suatu permasalahan, yaitu :

1. Apakah algoritma K-Nearest Neighbor(K-NN) dapat memprediksi adanya penyakit Diabetes Mellitus
2. Berapa tingkat akurasi yang dihasilkan dari metode algoritma K-Nearest Neighbor(K-NN) dalam memprediksi penyakit Diabetes Mellitus ini?
3. Bagaimana cara memperoleh dataset untuk metode algoritma K-Nearest Neighbor(K-NN)?

## **1.3 Batasan Masalah**

1. Pendeteksi dari penyakit ini berbasis sebuah website, dengan menggunakan Bahasa pemrograman Python dan HTML untuk pembuatan sistemnya.

2. Metode dan algoritma yang digunakan untuk memprediksi diabetes dengan metode algoritma K-NN.
3. Metode evaluasi akurasi hasil dengan klasifikasi menggunakan Confusion Matrix.
4. Hasil dari prediksi awal penyakit Diabetes Mellitus adalah dengan menampilkan nilai positif dan negatif.
5. Jumlah dataset digunakan 768 data penyakit diabetes

#### **1.4 Tujuan**

Berdasarkan latar belakang tersebut, tujuan dari adanya pembuatan pendeteksi penyakit diabetes menggunakan machine learning adalah

1. Merancang Pendeteksi Penyakit Diabetes dengan Menggunakan Machine Learning
2. Menerapkan Metode Machine Learning dalam Pembuatan Aplikasi
3. Menerapkan Algoritma K-NN dalam Pembuatan Aplikasi

#### **1.5 Manfaat**

##### **A. Bagi Masyarakat**

Memudahkan masyarakat untuk memprediksi penyakit Diabetes, dan kemudian dapat melakukan pemeriksaan sedini mungkin.

##### **B. Bagi Ilmu Pengetahuan**

1. Dapat mengimplementasi algoritma K-Nearest Neighbor dalam pemrograman Python.

2. Mengetahui cara kerja algoritma K-Nearest Neighbor dalam memprediksi suatu kondisi.





## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Kajian Pustaka**

Beberapa hipotesis investigasi terkait pemeriksaan akan diuraikan dalam bab ini. Hipotesis penelitian terdahulu yang telah dilakukan akan dijadikan acuan penting untuk membatasi penerapan spekulasi terkait dalam penelitian. Referensi terkait dan memanfaatkan dataset yang sama, khususnya dataset dari Kaggle yang berjudul Pima Indians Diabetes Dataset adalah sebagai berikut:

Penelitian dilakukan oleh Umikulsum Indah Lestari, Anis Yusrotun Nadhiroh, dan Cahyuni Novia pada tahun 2021 dengan judul “Penerapan Strategi K-Nearest Neighbor untuk Kerangka Pilihan Kembali untuk Bukti Diakui Diabetes Mellitus”. Dapat memberikan kesimpulan berdasarkan penggunaan dan pengujian, dengan harga  $k = 23$  menghasilkan tingkat ketepatan sebesar 0,96 atau 96%, tingkat ketelitian tersebut dianggap cukup tinggi sehingga alamat ini dianggap menarik dalam mewujudkan penggunaan K-NN Pendekatan Sistem Pengambilan Pilihan yang Mendukung untuk mengenali bukti awal diabetes [4].

Investigasi yang dilakukan oleh Roshi Saxena, Dr. Sanjay Kumar Sharma, dan Manali Gupta pada tahun 2021 dengan judul “Bagian K-tetangga terdekat di lokasi Diabetes Mellitus”. Kesimpulan: Diabetes Mellitus merupakan penyakit yang dapat terjadi pada siapa saja yang mengalami kelebihan berat badan, memiliki gaya hidup yang tidak sehat, terlalu banyak beban kerja, dan fokus. Dalam perkiraan yang dibuat menggunakan strategi K-Nearest Neighbor, yang merupakan perhitungan pembelajar apatis dengan menerapkan strategi preprocessing dan mengoptimalkan parameter pengklasifikasi apatis, Anda akan dapat menemukan nilai presisi tertinggi sebesar 8,48%, yang sudah mendapatkan ketepatan hasil sebesar 70.1% dengan memanfaatkan nilai  $k = 45$ , namun setelah dilakukan pre-processing karena adanya informasi yang hilang pada informasi tersebut, maka ketepatan akhir menjadi sebesar 78.58% [5].

Tanyakan tentang yang dilakukan oleh Raharjo Putra Kurniadi, Rd. Rohmad Saedudin, dan Vandha Pradwiyasma Widartha pada tahun 2021 dengan judul “Perbandingan Ketepatan Perhitungan K-Nearest Neighbor dan Perhitungan Relapse untuk Klasifikasi Diabetes”. Memberikan kesimpulan bahwa perhitungan KNN menghasilkan valuasi ketepatan sebesar 85,06% dibandingkan dengan perhitungan Calted Backslide dengan valuasi sebesar 77,92%. Dalam perhitungan valuasi AUC, valuasi perhitungan KNN sebesar 0,915 dan perhitungan Perhitungan Backslide sebesar 0,858. Tampaknya perhitungan K-Nearest Neighbor menghasilkan nilai akurasi dan eksekusi yang jauh lebih unggul dibandingkan perhitungan Calted Backslide dalam mengklasifikasikan diabetes. [6].

Dengan referensi yang dijelaskan di atas, Pertimbangan ini akan

menggunakan perhitungan K-Nearest Neighbor yang dikombinasikan dengan teknik Perplexity untuk menentukan ketepatan sistem. Dan penelitiannya menggunakan data yang sama, yang berasal dari Kaggle dengan judul Pima Indians Database. Untuk prognosis penyakit Diabetes Melitus akan dibedakan menjadi dua kelompok yaitu positif dan negatif. Pada tabel 2.1 dibawah adalah tabel penelitian sebelumnya.

**Tabel 2.1 Penelitian Perbandingan**

No	Judul Penelitian	Hasil Penelitian	Perbandingan
1.	Penerapan Metode <i>K-Nearest Neighbor</i> Untuk Sistem Pendukung Keputusan Identifikasi Penyakit <i>Diabetes Melitus</i> [4].	Penelitian ini mendapatkan hasil akurasi tertinggi yaitu sebesar 96% dengan menggunakan nilai $k = 23$ , dataset yang digunakan berjumlah 394 data, dibagi menjadi 50 data uji dan 344 data latih. Kemudian pada pembuatan sistem nya belum dikembangkan dalam bentuk web.	Penelitian ini akan menggunakan 768 data, yang akan dibagi menjadi 691 data latih dan 77 data uji. Pada pengujian sistem akan dibuat web menggunakan Bahasa python dan HTML.

2.	Role of K-nearest neighbor in detection of Diabetes Mellitus [5].	<p>Penelitian ini mendapatkan hasil akurasi tertinggi sebesar 78,58%. Pada perhitungan fungsi jarak pada pengguna K-NN menggunakan fungsi <i>Manhattan Distance</i>.</p>	<p>Penelitian ini akan menggunakan perhitungan fungsi pengukuran jarak yaitu dengan <i>Euclidean Distance</i>, dikarenakan pada pengukuran ini dapat diterapkan untuk berbagai dimensi.</p>
----	---	--	---

**Tabel 2.2 Penelitian Perbandingan (lanjutan)**

No	Judul Penelitian	Hasil Penelitian	Perbandingan
3.	Perbandingan Akurasi Algoritma <i>K-Nearest Neighbor</i> dan <i>Logistic Regression</i> Untuk Klasifikasi Penyakit Diabetes [6].	Penelitian ini mendapatkan hasil akurasi tertinggi dengan menggunakan K-NN yaitu sebesar 85,06%, untuk mengukur parameter tertinggi pada penelitian ini telah menggunakan <i>library GridSearchCV</i> yaitu mencari nilai <i>k</i> tertinggi secara otomatis ( <i>tuning</i> ).	Penelitian ini tidak menggunakan <i>tuning</i> , dikarenakan pada penginputan nilai <i>k</i> secara manual agar lebih dapat memilih nilai <i>k</i> tertinggi dan mendapatkan hasil secara optimal tanpa ada batasan.

### 2.1 Machine Learning

Machine Learning atau pembelajaran mesin merupakan suatu cabang dari ilmu Artificial Intelligence (AI) atau dikenal sebagai Kecerdasan Buatan. Peran machine learning sangat banyak membantu manusia dalam berbagai bidang, Machine Learning sendiri berfokus kepada pengembangan suatu sistem yang mampu belajar sendiri serta mampu memeriksa data yang ada dan melaksanakan tugas-tugas tertentu sesuai dengan apa yang dipelajarinya tanpa arahan pengguna., tanpa harus berulang kali diprogram oleh manusia[3]. Kemampuan dari Machine Learning dapat secara otomatis membaca data yang diberikan serta dapat

meningkatkan kinerja dari pengguna tanpa harus secara eksplisit ditulis kembali pada program komputernya[2].

## 2.2 Diabetes Mellitus

Kadar gula darah (glukosa) yang tinggi merupakan ciri khas diabetes melitus, suatu kondisi kronis. Sumber sel terpenting dalam tubuh manusia adalah glukosa. Apabila glukosa yang menumpuk didalam darah dikarenakan sel tubuh tidak dapat menyerap dengan baik maka akan menyebabkan berbagai gangguan didalam tubuh. Apabila tidak dapat ditangani dengan baik atau dikendalikan, maka dapat menyebabkan bahaya tubuh penderita hingga berujung pada nyawa dari penderita[1].

Pankreas, organ di belakang perut, bertanggung jawab untuk memproduksi kadar gula darah. Pankreas tidak mampu memproduksi insulin sesuai kebutuhan pada penderita diabetes melitus. Sel-sel tubuh tidak akan mampu mengubah glukosa menjadi energi jika tidak ada insulin di dalam tubuh. Ada 2 faktor genetik, yang menyebabkan tubuh memproduksi insulin tidak cukup atau tidak sama sekali, adalah penyebab kadar insulin kita menjadi rendah atau berkurang. Seperti infeksi dan faktor usia. Kemudian faktor kedua yaitu tubuh yang terus memproduksi hormon insulin, namun tidak bekerja secara baik dikarenakan tubuh mengalami resistensi atau kebal terhadap insulin. Contoh kasusnya seperti yang dialami pada orang yang obesitas atau gemuk[1].

**Tabel 2.3** Gejala Penyakit

NO.	GEJALA PENYAKIT
1.	Sering merasa haus
2.	Sering buang air kecil, terutama di malam hari
3.	Sering merasa sangat lapar
4.	Berat badan turun secara tiba-tiba tanpa sebab yang jelas
5.	Berkurangnya massa otot

6.	Terdapat (keton) hasil sisa pemecahan otot dan lemak dikarenakan tubuh tidak dapat menggunakan gula sebagai sumber energi tubuh yang terdapat pada urine.
7.	Lemas
8.	Pandangan kabur
9.	Luka yang sulit sembuh
10.	Seringkali merasakan infeksi, misalnya di gusi, kulit, vagina, atau saluran kemih. Beberapa gejala lain juga bisa menjadi ciri-ciri bahwa seseorang mengalami diabetes
11.	Mulut kering
12.	Rasa terbakar, kaku, dan nyeri pada kaki
13.	Gatal-gatal
14.	Disfungsi ereksi atau impotensi
15.	Mudah tersinggung
16.	Mengalami hipoglikemia reaktif, yaitu hipoglikemia yang terjadi beberapa jam setelah makan akibat produksi insulin yang berlebihan
17.	Munculnya bercak-bercak hitam di sekitar leher, ketiak, dan selangkangan, (akantosis nigrikans) sebagai tanda terjadinya resistensi insulin

Beberapa orang dapat mengalami kondisi prediabetes, yaitu kondisi ketika glukosa dalam darah di atas normal, namun tidak cukup tinggi untuk didiagnosis sebagai diabetes. Seseorang yang menderita prediabetes dapat menderita diabetes tipe 2 jika tidak ditangani dengan baik [1].

### 2.3 K-Nearest Neighbor(K-NN)

K-Nearest Neighbor (KNN) merupakan satu dari banyaknya algoritma yang digunakan untuk digunakan klasifikasi berdasarkan atribut dan sampel-sampel yang berasal dari data training. Pada tahun 1967, T. Cover dan P. Hart memperkenalkan algoritma KNN, yang menggunakan kelas tetangga terdekat untuk mengklasifikasikan kelas sampel[3]. Algoritma KNN digunakan untuk menentukan hasil akhir klasifikasi tetangga yang baru. Pada klasifikasi baru di

instance dapat dilakukan berdasarkan mayoritas kategori K-tetangga terdekat (nearest neighbors). Sedangkan jauh dekatnya dapat dihitung dengan jarak euclidean (d) formula untuk mencari jarak antara 2 titik ruang dua dimensi sebagai berikut[2].

