BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

4.1 Hasil Penelitian

Dari metodologi yang telah dirancang untuk membandingkan tingkat akurasi yang paling tinggi dalam memprediksi penyakit diabetes dengan menggunakan algoritma *neural network*, *support vector machine* dan *random forest*, maka hasil yang didapatkan adalah sebagai berikut:

4.1.1 *Exploratory Data Analysis*

Data yang diperoleh dari Kaggle berjumlah 100.000 individu yang terkena penyakit diabetes dan tidak terkena penyakit diabetes yang disimpan dalam format *csv*.

	gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes
0	Female	80.0	0	1	never	25.19	6.6	140	0
1	Female	54.0	0	0	No Info	27.32	6.6	80	0
2	Male	28.0	0	0	never	27.32	5.7	158	0
3	Female	36.0	0	0	current	23.45	5.0	155	0
4	Male	76.0	1	1	current	20.14	4.8	155	0
5	Female	20.0	0	0	never	27.32	6.6	85	0
6	Female	44.0	0	0	never	19.31	6.5	200	1

Gambar 4.1 1 Contoh sampel data

Tabel 4. 1 Tabel meta data

Attribute	*Description*	*Value*	Tipe data
Gender	Jenis kelamin	*Male*, *female*, *other*	Kategorikal
Hypertension	Apakah pasien memiliki riwayat darah tinggi (0 = tidak, 1 = iya)	1 dan 0	Numerikal
Heart disease	Apakah pasien memiliki riwayat serangan jantung (0 = tidak, 1 = iya)	1 dan 0	Numerikal

*Smoking histor*y	Riwayat merokok (*current*: sedang merokok, *ever*: pernah merokok, *former*: pernah merokok dan tidak merokok lagi, *never*: tidak pernah merokok, *no info*: tidak ada informasi, not *current*: saat ini sedang tidak merokok)	*Current, ever, former, never, no info, not current*	Kategorikal
BMI	Indeks masa tubuh (kg)	10,01 – 95,69	Numerikal
*HbA1c l*evel	Persentase *hemoglobin* yang terikat dengan glukosa	3,5 - 9	Numerikal
*Blood gluc*ose	Gula dalam darah yang dicek pada saat itu	80 - 300	Numerikal
Diabetes	Apakah seseorang menderita penyakit diabetes (0=tidak, 1=iya)	0 dan 1	Numerikal

a. *Missing Val*ue

Melihat apakah terdapat *missing value* pada masing-masing variabel. Pengecekan *missing value* pada setiap variabel sangat diperlukan karena akan mempengaruhi hasil dari *training*, sehingga jika ada *missing value* pada variabel harus segera ditangani dengan menghapusnya atau mengisi dengan nilai tertentu. Hasil pengecekan pada setiap variabel adalah:

	0
gender	0
age	0
hypertension	0
heart_disease	0
smoking_history	0
bmi	0
HbA1c_level	0
blood_glucose_level	0
diabetes	0

dtype: int64

Gambar 4.1 2 Hasil *missing value*

Setelah mengecek *missing value*, ternyata *missing value* yang terdeteksi berjumlah 0 atau tidak ada sama sekali, sehingga dapat dilakukan proses selanjutnya.

b. Tipe data

Melihat dan cek tipe data dari masing-masing variabel karena ini akan mempengaruhi kualitas data dan mengetahui proses selanjutnya.

