

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/357478344>

# Penerapan Algoritma K-Nearest Neighbor untuk Mendeteksi Diabetes Berbasis Web Application

Research · January 2022

DOI: 10.13140/RG.2.2.14361.34400

CITATIONS

2

READS

1,695

3 authors, including:



[Mohammad Farrel Nur Rilwanu](#)

Indonesia University of Education

1 PUBLICATION 2 CITATIONS

SEE PROFILE



[Hibar Taufikurachman](#)

Indonesia University of Education

2 PUBLICATIONS 3 CITATIONS

SEE PROFILE

# Penerapan Algoritma *K-Nearest Neighbor* untuk Mendeteksi Diabetes Berbasis Web Application

Faris Huwaidi<sup>1)</sup>, Hibar Taufikurachman<sup>2)</sup>, dan Mohammad Farrel Nur Rilwanu<sup>3)</sup>

Rekayasa Perangkat Lunak, Kampus Daerah Cibiru  
Universitas Pendidikan Indonesia  
Bandung, Indonesia

[farishuwaidi@upi.edu](mailto:farishuwaidi@upi.edu), [hibartaufikurachman@upi.edu](mailto:hibartaufikurachman@upi.edu), [farrelnr@upi.edu](mailto:farrelnr@upi.edu)

## Abstract

*Indonesia is the fifth country with the most diabetes sufferers in the world. This is influenced by an unhealthy lifestyle and then coupled with a lack of public awareness to check whether he has diabetes or not. The KNN (K-Nearest Neighbor) algorithm can be used to predict whether a person has diabetes. By using a dataset from the Pima Indian Diabetes Database, the data training process was carried out using the KNN algorithm and obtained decent accuracy results using a jupyter notebook. From the results of the trained data set, it is then exported to be used in website development using the Python programming language. In the web application developed, the user is asked to input data on pregnancies (a person's pregnancy rate as long as he is alive), insulin levels, glucose levels, BMI, blood pressure, family history of diabetes, skin thickness, and age in the form of a slider. The input data is processed by the KNN algorithm to determine the outcome in the form of a positive or negative diabetes result based on the proximity of the new data entered with other data that has been trained.*

*Keywords: Diabetes, KNN, Prediction, Web Application, Dataset.*

## Abstrak

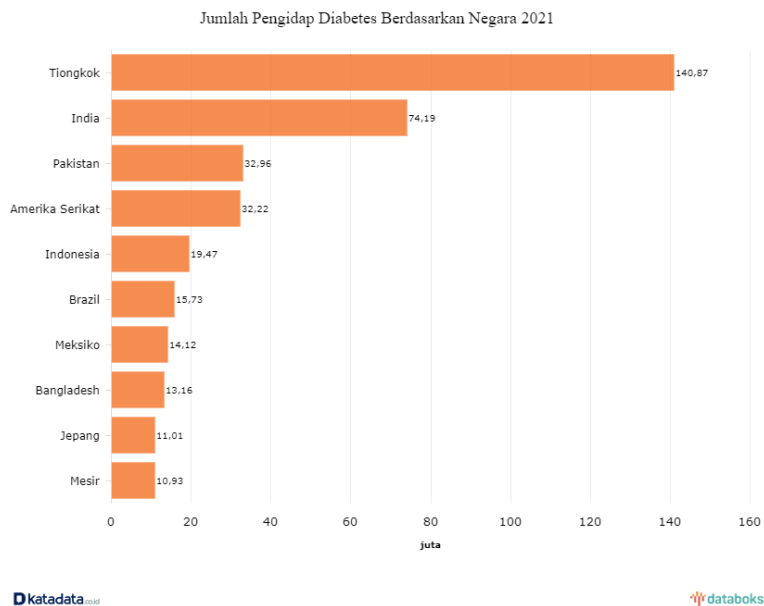
Indonesia merupakan negara peringkat kelima penderita diabetes terbanyak di dunia. Hal ini dipengaruhi pola hidup yang tidak sehat kemudian ditambah dengan kurangnya kesadaran masyarakat untuk mengecek apakah dia terkena diabetes atau tidak. Algoritma KNN (K-Nearest Neighbor) dapat digunakan untuk memprediksi apakah seseorang mengidap diabetes. Dengan menggunakan dataset dari Pima Indian Diabetes Database dilakukan proses pelatihan data menggunakan algoritma KNN dan didapatkan hasil akurasi yang lumayan menggunakan jupyter notebook. Dari hasil data set yang dilatih kemudian diekspor untuk dapat digunakan dalam pengembangan website menggunakan bahasa pemrograman python. Pada aplikasi web yang dikembangkan user diminta untuk menginputkan data pregnancies (angka kehamilan seseorang selama dia hidup), kadar insulin, kadar glukosa, BMI, Tekanan darah, riwayat diabetes dalam keluarga, ketebalan kulit, dan umur dalam bentuk slider. Data input diroses dengan algoritma KNN untuk menentukan hasil Outcome berupa hasil positif diabetes atau negatif berdasarkan kedekatan data baru yang di input dengan data lain yang telah dilatih.