Algoritma KNN akan mencari berdasarkan jarak terdekat dari data baru ke data training sehingga akan menentukan hasil dari nilai K. Nilai K tersebut merupakan hasil terbaik dari sejumlah data training. Untuk mendapatkan hasil dari nilai K dapat dilakukan dengan optimasi parameter. Nilai K yang tinggi umumnya mengurangi dampak derau pada setiap klasifikasi, tetapi mengaburkan garis yang memisahkannya. Kategori dasar untuk menentukan data baru menggunakan algoritma KNN dengan menggunakan fungsi jarak yang dipergunakan mengukur seberapa kedekatan dari data tersebut dengan data lain di dekatnya. Beberapa fungsi jarak yang umum digunakan untuk mengukur kedekatan 2 buah data  $X_1(x_{11}, x_{21}, x_{31} \dots x_{m1})$  dan  $X_2 = (x_{12}, x_{22}, x_{32} \dots x_{m2})$  yaitu.

#### 1) Jarak Euclidean:

Jarak Euclidean adalah metode untuk menghitung jarak antara dua titik dalam ruang Euclidean (mencakup dua, tiga, atau bahkan lebih bidang Euclidean). Rumus berikut digunakan dengan rumus jarak Euclidean untuk menentukan tingkat kesamaan data.

$$d(x, y) = \sqrt{\sum_{i=1}^m (x_i - y_i)^2}$$

dimana,

d = jarak antara x dan y

x = data pusat klaster

y = data pada atribut

i = setiap data n = jumlah data

$x_i$  = data pada pusat klaster ke i

$y_i$  = data pada setiap data ke i

## 2) Jarak Manhattan:

Perbedaan mutlak (absolut) antara dua koordinat objek dapat ditentukan dengan menggunakan jarak Manhattan. Rumusnya adalah sebagai berikut:

$$d(x, y) = \sum_{i=1}^m |x_i - y_i|$$

dimana,

d = jarak diantara x dan y

x = data pusat kluster

y = data pada atribut i = setiap data

n = jumlah data

$x_i$  = data pada pusat kluster ke i

$y_i$  = data pada setiap data ke i

## 3) Jarak Minkowski:

Minkowski, yang mungkin merupakan metrik dalam ruang vektor di mana standar dikarakterisasi dianggap perluasan dari Jarak perhitungannya. Nilai p biasanya satu atau dua saat menggunakan penghapusan Minkowski untuk mengukur pemisahan pertanyaan. Dalam strategi ini, penghapusan dihitung menggunakan persamaan di bawah ini

$$d(x, y) = (\sum_{i=1}^n |x_i - y_i|^p)^{1/p}$$

dimana, d = jarak antara x dan y

x = data pusat kluster

y = data pada atribut

i = setiap data

n = jumlah data

$x_i$  = data pada pusat kluster ke i



$y_i$  = data pada setiap data ke  $i$

$p$  = power

Alasan menggunakan algoritma KNN (K-Nearest Neighbor). pada penggunaan klasifikasi pada penyakit diabetes adalah penggunaan dalam mengambil sebuah keputusan atau menentukan data terbaru tersebut didiagnosa mengidap penyakit diabetes atau tidak, dan dapat dilihat dari mayoritas keputusan pasien-pasien berdasarkan kedekatannya dengan data pasien yang telah ada didalam data training dapat diketahui hasil diagnosis apakah orang itu mendapatkan hasil positif ataupun negatif.

## 2.4 Euclidean Distance

Euclidean Distance merupakan suatu teknik untuk menghindari agar hasil probabilitas pada perhitungan tidak menghasilkan nilai 0 yang dikarenakan tidak adanya data untuk suatu atribut tertentu dalam suatu kelas [7]. Dalam teknik ini dilakukan penambahan nilai satu data pada setiap perhitungannya. Dengan cara ini tidak akan mengubah nilai probabilitas secara signifikan, memungkinkannya untuk menghindari skenario nilai probabilitas nol.. Laplace Correction memiliki rumus sebagai berikut:

$$P_i = \frac{m_i + 1}{n + k}$$

dimana  $K$  adalah sebagai jumlah bin ataupun kelas dari atribut  $m_i$ .

## 2.5 Normalisasi Data

Normalisasi data atau scalling data untuk memfasilitasi analisis statistik, proses normalisasi data melibatkan memastikan bahwa semua variabel memiliki rentang nilai yang sama—tidak ada yang terlalu besar atau terlalu kecil. Min-Max adalah teknik normalisasi data. Setiap nilai dalam sebuah fitur dikurangi dengan nilai minimum fitur tersebut, yang kemudian dibagi dengan rentang nilai, atau nilai maksimum dikurangi nilai minimum fitur tersebut. Nilai normalisasi baru antara 0 dan 1 akan diperoleh dengan menggunakan metode ini. Method  $\max()$  digunakan untuk mengambil nilai maksimal dari data sedangkan metode  $\min()$  digunakan

untuk mengambil nilai minimum yang berasal dari data yang digunakan. Berikut ini rumus normalisasi data.

$$\text{Normalisasi} = \frac{\text{data}_x - \text{datamin}}{\text{datamax} - \text{datamin}}$$

dimana:

*data<sub>x</sub>* : data dari variabel yang akan dihitung *d*

*datamin* : data Minimal dari data yang digunakan

*datamax* : data maksimal dari data yang digunakan

## 2.6 Confusion Matrix

Model klasifikasi menguji atau memprediksi jumlah baris data yang benar atau salah dalam tabel Confusion Matrix. Tujuannya adalah untuk memastikan seberapa baik model klasifikasi bekerja.

**Tabel 2.4** Kinerja Klasifikasi

		Kelas Prediksi	
		+	-
Kelas Aktual	+	True Positive (TP)	False Negative (FN)
	-	False Positive (FP)	True Negative (TN)

Keterangan:

1. True Positive (TP) = hasil dari prediksi sistem yang positif dan sesuai dengan aktualnya positif
2. True Negative (TN) = hasil dari prediksi sistem yang negatif dan sesuai dengan aktualnya negatif.

3. False Positive (FP) = hasil dari prediksi sistem yang positif, sedangkan aktualnya negatif.

4. False Negative (FN) = hasil dari prediksi sistem yang negatif, sedangkan aktualnya positif.

Precision adalah klasifikasi positif yang benar (true positive) dan keseluruhan data yang diprediksikan sebagai kelas positif. Persamaan 19 precision, yaitu:

$$Precision = TP / TP + FP$$

Recall adalah jumlah dokumen yang memiliki klasifikasi positif yang benar (true positive) dari semua data yang benar-benar positif (termasuk didalamnya false negative). Persamaan recall, yaitu:

$$Recall = TP / TP + FN$$

Accuracy adalah nilai kedekatan antara jumlah dokumen yang memiliki klasifikasi benar (true positive dan true negative) dengan jumlah semua data .Persamaan accuracy, yaitu:

$$Accuracy = TP + TN / TP + TN + FP + FN$$

## 2.7 Flowchart

Flowchart merupakan suatu teknik untuk memeriksa langkah-langkah sistem atau program. Flowchart adalah deskripsi logis dari simbol-simbol yang menentukan aliran data program. Pemrogram telah menggunakan strategi ini untuk membantu pengembangan sistem pemrosesan data[5].

## 2.8 Python

Guido van Rossum menciptakan bahasa pemrograman tingkat tinggi Python lebih tepatnya tahun 1989, dan dirilis untuk umum pertama kali tahun 1991. Python dikembangkan dengan tujuan untuk memudahkan pemrogram menyelesaikan pekerjaannya dengan lebih cepat. Pemrogram akan menganggap Python sangat ramah pengguna dan hemat waktu, serta mudah digunakan untuk pengembangan program dan kompatibel dengan sistem. Baik program mandiri maupun pemrograman skrip dapat ditulis dengan Python (Pemrograman Skrip)[8].

Beberapa feature yang dimiliki Python antara lain :

**Tabel 2.5** Fitur Python

NO	Fitur Python
1	Memiliki library yang luas, dalam distribusi Python telah disediakan modul-modul siap pakai untuk berbagai keperluan.
2	Memiliki tata bahasa yang jernih dan mudah dipelajari.
3	Memiliki aturan layout source code, layout source code yang memudahkan pengecekan, pembacaan kembali, dan penulisan ulang source code tersebut.
4	Memiliki modular, mudah dikembangkan dengan menciptakan modul-modul baru, dimana modul-modul tersebut dapat dibangun dengan bahasa Python maupun C/C++
5	Memiliki fasilitas pengumpulan sampah secara auto seperti halnya pada bahasa pemrograman Java, Python memiliki fasilitas pengaturan penggunaan memory komputer sehingga para programmer tidak perlu melakukan pengaturan memory komputer.

## 2.9 Library Pada Python

Di Python, library adalah kumpulan paket dan modul dengan fungsionalitas identik yang bertujuan untuk menyederhanakan pengembangan aplikasi tanpa memerlukan penulisan kode yang ekstensif..

Dalam Python, istilah "library" mengacu pada kode program tambahan yang digunakan untuk tujuan tertentu. Lebih dari 140.000 pustaka sumber terbuka telah dikembangkan untuk Python. Tanpa harus menulis seluruh skrip, pustaka Python akan menghasilkan kode dengan cepat dan efisien, menghemat waktu.

## 2.10 Dataset

Dataset yang digunakan berasal dari Pima Indians Diabetes Database, sebuah situs publik yang digunakan untuk mempelajari populasi Indian Pima.

Salah satu suku Indian Amerika adalah Indian Pima. Pima, juga dikenal sebagai Akimel O'odham atau Akimel O'otham, adalah sekelompok penduduk asli Amerika yang tinggal di Arizona selatan di sepanjang sungai Gila dan Colorado[11]. Mereka disebut sebagai "Orang Sungai". Sampel dari semua pasien dalam kumpulan data ini kemudian secara eksklusif adalah wanita dan berusia lebih dari 20 tahun.

Untuk memanfaatkan algoritma K-Nearest Neighbor (K-NN) untuk pengujian dan pelatihan data, dataset ini akan dimasukkan ke dalam sistem database. Ada 768 data dan atributnya dalam dataset. Data pelatihan dan data pengujian akan dipisahkan dari kumpulan data ini di masa mendatang. Diketahui sembilan variabel dalam data diabetes, dengan 1 variabel terikat dan 8 variabel bebas[11].

**Tabel 2.6** Variabel

NO.	VARIABEL
1	Kehamilan: Mempresentasikan bahwa berapa kali wanita tersebut mengalami kehamilan semasa hidupnya.
2	Glukosa: Mempresentasikan kadar dari glukosa plasma pada 2 jam dalam tes glukosa.
3	Tekanan darah: Tekanan darah adalah cara yang sangat terkenal untuk mengukur kesehatan jantung seseorang, ada juga ukuran tekanan darah yaitu diastolik dan sistolik. Dalam data ini, kita memiliki tekanan darah diastolik dalam (mm / Hg) ketika jantung rileks setelah kontraksi.
4	4. Lemak tubuh: Nilai yang digunakan untuk memperkirakan lemak tubuh (mm) yang diukur pada lengan kanan setengah antara proses olecranon dari siku dan proses akromial skapula.
5	Insulin: Tingkat insulin 2 jam insulin serum dalam satuan mu (U/ml).
6	BMI: Indeks Massa Tubuh (berat dalam kg / (tinggi dalam meter kuadrat), dan merupakan indikator kesehatan seseorang.
7	Riwayat diabetes: Indikator riwayat diabetes dalam keluarga.
8	Umur: Umur wanita suku indian pima (years).

## 2.11 Jupyter Notebook

Perangkat lunak Jupyter Notebook memudahkan untuk melihat dan menganalisis data. Dokumen bernama Jupyter Notebook (file dengan ekstensi ipynb) dibuat oleh Jupyter Notebook. Ini berisi elemen teks kaya seperti paragraf, persamaan matematika, gambar, dan tautan serta kode komputer[13]. Jupyter Lab akan segera menggantikan Notebook Jupyter, yang sebelumnya dikenal sebagai Notebook IPython. Ilustrasi yang bagus dalam memperkenalkan sains terbuka kepada akademisi di universitas adalah penggunaan klasifikasi Notebook Jupyter di bidang pembelajaran mesin.

## 2.12 Visual Studio Code

Microsoft Visual Studio Code (VS Code) adalah editor konten mumpuni yang dirancang untuk digunakan pada sistem kerja yang berbeda, termasuk Windows, Linux, dan Mac OS X. Dengan bantuan plug-in yang dapat diinstal melalui pusat komersial Visual Studio Code (seperti C++, C# , Python, Go, Java, dll.)[8], editor konten ini mendukung dialek pemrograman yang berbeda, termasuk JavaScript, TypeScript, dan Node.js.