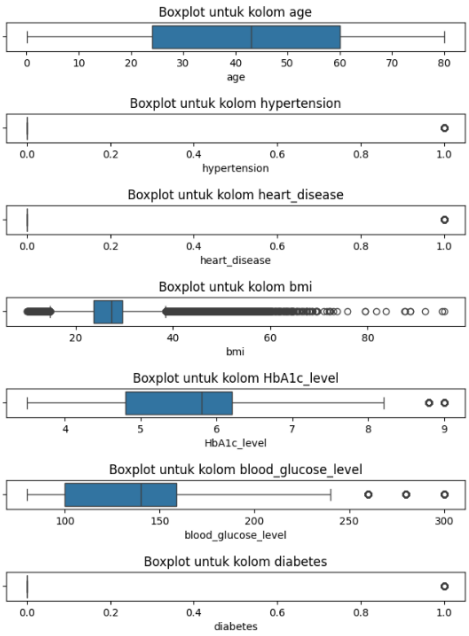
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100000 entries, 0 to 99999
Data columns (total 9 columns):
 #   Column               Non-Null Count   Dtype  
---  ------               --------------   -----  
 0   gender               100000 non-null  object 
 1   age                  100000 non-null  float64
 2   hypertension         100000 non-null  int64  
 3   heart_disease        100000 non-null  int64  
 4   smoking_history      100000 non-null  object 
 5   bmi                  100000 non-null  float64
 6   HbA1c_level          100000 non-null  float64
 7   blood_glucose_level  100000 non-null  int64  
 8   diabetes             100000 non-null  int64  
dtypes: float64(3), int64(4), object(2)
memory usage: 6.9+ MB
```

Gambar 4.1 3 Hasil tipe data

Setelah dilakukan cek tipe data, ternyata mayoritas tipe data bertipe angka dan hanya variabel *gender* dan *smoking history* yang tidak bertipe angka.

c. Outlier

Mengecek kemungkinan adanya *outlier* dengan memvisualisasikannya menggunakan *boxplot*, hasil visualisasinya adalah sebagai berikut:

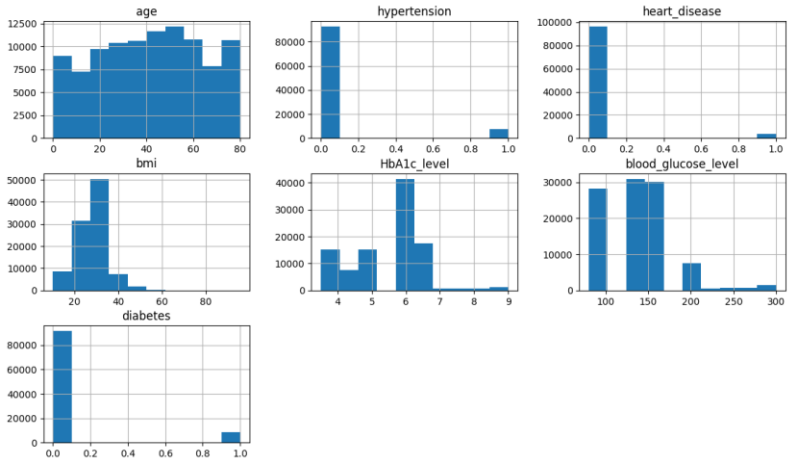


Gambar 4.1 4 Hasil *outlier*

Dari visualisasi tersebut, *outlier* yang terdeteksi tidak dihapus dikarenakan masih *outlier* tersebut tidak termasuk dalam kesalahan.Contohnya adalah *outlier* pada kolom *hypertension*, angka 1 dianggap sebagai outlier karena mayoritas adalah angka 0, sehingga angka 1 dianggap sebagai angka yang asing. Tetapi karena kolom *hypertension* hanya berisi 0 dan 1, maka angka 1 tidak dianggap sebagai *outlier*. Selain itu juga, penghapusan *outlier* dikhawatirkan akan menyebabkan penurunan performa dalam *training*.

d. Distribusi data

Memvisualisasikan distribusi data dengan memanfaatkan *histogram* agar lebih mudah dilihat dan dimengerti.

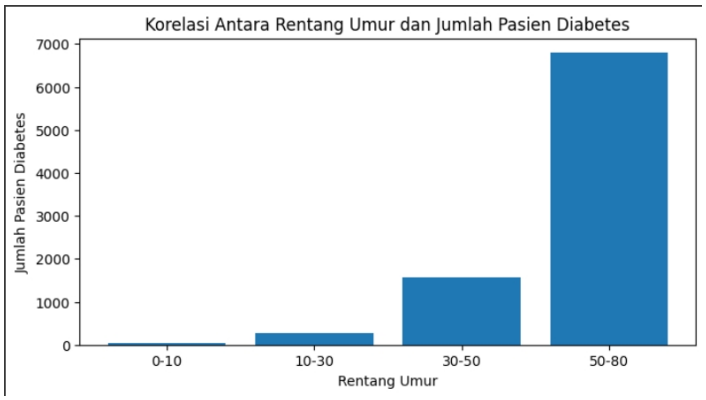


Gambar 4.1 5 Penyebaran/distribusi data

Pada gambar visualisasi distribusi data di atas dapat dilihat bahwa penyebaran atau pendistribusian data tidak seimbang dan lebih condong ke kiri atau biasa disebut juga dengan *right-skewed*. Hal ini menandakan bahwa *mean* atau nilai rata-rata lebih besar daripada *median*. Karena variabel target lebih banyak yang bernilai 0 sehingga menyebabkan data yang tidak seimbang, maka akan dilakukan *undersampling* data, sehingga menghasilkan keseimbangan data target yang bernilai 1 dan 0.

e. Visualisasi korelasi antara masing-masing variabel dan target

1) Korelasi umur dan penderita diabetes

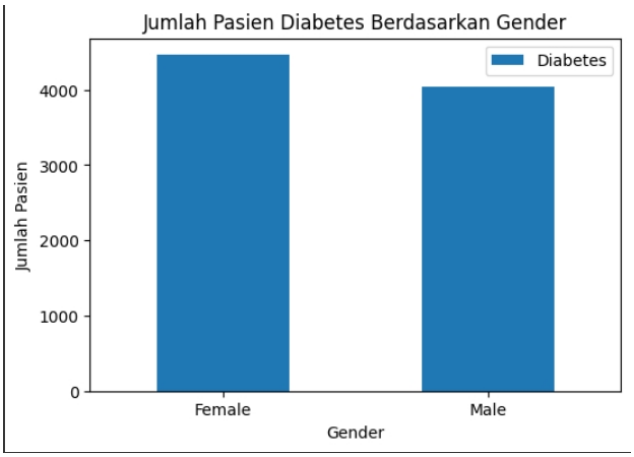


Gambar 4.1 6 Korelasi antara umur dan diabetes

Dari visualisasi di atas, diketahui bahwa penderita diabetes terbanyak adalah yang berumur 50-80 tahun dengan jumlah kasus mencapai 6791 dari 8500. Tetapi hal itu tidak menjadikannya parameter bahwa penyakit diabetes hanya terjadi ketika

