



Pengembangan Sistem Pendukung Keputusan Untuk Prediksi Diabetes

Achmad Farhan Aldyno¹, Faiza Ulinnuha Junaidi², Haidar Rabbani³, Ahlam Nauf Oda⁴, Achmad Pratama Rifai⁵

^{1,2,3,4,5}Departemen Teknik Mesin dan Industri, Universitas Gadjah Mada, Indonesia
Email: achmad.f.a@mail.ugm.ac.id¹, achmad.p.rifai@ugm.ac.id⁵

Abstract

Diabetes is one of the major health issues worldwide, affecting 10.5% of the total adult population (20-79 years old). Often referred to as the silent killer, nearly half of those affected by diabetes are unaware of their condition. Diabetes is categorized into several types, namely type 1 diabetes mellitus, type 2 diabetes mellitus, and gestational diabetes. Detection of diabetes can be carried out through various methods, including blood sugar level tests, Hemoglobin A1c (HbA1c) tests, oral glucose tolerance tests, as well as physical examinations and medical history reviews by doctors. Interpreting the results of these tests can be used to identify the potential for an individual to have diabetes, employing a machine learning approach as a decision support system for doctors to make informed decisions, and also providing patients with reminders to consult with a doctor. In the machine learning model we've developed, we trained and tested algorithms using the 'Diabetes prediction dataset,' consisting of 8 variables: age, gender, Body Mass Index (BMI), hypertension, heart disease, smoking history, HbA1c level, and blood glucose level. The algorithm employed was the Artificial Neural Network (ANN) with the optimizer using Stochastic Gradient Descent (SGD). This application is intended to serve as a decision support system for doctors and the general public. It's designed using Anvil for 8 types of input variables, providing 2 output variables: the percentage of an individual's potential to have diabetes and suggestions for preventing such risks.

Keywords: Diabetes, prediction, artificial neural networks, decision support systems

Abstrak

Diabetes merupakan salah satu masalah kesehatan utama di seluruh dunia, mencapai 10,5% dari total populasi orang dewasa (20-79 tahun). Diabetes seringkali disebut sebagai silent killer karena hampir separuh penderitanya tidak menyadari bahwa mereka menderita diabetes. Diabetes digolongkan menjadi beberapa jenis, yaitu diabetes melitus tipe 1, diabetes melitus tipe 2 dan diabetes gestasional. Deteksi diabetes dapat dilakukan dengan berbagai cara, antara lain pemeriksaan kadar gula dalam darah, tes Hemoglobin A1c (HbA1c), tes toleransi glukosa oral, serta pemeriksaan fisik dan riwayat kesehatan oleh dokter. Interpretasi hasil tes tersebut dapat digunakan untuk mendeteksi potensi seseorang menderita diabetes dengan pendekatan machine learning sebagai sistem pendukung keputusan bagi dokter untuk mengambil keputusan, juga bermanfaat bagi pasien sebagai pengingat untuk memeriksakan diri ke dokter. Pada model machine learning yang kami buat, kami melatih dan menguji algoritma menggunakan dataset "Diabetes prediction dataset", yang terdiri dari 8 variabel, yaitu age, gender, Body Mass Index (BMI), hypertension, heart disease, smoking history, HbA1c level, dan blood glucose level. Algoritma yang digunakan adalah Artificial Neural Network (ANN) dengan optimizer menggunakan Stochastic Gradient Descent (SGD). Aplikasi ini diharapkan dapat menjadi sistem pendukung keputusan bagi dokter dan masyarakat umum, yang dirancang dengan menggunakan Anvil untuk 8 jenis variabel input dan memberikan 2 variabel output berupa persentase potensi seseorang menderita diabetes serta saran yang dapat dilakukan untuk mencegah risiko tersebut.

Kata kunci: Diabetes, prediksi, artificial neural networks, sistem pendukung keputusan

1. PENDAHULUAN

Diabetes merupakan salah satu masalah kesehatan utama di seluruh dunia, yang dapat menyebabkan berbagai komplikasi serius jika tidak didiagnosis dan dikelola dengan tepat. Berdasarkan laporan International Diabetes Federation Atlas 2021, jumlah penderita diabetes di dunia mencapai 10,5% dari total populasi orang dewasa (20-79 tahun), dengan hampir separuhnya tidak menyadari bahwa mereka menderita diabetes. Jumlah penderita diabetes terus mengalami peningkatan dari tahun ke tahun. IDF memprediksi bahwa pada tahun 2045 jumlah penderita diabetes akan mengalami peningkatan sebesar 46% [1]. Diabetes seringkali disebut juga sebagai *silent killer*, mengingat kebanyakan dari penderita tidak mengetahui jika mereka memiliki penyakit tersebut dari awal dan secara mendadak mengalami komplikasi pada beberapa organ tubuh yang dimiliki. Diabetes sendiri dapat digolongkan menjadi beberapa jenis, yaitu diabetes melitus tipe 1, diabetes melitus tipe 2 dan diabetes gestasional.