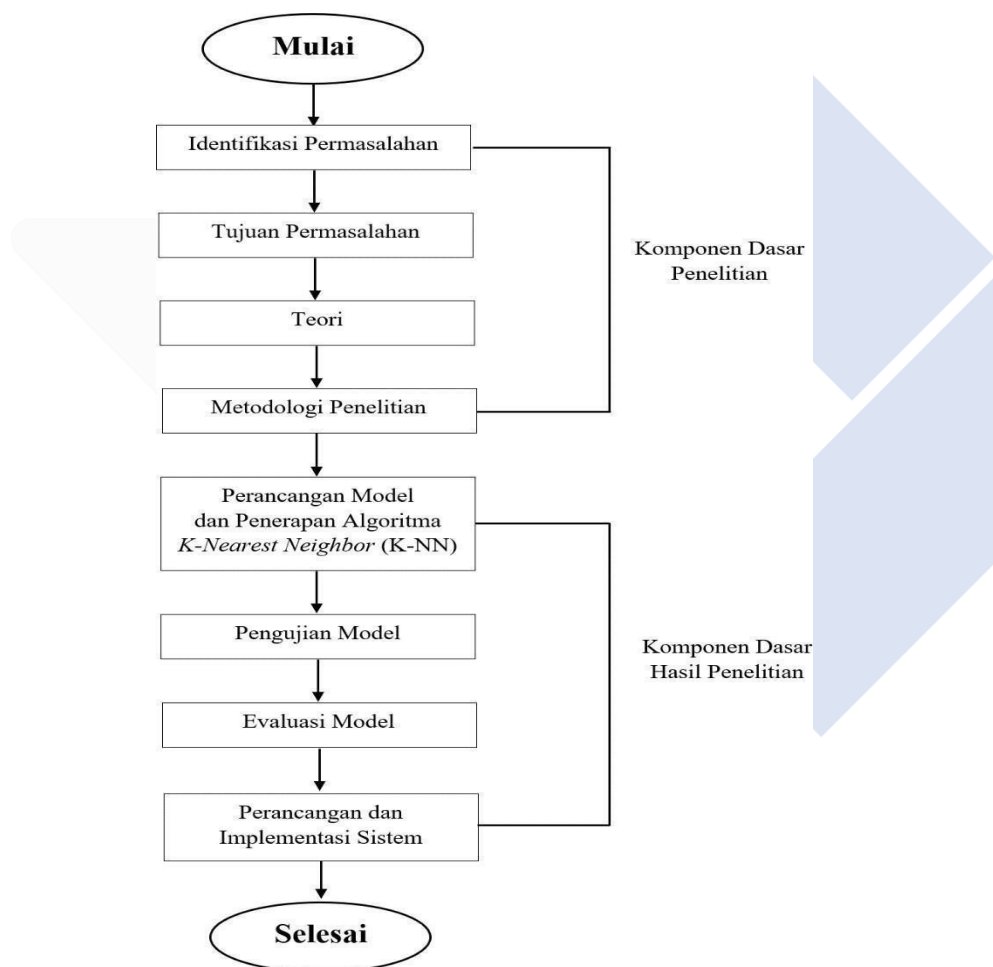
Banyak fitur yang diiklankan oleh Visual Studio Code, termasuk Investigasi, Intellisense, Integrasi Git, dan fitur ekspansi yang meningkatkan kemampuan editor konten. Seiring bertambahnya bentuk Visual Studio Code, jumlah sorotan ini juga akan bertambah. Bentuk Visual Studio Code ini menonjol dari editor konten lainnya karena mendapat pembaruan standar dari bulan ke bulan[14].

Selain itu, editor konten VS Code bersifat open source, memungkinkan Anda melihat dan berkontribusi pada kemajuannya. Antarmuka Github juga menyediakan akses ke kode sumber VS Code. Karena para insinyur aplikasi mungkin tertarik dengan kemajuan VS Code di masa depan, hal ini juga menjadikannya aplikasi favorit para pengembang [14].

## BAB III METODE PELAKSANAAN

### 3.1 Alur Penelitian

Alur ini dijadikan arahan bagi pembuatnya dalam melaksanakan penelitian ini agar hasil yang dicapai tidak melenceng dari tujuan yang telah ditetapkan. Pada gambar 3.1 dibawah ini menunjukkan alur penelitian.



**Gambar 3.1** Alur Penelitian

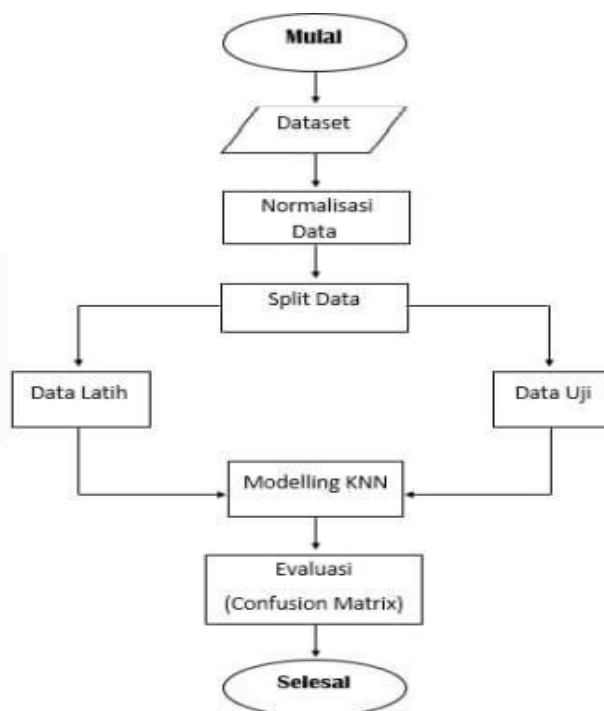


Penjelasan dari gambar 3.1 diatas adalah sebagai berikut:

1. Identifikasi masalah merupakan tahap untuk menjelaskan masalah dan membuat penjelasan yang dapat diukur. Tahap ini merupakan tahap awal untuk melakukan penelitian.
2. Tujuan masalah atau rumusan masalah merupakan pertanyaan- pertanyaan yang dibuat diawal proses penelitian yang menjadi fokus analisis dalam penelitian.
3. Teori atau landasan teori dapat berupa sebuah konsep dengan artikulasi yang terorganisir dengan sempurna dan sistematis dalam penelitian, karena dapat menjadi landasan yang kokoh dalam penyelidikan..
4. Metodologi penelitian merupakan tahap untuk mendapatkan informasi dan melakukan investigasi pada data
5. Pada tahap berikutnya yaitu komponen dasar hasil penelitian, yaitu melakukan tahap perancangan model yang menggunakan software *jupyter notebook* dengan menggunakan algoritma *K-Nearest Neighbor*
6. Pengujian model dilakukan untuk mengetahui apakah pemodelan yang dilakukan dapat berjalan dengan baik sesuai dengan yang diharapkan
7. Evaluasi model adalah tahap untuk melakukan perhitungan dari pengujian model yang menggunakan *Confusion Matrix*
8. Pada tahap perancangan dan implementasi sistem yaitu tahap untuk pembuatan sistem yang berbasis website

### 3.2 Flowchart Proses

Berikut merupakan diagram alur proses yang dimulai dari pengumpulan dataset dari penelitian yang akan dilakukan hingga menentukan hasil akurasi dari pemodelan. Pada gambar 3.2 menunjukkan langkah atau alur tahapan yang digambarkan dengan flowchart



**Gambar 3.2** Flowchart Proses

1. Dataset yang didapatkan dari situs Kaggle yang terdiri dari 8 variabel dan 1 hasil. Berjumlah keseluruhan yaitu 768 data.
2. Normalisasi atau scaling data merupakan tahap perhitungan untuk mencari rentang dari keseluruhan nilai pola memiliki nilai yang sama, antara 0 sampai 1. Perhitungan normalisasi yang digunakan adalah metode min – max.
3. Pada tahap Scaling data atau penggabungan data merupakan tahap pembagian dataset menjadi 2 jenis, yaitu data latih (training) dan data uji (testing). Data uji adalah kumpulan data yang digunakan untuk menguji keakuratan K-Nearest

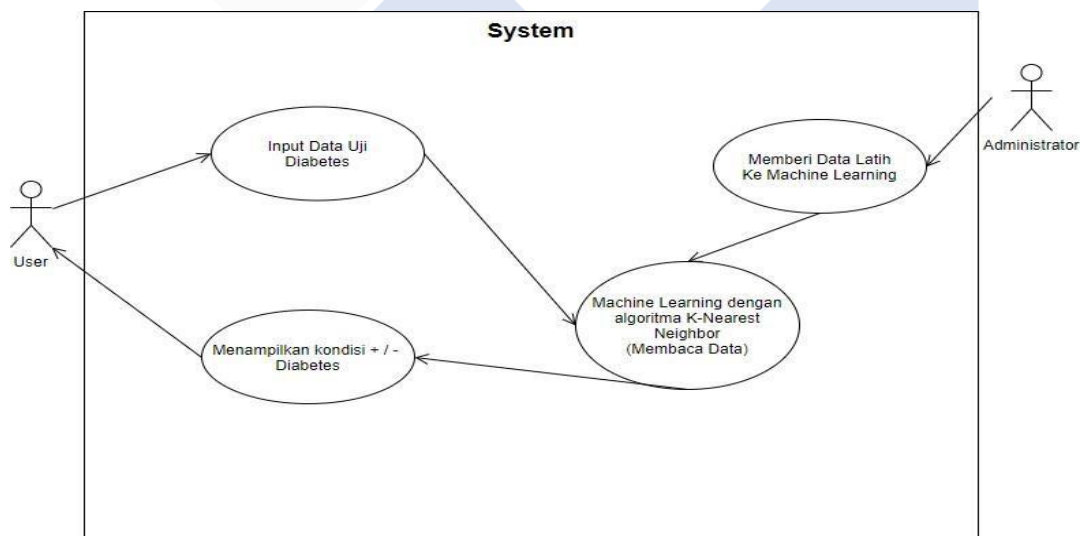
Neighbor (semakin banyak data akan semakin akurat). Data Latih adalah kumpulan data yang diuji menggunakan Modeling KNN.

4. Setelah itu, melakukan pemodelan data. Pada pemodelan ini akan memasukkan nilai k, yaitu nilai tetangga terdekat yang akan dilakukan untuk melakukan pengklasifikasian data. Dalam penggunaan nilai k, harus sesuai dengan banyaknya data. Jika jumlah data bernilai genap maka harus bernilai ganjil, begitu juga sebaliknya. Dikarenakan agar berfungsi untuk menghindari dari munculnya total jarak yang bisa bersamaan dalam proses pengklasifikasian.

5. Setelah itu, data prediksi akan dievaluasi untuk mengetahui tingkat keakuratan sistem dengan menggunakan Confusion Matrix. Hasil yang didapatkan dari evaluasi Confusion Matrix meliputi nilai accuracy, recall, dan precision

### 3.3 Use Case Diagram

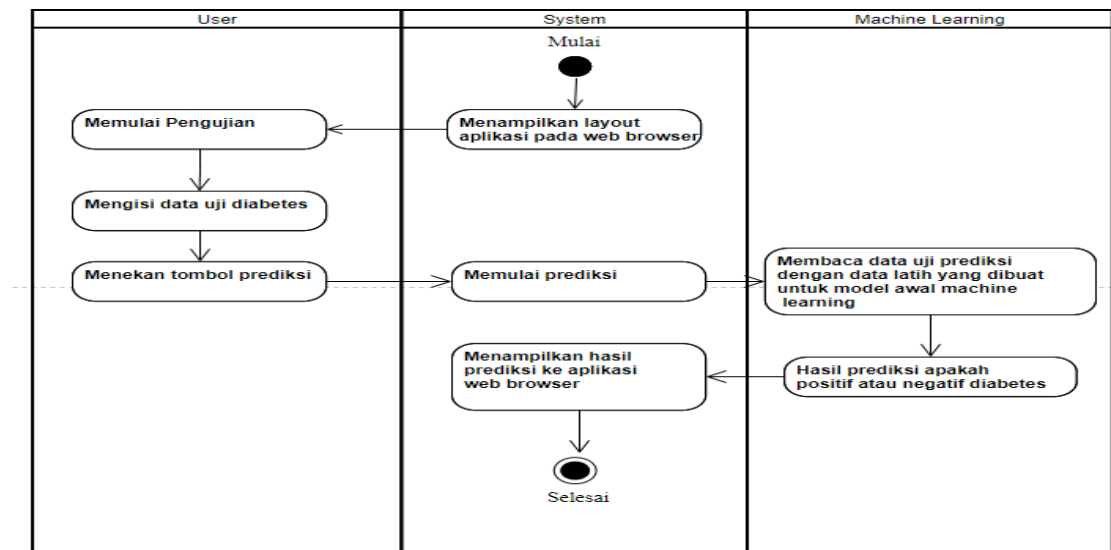
Berikut merupakan Use Case Diagram atau merupakan gambaran atau representasi dari interaksi yang terjadi antara sistem dan lingkungannya yang dimana memperlihatkan proses aktivitas secara urut dalam sistem yang dibuat yang ditunjukkan gambar 3.3



**Gambar 3.3** Use Case Diagram

### 3.4 Activity Diagram

Berikut merupakan Activity Diagram atau merupakan rancangan aliran aktivitas dalam sebuah sistem yang dijalankan dimulai dari awal sistem hingga selesai yang ditunjukkan pada gambar 3.4



Gambar 3.4 Activity Diagram

### 3.5. Analisis Kebutuhan

Analisis kebutuhan terhadap suatu sistem dibagi menjadi dua jenis, yaitu fungsional dan non fungsional.