berusia di atas 50 tahun. Berdasarkan visualisasi di atas, dapat dikeathui bahwa faktor usia seseorang juga menentukan faktor terkena diabetes

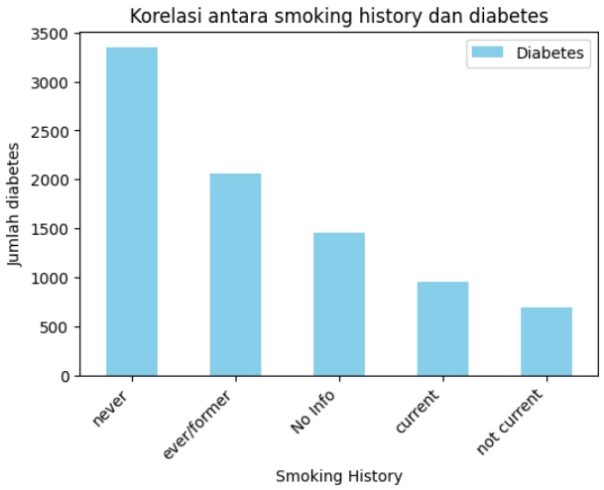
2) Korelasi antara gender dan penderita diabetes



Gambar 4.1 7 Korelasi antara *gender* dan diabetes

Dari grafik ini, individu berjenis kelamin perempuan lebih banyak yang berisiko terkena diabetes dibandingkan dengan laki-laki dengan jumlah masing-masing yaitu 4.461 dan 4.039. Namun, selisih antara penderita diabetes jenis kelamin laki-laki dan perempuan tidaklah banya, yaitu hanya 422.

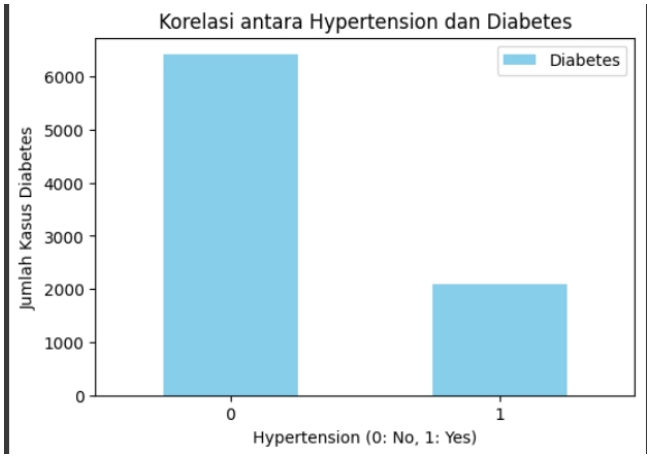
3) Korelasi antara *smoking history* dan penderita diabetes



Gambar 4.1 8 Korelasi antara *smoking history* dan diabetes

Dari variabel *smoking history*, dapat dilihat bahwa *never* atau orang yang tidak pernah merokok justru lebih banyak yang terkena diabetes meskipun disusul dengan yang sudah pernah merokok dan tidak merokok lagi. Dan untuk yang sedang tidak merokok justru lebih sedikit memiliki penyakit diabetes.

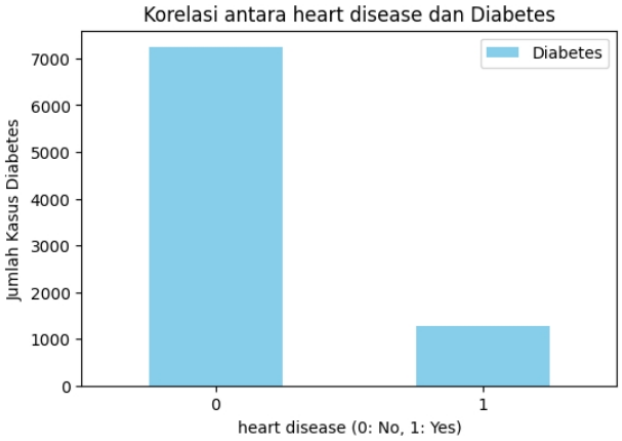
4) Korelasi antara *hypertension* dan diabetes



Gambar 4.1 9 Korelasi antara *hypertension* dan diabetes

Dari visualisasi di atas, terlihat bahwa 6412 orang yang tidak memiliki penyakit darah tinggi justru tidak terkena diabetes dan 2088 orang yang memiliki penyakit darah tinggi juga memiliki penyakit diabetes. Hal ini menunjukkan bahwa darah tinggi tidak selalu berbanding lurus dengan diabetes.

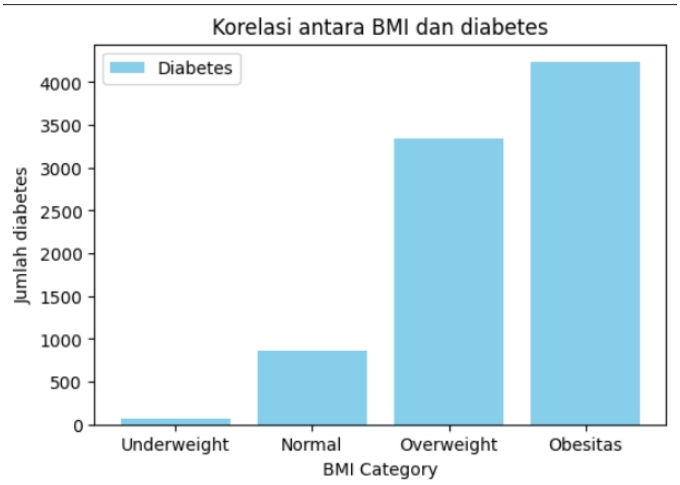
5) Korelasi antara *heart disease* dan diabetes



Gambar 4.1 10 Korelasi antara *heart disease* dan diabetes

Dapat dilihat bahwa individu yang memiliki penyakit diabetes dan tidak memiliki masalah dengan jantung berjumlah 7233 kasus, sementara individu yang memiliki serangan jantung dan juga memiliki penyakit diabetes berjumlah 1267 kasus. Dari data ini juga dapat disimpulkan bahwa penyakit diabetes juga tidak selalu berbanding lurus dengan penyakit diabetes, karena banyak yang terkena diabetes tetapi tidak ada serangan jantung dan memiliki riwayat penyakit jantung tetapi tidak terkena diabetes.

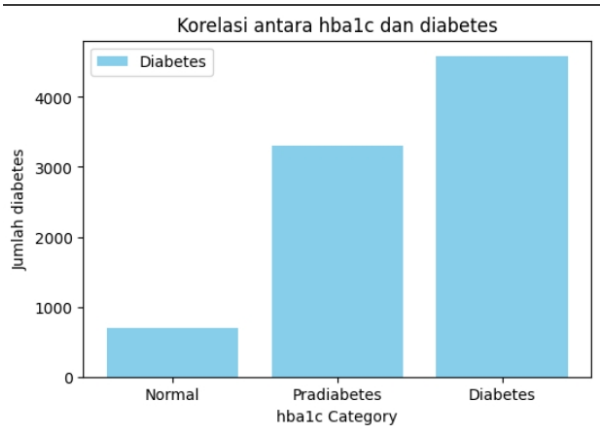
6) Korelasi antara *body massa index* dan diabetes



Gambar 4.1 11 Korelasi antara *BMI* dan diabetes

Individu yang paling sedikit untuk terkena diabetes dilihat dari *body massa index* adalah yang tergolong *underweight* atau individu yang berat badannya di bawah normal dengan jumlah 64 orang, sementara yang tergolong normal terdapat 867 orang. Kemudian kasus yang paling banyak menderita penyakit diabetes adalah individu yang tergolong *overweight* dan diabetes dengan masing-masing adalah 3341 individu dan 4232 individu. Dapat disimpulkan bahwa berat badan yang berlebih semakin meningkatkan peluang untuk terkena diabetes.