Kata kunci: Diabetes, KNN, Prediksi, Aplikasi web, Dataset

## PENDAHULUAN

Diabetes adalah penyakit kronis yang terjadi ketika pankreas tidak lagi mampu membuat insulin atau ketika tubuh tidak dapat menggunakan insulin yang dihasilkannya dengan baik. Insulin adalah hormon yang dibuat oleh pankreas, yang bertindak seperti kunci untuk membiarkan glukosa dari makanan yang kita makan melewati aliran darah ke dalam sel-sel dalam tubuh untuk menghasilkan energi. Semua makanan karbohidrat dipecah menjadi glukosa dalam darah. Insulin membantu glukosa masuk ke dalam sel.

Indonesia merupakan salah satu dari 10 besar negara dengan jumlah penderita penyakit diabetes terbanyak di dunia. Pada tahun 1995 Indonesia yang masih tergolong negara yang baru berkembang dan menempati peringkat ke-7 dengan jumlah penderita diabetes sebanyak 4,5 juta jiwa. Peringkat tersebut diprediksi akan naik menjadi peringkat ke-5 pada tahun 2025 dengan perkiraan jumlah penderita 12,4 juta jiwa. Pada tahun 2021, *International Diabetes Federation* (IDF) mencatat bahwa Indonesia telah berada di posisi kelima dengan jumlah pengidap diabetes 19,47 juta dengan jumlah penduduk 179,72 juta. Ini berarti prevalensi diabetes di Indonesia sebesar 10,6% [1].



**Gambar 1. 1 Jumlah Pengidap Diabetes Berdasarkan Negara 2021**

Masyarakat umum pada kenyataannya tidak menyadari bahwa dirinya menderita penyakit diabetes sehingga terlambat untuk melakukan diagnosa sejak dini. Sehingga penderita dikhawatirkan tidak menjaga pola hidupnya dan tidak merawat diri karena tidak menyadari sebelumnya. Selain itu pola hidup yang tidak sehat juga dapat menyebabkan terjangkit diabetes di usia muda. Akibatnya ketika mereka datang ke layanan kesehatan sudah dalam kondisi yang parah. Oleh karena itu dibutuhkan sistem yang dapat mendeteksi apakah seseorang telah terjangkit diabetes atau tidak[2].

K-Nearest Neighbor (KNN) merupakan salah satu algoritma dengan teknik lazy learning yang menggunakan metode klasifikasi terhadap objek berdasarkan data yang jaraknya paling dekat dengan objek tersebut. KNN juga termasuk dalam kelompok *instance-based learning*[3].

Pada algoritma KNN, data berdimensi  $q$ , jarak dari data tersebut ke data yang lain dapat dihitung. nilai dari jarak inilah yang digunakan sebagai nilai kedekatan antara data uji dengan data latih[4]. Terdapat berbagai macam metode yang dapat diterapkan pada algoritma KNN dalam menghitung jarak kedekatan objek dengan data terdekat, namun yang paling umum digunakan adalah metode *Euclidean Distance*, bahkan digunakan sebagai *default* pada *library* SKLearn pada bahasa pemrograman Python saat menggunakan algoritma KNN. Metode penghitungan jarak lainnya yang umum digunakan pada algoritma KNN yaitu *Minkowski Distance* dan *Manhattan Distance*[5].