#### 3.5.1 Analisis Kebutuhan Fungsional

Kebutuhan fungsional merupakan kebutuhan yang harus terpenuhi di sistem karena berisi proses – proses dan layanan yang disediakan oleh sistem. Kebutuhan fungsional meliputi:

1. Mampu menampilkan interface yang dapat dengan mudah dimengerti dan dipahami serta digunakan oleh pengguna.
2. Mampu untuk memprediksi awal penyakit Diabetes menggunakan algoritma metode K-Nearest Neighbor.

3. Mampu menghasilkan hasil dari prediksi berupa hasil positif atau negatif dari sebuah aplikasi web.

### 3.5.2 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional merupakan langkah untuk mengetahui spesifikasi kebutuhan untuk sistem. Spesifikasi ini mencakup kebutuhan perangkat keras (hardware) dan perangkat lunak (software) untuk membangun dan menjalankan sistem.

**Tabel 3.1** Perangkat lunak membangun sistem

<b>Perangkat Lunak</b>
OS Windows 10
Jupyter Notebook
Python
Web Browser

**Tabel 3.2** Perangkat keras membangun sistem

<b>Perangkat Keras</b>
Processor AMD Ryzen 5 3400 G 3.7 GHz
Memory RAM 16GB DDR4
SSD 128GB HDD 500GB
Radeon Vega 11 Graphics 2GB

### 3.6 Analisis Prediksi Dengan Metode K-Nearest Neighbor(K-NN)

Berikut langkah – langkah dari analisis prediksi awal penyakit Diabetes Mellitus dengan algoritma K-Nearest Neighbor: Sebelum melakukan analisis prediksi awal penyakit Diabetes Mellitus dengan algoritma K-Nearest Neighbor, terlebih dahulu harus membuat data training dari dataset.

**Tabel 3.3 Model Data Training**

<b>Kehamilan</b>	<b>Glukosa</b>	<b>Tekanan Darah (mm/Hg)</b>	<b>Lemak Tubuh (mm)</b>	<b>Insulin (U/ml)</b>	<b>BMI (kg)</b>	<b>Riwayat Penyakit Diabetes (%)</b>	<b>Umur (Tahun)</b>	<b>Hasil</b>
8	188	78	0	0	47,9	0,137	43	Positif
7	152	88	44	0	50	0,337	36	Positif
2	99	52	15	94	24,6	0,637	21	Negatif
1	109	56	21	135	25,2	0,833	23	Negatif
2	88	74	19	53	29	0,229	22	Negatif
8	179	72	42	130	32,7	0,719	36	Positif
4	151	90	38	0	29,7	0,294	36	Negatif
7	102	74	40	105	37,2	0,204	45	Negatif
0	131	88	0	0	31,6	0,743	32	Positif
6	104	74	18	156	29,9	0,722	41	Positif
3	148	66	25	0	32,5	0,256	22	Negatif
4	120	68	0	0	29,6	0,709	34	Negatif
6	102	82	0	0	30,8	0,18	36	Positif
6	134	70	23	130	35,4	0,542	29	Positif
2	87	0	23	0	28,9	0,773	25	Negatif
1	79	60	42	48	43,5	0,678	23	Negatif

Berdasarkan tabel 3.5 diatas, kualitas atribut dan kelas ditentukan dalam kumpulan data kasus Diabetes Mellitus ini. Untuk kelas ini terdapat hasil yaitu angka 1 = positif dan angka 0 = negatif. Sedangkan untuk atribut nya atau variabel terdiri dari 8 buah, Meliputi jumlah kelahiran, kadar glukosa, berat tekanan darah, ketebalan kulit, insulin, berat badan (BMI), riwayat diabetes keluarga, dan usia. Pada data uji Informasi persiapan ini akan dianalisis menggunakan perhitungan K-Nearest Neighbor dan akan menjadi dasar perhitungan ekspektasi Diabetes Mellitus.

### **3.7 K Nearest-Neighbor**

Pada kasus diabetes melitus dengan algoritma K-NN dai dataset ini diketahui bahwa hasil dari variabel “outcome” dalam bentuk kategori yang muncul apakah positif mengidap diabetes atau tidak. K-NN memiliki kelebihan antara lain :

1. K-NN sangat intuitif dan langsung: penghitungan K-NN sangat mudah dibuat dan juga mudah dijalankan. Akurasi yang tinggi (relatif).
2. K-NN tidak curiga: K-NN mungkin merupakan perhitungan non- parametrik yang menunjukkan adanya asumsi-asumsi yang harus dipenuhi untuk mewujudkan K-NN. Model parametrik seperti straight backslide memiliki banyak keraguan yang harus dipenuhi karena data akhir-akhir ini dapat diaktualisasikan yang tidak terjadi di K-NN.
3. Dapat dimanfaatkan untuk Klasifikasi dan Pengulangan: salah satu titik fokus terbesar K-NN adalah dapat digunakan untuk klasifikasi dan masalah berulang.

Setelah memiliki data training, diperlukan data uji sebagai bahan prediksiakan hasilnya. Pada tabel 3.4 merupakan contoh model data testing.

**Tabel 3.4 Model Data Testing**

Kehamilan	Glukosa	Tekanan Darah	Lemak Tubuh	Insulin	BMI	Riwayat Penyakit Diabetes	Umur	Hasil
3	126	88	41	235	39,3	0,704	27	Negatif
8	155	62	26	495	34	0,543	46	Positif
8	183	64	0	0	23,3	0,672	32	Positif
1	89	66	23	94	28,1	0,167	21	Negatif
0	137	40	35	168	43,1	2,288	33	Positif
5	116	74	0	0	25,6	0,201	30	Negatif
3	78	50	32	88	31	0,248	26	Positif
10	95	85	25	36	37,4	0,247	24	Positif

### 3.8 Normalisasi Data

Pada tahap ini akan dilakukan normalisasi untuk semua pola mulai dari data training ataupun data testing. Hal ini bertujuan supaya rentang dari keseluruhan nilai pola memiliki nilai yang sama, antara 0 sampai 1. Perhitungan normalisasi yang digunakan adalah metode min-max.

**Tabel 3.5 Nilai minimal dan maksimal data latih**

<b>Max</b>	8	188	90	44	285	50	0,833	45
<b>Min</b>	0	79	0	0	0	24,6	0,137	21

**Tabel 3.6 Nilai minimal dan maksimal data uji**

<b>Max</b>	10	183	88	41	495	43,1	2,288	46
<b>Min</b>	0	78	40	0	0	23,3	0,167	21



**Tabel 3.7** Model Data Latih dan Data Uji

<b>Kehamilan</b>	<b>Glukosa</b>	<b>Tekanan Darah</b>	<b>Lemak Tubuh</b>	<b>Insulin</b>	<b>BMI</b>	<b>Riwayat Penyakit Diabetes</b>	<b>Umur</b>
1	1	0,866667	0	0	0,917323	0	0,916667
0,875	0,669725	0,977778	1	0	1	0,287356	0,625
0,25	0,183486	0,577778	0,340909	0,329825	0	0,718391	0
0,125	0,275229	0,622222	0,477273	0,473684	0,023622	1	0,083333
0,25	0,082569	0,822222	0,431818	0,185965	0,173228	0,132184	0,041667
1	0,917431	0,8	0,954545	0,45614	0,318898	0,836207	0,625
0,5	0,66055	1	0,863636	0	0,200787	0,225575	0,625
0,875	0,211009	0,822222	0,909091	0,368421	0,496063	0,096264	1
0	0,321101	0,888889	0,772727	1	0,771654	0,043103	0,25
0,25	0,192661	0,711111	0,522727	0	0,200787	0,331897	0
0	0,477064	0,977778	0	0	0,275591	0,87069	0,458333
0,75	0,229358	0,822222	0,409091	0,547368	0,208661	0,840517	0,833333
0,375	0,633028	0,733333	0,568182	0	0,311024	0,170977	0,041667
0,5	0,376147	0,755556	0	0	0,19685	0,821839	0,541667
0,5	0,284404	0,733333	0	0	0,287402	0,479885	0,333333
0,375	0,293578	1	0,272727	0,273684	0,149606	0,514368	0,333333
0,75	0,211009	0,911111	0	0	0,244094	0,061782	0,625
0,75	0,504587	0,777778	0,522727	0,45614	0,425197	0,581897	0,333333
0,25	0,073394	0	0,522727	0	0,169291	0,913793	0,166667
0,125	0	0,666667	0,954545	0,168421	0,744094	0,777299	0,083333

**Tabel 3.8** Model data uji setelah normalisasi data

<b>Kehamilan</b>	<b>Glukosa</b>	<b>Tekanan Darah</b>	<b>Lemak Tubuh</b>	<b>Insulin</b>	<b>BMI</b>	<b>Riwayat Penyakit Diabetes</b>	<b>Umur</b>
0,3	0,457143	1	1	0,474747	0,808081	0,253182	0,24
0,8	0,733333	0,458333	0,634146	1	0,540404	0,177275	1
0,8	1	0,5	0	0	0	0,238095	0,44
0,1	0,104762	0,541667	0,560976	0,189899	0,242424	0	0
0	0,561905	0	0,853659	0,339394	1	1	0,48
0,5	0,361905	0,708333	0	0	0,116162	0,01603	0,36
0,3	0	0,208333	0,780488	0,177778	0,388889	0,03819	0,2
1	0,161905	0,9375	0,609756	0,072727	0,712121	0,037718	0,12

Pada tabel 3.8 merupakan hasil dari perhitungan normalisasi pada data latih, yang berjumlah ddari 20 sample data. Dan tabel 3.8 hasil normalisasi data uji yang berjumlah 8 sampel data.

Proses perhitungan normalisasi data pada sesuai atribut. Pada hasil tabel 3.7.5 yang menunjukkan perhitungan normalisasi data padamodel data training. Dengan mengambil contoh 3 tabel diatas.

**Tabel 3.9** Model data latih setelah normalisasi data

<b>3</b>	$\frac{2-0}{8-0} = 0,25$	$\frac{99-79}{188-79} = 0,183486$	$\frac{52-0}{90-0} = 0,577778$
----------	--------------------------	-----------------------------------	--------------------------------

### 3.9 Euclidean Distance atau Jarak

Tahap berikutnya yaitu menentukan jarak atau Euclidean Distance, dengan rumus

seperti berikut ini : 
$$d_{Euclidean}(x,y) = \sqrt{\sum_i(x_i - y_i)^2}$$

Xi : Data atribut yang telah dinormalisasi, Yi : Data uji baru selain data latih. Pada tabel 3.10 merupakan hasil dari perhitungan Euclidean Distance dengan memperhatikan rumus diatas dan menggunakan 8 bahan uji dari sample data testing yang telah digunakan.

**Tabel 3.10** Hasil dari perhitungan *Euclidean Distance*

1	2	3	4	5	6	7	8
1,600553	1,363975	1,140099	1,834011	1,996114	1,27837	1,832616	1,327675
0,887649	1,333159	1,542605	1,544644	1,548803	1,49432	1,437761	0,912648
1,279874	1,655012	1,274571	0,820283	1,453353	0,986299	1,021231	1,347433
1,29512	1,666787	1,468431	1,085602	1,324887	1,288807	1,236645	1,567657
1,018652	1,617793	1,291879	0,377356	1,6729	0,701879	0,766961	0,965419
1,213145	1,101339	1,322761	1,690759	1,517221	1,571703	1,616133	1,379437
0,918231	1,303711	1,129509	1,117979	1,62312	1,020501	1,180128	1,060224
1,068625	0,945451	1,507957	1,384658	1,705816	1,303422	1,208202	1,016302
0,70282	1,283938	1,875851	1,105853	1,520445	1,524426	1,217129	1,39089
1,020629	1,67642	1,230781	0,456455	1,498241	0,775196	0,743031	0,99497
1,424362	1,794991	1,272083	1,287955	1,532687	1,049472	1,524149	1,57382
1,302593	1,10551	1,314666	1,438625	1,538352	1,213419	1,408482	1,323893
0,90149	1,497238	0,97595	0,680599	1,494741	0,758883	0,898564	0,919813

**Contoh Perhitungan :**

Uji\_1 Baris 1

*dEuclidean*

$$= \sqrt{\sum i} = (1 - 0,3)^2 + (1 - 0,457143)^2 + (0,866667 - 1)^2 + (0 - 1)^2 + (0 - 0,474747)^2 + (0,9177323 - 0,808081)^2 + (0 - 0,253182)^2 + (0,916667 - 0,24)^2 =$$

**1, 600553**

**3.10 Menghitung Nilai Tetangga atau K**

Pengorganisasian ini dapat berupa cara untuk menentukan atau memilih nilai hapus terkecil yang akan masuk ke dalam posisi sejumlah nilai k. Misalnya, jika kita memasukkan nilai k sebagai 2, maka kita harus menemukan nilai terkecil yang terpisah dari dua nilai..