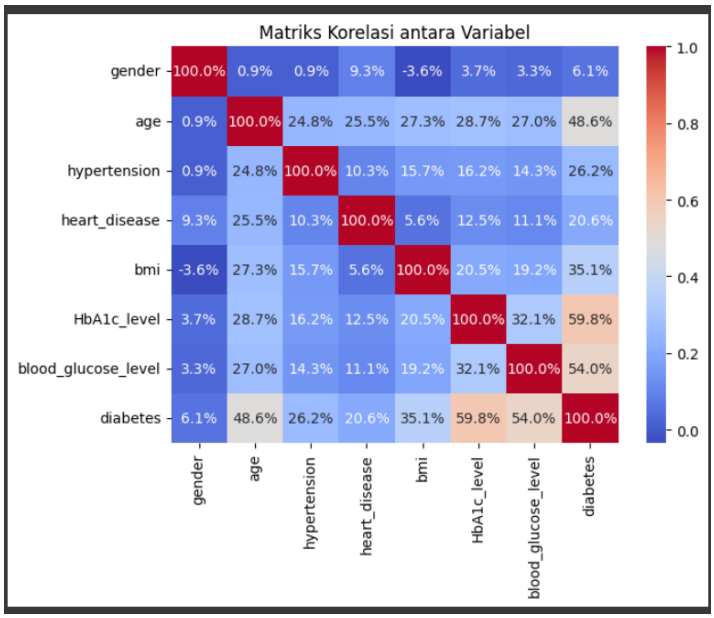
7) Korelasi antara *HbA1c* dan diabetes



Gambar 4.1 12 Korelasi antara *HbA1c* dan diabetes

Individu yang *HbA1c* levenya tergolong normal cenderung untuk terhindar dari diabetes. Dari visualisasi di atas, inidividu yang memiliki *HbA1c* normal dan terkena diabetes hanya berjumlah 704 orang. Kemudian semakin tinggi kadar *HbA1c* nya, maka kemungkinan untuk terkena diabetes juga semakin tinggi, karena individu yang memiliki *HbA1c* nya di atas 6,5 dan terkena diabetes berjumlah 4578 orang.

8) Metriks korelasi



Gambar 4.1 13 Metrik korelasi antar variabel

Pada penelitian ini, grafik yang digunakan untuk melihat nilai korelasi antar variabel menggunkan diagram *heatmap*. Warna merah pada diagram ini menunujukkan bahwa variabel memiliki korelasi yang positif atau kuat dengan variabel lainnya, sementara warna biru menunjukan nilai satu variabel dengan variabel lain cenderung turun.

Korelasi variabel *HbA1c_level* dengan variabel diabetes sangat tinggi. Dapat dilihat bahwa korelasi variabel *HbA1c_level* dengan variabel diabetes mencapai 59,8%. Hal ini dapat dideskripsikan bahwa variabel *HbA1c_level* memainkan peran penting untuk mempengaruhi individu diabetes. Variabel *HbA1c_level* juga memiliki korelasi yang cukup tinggi dengan variabel *blood_glucose_level*, yaitu 32,1%. Selain itu, korelasi dengan variabel *bmi* dan *age* juga cukup tinggi, yaitu berkisar di angka 20%. Sementara korelasi terendah yaitu dengan *hypertension*, *heart_disease* dan *gender* dengan masing-masing 16,2%, 12,5% dan 3,7%.

Kemdian variabel dengan korelasi tertinggi dengan variabel diabetes, yaitu *blood_glucose_level* dengan 54%. Hal ini membuktikan bahwa penyakit diabetes berkorelasi dengan kadar gula darah. Korelasi variabel tertinggi selain diabetes yaitu dengan variabel *HbA1c_level* dengan masing-masing yaitu 32,1% dan 27%. Dapat dideskripsikan bahwa umur memiliki pengaruh dengan kadar gula darah seseorang. Sementara korelasi dengan variabel lainnya berada di kisaran 11%-19% yang berkorelasi dengan variabel *bmi* dengan 19,2%, *hypertension* dengan 14,3% dan *heart_disease* dengan 11,1%. Sementara korelasi dengan umur adalah yang paling rendah dengan nilai 3,3%.

Variabel *age* juga memiliki tingkat korelasi yang paling tinggi dengan diabetes, yaitu mencapai 48,6%. Hal ini sesuai dengan penjelasan sebelumnya yang menjelaskan bahwa semakin tinggi umur seseorang, maka akan semakin tinggi resiko untuk terkena diabetes. Variabel *age* cenderung memiliki korelasi yang seimbang dengan variabel lainnya, yaitu dengan *HbA1c_level* 28,7%, dengan *bmi* 27,3%, dengan *blood_glucose_level* 27%, *heart_disease* 25,5% dan 24,8%. Sementara dengan variabel *gender* hanya berada di angka 0.9%.