Deteksi diabetes pada umumnya dilakukan dengan menggunakan beberapa cara, yang biasanya meliputi pemeriksaan kadar gula darah, tes Hemoglobin A1c (HbA1c), tes toleransi glukosa oral, serta pemeriksaan fisik dan riwayat kesehatan oleh dokter. Tren terkini dalam deteksi diabetes adalah dengan ikut menggabungkan *machine learning* sebagai sistem pendukung keputusan dan bahan pertimbangan tambahan dokter. Algoritma *Machine Learning* (ML) dapat menjadi alat bantu yang berguna bagi dokter dalam menyusun diagnosis yang lebih akurat dan rencana perawatan yang sesuai. Dalam praktek klinis, kombinasi antara kecerdasan buatan (seperti ML) dengan pengetahuan medis dan pengalaman dokter adalah yang terbaik untuk memberikan perawatan yang optimal kepada pasien. Kedua aspek ini, bila digabungkan, dapat meningkatkan kualitas diagnosa dan perawatan yang diberikan oleh dokter kepada pasien.

Dengan pengecekan dan diagnosa diabetes yang lebih baik, pasien berisiko tinggi dapat diberikan pendidikan kesehatan, nasihat nutrisi, dan rencana pengelolaan gaya hidup yang dapat secara signifikan mengurangi kemungkinan berkembangnya diabetes, utamanya diabetes tipe 2. Hal ini sangat penting di Indonesia yang sistem kesehatannya menghadapi berbagai tantangan, termasuk keterbatasan sumber daya dan akses terhadap perawatan kesehatan.

2. METODOLOGI PENELITIAN

Pada beberapa tahun terakhir *Machine Learning* (ML) sering diaplikasikan dalam sistem pendukung keputusan yang bersifat klinis pada ranah diabetes. Banyak penelitian menggunakan dataset medis yang berkaitan dengan diabetes untuk mendapatkan tujuan berupa melakukan diagnosa, penilaian risiko, prediksi awal, dan prognosis pada pasien [2]. Dataset ini sering kali terdiri dari informasi seperti riwayat medis, hasil tes laboratorium, informasi genetik, dan data lainnya yang relevan dengan kondisi kesehatan pasien.

Berbagai jenis algoritma ML dapat diterapkan untuk deteksi diabetes, termasuk Regresi Logistik [3-4], Decision Trees [5-6], Random Forests [7], Support Vector Machines [8-9], Artificial Neural Networks [10-12], dan lain-lain. Setiap algoritma memiliki kelebihan dan kelemahan yang berbeda dalam menangani

dataset tertentu. Identifikasi fitur-fitur penting yang berhubungan dengan diabetes juga menjadi beberapa fokus utama dari penelitian lain. Faktor-faktor yang memiliki korelasi signifikan dengan kemungkinan seseorang terkena diabetes, seperti kadar glukosa darah, BMI, riwayat keluarga, tekanan darah, dan lain sebagainya menjadi perhatian penelitian beberapa tahun ke belakang [13-15].

Setelah mendapatkan algoritma ML dan faktor-faktor yang akan digunakan, penting untuk melakukan validasi terhadap model yang digunakan. Untuk kasus kategorisasi biner, metrik validasi yang biasanya digunakan adalah *Log Loss (Binary-Entropy)*, serta metrik *'accuracy'*, *'precision'*, dan *'recall'*. Selama pelatihan model, metrik dihitung pada dataset validasi terpisah untuk membantu mengukur kinerja model dalam mengklasifikasikan pasien dengan atau tanpa diabetes. Selain membangun model yang akurat, interpretasi hasil juga menjadi perhatian penting. Meskipun ada banyak penelitian yang menunjukkan hasil yang menjanjikan dalam deteksi diabetes menggunakan ML, implementasi klinis dari model-model ini masih memerlukan validasi lebih lanjut dan integrasi dengan praktik klinis yang ada. Untuk mengintegrasikan model ini ke dalam praktik medis secara efektif, diperlukan pengembangan sistem yang intuitif dan mudah digunakan, terutama bagi dokter yang mungkin tidak memiliki latar belakang teknis yang mendalam. Adanya Sistem Pendukung Keputusan (*Decision Support System / DSS*) dengan antarmuka grafis yang mudah dapat menjadi solusi yang sangat berguna dalam membantu dokter dalam proses diagnosa dan masyarakat dalam mengenali kondisi kesehatannya sendiri.