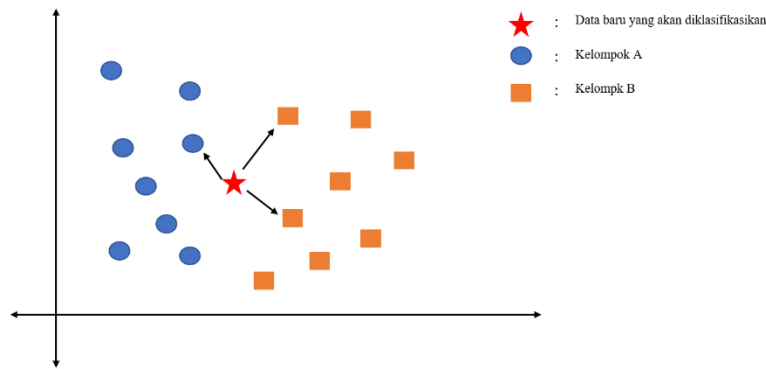
Dataset yang digunakan adalah **Pima Indians Diabetes Database** yang dipublikasikan oleh *UCI Machine Learning* pada situs *kaggle*. Alasan kami memilih dataset tersebut karena telah dipakai dan di *vote count* hampir 3 ribu pengguna pada situs *kaggle* yang merupakan sebuah portal komunitas ilmu data dan pembelajaran mesin di seluruh dunia.

Untuk dapat mendeteksi apakah seseorang menderita diabetes atau tidak maka dibuatlah sebuah sistem pendeteksi diabetes dengan menggunakan algoritma klasifikasi *k-nearest neighbor* (KNN).

## KONSEP DASAR

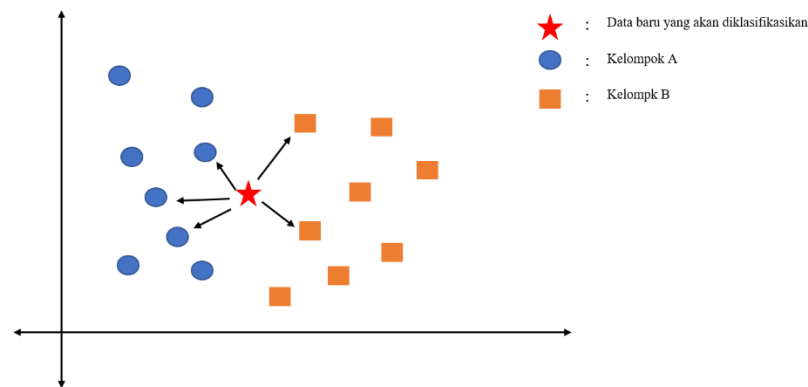
### Menentukan Nilai $k$

Menentukan nilai  $k$  atau jumlah tetangga merupakan langkah awal untuk membangun model pembelajaran berdasarkan algoritma K-Nearest Neighbor. Nilai  $k$  merupakan jumlah data yang diperhitungkan dalam menentukan klasifikasi saat ada input data yang baru. Misal jika kita ingin mengelompokkan sebuah nilai yang dapat digambarkan pada bidang kartesius, lalu kita tentukan nilai  $k$  sebanyak tiga (tiga tetangga terdekat), maka nilai tersebut akan menetapkan tiga tetangga terdekatnya.



**Gambar 2. 1 Contoh penerapan Algoritma K-NN dengan 3 objek**

Data/objek baru tersebut akan diklasifikasikan sebagai Kelompok B karena data tersebut memiliki tetangga Kelompok B yang lebih banyak (dua tetangga), sedangkan data tersebut hanya memiliki satu tetangga Kelompok A. Jika kita ubah nilai k-nya, maka penentuan jumlah tetangganya pun akan sedikit berubah. Semisal kita ubah nilai k menjadi 5.