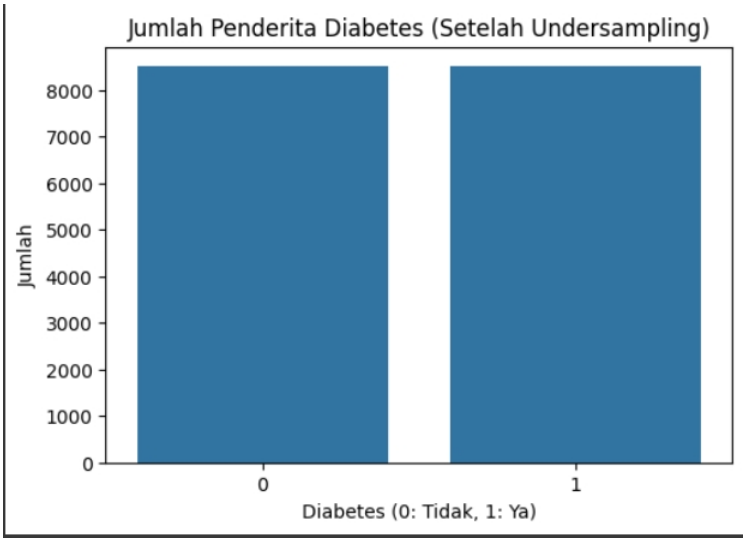
Kemudian variabel *bmi* mempunyai korelasi yang cukup tinggi juga dengan variabel diabetes, yaitu 35,1%. Sementara korelasi dengan variabel *heart_disease* tergolong cukup rendah, yaitu berkisar di angka 5,6% dan yang terendah adalah korelasi dengan variabel *gender*. Sementara variabel yang paling kecil korelasi dengan variabel diabetes yaitu, gender yang hanya berkisar 6,1%.

Dari deskripsi di atas, dapat disimpulkan bahwa variabel diabetes yang juga variabel dependen sangat berkorelasi dengan variabel independen dengan korelasi paling atas yaitu variabel *HbA1c_level* dan *blood_glucose_level* dengan angka 59,8% dan 54%, kemudian variabel *age*, *bmi hypertension* dan *heart_disease* dengan persentase 48,6%, 35,1%, 26,2% dan 20,6%, dan yang terakhir adalah *gender* dengan 6,1%. Hal ini menandakan bahwa variabel yang paling berhubungan dengan variabel diabetes adalah *HbA1c_level*, sementara yang paling tidak berhubungan dengan variabel diabetes adalah *gender*.

4.1.2 *Preprocessing data*

a. Undersampling data

Dalam penelitian ini, digunakan teknik *undersampling* untuk menyeimbangkan jumlah data di kolom target. Alasan digunakan *undersampling* ini adalah karena perbedaan kelas yang sangat jauh, yaitu 91,5% untuk kelas 1 dan 8,5% untuk kelas 0. *Undersampling* juga dilakukan untuk mengurangi terjadinya bias dalam model *machine learning*. *Undersampling* dilakukan dengan cara mengurangi data dari kelas 0 untuk menyamakan dengan kelas 1, sehingga hasil akhirnya adalah 1:1. Hasil akhirnya adalah tersisa 17.000 data yang akan diolah nantinya dengan pembagian 8.500 adalah data dengan target bernilai 1 dan 8500 data yang bernilai 0 dari fitur diabetes. Dengan melakukan *undersampling*, maka diharapkan dapat meningkatkan kemampuan model dalam mengenali pola yang bernilai 1 maupun 0 sehingga dapat memberikan hasil prediksi yang lebih akurat dan seimbang.



Gambar 4.1 14 Hasil *undersampling* fitur diabetes

b. Data *encoding*

Sebelum data dibagi, data dilakukan *encoding* atau atau mentransformasi data menjadi angka pada variabel smoking_history, dengan cara melakukan proses *dummy data*. sehingga data dapat dilakukan proses *training*.

	0	1	2	3	4
gender	0	0	1	0	1
age	80.0	54.0	28.0	36.0	76.0
hypertension	0	0	0	0	1
heart_disease	1	0	0	0	1
bmi	25.19	27.32	27.32	23.45	20.14
HbA1c_level	6.6	6.6	5.7	5.0	4.8
blood_glucose_level	140	80	158	155	155
diabetes	0	0	0	0	0
smoking_history_No Info	False	True	False	False	False
smoking_history_current	False	False	False	True	True
smoking_history_ever	False	False	False	False	False
smoking_history_former	False	False	False	False	False
smoking_history_never	True	False	True	False	False
smoking_history_not current	False	False	False	False	False

Gambar 4.1 15 Hasil *one-hot encoding* fitur smoking history