Pada tabel 3.13 akan menunjukkan hasil dari k, dan telah melakukan pengujian pada 8 sampel. Tabel dibawah ini akan menunjukkan hasil serta jumlah dari variable nya. Dimasukkan pada table berikut nilai k adalah 5, maka :

**Tabel 3.11 Hasil Ujian K = 5**

1	2	3	4	5	6	7	8
5	5	5	5	5	5	5	5
Positif							Positif
			Negatif	Negatif			
				Negatif			
			Negatif		Negatif	Negatif	Negatif
	Positif						
	Negatif						
Negatif	Negatif						
			Negatif			Negatif	
	Positif						
Negatif		Negatif	Negatif		Negatif	Negatif	Negatif
		Negatif					
		Negatif			Negatif		

			Negatif		Negatif		
		Positif			Positif		Positif
Positif	Positif	Positif		Positif			Positif
				Negatif		Negatif	

Apabila hasil dari nilai k telah ditemukan, maka langkah selanjutnya adalah mengambil jumlah kelas terbanyak dari nilai k yang telah ditentukan untuk hasil prediksi. Yaitu dengan tabel 3.12 dibawah ini.

**3.12 Tabel Jumlah Kelas dan Hasil Prediksi**

Data Uji ke-n	Asli	Prediksi	Positif	Negatif
1	Negatif	Negatif	2	3
2	Positif	Positif	3	2
3	Positif	Negatif	2	3
4	Negatif	Negatif	0	5
5	Positif	Negatif	1	4
6	Negatif	Negatif	1	4
7	Positif	Negatif	0	5
8	Positif	Positif	3	2

Pada tabel 3.12, telah ditentukan bahwa jumlah kelas dan hasil prediksi yang telah dilakukan menampilkan hasil sesuai dengan perhitungan diatas. Pada variable asli, menunjukkan hasil dari sampel data uji atau data testing yang asli, kemudian dilakukan dengan hasil prediksi dengan perhitungan nilai, apabila jumlah kasus positif diabetes lebih besar dibandingkan kasus negatif diabetes maka akan menghasilkan data positif, jika sebaliknya maka akan menunjukkan hasil negatif diabetes.

**Positif > Negatif = Positif**

**Negatif > Positif = Negatif**

### 3.11 Evaluasi

Evaluasi data dilakukan setelah dilakukannya proses prediksi dengan algoritma K-Nearest Neighbor. Metode evaluasi data yang digunakan pada penelitian ini adalah Confusion Matrix. Dengan metode ini akan menghasilkan nilai accuracy, recall, dan precision. Pada tabel 3.14 menunjukkan contoh hasil perbandingan prediksi dengan aktual.

**3.13 Tabel Contoh hasil perbandingan prediksi dengan aktual**

Data	Diabetes Asli	Diabetes Prediksi
1	Negatif	Negatif
2	Positif	Positif
3	Positif	Negatif
4	Negatif	Negatif
5	Positif	Negatif
6	Negatif	Negatif
7	Positif	Negatif
8	Positif	Positif

Dari hasil prediksi diatas, selanjutnya akan diproses dengan Confusion Matrix

### 3.14 Tabel Confusion Matrix

Data	Diabetes Aktual	Diabetes Prediksi	Confusion Matrix
1	Negatif	Negatif	TN
2	Positif	Positif	TP
3	Positif	Negatif	FN
4	Negatif	Negatif	TN
5	Positif	Negatif	FN
6	Negatif	Negatif	TN
7	Positif	Negatif	FN
8	Positif	Positif	TP

### 3.15 Tabel Confusion Matrix dengan perhitungan Excel

	PREDIKSI	
AKTUAL	POSITIF	NEGATIF
POSITIF	2	3
NEGATIF	0	3

Dari tabel diatas, maka dapat dilakukan perhitungan sebagai berikut :

### 1. Accuracy

Rasio prediksi benar dengan keseluruhan dari data.

$$\text{Accuracy} = (\text{TP}+\text{TN}) / (\text{TP}+\text{TN}+\text{FP}+\text{FN})$$

$$= (2+3) / 2+3+0+3)$$

$$= 5/ 8 = 0.625 * 100 = 62.5 \%$$

### 2. Recall

Rasio prediksi positif benar yang dibandingkan dari keseluruhan hasil yang positif benar.

$$\text{Recal l} = \text{TP} / (\text{TP}+\text{FN})$$

$$= 3 / (2+3)$$

$$= 2 / 5 = 0.4 * 100 = 40.0\%$$

### 3. Precision

Rasio prediksi benar positif dibandingkan dengan keseluruhan dari hasil yang diprediksi positif.

$$\text{Precision} = \text{TP} / (\text{TP}+\text{FP})$$

$$= 3 / (3+1)$$

$$= 2 / 2 = 1 * 100 = 100 \%$$



### 3.12 Analisis Kebutuhan Sistem

Sistem ini akan dibangun pada media web dan terdiri dari beberapa halaman antarmuka pengguna. Halaman web menggunakan HTML dan CSS untuk tampilannya, serta Python untuk pemrosesannya.

#### 3.12.1 Perancangan Antarmuka Halaman Beranda

Halaman beranda adalah halaman yang akan ditampilkan pertama kali saat pengguna mengakses web ini. Pada halaman ini terdapat banner yang dapat digeserkan ke kiri, serta gambar dan info pendukung di bawahnya. Selain itu, di setiap halaman pengguna terdapat header dan footer. Header web yang memiliki dua bagian yaitu top-header dan navigation-header.

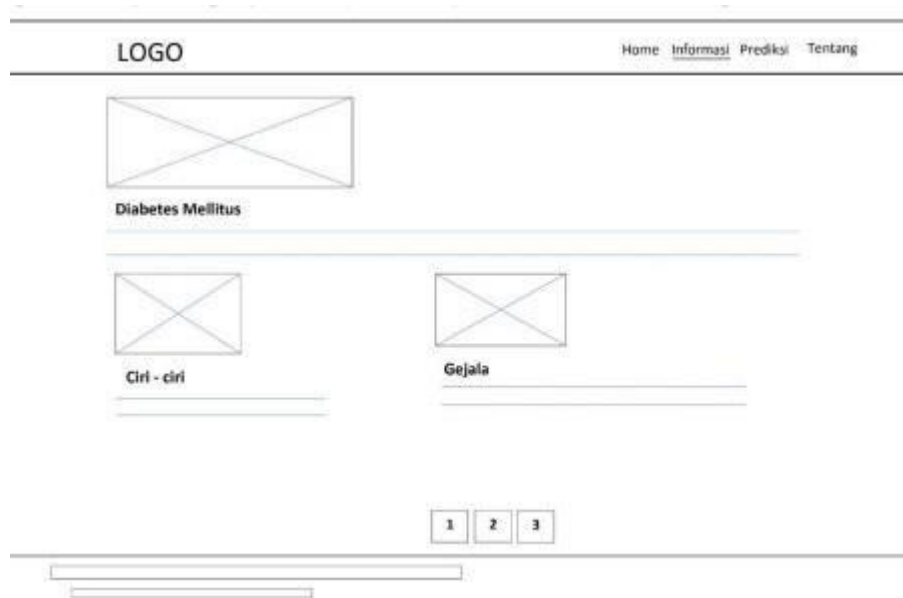
Bagian top - header berisi logo web dan menu – menunya. Sedangkan di bawah nya terdapat kolom prediksi, jika ingin dengan langsung melakukan prediksi, kemudian di bawahnya terdapat pengertian dan ciri dari Diabetes itu sendiri. Bagian footer berisi tahun pembuatan web.



**Gambar 3.5** Rancangan antarmuka halaman beranda

### 3.12.2 Perancangan Antarmuka Halaman Informasi

Halaman informasi adalah halaman yang akan menampilkan informasi tentang penyakit Diabetes Mellitus. Pada halaman ini terdapat gambar pendukung serta penjelasannya. Pada gambar 3.13 adalah tampilan informasi



**Gambar 3.6** Rancangan antarmuka halaman informasi

### 3.12.3 Perancangan Antarmuka Halaman Diagnosis

Halaman diagnosis adalah halaman pengguna untuk memulai melakukan prediksi penyakit Diabetes Mellitus. Pada halaman ini user akan diminta untuk mengisi seluruh variabel yang akan digunakan untuk memasukkan data-data dari pengguna. Jika telah selesai maka akan diminta untuk menekan “Mulai Prediksi”. Kemudian pada top header terdapat menu bantuan jika pengguna mengalami kesusahan dalam menggunakan aplikasi. Gambar 3.14 menunjukkan rancangan antarmuka halaman proses diagnosis

LOGO Home Informasi Prediksi Tentang

---

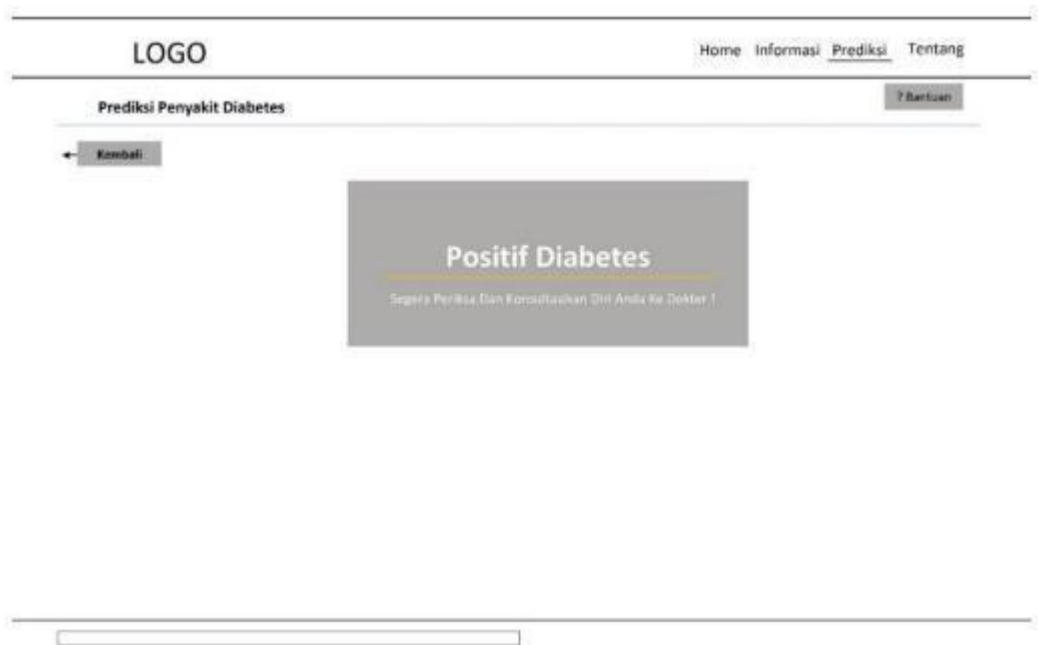
**Prediksi Penyakit Diabetes** ? Bantuan

<b>Kehamilan</b> <input type="text" value="Jumlah Kehamilan"/>	<b>Insulin</b> <input type="text" value="Jumlah Insulin (U/inj)"/>
<b>Glukosa</b> <input type="text" value="Jumlah Kadar Glukosa"/>	<b>BMI</b> <input type="text" value="Jumlah Berat Badan (Kg)"/>
<b>Tekanan Darah</b> <input type="text" value="Jumlah Tekanan Darah (mm/Hg)"/>	<b>Riwayat</b> <input type="text" value="Jumlah Riwayat Penyakit dari Keluarga"/>
<b>Lemak Tubuh</b> <input type="text" value="Jumlah Lemak Tubuh (mm)"/>	<b>Umur</b> <input type="text" value="Umur (Tahun)"/>

**Gambar 3.6** Rancangan antarmuka halaman proses diagnosis

### 3.12.4 Perancangan Antarmuka Halaman Hasil Prediksi

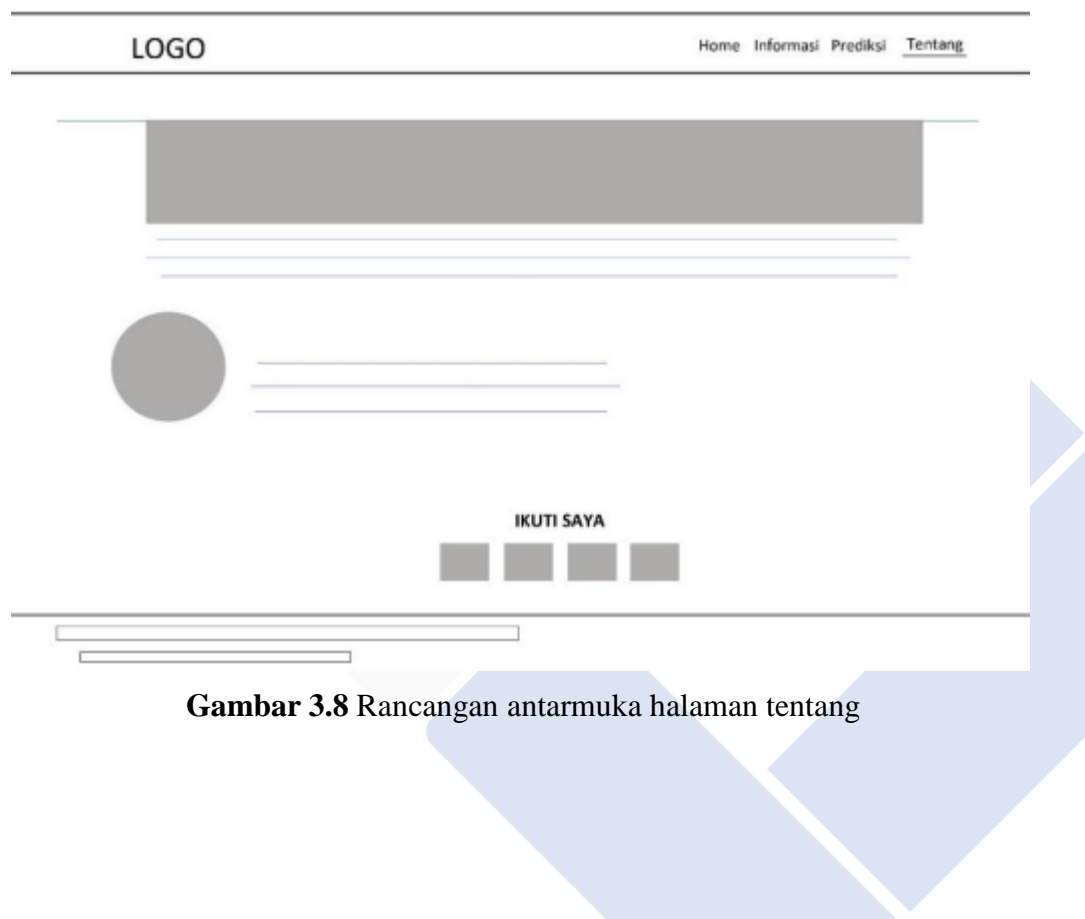
Halaman hasil prediksi adalah halaman untuk menentukan hasil dari prediksi, pada tampilan tersebut akan menampilkan hasil akhir dari pemrosesan data, pasien tersebut mengidap penyakit diabetes atau tidak.