**Gambar 2. 2 Contoh penerapan Algoritma K-NN dengan 5 objek**

Data tersebut akan diklasifikasikan sebagai Kelompok A karena memiliki tetangga anggota Kelompok A sebanyak 3, dibandingkan dengan kedekatannya dengan anggota Kelompok B yang hanya 2. Penentuan k atau jumlah tetangga terdekat ini merupakan konsep yang penting pada algoritma KNN.

#### Distance Metrics

Sebelumnya sedikit disinggung mengenai matriks penghitungan jarak atau distance metrics yang umum digunakan pada algoritma KNN untuk mengukur kedekatan objek dengan data terdekat yaitu Euclidean distance, Minkowski distance, dan Manhattan distance. Untuk membuat sebuah model pembelajaran mesin dengan performa yang baik tentu tidak cukup menggunakan parameter atau metode default saja. Dengan ketiga metode matriks perhitungan jarak yang berbeda-beda, memungkinkan kita mendapatkan model pembelajaran performa terbaik.

#### Euclidean Distance

Euclidean distance adalah metode penghitungan jarak garis lurus terhadap dua objek yang berbeda. Metode ini dapat diterapkan ke dalam ruang 1, 2, dan 3 dimensi[6]. Penghitungan jarak pada ruang 1-dimensional dapat digambarkan dengan formula sebagai berikut.

$$d(x_1, x_2) = \sqrt{\sum_{i=0}^n (x_{1i} - x_{2i})^2}$$

Dalam implementasinya pada model pembelajaran mesin, rumus penghitungannya dapat berubah tergantung banyaknya independent variabel pada dataset yang digunakan sebagai data latih. Jika dataset memiliki dua atau lebih independent variabel begitu juga dimensi yang dihitungnya bertambah, maka formulanya menjadi seperti berikut.

$$d = \sqrt{\sum_{i=1}^n (x_{1i} - x_{2i})^2 + (y_{1i} - y_{2i})^2 + (z_{1i} - z_{2i})^2 + \dots}$$

### Manhattan Distance

Manhattan distance digunakan untuk mengambil kasus yang cocok dari basis kasus dengan menghitung jumlah bobot absolute dari perbedaan antara kasus yang sekarang dan kasus yang lain[7]. Untuk menghitung bobot digunakan persamaan berikut:

$$d_{ij} = \sum W_k |x_{ik} - c_{jk}|$$

Dimana diketahui dij adalah jarak antara kasus antara i th dan jth dengan semua parameternya. W merepresentasikan jumlah dari bobot. X adalah kasus yang baru dikurangi dengan C yaitu history (kasus yang ada dalam Casse Base).

### Minkowski Distance

Minkowski distance merupakan sebuah metrik dalam ruang vektor di mana suatu norma didefinisikan (normed vector space) sekaligus dianggap sebagai generalisasi dari Euclidean distance dan Manhattan distance. Dalam pengukuran jarak objek menggunakan Minkowski distance biasanya digunakan nilai p adalah 1 atau 2[8]. Berikut rumus yang digunakan menghitung jarak dalam metode ini.

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

Dimana.