c. Normalisasi data

Selanjutnya adalah melakukan normalisasi data dengan membuat skala dari 0 hingga 1, sehingga data berada dalam posisi yang sama dan terhindar dari *outlier*

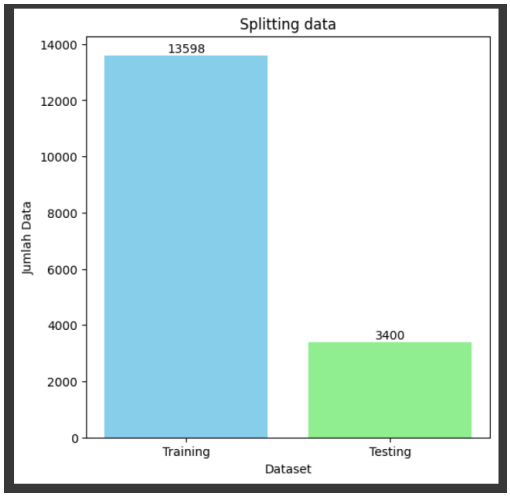
dan pelatihan dapat berjalan dengan lebih baik. Variabel yang dinormlisasi yaitu, *age*, *bmi*, *HbA1c_level* dan *blood_glucose_level*.

	0	1	2	3	4
gender	0	0	1	0	1
age	1.0	0.674675	0.349349	0.449449	0.94995
hypertension	0	0	0	0	1
heart_disease	1	0	0	0	1
bmi	0.177171	0.202031	0.202031	0.156863	0.118231
HbA1c_level	0.563636	0.563636	0.4	0.272727	0.236364
blood_glucose_level	0.272727	0.0	0.354545	0.340909	0.340909
diabetes	0	0	0	0	0
smoking_history_No Info	False	True	False	False	False
smoking_history_current	False	False	False	True	True
smoking_history_ever	False	False	False	False	False
smoking_history_former	False	False	False	False	False
smoking_history_never	True	False	True	False	False
smoking_history_not current	False	False	False	False	False

Gambar 4.1 16 *Hasil* normalisasi

d. Splitting data

Dataset akan dibagi untuk digunakan dalam *training* dan *testing* dengan rasio 80:20, itu berarti 13598 data akan dilatih dan 3400 data akan menjadi *testing*. *Training* data adalah data yang digunakan untuk model belajar sehingga dapat mengenali pola atau hubungan antar fitur dan hasil. *Training data* berfungsi untuk membangun dan mengoptimalkan model. Sementara itu, *testing* merupakan data yang dipakai untuk menilai kinerja model setelah melalui proses pelatihan dengan menggunakan data *training*. *Data testing* juga merupakan data yang tidak pernah dikenali oleh model sebelumnya yang bertujuan untuk mengukur seberapa optimal model dapat menggeneralisasi data baru yang tidak ada dalam proses pelatihan.



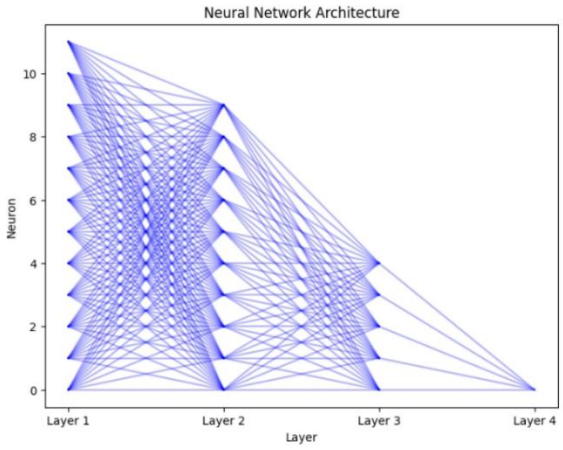
Gambar 4.1 17 Visualisasi *splitting data*

4.1.3 *Validation* dan *training model*

Model algoritma yang digunakan untuk melatih data yaitu, *support vector machine*, *random forest* dan *neural network* yang juga dengan memanfaatkan 10 *fold cross validation*.

a. Neural network

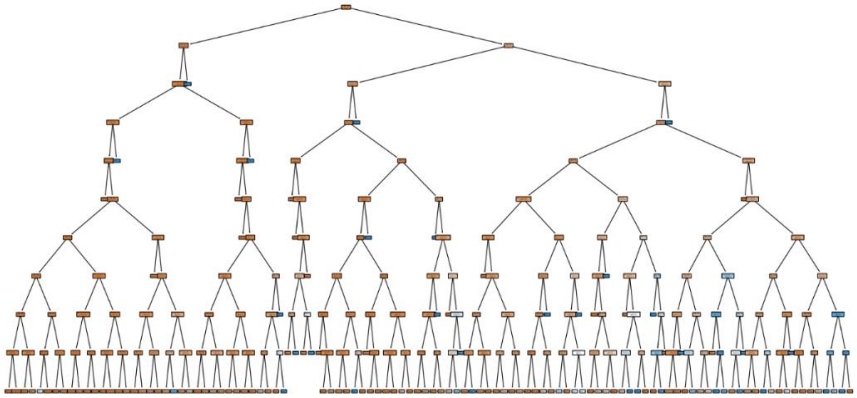
Neural network yang dibangun dari data latih menggunakan metode *neural network* dengan *multilayer perceptron. Neural* ini memiliki 3 lapisan, yaitu lapisan *input (*12 *neuron),* 2 lapisan tersembunyi *(*10 *neuron* pada lapisan kedua dan 5 *neuron* pada lapisan ketiga*),* serta lapisan *output* dengan 1 *neuron* yang menghasilkan nilai *biner*.



Gambar 4.1 18 Arsitektur *neural network*

b. *Random forest*

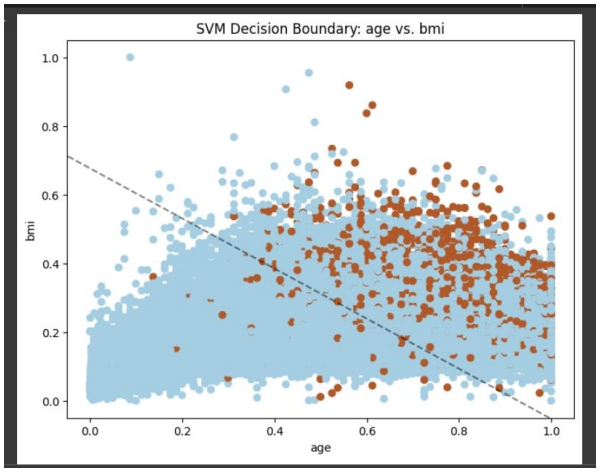