**Gambar 3.7** Rancangan antarmuka hasil prediksi

### 3.12.5 Perancangan Antarmuka Halaman Tentang

Halaman tentang adalah halaman yang berisi informasi tentang web ini, meliputi penjelasan web, layanan web, dan pemilik web. Gambar 3.16 menunjukkan rancangan antarmuka halaman tentang.



**Gambar 3.8** Rancangan antarmuka halaman tentang

## **BAB IV**

### **PEMBAHASAN**

#### **4.1 Deskripsi**

Proyek akhir ini menggunakan metode dari algoritma K-Nearest Neighbor (K-NN) untuk melakukan perkiraan prediksi dari penyakit Diabetes Mellitus. Sistem web dibuat dengan menggunakan jupyter notebook untuk melihat nilai hasil akurasi serta prediksi dari dataset, kemudian menggunakan HTML dan CSS untuk mengatur tampilannya dan bahasa pemrograman PHP untuk pemrosesan datanya. Berdasarkan pada analisa dan perancangan yang terdapat pada bab – bab sebelumnya, implementasi merupakan tahap dimana sistem digunakan dalam keadaan sebenarnya.

Sebelum melakukan perancangan sistem, penulis akan mengumpulkan dataset penyakit Diabetes Mellitus yang akan digunakan untuk kebutuhan sebagai data latih dan data uji pada sistem ini. Dataset berasal dari Kaggle yang berjumlah 768 data.

#### **4.2 Implementasi Model**

Proyek akhir ini menggunakan metode dari algoritma K-Nearest Neighbor untuk melakukan perkiraan prediksi penyakit Diabetes Mellitus. Sistem web dibuat dengan menggunakan jupyter notebook untuk melihat nilai hasil akurasi serta prediksi dari dataset kemudian menggunakan HTML dan CSS untuk mengatur tampilannya dan bahasa pemrograman PHP untuk pemrosesan datanya. Berdasarkan pada analisa dan perancangan yang terdapat pada bab – bab sebelumnya. Berikut pembuatan model pada pendeteksi penyakit Diabetes:

##### **4.2.1 Pembuatan Library**

Langkah awal yang dilakukan untuk membangun sistem pada pembuatan model sistem pada jupyter notebook adalah dengan membuat library yang berfungsi untuk memudahkan untuk membuat suatu aplikasi tanpa menulis banyak kode.

```
C:\Users\user>pip install numpy
Collecting numpy
  Downloading numpy-1.23.5-cp39-cp39-win_amd64.whl (14.7 MB)
----- 14.7/14.7 MB 10.2 MB/s eta 0:00:00
Installing collected packages: numpy
Successfully installed numpy-1.23.5
```

**Gambar 4.1** Peninginstallan Numpy

Apabila semua library pada python telah diinstal melalui cmd, maka langkah selanjutnya adalah mengimport seluruh library tersebut ke jupyter notebook. Library yang digunakan untuk menyimpan dan membaca data dari dan ke dalam file ialah library pickle. Kemudian library pandas adalah untuk mempermudah mengolah dan menganalisa data-data terstruktur. Library seaborn ialah untuk membuat stasistik dan juga grafik dari python. Library matplotlib.pyplot digunakan untuk membuat fungsi visualisasi ke dalam bentuk grafik, dan library sklearn untuk membangun sebuah model dari machine learning. Menyediakan banyak metode algoritma pembelajaran regresi, klasifikasi, dan pengelompokan .

```
import pickle
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
from sklearn.metrics import f1_score
```

**Gambar 4.2** Import library Python yang dibutuhkan

Pada gambar 4.2, merupakan library yang digunakan pada model klasifikasi terhadap penyakit diabetes menggunakan metode algoritma K-Nearest Neighbor. Langkah import library yang dibutuhkan untuk mengeksekusi perintah - perintah pada pemodelan KNN.

#### 4.2.2 Pemuatan Data

Pada tahap ini, dataset yang ada pada Local PC dimuat kedalam aplikasi jupyter notebook dikarenakan jupyter itu sendiri bersifat offline agar dapat digunakan pada pembuatan model:

```
#Membaca Dataset  
df = pd.read_csv('dataset_diabetes.csv')
```

**Gambar 4.3** Memasukkan dataset pada model python

Pada gambar 4.3 , digunakan untuk memasukkan dataset yang ada pada file PC agar dapat diunggah ke Jupyter Notebook. Untuk menampilkan dataset maka diperlukan kode `df.head()`.Maka file data akan menampilkan 5 data teratas dari dataset.

#### 4.2.3 Pendeskripsian Data

Pada tahap pendeskripsian data merupakan untuk melihat informasi yang ada pada data yang digunakan.

```
#Menampilkan Data  
df.head()  
#Jumlah Nilai hasil positif dan negatif pada data  
df.Hasil.value_counts()
```

**Gambar 4.4** Menampilkan informasi dari dataset

Pada informasi data terdapat 768 baris data serta 9 variabel yang bermacam-macam digunakan untuk mengklasifikasikan penyakit diabetes. Kemudian pada jumlah hasil nilai positif pada 768 data yang digunakan terdiri 500 data pasien positif dan 268 data pasien negatif.



#### 4.2.4 Pengecekan Data

Sebelum masuk pada pemodelan, maka langkah awal adalah mengecek data apakah data tersebut ada yang hilang atau kosong.

```
#mengecek apakah ada data yg kosong atau hilang data  
df.isnull().any().sum()
```

**Gambar 4.5** Mencari nilai yang kosong pada data

Setelah data di cek, maka tidak didapatkan data yang kosong. Maka data yang digunakan bisa dilanjutkan ke tahap selanjutnya. Source Code diatas ialah untuk memanggil dan mengecek data yang hilang pada dataset yang digunakan.

#### 4.2.5 Ubah Data

Pada data yang akan diuji, terdapat 4 variabel yang data nya bernilai 0. Pada 4 variabel tersebut akan diisi oleh nilai dari rata-rata data yang digunakan. Dikarenakan nilai 0 pada data itu sendiri dapat mempengaruhi pada perhitungan akurasi.

```
#mengubah nilai yang kosong menjadi nilai rata rata data  
col=('Glukosa', 'Darah', 'LemakTubuh','Insulin', 'BMI')  
for i in col:  
    df[i].replace(0,df[i].mean(),inplace=True)
```

**Gambar 4.6** Mengubah data 0 menjadi nilai rata-rata

#### 4.2.6 Pemisahan Variabel Hasil

Selanjutnya, memisahkan variabel hasil outcome positif atau negatif yang akan digunakan menjadi data (y) pada dataset. Pemisahan pada data (x) dan (y) sangat diperlukan untuk melakukan klasifikasi. Pada data ini untuk data (x) terdiri atas 8 variabel yang memiliki fungsi yang berbeda-beda. Dan data (y) terdiri atas 1 variabel output yang merupakan hasil dari data yang akan digunakan. Pada gambar 4.7 dibawah menampilkan pemisahan data (x) dan (y).

```
#Pembagian data X dan Y pada data  
X = df.drop('Hasil', axis=1)  
y = df['Hasil']
```

**Gambar 4.7** Memisahkan data menjadi x dan y

#### 4.2.7 Normalisasi Data

Setelah melakukan pembagian data (x) dan data (y), selanjutnya yaitu melakukan normalisasi data atau scaling data pada data. Normalisasi data digunakan agar supaya dapat hasil maksimal, oleh karena itu seluruh data akan diubah range angka menjadi 0-1. Gambar 4.8 merupakan hasil normalisasi.

```
#Normalisasi Data

from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler(feature_range = (0,1))
X = scaler.fit_transform(X)
X = pd.DataFrame(X)
X.head()
```

	0	1	2	3	4	5	6	7
0	0.470588	0.929032	0.551020	0.147135	0.079086	0.607362	0.025192	0.366667
1	0.411765	0.696774	0.653061	0.402174	0.079086	0.650307	0.110589	0.250000
2	0.117647	0.354839	0.285714	0.086957	0.096154	0.130879	0.238685	0.000000
3	0.058824	0.419355	0.326531	0.152174	0.145433	0.143149	0.322374	0.033333
4	0.117647	0.283871	0.510204	0.130435	0.046875	0.220859	0.064475	0.016667

Gambar 4.8 Normalisasi Data

#### 4.2.8 Split Data

Apabila data tersebut telah dilakukan normalisasi data, maka langkah selanjutnya melakukan pemisahan data atau split data. Pemisahan data ini sendiri digunakan untuk memisahkan dataset utuh menjadi 2 jenis data. Dibagi menjadi 2 jenis data yaitu data uji (testing) dan data latih (training). Fungsi dari data tersebut dipisahkan ialah untuk memaksimalkan hasil dari prediksi pada model machine learning. Data latih digunakan untuk membuat model dalam pembelajaran mesin, dan data uji digunakan untuk memastikan model berfungsi dan benar (terhadap korelasi). dalam model yang akan dibuat.

```
#Bagi data menjadi data uji dan data latih

x_train, x_test, y_train, y_test = train_test_split(X,y, test_size =0.1, random_state = 0)
print("Jumlah Data Latih      :", x_train.shape)
print("Jumlah Data Uji       :", x_test .shape)
print("Jumlah Hasil Data Latih :", y_train.shape)
print("Jumlah Hasil Data Uji  :", y_test.shape)

Jumlah Data Latih      : (691, 8)
Jumlah Data Uji       : (77, 8)
Jumlah Hasil Data Latih : (691,)
Jumlah Hasil Data Uji  : (77,)
```

**Gambar 4.9** Pembagian data uji dan data latih

Pada gambar 4.9 diatas, menampilkan hasil perhitungan untuk memisahkan data, menjadi data uji dan data latih. Jumlah dari keseluruhan data berjumlah 768 data, maka disini tiap tersebut akan dibagi 90% data latih dan 10% data uji, menjadi 691 data latih dan 77 data uji. Dan random\_state merupakan pembuat angka menjadi acak, dalam kasus ini akan dikosongkan.

#### 4.2.9 Modeling KNN

Kemudian masuk pada pemodelan KNN, pada pemodelan menggunakan KNN harus memasukkan nilai k untuk melakukan klasifikasi. Nilai k tersebut merupakan banyaknya tetangga terdekat. Pada klasifikasi yang akan dicoba, akan menggunakan nilai  $k = 14$ , agar mendapatkan akurasi yang maksimal. Dalam upaya agar dapat menentukan nilai dari k, apabila jumlah klasifikasi kita genap maka sebaiknya kita menggunakan nilai k yang ganjil, begitu juga sebaliknya, dikarenakan ada kemungkinan jika tidak, akan tidak mendapatkan jawaban. Pada gambar 4.10 merupakan script pembuatan model KNN.

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

knn = KNeighborsClassifier(n_neighbors=14) #input nilai k
knn.fit(x_train,y_train)

knn_acc = accuracy_score(y_test,knn.predict(x_test))
```

**Gambar 4.10** Pembuatan KNN

#### 4.2.10 Evaluasi

Evaluasi akan dilakukan dengan menggunakan metode Confusion Matrix untuk mencari nilai dari 3 aspek yaitu accuracy, recall, dan precision. Data uji yang sudah diperoleh hasil prediksinya akan dibandingkan dengan aktual, kemudian diberikan label keterangan Confusion Matrix. Pada gambar 4.11 dibawah merupakan script untuk menampilkan tabel Confusion Matrix yang digunakan untuk menghitung nilai hasil akhir dari model klasifikasi pada pendeteksi penyakit diabetes.