Dengan demikian, integrasi model ML ke dalam sistem pendukung keputusan yang memiliki antarmuka grafis yang ramah pengguna akan membantu dokter maupun masyarakat awam dalam membuat keputusan yang lebih baik dan lebih terinformasi terkait kesehatan badan. Hal ini dapat meningkatkan efisiensi proses medis serta akurasi diagnosa, yang pada gilirannya dapat memberikan manfaat yang besar bagi pasien.

3. HASIL DAN PEMBAHASAN

3.1. Pengembangan Model Prediksi Diabetes

Dataset yang digunakan terdiri dari 8 variabel, yaitu : *age, gender, Body Mass Index (BMI), hypertension, heart disease, smoking history, HbA1c level, dan blood glucose level*, dengan jumlah total 100.000 sampel. Pada dataset ini tidak ditemukan nilai *null*. Dataset yang didapat diasumsikan bersih dan sudah *ter-encode* dengan benar, tanpa perlu melakukan data *cleaning* ataupun data *preprocessing* yang kompleks. Dataset diabetes didapatkan dari laman Kaggle [16] dalam format *.csv*. Penjelasan masing-masing variabel dapat dilihat pada Tabel 1.

Tabel 1. Variasi Dataset

Variable	Data Type	Data Range / Category	Label Encoded (for categorical data)
Gender	Categorical	{Male, Female, Other}	Male = 1, Female = 2, Other = 3
Age	Numerical	0 - 80	
Hypertension	Categorical	{have hypertension, don't have}	1 = have, 0 = don't have

Variable	Data Type	Data Range / Category	Label Encoded (for categorical data)
Heart_disease	Categorical	{hypertension}, {have heart disease, don't have heart disease}	1 = have, 0 = don't have
Smoking_history	Categorical	{'never': 1, 'former': 2, 'not current': 3, 'current': 4, 'ever': 5, 'No Info': 6}	{1,2,3,4,5,6} sequentially
Bmi	Numerical	10.16 - 71.55	
HbA1c_level	Numerical	3.5 - 9	
Blood_glucose_level	Numerical	80 - 300	
Diabetes	Categorical	{presence of diabetes, absence of diabetes}	1 = presence, 0 = absence

Data kemudian dikelompokkan menjadi dua kelompok terpisah, yakni *training dataset* yang digunakan untuk melatih algoritma, dan *test dataset* yang digunakan untuk menguji algoritma dengan jumlah *test size* sebanyak 20% atau sekitar 20.000 dari 100.000 data. Algoritma yang digunakan adalah *Artificial Neural Network* (ANN). Algoritma ini berusaha menyerupai cara kerja neuron yang ada pada otak manusia. ANN ini dibuat dengan menggunakan hyperparametric berupa *Stochastic Gradient Descent* (SGD) dari *library* Keras dan disimpan dalam format .h5 untuk kemudian diunggah dan disimpan di *server module* Anvil. Algoritma ini kemudian diuji dan dicoba dengan jumlah *hidden layer* sebanyak 10, *batch size* sebanyak 5000, dan jumlah iterasi (epoch) setiap *training*-nya sebanyak 32. *Library* yang digunakan adalah NumPy, Pandas, TensorFlow, Joblib, ScikitLearn dan Anvil Server. Dalam melatih algoritma ini juga menggunakan *feature scaling* yang kemudian skalanya disimpan dalam bentuk .pkl dengan menggunakan *library* StandardScaler serta joblib. Untuk *coding* dapat dilihat pada Gambar 1 dan Gambar 2. Pelatihan model dapat dilihat pada Gambar 3 dan Gambar 4.

```

import numpy as np
import pandas as pd

import tensorflow as tf
from tensorflow.keras.metrics import Precision, Recall
import keras

from sklearn import datasets
from sklearn.model_selection import train_test_split as tts
from sklearn.preprocessing import StandardScaler
    
```

Gambar 1. Sebagian *library* yang digunakan



```
from sklearn.preprocessing import StandardScaler
import joblib

sc = StandardScaler()

# Developing ANN model
# Sequential itu simple input-hidden-output layer
ann = tf.keras.models.Sequential()

# Adding Hidden Layer
# units = neuron number
ann.add(tf.keras.layers.Dense(units=10, activation='relu',
input_shape=(X.shape[1],)))

# Adding the output layer with sigmoid activation for binary
classification
ann.add(tf.keras.layers.Dense(units=1, activation='sigmoid'))
```