Pada penelitian kali ini, model *random forest* dikonfigurasi dengan 100 pohon keputusan (*n_estimators*=100). Setiap pohon dibatasi hingga kedalaman maksimum 10 *level* (*max_depth*=10) untuk mencegah pembentukan pohon yang terlalu kompleks. Pembatasan ini bertujuan untuk mengurangi risiko *overfitting* dan meningkatkan kemampuan model dalam menggeneralisasi data baru. Selain itu, parameter *random_state*=42 diterapkan untuk memastikan konsistensi hasil setiap kali model dilatih, sehingga evaluasi dan perbandingan model dapat dilakukan secara lebih akurat.



Gambar 4.1 19 Salah satu contoh *tree*

c. Support Vector Machine

Pada penelitian kali ini, model *support vector machine* menggunakan *kernel linear*. *Kernel linear* berarti model mencoba memisahkan data dengan menggunakan garis lurus sebagai batas antar kelas. Pendekatan ini dipilih karena sederhana dan cocok untuk data yang mudah dipisahkan. Dengan cara ini, model dapat menemukan garis terbaik yang memisahkan kelompok data sehingga mampu membuat prediksi yang lebih akurat. Metode ini juga lebih cepat dibandingkan metode lain, sehingga efisien untuk digunakan dalam penelitian ini.

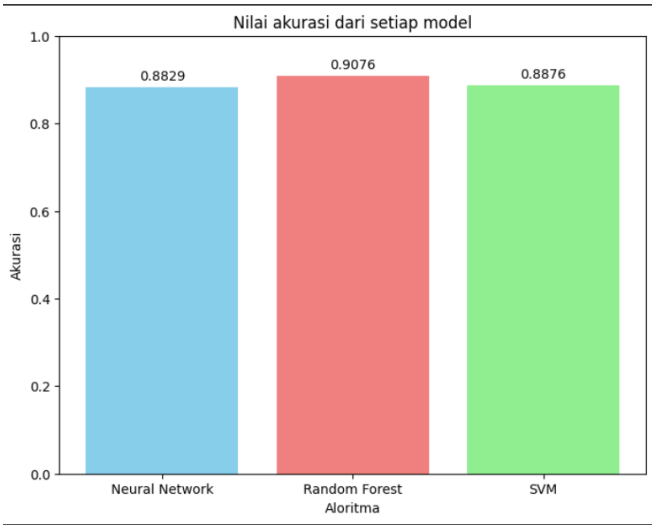


Gambar 4.1 20 Contoh *support vector machine*

4.1.4 Evaluasi

a. Akurasi

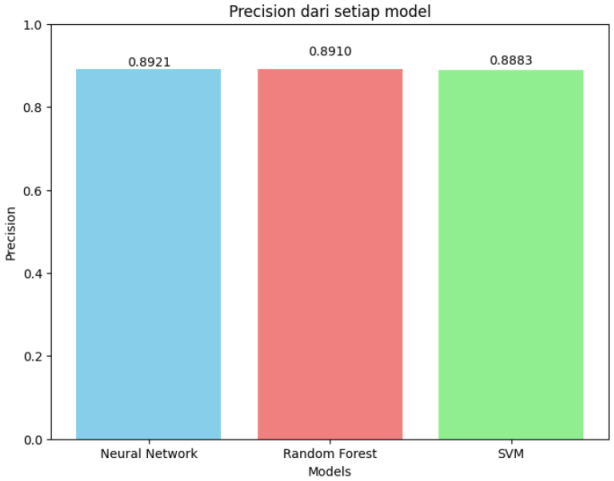
Nilai akurasi yang didapatkan dari masing-masing algoritma cenderung besar karena berada di kisaran 90%. Algoritma *neural network* memiliki nilai akurasi yang paling kecil di antara 3 algoritma lainnya, yaitu sebesar 0,8829 atau 88%, sementara algoritma *support vector machine* 0.8876 atau 89%. Dan yang memiliki nilai tertinggi adalah *random forest* dengan nilai akurasi mencapai 0,9076 atau 91%.



Gambar 4.1 21 Hasil akurasi dari setiap algoritma

b. *Precision*

Hasil dari *precision* dari masing-masing algoritma juga dapat dikatakan bagus dikarenakan nilai yang paling rendah adalah 0,8883 atau 89% yang dimiliki oleh algoritma *support vector machine*. Kemudian untuk algoritma *random forest* yang mempunyai nilai *precision* 0,8910 atau 89% dan *neural network* adalah dengan nilai yang tertinggi yaitu 0.8921 atau 89%. Hasil precision dari ketiga algoritma tersebut hanya dibedakan oleh angka akhirnya dan ketika dibulatkan akan menjadi sama. Hal ini kemudian dapat disimpulkan bahwa setiap model dapat mengklasifikasikan dan memprediksi variabel target *true positive* dengan sangat baik. Kemudian yang paling baik dalam memprediksi variabel target yang positif yaitu algoritma *neural network*.

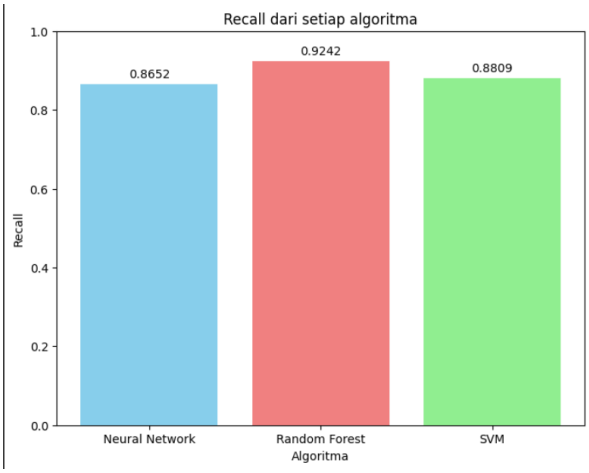


Gambar 4.1 22 Hasil *precision* dari setiap algoritma

c. *Recall*

Untuk *recall* sendiri juga mendapatkan hasil yang cukup baik. Algoritma yang memiliki nilai *recall* terbaik adalah *random forest* dengan nilai recall 0.9242 atau 92%. Ini berarti algoritma *random forest* cukup baik dalam mengidentifikasi kasus positif. Sementara nilai recall dari algoritma *support vector machine*, yaitu sebesar 0.8809 atau 88%, hal ini menunjukkan bahwa model *support vector machine*. Sementara nilai yang paling rendah didapatkan oleh algoritma *neural network* dengan nilai 0.8652 atau 87%, hal ini menunjukkan bahwa algoritma *neural*