d = jarak antara x dan y

x = data pusat klaster

y = data pada atribut

i = setiap data

n = jumlah data,

xi = data pada pusat klaster ke i

yi = data pada setiap data ke i

p = power

### Variabel Pengujian

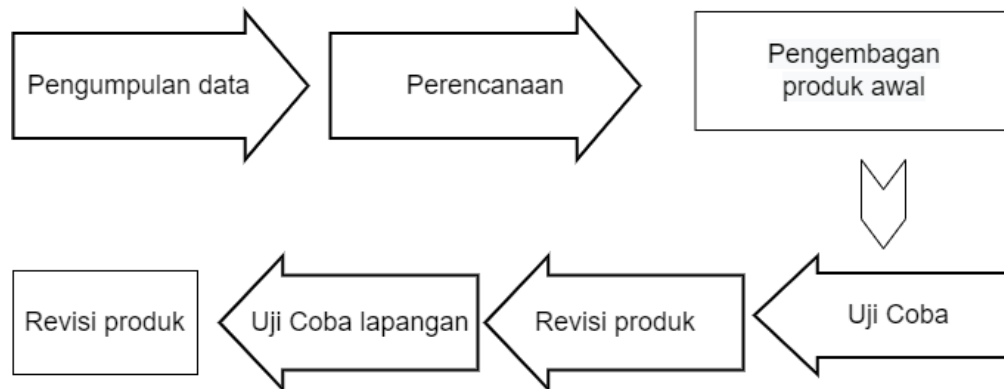
Dalam data set untuk menentukan apakah seseorang menderita diabetes atau tidak terdapat 9 variabel yang terdiri dari 8 variabel independen(x) dan 1 variabel dependen(y). Adapun variabel independen (x) sebagai berikut:

- Pregnancies: persentasi berapa kali seorang wanita pernah hamil selama hidupnya
- Insulin: tingkat insulin dalam 2 jam insulin serum dengan satuan mu U/ml
- Glucose: persentasi gula darah pada 2 jam dalam tes toleransi glukosa
- BMI: indeks massa tubuh (berat dalam kg/tinggi dalam meter)
- Blood Presure: tekanan darah dalam satuan mm/hg
- Diabetes Pidegree Function: indikator riwayat diabetes dalam keluarga atau keturunan
- Skin Thickness: nilai yang digunakan untuk mengukur lemak tubuh yang diukur pada lengan kanan setengah antara proses olecranon dari siku dan proses akromial
- Age: umur dari seorang sample

sedangkan variabel dependen(y) merupakan output dari hasil prediksi dengan class variable 0 atau 1, 0 untuk tidak mengidap diabetes dan 1 untuk hasil positif mengidap diabetes

### METODE PENELITIAN

Metode penelitian yang digunakan adalah R&D atau Research and Development, yaitu metode yang bertujuan untuk dapat mengembangkan produk lalu menguji dan mengembangkannya sampai produk tersebut dapat berjalan sesuai yang diharapkan [9].



**Gambar 3. 1 Alur tahap penelitian**

Dari gambar diatas dapat dilihat tahapan yang dilakukan pada penelitian adalah sebagai berikut.

### **Pengumpulan Data**

Pada tahap ini, dilakukan beberapa langkah kegiatan yaitu mengumpulkan informasi terkait penyakit diabetes, algoritma *K-Nearest Neighbor* beserta implementasinya, juga mencari dataset yang sesuai dengan kasus terkait dan cocok untuk melakukan pelatihan model pembelajaran mesin serta pengimplementasiannya. Selain itu, tahap ini juga menganalisa *tools* apa saja yang sekiranya dibutuhkan saat proses pengembangan *web application* sebagai implementasi dari model pembelajaran mesin dengan algoritma *K-Nearest Neighbor*.

### **Perencanaan**

Pada tahap perencanaan, desain produk dirancang dengan memperhatikan aspek-aspek yang penting mulai dari tujuan produk dirancang yaitu untuk dapat memprediksi apakah seseorang mengidap diabetes atau tidak. Kemudian menemukan target penggunanya yaitu orang dewasa dengan umur diatas 20 tahun. Selain itu data yang diperoleh pada tahap sebelumnya akan diolah untuk dijadikan pertimbangan pengambilan keputusan mengenai *tools* apa yang akan digunakan untuk mengembangkan aplikasi ini, kemudian bahasa pemrograman yang cocok dengan *tools* dan kasus yang ada.

### **Pengembangan produk awal**

Pengembangan produk awal dimulai dengan melatih dan membangun model pembelajaran mesin berdasarkan *dataset* yang dipakai dengan melewati berbagai tahap yaitu: *import library* dan *dataset*, *exploratory data analysis*, *data preprocessing*, *splitting*, dan *modeling*. Setelah itu dilakukan perancangan aplikasi web menggunakan template yang sudah dengan bahasa pemrograman Python. Kemudian hasil dari pelatihan data menggunakan algoritma KNN di ekspor ke dalam aplikasi web sehingga web dapat menghasilkan output berdasarkan inputan dari pengguna.