Gambar 2. Pembentukan *hidden* dan *output layer*

```
X_train, X_test, Y_train, Y_test = tts(X, Y, test_size = 0.2,
random_state=1)

# Feature scaling
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)

print("X_train:", X_train,
      '\n', "X_test", X_test)

# Save feature scaler
scalerSavePath = '/content/drive/MyDrive/UGM - DRIVE
SYNC/Semester 7/Sistem Pendukung Keputusan/Proyek Pasca
UTS/Cek Diabetes/cekDiabetesScaler_02.pkl'
joblib.dump(sc, scalerSavePath)
```

Gambar 3. Pembagian data untuk *training* dan untuk *test* serta *feature scaling*

```
#determine hyperparameter
ann.compile(optimizer=tf.keras.optimizers.SGD(learning_rate=0.01, momentum=0.0),
            loss='binary_crossentropy',
            metrics=['accuracy', tf.keras.metrics.Precision(),
tf.keras.metrics.Recall()])

# training & evaluate
ann.fit(X_train, Y_train, class_weight=class_weight_dict, batch_size = 2000,
epochs = 32)

##Stop the code manually if it's good enough.
##Personally we stopped at 11 - 90 - 85 - 65 percent-ish

precision = 0
recall = 0

while precision <= 75 or recall <= 75: # Continue training until both Precision
and Recall > 75
    ann.compile(optimizer=tf.keras.optimizers.SGD(learning_rate=0.01,
momentum=0.0),
                loss='binary_crossentropy',
                metrics=['accuracy', tf.keras.metrics.Precision(),
tf.keras.metrics.Recall()])

    ann.fit(X_train, Y_train, batch_size=5000, epochs=32, verbose=0)

    _, accuracy, precision, recall = ann.evaluate(X_test, Y_test, verbose=0)
    print(f'Precision: {precision*100:.2f}%, Recall: {recall*100:.2f}%')

print("Desired Precision and Recall achieved!")
```

Gambar 4. *Training model*

Model kemudian dievaluasi untuk melihat hasil evaluasi dari model ANN dengan menggunakan metrik yang dijelaskan dalam Tabel 2. Ada empat metrik yang digunakan dalam melakukan evaluasi model, yakni *loss* untuk variabel X_{test} dan *accuracy*, *precision*, dan *recall* untuk variabel Y_{test} . Masing-masing metrik memiliki kelebihan dan kekurangannya masing-masing, sehingga untuk menguji validasi model ML ada baiknya menggunakan beragam metrik dan algoritma. Metrik *loss* menggunakan algoritma Fungsi *cross-entropy loss* untuk mengukur perbedaan antara label sebenarnya (*true labels*) dan label yang diprediksi (*predicted labels*). Fungsi ini cocok digunakan pada aplikasi klasifikasi *biner* (0 atau 1, menderita diabetes atau tidak), yang mana kita ingin menilai seberapa baik model telah memprediksi label yang benar untuk setiap sampel data. Adapun ilustrasi *confusion matrix* dan rumus dari *metrik accuracy*, *precision*, dan *recall* dapat dilihat pada Gambar 5.

Tabel 2. Hasil evaluasi model ML

Metrik	Deskripsi	Nilai	Penjelasan dalam Konteks Diabetes
<i>Loss</i>	Mewakili kesalahan antara prediksi & aktual	0,1122	<i>Loss</i> yang lebih rendah menandakan akurasi model dalam memprediksi hasil terkait diabetes lebih dekat dengan nilai aktual.
<i>Accuracy</i>	Keakuratan secara keseluruhan dari prediksi model	96,15%	Dari semua individu yang diuji untuk diabetes, model memprediksi dengan benar sekitar 96,15% dari dataset yang ada
<i>Precision</i>	Akurasi prediksi positif	~87,82 %	Di antara yang diprediksi menderita diabetes, sekitar 87,82% dari dataset yang ada dan benar-benar menderita diabetes, mengurangi kesalahan prediksi yang tidak benar.
<i>Recall</i>	Kemampuan model untuk mengidentifikasi semua kasus positif	~63,61 %	Model berhasil mengidentifikasi sekitar 63,61% dari semua kasus diabetes aktual dalam dataset, mengurangi kasus yang terlewatkan