```

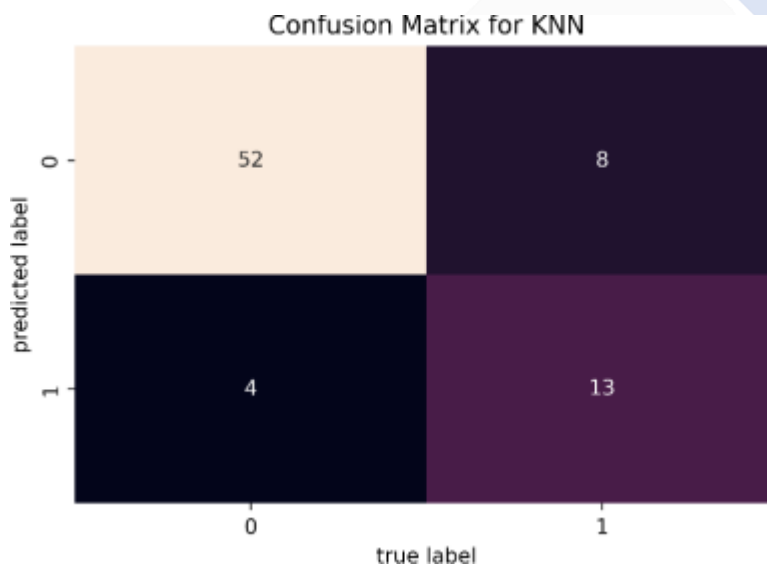
from sklearn.metrics import confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(dpi = 400)
mat = confusion_matrix(y_test, test_pred)
sns.heatmap(mat.T, annot=True, fmt='d', cbar=False)

plt.title('Confusion Matrix for KNN')
plt.xlabel('true label')
plt.ylabel('predicted label')
plt.savefig("confusion_matrix.png")
plt.show()

```

**Gambar 4.11** Menampilkan tabel Confusion Matrix



**Gambar 4.12** Label Confusion Matrix

```
print(classification_report(y_test, test_pred))
```

	precision	recall	f1-score	support
0	0.87	0.93	0.90	56
1	0.76	0.62	0.68	21
accuracy			0.84	77
macro avg	0.82	0.77	0.79	77
weighted avg	0.84	0.84	0.84	77

**Gambar 4.13** Hasil Klasifikasi

### 4.3 Implementasi Pembuatan Sistem

Setelah dilakukan pemodelan pada jupyter notebook dan mendapatkan hasil akurasi yang maksimal, sebesar 84,41% dengan nilai  $k = 14$ . Maka langkah selanjutnya ialah pembuatan sistem untuk menjalankan dari pemodelan yang telah dibuat.

Sistem yang akan dibuat pada pendeteksi diabetes ini menggunakan HTML dan CSS untuk mengatur tampilannya dan Bahasa pemrograman PHP untuk pemrosesan datanya. Kemudian aplikasi pengolah data yang digunakan untuk melakukan pembuatan sistem yang berbasis web ini ialah Visual Studio Code, Yang nantinya model pada jupyter akan diekspor menggunakan pickle.

#### 4.3.1 Penyimpanan Pemodelan

Setelah melakukan modeling, langkah selanjutnya yaitu melakukan simpan model agar hasil dari akurasi yang dibuat dapat digunakan pada web. Pada kasus ini untuk menyimpan model pada jupyter notebook menggunakan pickle yang ada pada fitur sklearn.

Pada gambar 4.14 merupakan script untuk menyimpan model pada pemodelan yang ada pada jupyter notebook yang digunakan. Nama unduhan diberi nama "knn\_pickle", dan wb merupakan write binary, yaitu untuk mengubah menjadi biner.

```
import pickle

with open('knn_pickle', 'wb') as r:
    pickle.dump(knn,r)
```

**Gambar 4.14** Penyimpanan Model

### 4.3.2 Penginstalan Kebutuhan Interface

Sebelum melakukan pembuatan sistem pada interface maka langkah sebelumnya ialah membuat folder tersendiri untuk project yang akan digunakan. Kemudian akses menggunakan command prompt (cmd) masuk ke folder yang telah dibuat, kemudian buat virtual env, digunakan untuk memastikan bahwa library yang dipasang pada sistem hanya untuk diinstal pada project ini saja. Pada gambar 4.15 menampilkan penginstall virtual env.

```
F:\Proyek Akhir\Diabetes_KNN>pip3 install virtualenv
Collecting virtualenv
  Downloading virtualenv-20.17.1-py3-none-any.whl (8.8 MB)
----- 8.8/8.8 MB 11.3 MB/s eta 0:00:00
Collecting distlib<1,>=0.3.6
  Downloading distlib-0.3.6-py2.py3-none-any.whl (468 kB)
----- 468.5/468.5 kB 28.6 MB/s eta 0:00:00
Collecting filelock<4,>=3.4.1
  Downloading filelock-3.8.2-py3-none-any.whl (10 kB)
Collecting platformdirs<3,>=2.4
  Downloading platformdirs-2.6.0-py3-none-any.whl (14 kB)
Installing collected packages: distlib, platformdirs, filelock, virtualenv
Successfully installed distlib-0.3.6 filelock-3.8.2 platformdirs-2.6.0 virtualenv-20.17.1
F:\Proyek Akhir\Diabetes_KNN>
```

**Gambar 4.15** Penginstallan virtualenv

Setelah virtualenv berhasil diinstal, langkah selanjutnya yaitu mengakses active.bat untuk mengunduh library yang akan digunakan untuk membuat website. Disini akan menggunakan library flask, menggunakan flask-sqlalchemy yang berfungsi untuk pembuatan database. Pada kali ini tetap digunakan walaupun tidak menggunakan database. Pada gambar 4.16 merupakan tahap penginstallan flask.



```
F:\Proyek Akhir\Diabetes_KNN>pip install flask
Requirement already satisfied: flask in c:\users\user\appdata\local\programs\python\python39\lib\site-packages (2.2.2)
Requirement already satisfied: Werkzeug>=2.2.2 in c:\users\user\appdata\local\programs\python\python39\lib\site-packages
(from flask) (2.2.2)
Requirement already satisfied: importlib-metadata>=3.6.0 in c:\users\user\appdata\local\programs\python\python39\lib\sit
e-packages (from flask) (5.0.0)
Requirement already satisfied: Jinja2>=3.0 in c:\users\user\appdata\local\programs\python\python39\lib\site-packages (fr
om flask) (3.1.2)
Requirement already satisfied: click>=8.0 in c:\users\user\appdata\local\programs\python\python39\lib\site-packages (fro
m flask) (8.1.3)
Requirement already satisfied: itsdangerous>=2.0 in c:\users\user\appdata\local\programs\python\python39\lib\site-packag
es (from flask) (2.1.2)
Requirement already satisfied: colorama in c:\users\user\appdata\local\programs\python\python39\lib\site-packages (from
click>=8.0->flask) (0.4.4)
Requirement already satisfied: zipp>=0.5 in c:\users\user\appdata\local\programs\python\python39\lib\site-packages (from
importlib-metadata>=3.6.0->flask) (3.10.0)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\user\appdata\local\programs\python\python39\lib\site-packages
(from Jinja2>=3.0->flask) (2.1.1)
```

**Gambar 4.16** Penginstallan Flask

### 4.3.3 Pembuatan Python

Kemudian langkah berikutnya adalah membuka visual studio code, membuat file app.py sebagai python, pada file app.py model yang telah diekspor dari akurasi sebesar 84% yang sebelumnya sudah digunakan akan dimasukkan pada file ini. Pada gambar 4.17 dibawah ini merupakan script untuk memasukkan hasil dari modeling ke sistem web yang akan digunakan.

```

F: > Proyek Akhir > DiabetesProgram > app.py > ...
1  from flask import Flask, render_template, request, redirect
2  import pickle
3  import sklearn
4  import numpy as np # numpy==1.19.3
5  import sklearn.neighbors_classification
6
7  app = Flask(__name__)
8
9  @app.route('/home')
10 @app.route('/')
11
12 def home():
13     return render_template('home.html')
14
15 @app.route('/informasi')
16 def informasi():
17     return render_template('informasi.html')
18
19 @app.route('/about')
20 def about():
21     return render_template('about.html')
22
23 @app.route('/informasi1')
24 def informasi1():
25     return render_template('informasi1.html')
26
27 @app.route('/index')
28 @app.route('/', methods=['POST', 'GET'])
29
30 def index():
31     if request.method == 'POST':
32
33         with open('knn_pickle', 'rb') as r:
34             model = pickle.load(r)
35
36         kehamilan = float(request.form['kehamilan'])
37         glukosa = float(request.form['glukosa'])
38         darah = float(request.form['darah'])
39         kulit = float(request.form['kulit'])
40         insulin = float(request.form['insulin'])
41         bmi = float(request.form['bmi'])
42         riwayat = float(request.form['riwayat'])
43         umur = float(request.form['umur'])
44
45         datas = np.array((kehamilan,glukosa,darah,kulit,insulin,bmi,riwayat,umur))
46         datas = np.reshape(datas, (1, -1))
47
48         isDiabetes = model.predict(datas)
49

```

Gambar 4.17 App.py

```

    return render_template('hasil.html', finalData=isDiabetes)
else:
    return render_template('index.html')