### **Uji coba**

Pengujian dibagi ke dalam dua bagian. Pengujian pertama dilakukan dalam membangun dan mengembangkan model pembelajaran mesin. Pengujian pertama dilakukan dengan melihat dan memahami performa model pembelajaran mesin dengan memperhatikan beberapa aspek yaitu: akurasi atau *score* model, *dataset* yang sudah diolah dan dirapikan, dan parameter yang digunakan pada algoritma KNN. Dengan memperhatikan ketiga aspek tersebut dapat memaksimalkan kinerja dan performa dari model pembelajaran mesin yang merupakan inti dari pengembangan produk.

Pengujian kedua dilakukan pada proses pengembangan *web application* dan *deploying* produk dengan *hosting* produk dengan layanan *Heroku*. Aspek yang diperhatikan adalah kelancaran dalam membangun *web application*, keberhasilan *hosting* produk, dan mampu atau tidaknya produk melakukan prediksi melalui *web application* yang telah di *hosting*.

### **Revisi produk**

Melakukan revisi produk untuk meningkatkan kualitas produk dengan menyempurnakan performa model pembelajaran mesin. Penyempurnaan model pembelajaran mesin dilakukan dengan mencari parameter terbaik agar model menghasilkan *score* yang lebih tinggi dari sebelumnya (*model evaluation* dan *hyperparameter tuning*). Selain itu melakukan beberapa revisi pada pengembangan *web application* untuk meningkatkan kualitas UI/UX agar pengguna dapat dengan mudah dan nyaman menggunakan *web application* tersebut.

### **Uji coba lapangan**

Melakukan uji coba terhadap produk untuk dapat melakukan prediksi diagnosis diabetes dengan *input* pada *form* yang ada pada *web application* dan melihat hasil/output yang diberikan oleh produk.

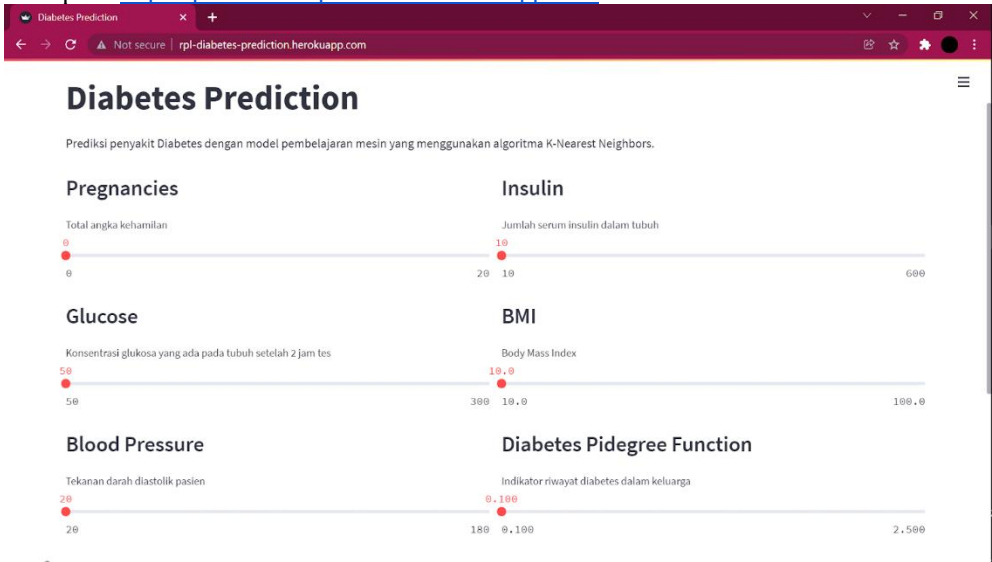
### **Revisi Produk**

Revisi produk dilakukan kembali setelah melakukan pengujian lapangan berdasarkan beberapa pertimbangan diantaranya adalah hasil analisa uji, saran, dan masukan saat pengujian lapangan dilakukan.