		Predicted		
		Positive	Negative	
Actual	Positive	True Positive	False Negative	Recall/Sensitivity $\frac{TP}{TP + FN}$
	Negative	False Positive	True Negative	Specificity $\frac{TN}{TN + FP}$
		Precision $\frac{TP}{TP + FP}$	Negative Predictive Value $\frac{TN}{TN + FN}$	Accuracy $\frac{TP + TN}{TP + FN + TN + FP}$

Gambar 5. Ilustrasi *confusion matrix* dan rumus dari metrik *accuracy*, *precision*, dan *recall*

Model kemudian diuji coba dengan dua data baru yang dapat dilihat pada Tabel 3. Dengan data baru tersebut, didapatkan hasil yang menunjukkan bahwa pasien dengan data tersebut memiliki kemungkinan diabetes sebesar 0.184901% dan 0.384602%. Hasil dapat dilihat pada Gambar 6.

Tabel 3. Uji data baru

Variabel	Nilai New_Data	Nilai New_Data2
Gender	2	2
Age	40	50
Hypertension	1	1
Heart_disease	1	1
Smoking_history	1	1
Bmi	30	28
HbA1c_level	4	4.3
Blood_glucose_level	120	125
Hasil	[[0.00184901]]	[[0.00384602]]

```

#testing loaded model
new_data = np.array([[2,40,1,1,1,30,4,120]])
new_data = sc.transform(new_data)

new_data2 = np.array([[2,50,1,1,1,28,4.3,125]])
new_data2 = sc.transform(new_data2)

print("prediksi kemungkinan diabetes dari data baru 1:", ann.predict(new_data))
print("prediksi kemungkinan diabetes dari data baru 2:", ann.predict(new_data2))
    
```

Gambar 6. Hasil uji coba dengan data baru di *back-end*

Dalam perancangan aplikasi DSS ini juga sempat ditemui beberapa *error* yang beragam jenisnya, dengan yang paling sering ditemui adalah sebagai berikut:

1. **TypeError:** Kesalahan ini terjadi saat suatu operasi atau fungsi dilakukan pada tipe data yang tidak mendukung operasi tersebut. Kasus ini terjadi saat mencoba mengonversi hasil dari model dengan persentase tanpa *coding* yang baik, yang menyebabkan penambahan sebuah *string* (%) dengan sebuah bilangan bulat (hasil model).
2. **ValueError:** Kesalahan ini terjadi saat sebuah fungsi menerima argumen dengan tipe data yang sesuai, namun memiliki nilai yang tidak sesuai. Pada kasus ini, terjadi *ValueError* saat secara tidak sengaja mencoba untuk mengonversi sebuah *string* yang bukan merupakan representasi dari angka menjadi bilangan bulat.
3. **AppOfflineError:** Kesalahan ini mengindikasikan bahwa aplikasi sedang *offline* atau tidak dapat terhubung ke *server*, mungkin karena adanya pemeliharaan atau masalah koneksi.
4. **SyntaxError:** Kesalahan ini menandakan bahwa kode memiliki masalah sintaksis. Ini terjadi ketika *interpreter* Python menemukan kode yang tidak sesuai dengan aturan sintaksis bahasa tersebut. Penyebab umumnya termasuk kurangnya tanda kurung atau tanda kutip, indentasi yang salah, atau variabel yang salah ejaan.

5. **DenseLayerExpected**: Kesalahan ini merupakan salah satu kesalahan yang cukup signifikan, karena model yang harusnya memiliki *hidden layer* dengan jumlah yang telah ditentukan, namun saat dicek ternyata tidak ada. Hal ini bisa terjadi karena model yang disimpan tidak dengan sempurna, atau *library* yang terkait dengan model tidak ikut diunggah, misalnya *library StandardScaler*.

3.2. Perancangan Antarmuka Grafis DSS

Antarmuka grafis *Decision Support System* (DSS) yang telah dirancang menggunakan tiga jenis input, yakni *text box*, *dropdown*, dan *radio button* yang dapat dilihat pada Gambar 7. *Value* dari *text box* dan *dropdown* bergantung pada angka dan pilihan yang dimasukkan, sedangkan *value* dari *radio button* memiliki nilai 1 untuk “Ya” dan 0 untuk “Tidak” sebagaimana pada Gambar 8. Setelah semua *value* dimasukkan, maka pengguna dapat menekan tombol “*Submit*” untuk memproses nilai. Hasil akan muncul di dalam *text box* dengan keterangan yang menunjukkan bahwa pengguna memiliki potensi diabetes atau tidak dengan persentase tertentu seperti yang dapat dilihat pada Gambar 7.