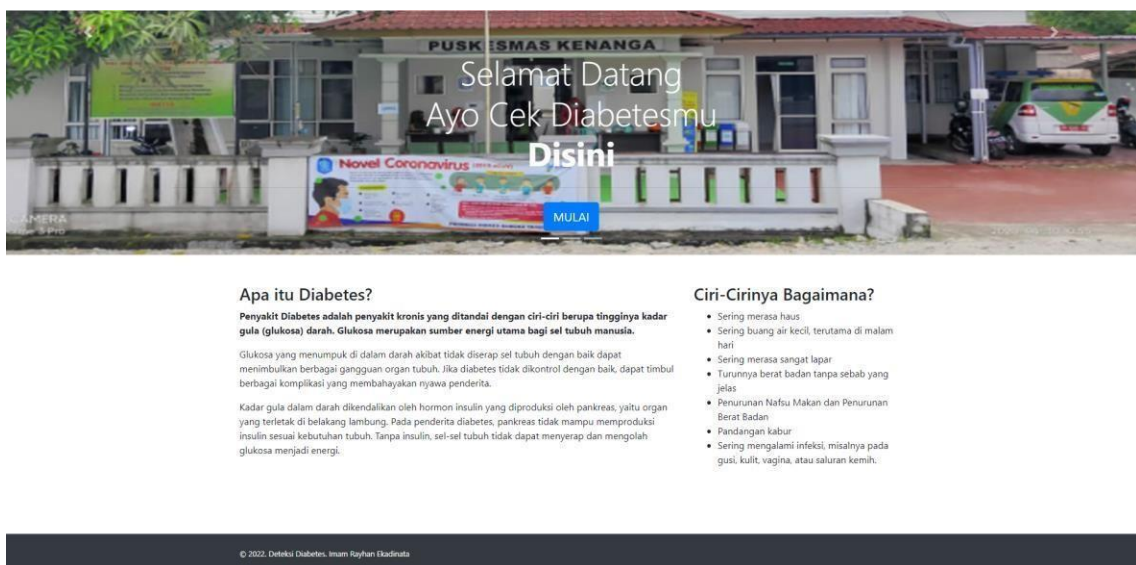
if __name__ == "__main__":
    app.run(debug=True)

```

Gambar 4.18 App.py

### 4.3.4 Halaman Beranda

Halaman beranda merupakan halaman pada sistem deteksi diabetes. Pada sistem ini diberi nama “Deteksi Diabetes”. Halaman beranda ini akan ditampilkan pertama kali saat pengguna akan menggunakan atau mengakses halaman sistem web ini. Terdiri atas tombol mulai untuk langsung melakukan deteksi diabetes, atau memiliki 4 bar yang berbeda isinya, serta memiliki definisi dan ciri dari diabetes itu sendiri. Pada gambar 4. 19 merupakan tampilan halaman beranda.



**Gambar 4.19** Halaman Beranda

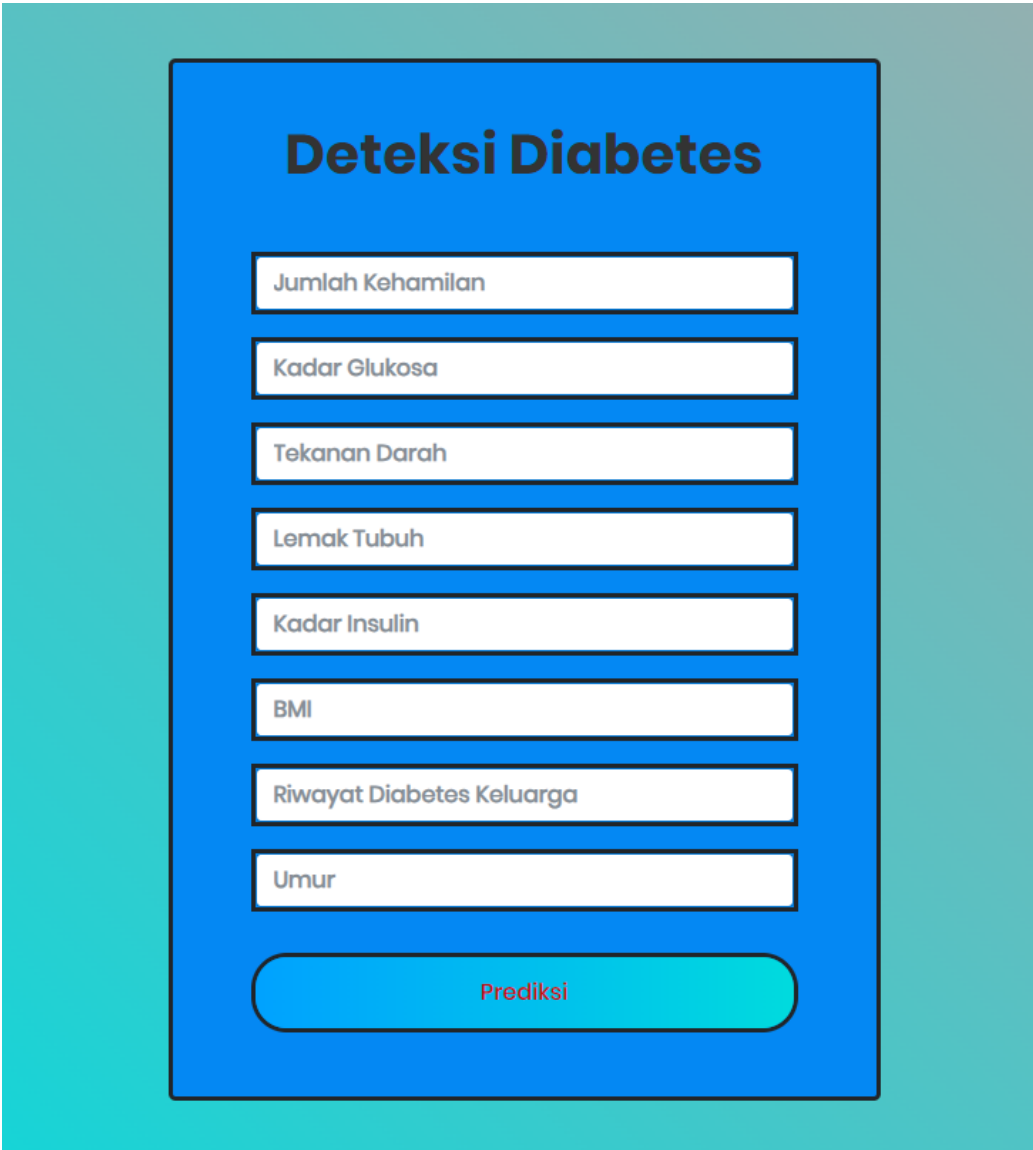
### 4.3.5 Halaman Informasi

Halaman informasi adalah halaman yang menampilkan beberapa informasi – informasi tentang penyakit Diabetes Mellitus. Pada halaman informasi berisi tentang berita-berita terbaru tentang penyakit diabetes, hingga permasalahan yang bisa diatasi dari pengguna yang terkena penyakit diabetes.



Gambar 4.20 Halaman Informasi

#### 4.3.6 Halaman Prediksi



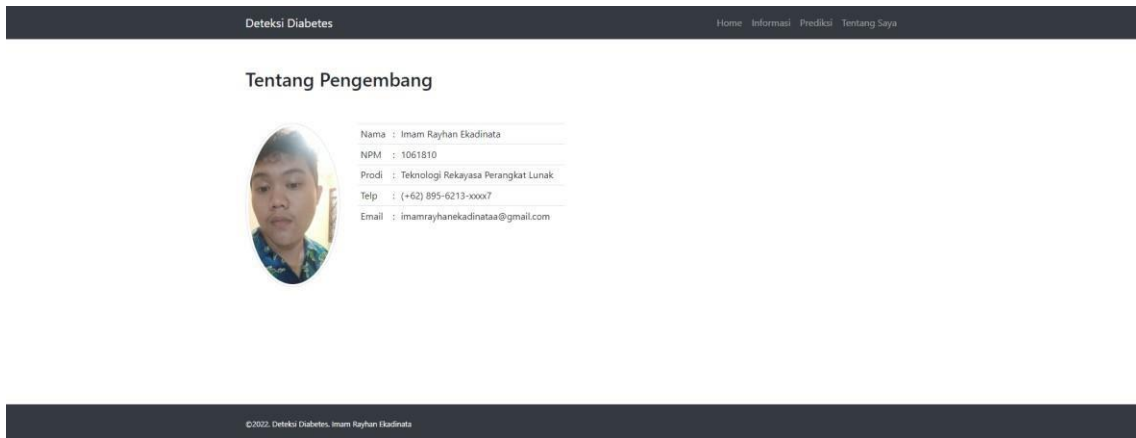
The image shows a digital form titled "Deteksi Diabetes" (Diabetes Detection) on a blue background. The form contains several input fields for user data: "Jumlah Kehamilan" (Number of Pregnancies), "Kadar Glukosa" (Glucose Level), "Tekanan Darah" (Blood Pressure), "Lemak Tubuh" (Body Fat), "Kadar Insulin" (Insulin Level), "BMI", "Riwayat Diabetes Keluarga" (Family History of Diabetes), and "Umur" (Age). At the bottom of the form is a red button labeled "Prediksi" (Predict).

**Gambar 4.21** Halaman Prediksi

Pada gambar 4.21 diatas merupakan halaman informasi yang menampilkan informasi tentang penyakit Diabetes Mellitus. Pada halaman informasi berisi tentang berita-berita terbaru tentang penyakit diabetes, hingga permasalahan yang bisa diatasi dari pengguna yang terkena penyakit diabetes.

### 4.3.7 Halaman Tentang

Halaman tentang adalah halaman tentang pembuat web sistem diabetes.



**Gambar 4.22** Halaman Tentang

### 4.3.8 Halaman Hasil Prediksi

Halaman hasil diagnosis adalah halaman yang menampilkan hasil prediksi dari sistem.



**Gambar 4.23** Halaman Hasil Prediksi

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Berdasarkan penelitian yang telah dilakukan sebelumnya, maka dapat diperoleh kesimpulan sebagai berikut :

1. Algoritma K-Nearest Neighbor dapat memprediksi penyakit Diabetes Mellitus berdasarkan data – data dari pengguna. Hasil prediksi yang didapat adalah kelas positif dan kelas negatif.
2. Pengujian yang telah dilakukan pada dataset dengan menggunakan algoritma K-Nearest Neighbor, setelah dilakukan evaluasi dengan Confusion Matrix mendapatkan nilai accuracy tertinggi sebesar 84,41 % dengan nilai  $k = 14$ .
3. Jumlah komposisi data kelas positif dan negatif yang ada dalam data latih akan mempengaruhi akurasi sistem. Semakin sedikit jumlah data latih, maka akan semakin bervariasi komposisi datanya, sehingga dapat menghasilkan akurasi yang semakin tinggi.

#### **5.2 Saran**

Pada penelitian ini terdapat beberapa kekurangan dan keterbatasan. Kekurangan dan keterbatasan ini dapat dijadikan acuan dan pertimbangan untuk penelitian selanjutnya. Berikut adalah saran untuk penelitian selanjutnya berdasarkan penelitian yang telah dilakukan :

1. Dalam penelitian ini, dataset diperoleh dari situs Kaggle, tanpa didampingi pakarnya. Oleh karena itu, dapat dikembangkan lagi dengan cara bekerja sama dengan pakar sehingga prediksi yang didapatkan lebih akurat.
2. Pada penelitian ini perlu adanya analisis menggunakan metode Machine Learning jenis lainnya untuk menganalisis data yang sama dengan nilai akurasi yang lebih tinggi dibandingkan dengan menggunakan KNN.

## DAFTAR PUSTAKA

- [1] [1] Lestari, Zulkarnain and ST.Aisyah Sijid, " Diabetes Melitus: Review Etiologi, Patofisiologi, Gejala, Penyebab, Cara, ," *Jurnal Prosiding Biologi Achieving The Sustainable Development Goals With Biodiversity in ConfortingClimate Change*, vol. 7, no. 1, pp. 237-241, November 2021.
- [2] W. Teguh, *Fundamental Of Python For Machine Learning*, vol. 1, Klitren LorGK III, Yogyakarta: Penerbit Gava Media, 2018, p. x+150.
- [3] H. Dr. Yaya and W. Teguh, *Machine Learning (Konsep dan Implementasi)*,vol. 1, Klitren Lor GK III, Yogyakarta: Penerbit Gava Media, 2020, p. xiv+178.
- [4] U. I. Lestari, A. Y. Nadhiroh and C. Novia, "Penerapan *Metode K-Nearest Neighbor* Untuk Sistem Pendukung Keputusan Identifikasi Penyakit DiabetesMelitus," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 8, no. 4, pp. 2071-2082, Desember 2021.
- [5] R. Saxena, D. S. Khumar and M. Gupta, "Role of K-nearest neighbour in detection of Diabetes Mellitus," *Turkish Journal of Computer and MathematicsEducation*, vol. 12, no. 10, pp. 373-376, Januari 2021.
- [6] R. Kurniadi, R. Saedudin and V. P. Widartha, "Perbandingan Akurasi Algoritma *K-Nearest Neighbor Dan Logistic Regression* Untuk Klasifikasi Penyakit Diabetes," *e-Proceeding of Engineering*, vol. 8, no. 5, pp. 9757- 9764,Oktober 2021.
- [7] M. Nishom, "Perbandingan Akurasi *Euclidean Distance, Minkowski Distance,dan Manhattan Distance* pada Algoritma *KMeans Clustering berbasis Chi- Square*," *Jurnal Pengembangan IT (JPIT)*, vol. 4, no. 1, pp. 20-24, Januari 2019.
- [8] F. Yunita, "Sistem Klasifikasi Penyakit Diabetes Mellitus Menggunakan Metode *K-Nearest Neighbor (K-NN)*," *Jurnal BAPPEDA*, vol. 2, no. 1, pp. 223-230, April 2016.
- [9] S. Alifatul Q, S. Kautsar and Y. Ferry H, "Aplikasi Pembelajaran Matematika Dasar Bangun Datar Menggunakan Python Pada Perangkat Bergerak," vol. 11, no.3, pp. 13-22, Oktober 2016.
- [10] M.Salsabila, "Library Python Kenali Perbedaan *Module, Package*, dan



*Library Pada Python,*” [Online]. Available: [https:// https://www.dqlab.id/library-python-kenali-perbedaan-module-packagedan-library-pada-python](https://www.dqlab.id/library-python-kenali-perbedaan-module-packagedan-library-pada-python). [Accessed 2 Desember 2021].

[11] Y. Permana and R. Puji, "Perancangan Sistem Informasi Penjualan Perumahan Menggunakan Metode SDLC Pada PT. Mandiri Land Prosperous Berbasis Mobile," *Jurnal Teknologi Pelita Bangsa*, vol. 10, no. 2, pp. 153-167, Desember 2019.

[12] R. Resky, M. Rani and A. H. Yudi, "Implementasi Metode *Machine Learning* Menggunakan Algoritma *Evolving Artificial Neural Network* Pada Kasus Prediksi Diagnosis Diabetes," *Jurnal Aplikasi dan Teori Ilmu Komputer*, vol. 3, no. 2, pp. 85-97, September 2020.

[13] D. Setiabudidaya, "Penggunaan Piranti Lunak Jupyter Notebook dalam Upaya Mensosialisasikan Open Science," Palembang: Jurusan Fisika Universitas Sriwijaya, vol. 3, no. 2, pp. 1-5, 2018.

[14] R. Endang, P. Rully, "Mengenal *Machine Learning* Dengan Teknik *Supervised* dan *Unsupervised* Learning Menggunakan Python," *Jurnal Bina Insani ICT*, vol. 7, no. 2, pp. 156-165, Desember 2020.

## LAMPIRAN

### Data Pribadi

Nama lengkap : Imam Rayhan Ekadinata  
Tempat&tanggal lahir : Sungailiat, 13 Maret 2001  
Alamat rumah : Jl Asoka blok 3c No 1 Rss Pemda  
Jenis kelamin : Laki-Laki  
Agama : Islam  
No HP : 0895621338237  
Email : [imamrayhanekadinata@gmail.com](mailto:imamrayhanekadinata@gmail.com)



### Pendidikan

2006-2012 : SD Negeri 8 Sungailiat  
2012-2015 : SMP Negeri 1 Sungailiat  
2015-2018 : SMK Negeri 1 Sungailiat  
2018-2023 : Politeknik Manufaktur Negeri Babel

Sungailiat, 18 Januari 2023

Tanda Tangan

Imam Rayhan Ekadinata