**HASIL DAN PEMBAHASAN**

**Nama Aplikasi**

Aplikasi yang dibangun diberi nama sesuai dengan fungsi utamanya yaitu Diabetes Prediction yang dapat diakses pada <http://rpl-diabetes-prediction.herokuapp.com>.



**Gambar 4. 1 Tampilan dari aplikasi web Diabetes Prediction**

**Tampilan dan Fitur Aplikasi**

Aplikasi Diabetes Prediction ini sudah sampai pada tahap finalisasi. Dapat dilihat dengan tampilan antarmuka pengguna pada lampiran 3. aplikasi ini memiliki fitur-fitur dijelaskan pada tabel dibawah ini.Implementasi Penggunaan Aplikasi

**Tabel 4. 1 Fitur dan deskripsi mengenai aplikasi**

No.	Fitur	Deskripsi
1	Deteksi Diabetes pada halaman utama	fitur ini meminta beberapa inputan dari user berupa variabel dependen yaitu mulai dari pregnancies (angka kehamilan seseorang selama dia hidup), kadar insulin, kadar glukosa, BMI, Tekanan darah, riwayat diabetes dalam keluarga, ketebalan kulit, dan umur. setelah user menginput semuanya dan menekan tombol prediksi maka data input tersebut dianalisis menggunakan KNN berdasarkan kedekatannya dengan data yang lain pada data latihan kemudian dikeluarkan hasil outcome berupa pernyataan apakah user tersebut positif diabetes atau negatif.
2	Setting	berupa fitur untuk pengaturan pada aplikasi yang terdiri dari development, appreerance (mengatur tampilan lebar atau sempit), dan theme untuk mengganti tema tampilan light, dark, atau mengikuti sistem.
3	Report a Bug	fitur ini berfungsi untuk melaporkan bug yang di dapatkan oleh user ke develper dengan melalui github
4	Get Help	fitur untuk melihat bantuan penjelasan atas apa yang dibutuhkan oleh user

**Implementasi dan Penggunaan Aplikasi**

Setelah melatih model pembelajarn mesin menggunakan algoritma K-Nearest Neighbor, kita dapat melakukan prediksi pada model tersebut untuk melihat keakuratan model berdasarkan diagnosa yang diprediksi. Kita dapat memasukkan data yang berisi data independent variabel. Semisal untuk melihat apakah

model dapat melakukan prediksi, kita dapat memasukkan data yaitu Pregnancies = 1, Glucose = 202, BloodPressure = 108, SkinThickness = 52, Insulin = 131, BMI = 48.5, DiabetesPidegreeFunction = 1.114, dan Age = 44.

```
# Preg: 1, Glu: 202, Bp: 108, SkinT: 52, Insulin: 131, BMI: 48.5, DBF: 1.114, Age: 44
input_data_1 = [[1, 202, 108, 52, 131, 48.5, 1.114, 44]]
```

```
pred = new_knn_model.predict(input_data)
```

```
pred[0]
```

1

**Gambar 4. 2 Implementasi Menggunakan Jupyter Notebook**

Hasil prediksi dari model berdasarkan data yang dimasukkan adalah '1' atau positif Diabetes. Tentu hasil diagnosis akan berbeda-beda tergantung inputan yang dimasukkan. Model yang dibuat juga diimplementasikan pada aplikasi berbasis website yang telah di-hosting agar model pembelajaran mesin dapat diakses dan dicoba oleh orang-orang.

The screenshot shows a web browser with the URL 'rpl-diabetes-prediction.herokuapp.com'. The application has six sliders for inputting data: Glucose (202), BMI (48.5), Blood Pressure (108), Diabetes Pidegree Function (1.114), Skin Thickness (52), and Age (44). Below the sliders is a 'PREDIKSI!' button. The result displayed is 'HASIL PREDIKSI: POSITIF'.