Cek Diabetes

Made by:
Achmad Farhan Aldyno
Ahlam Nauf Oda
Faiza Ulinnuha Junaidi
Haider Rabbani
Lathifah Nuraini

Gender: Pria

Usia: e.g. 28

Riwayat hipertensi: Ya Tidak

Riwayat penyakit jantung: Ya Tidak

Riwayat merokok: Tidak pernah

Body mass index (BMI): (misal 24.31)

HbA1c level: (misal 6.6)

Kadar gula darah (mg/dl): (misal 135)

Hasil: _____

Saran: _____

Gambar 7. Antarmuka grafis aplikasi

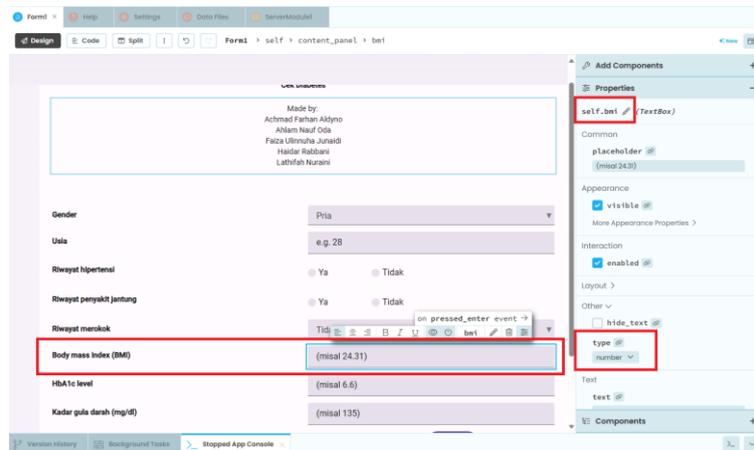
Gender: Pria

Usia: e.g. 28

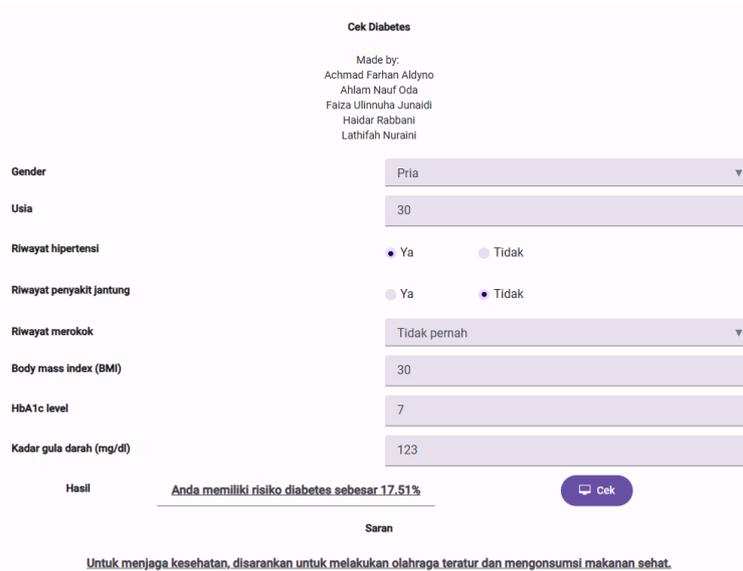
Riwayat hipertensi: Ya Tidak

Gambar 8. *Dropdown*, *text box*, dan *radio button*

Pemilihan antarmuka menggunakan jenis *dropdown*, *radio button*, dan *text box* didasarkan pada jenis *input* yang digunakan untuk mengantisipasi kesalahan dari pengguna. Konsep ini dikenal juga sebagai *data validation*. Untuk validasi *input* pada semua *text box*, digunakan fungsi yang berguna untuk membatasi tipe *input* berupa angka. Apabila memasukkan tipe lain, maka akan terjadi *error* dan diminta melakukan *input* kembali. Fungsi tersebut dapat dilihat pada Gambar 9 berikut. Adapun saat digunakan, aplikasi dapat memberitahukan risiko seseorang terkena diabetes dan saran untuk mencegah atau memeriksakan diri ke dokter sebagaimana terlihat di Gambar 9.



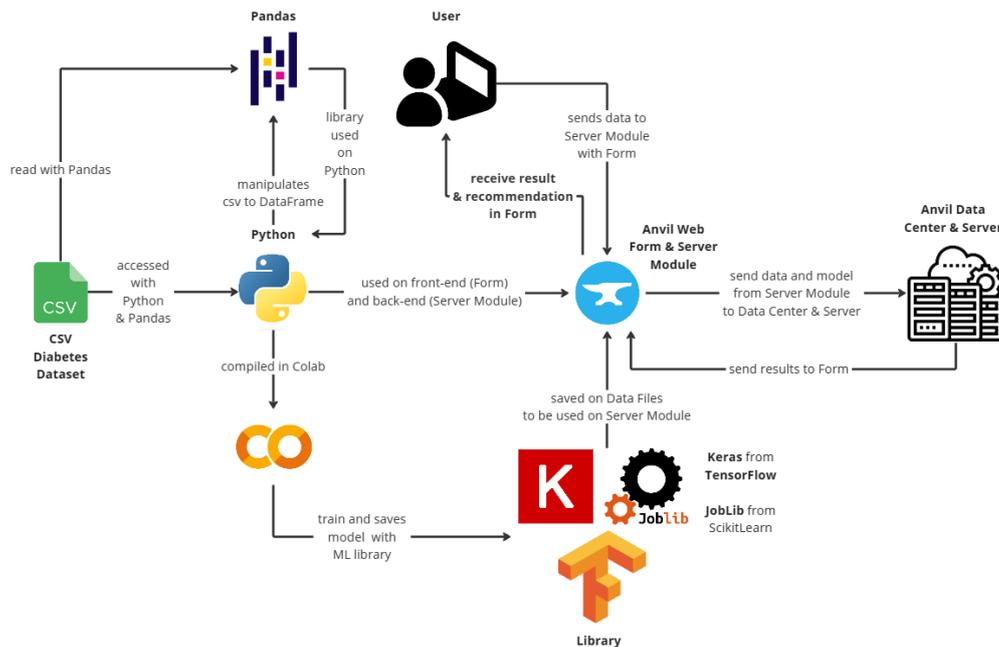
Gambar 9. Tipe input



Gambar 10. Hasil menggunakan aplikasi DSS

Adapun mekanisme Anvil yang berjalan diilustrasikan pada Gambar 11 di bawah ini. Data training berupa format .csv digunakan untuk melatih dan membuat model di Google Colab dengan ML library yang berasal dari Python. Model ML yang terbuat akan disimpan dalam Anvil di Server Module atau yang

biasa disebut juga sebagai *back-end*. Bagian *front-end* dan *back-end* diprogram dalam bahasa Python. Anvil kemudian menerima data dari pengguna yang kemudian mengirimkannya bersama model yang tersimpan tadi ke Data Center & Saver untuk diolah. Setelah mendapatkan hasilnya, Anvil kemudian menampilkannya ke pengguna melalui bagian *front-end* nya. Penjelasan lebih rinci terkait algoritma ML yang digunakan dibahas di bagian selanjutnya.



Gambar 11. Mekanisme Anvil yang berjalan

4. SIMPULAN

Pengembangan aplikasi ini menunjukkan bahwa terdapat potensi untuk mengembangkan DSS di bidang kesehatan, salah satunya pada kasus deteksi dini diabetes. Dengan adanya aplikasi ini, dokter dan masyarakat dapat mengenal lebih lanjut terkait kesehatan diri sendiri, gaya hidup sehat, serta pencegahan diabetes. Hasil pengujian untuk standar diagnosa medis mungkin masih rendah, utamanya pada metrik *recall*. Meskipun akurasi model adalah 96,15%, model hanya mampu mengidentifikasi kasus diabetes aktual dengan tingkat presisi sebesar 63,61%. Ini mengartikan bahwa meskipun model memiliki akurasi yang tinggi, namun dari semua kasus yang diprediksi sebagai diabetes, hanya sekitar 63,61% yang menderita diabetes. Sebagai suatu bentuk pembuktian konsep yang menjadi purwarupa fungsional, aplikasi ini masih dapat dikembangkan jauh lebih baik lagi, dengan beberapa di antaranya adalah:

1. Sisi tampilan / *front-end* dan juga sisi *back-end* yang dapat dikembangkan dalam bahasa lain,
2. *Error handling* dan *exceptions handling* yang lebih baik dan komprehensif,



3. Pemilihan dan performa model ML yang dapat direkayasa lebih lanjut untuk mendapatkan menghasilkan nilai metrik *precision* dan *recall* yang lebih tinggi,
4. Penambahan saran kesehatan yang lebih komprehensif dan fitur seperti tombol konsultasi dengan dokter atau pencarian klinik / rumah sakit terdekat,
5. Penyimpanan riwayat entri data untuk mencatat dan melihat risiko diabetes secara historis, apakah ada tren risiko yang konstan, menurun, atau meningkat,
6. Eksplorasi dataset yang lebih baik dan mendetail, termasuk penambahan dan pengurangan variabel-variabel baru, semisal genetika keluarga, tipe diabetes yang dialami, dan lain sebagainya, serta
7. Penggabungan DSS dengan alat fisik lain seperti timbangan BMI dan alat cek darah.