**Gambar 4. 3 Implementasi Menggunakan Aplikasi Website**

Hasil yang diinputkan melalui jupyter notebook dan juga aplikasi website akan sama hasilnya.

## KESIMPULAN

Aplikasi Diabetes Prediction secara keseluruhan telah berhasil dibangun dengan menggunakan Algoritma KNN dan menggunakan dataset Pima Indian Diabetes Database. Aplikasi web ini telah dapat berfungsi sebagaimana mestinya yaitu user diminta untuk menginputkan data pregnancies(angke kehamilan seseorang selama dia hidup), kadar insulin, kadar glukosa, BMI, Tekanan darah, riwayat diabetes dalam keluarga, ketebalan kulit, dan umur dalam bentuk slider. Data input diproses dengan algoritma KNN untuk menentukan hasil Outcome berupa hasil positif diabetes atau negatif berdasarkan kedekatan data baru yang di input dengan data lain yang telah dilatih.

## DAFTAR RUJUKAN

- [1] R. Yosmar, D. Almasdy, and F. Rahma, "Survei Risiko Penyakit Diabetes Melitus Terhadap Masyarakat Kota Padang," *J. Sains Farm. Klin.*, vol. 5, p. 134, Aug. 2018, doi: 10.25077/jsfk.5.2.134-141.2018.
- [2] I. Luthfa, "Implementasi Selfcare Activity Penderita Diabetes Mellitus di Wilayah Puskesmas Bangetayu Semarang," *Bul. Penelit. Kesehat.*, vol. 47, no. 1, pp. 23–28, 2019, doi: 10.22435/bpk.v47i1.779.
- [3] H. Leidiyana, "Penerapan Algoritma K-Nearest Neighbor Untuk Penentuan Resiko Kredit Kepemilikan Kendaraan Bermotor," *J. Penelit. Ilmu Komputer, Syst. Embed. Log.*, vol. 1, no. 1, pp. 65–76, 2013.
- [4] M. Lestari, "Penerapan Algoritma Klasifikasi Nearest Neighbor (K-NN) untuk Mendeteksi Penyakit Jantung," *Fakt. Exacta*, vol. 7, no. September 2010, pp. 366–371, 2014.
- [5] C. W. Yean *et al.*, "Analysis of the Distance Metrics of KNN Classifier for EEG Signal in Stroke



- Patients,” *2018 Int. Conf. Comput. Approach Smart Syst. Des. Appl. ICASSDA 2018*, pp. 1–4, 2018, doi: 10.1109/ICASSDA.2018.8477601.
- [6] C. A. Pamungkas, “Aplikasi Penghitung Jarak Koordinat Berdasarkan Latitude Dan Longitude Dengan Metode Euclidean Distance Dan Metode Haversine,” *J. Inf. Politek. Indonusa Surakarta*, vol. 5, no. 2, pp. 8–13, 2019.
  - [7] K. Latifah, “Kombinasi Algorithma K-NN dan Manhattan Distance untuk Menentukan Pemenang Lelang,” *J. Inform. Upgris*, vol. 1, pp. 49–58, 2015, [Online]. Available: <https://docplayer.info/34038947-Kombinasi-algorithma-k-nn-dan-manhattan-distance-untuk-menentukan-pemenang-lelang.html>.
  - [8] M. Nishom, “Perbandingan Akurasi Euclidean Distance, Minkowski Distance, dan Manhattan Distance pada Algoritma K-Means Clustering berbasis Chi-Square,” *J. Inform. J. Pengemb. IT*, vol. 4, pp. 20–24, Jan. 2019, doi: 10.30591/jpit.v4i1.1253.
  - [9] S. Haryati, “RESEARCH AND DEVELOPMENT (R&D) SEBAGAI SALAH SATU MODEL PENELITIAN DALAM BIDANG PENDIDIKAN,” *Maj. Ilm. Din.*, no. vol.37.No.1, 15 Januari 2012, 2012, [Online]. Available: <http://jurnal.utm.ac.id/index.php/MID/article/view/13>.