DAFTAR PUSTAKA

- [1] International Diabetes Federation Atlas, Brussels, Belgium, 2021, [Online]. Available: <https://idf.org/about-diabetes/diabetes-facts-figures/>.
- [2] A. Tuppada and S.D. Patil, "Machine learning for diabetes clinical decision support: a review," *Adv Comput Intell*, vol. 2, no. 2, p. 22, Apr. 2022, doi: 10.1007/s43674-022-00034-y. Epub 2022 Apr 13. PMID: 35434723; PMCID: PMC9006199.
- [3] P. Rahimloo and A. Jafarian, "Prediction of Diabetes by Using Artificial Neural Network, Logistic Regression Statistical Model and Combination of Them," *Bull. Société R. Sci. Liège*, vol. 85, pp. 1148-1164, 2016.
- [4] A. Mohebbi, T.B. Aradóttir, A.R. Johansen, H. Bengtsson, M. Fraccaro, and M. Mørup, "A deep learning approach to adherence detection for type 2 diabetics," in *Proceedings of the 2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Jeju, Korea, 11-15 July 2017, pp. 2896-2899.
- [5] N. Yuvaraj and K.R. SriPreethaa, "Diabetes prediction in healthcare systems using machine learning algorithms on Hadoop cluster," *Clust. Comput.*, vol. 22, pp. 1-9, 2017.
- [6] D. Sisodia and D.S. Sisodia, "Prediction of Diabetes using Classification Algorithms," *Procedia Comput. Sci.*, vol. 132, pp. 1578-1585, 2018.
- [7] J.P. Kandhasamy and S. Balamurali, "Performance Analysis of Classifier Models to Predict Diabetes Mellitus," *Procedia Comput. Sci.*, vol. 47, pp. 45-51, 2015.
- [8] Z. Tafa, N. Pervetica, and B. Karahoda, "An intelligent system for diabetes prediction," in *Proceedings of the 2015 4th Mediterranean Conference on Embedded Computing (MECO)*, Budva, Montenegro, 14-18 June 2015, pp. 378-382.
- [9] A. Negi and V. Jaiswal, "A first attempt to develop a diabetes prediction method based on different global datasets," in *Proceedings of the 2016 Fourth International Conference on Parallel, Distributed and Grid Computing (PDGC)*, Wagnaghat, India, 22-24 December 2016, pp. 237-241.

- [10] E.O. Olaniyi and K. Adnan, "Onset diabetes diagnosis using artificial neural network," *Int. J. Sci. Eng. Res.*, vol. 5, pp. 754–759, 2014.
- [11] Z. Soltani and A. Jafarian, "A New Artificial Neural Networks Approach for Diagnosing Diabetes Disease Type II," *Int. J. Adv. Comput. Sci. Appl.*, vol. 7, pp. 89–94, 2016.
- [12] M. Mamuda and S. Sathasivam, "Predicting the survival of diabetes using neural network," in *Proceedings of the AIP Conference Proceedings*, Bydgoszcz, Poland, 9–11 May 2017, vol. 1870, pp. 40–46.
- [13] F.K. Adli, "Diabetes Melitus Gestasional: Diagnosis Dan Faktor Risiko," *Jurnal Medika Utama*, vol. 3, no. 01 Oktober, pp. 1545-1551, 2021, [Online]. Available: <http://jurnalmedikahutama.com/index.php/JMH/article/view/312>.
- [14] S. Trisnawati, I.K.T. Widarsa, and K. Suastika, "Risk factors of type 2 diabetes mellitus of outpatients in the community health centres of South Denpasar Subdistrict," *Public Health and Preventive Medicine Archive*, vol. 1, no. 1, pp. 69–73, 2013, doi: 10.53638/phpma.2013.v1.i1.p14.
- [15] A. Sanda, F. Mangarengi, and R.D. Pakasi, "Kadar Asam Urat Berkorelasi dengan Kadar Hemoglobin Terглиkolisasi (HbA1c) Pasien Diabetes Melitus tipe 2," *Cermin Dunia Kedokteran*, vol. 45, no. 6, pp. 405–407, 2018, doi: 10.55175/cdk.v45i6.763.
- [16] M. Mustafa, "Diabetes prediction dataset, Version 1," Retrieved Nov. 24, 2023, from <https://www.kaggle.com/datasets/iammustafatz/diabetes-prediction-dataset>, Apr. 2023.