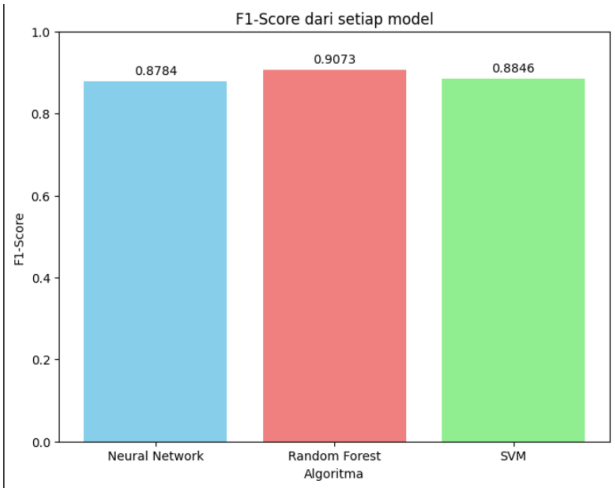
network adalah model yang paling banyak melewatkan kasus positif meskipun hanya berbeda sedikit dengan model *support vector machine*.



Gambar 4.1 23 Hasil *recall* dari setiap algoritma

d. *F1-score*

Nilai *f1-score* digunakan untuk mengukur keseimbangan antara *precision* dan *recall*, khususnya pada data dengan distribusi kelas yang tidak merata. Dalam penelitian ini, algoritma *random forest* memiliki *F1-score* tertinggi, yaitu sebesar 0,9073 atau 91%. Kemudian diikuti oleh *support vector machine* sebesar 0,8846 atau 88%, dan *support vector machine* sebesar 0,8784 atau 88%. Hasil ini menunjukkan bahwa *random forest* adalah model yang paling baik dalam mengklasifikasikan data secara keseluruhan. Algoritma *random forest* dapat membuat prediksi yang lebih akurat dibandingkan dengan *neural network* dan *support vector machine*. Meskipun *neural network* dan *support vector machine* memiliki performa yang sedikit lebih rendah dibandingkan *random forest*, namun keduanya tetap menunjukkan hasil yang cukup baik untuk tugas klasifikasi ini.



Gambar 4.1 24 Hasil *f1-score* dari setiap algoritma

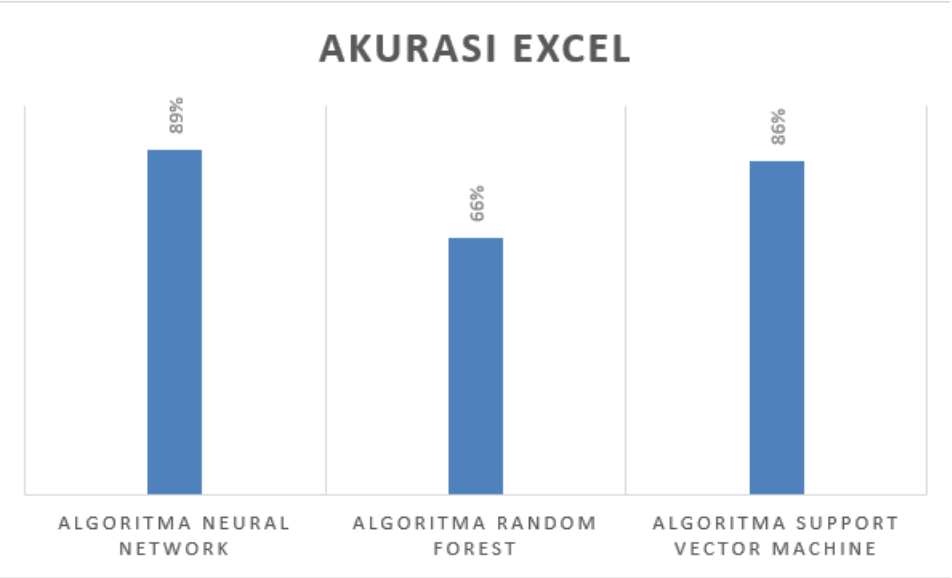
4.2 Pembahasan

Memprediksi penyakit diabetes dapat dilakukan dengan algoritma *neural network*, *support vector machine* dan *random forest* yang mendapatkan nilai akurasi yang cukup tinggi. Sementara nilai akurasi yang terbaik adalah nilai yang dihasilkan dari algoritma *random forest* dengan akurasi mencapai 91%. Hal ini menunjukan algoritma *random forest* mampu memprediksi dengan tingkat keakuratan yang tinggi dan optimal. Meski hasil tingkat akurasi tertinggi dimiliki oleh algoritma *random forest*, namun bukan berarti algoritma lain memiliki nilai akurasi yang sangat rendah, karena hanya berbeda 2%-3%. Kemudian untuk hasil *precision* tertinggi adalah algoritma *neural network* dengan nilai 89%, sementara nilai tertinggi *recall* adalah algoritma *random forest*, yaitu 92%. Sementara untuk hasil *f1-score* tertinggi adalah algoritma *random forest* dengan nilai 91%.

y_asli	y_pred_rf	y_pred_nn	y_pred_svm
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	1
0	1	1	1
0	0	0	0
0	0	0	0

Gambar 4.1 25 Hasil *y_pred* dan *y_act*

Penelitian ini juga menggunakan *tools excel* untuk menghitung secara manual dari masing-masing algoritma dan didapatkan nilai akurasi dengan hasil sebagai berikut:



Gambar 4.2 1 Nilai akurasi dengan *excel*

Untuk file *excel* dengan perhitungan lebih jelas, dapat dilihat melalui link berikut:

a. Algoritma *neural network* (*neural network*)
b. Algoritma *random forest* (*random forest*)
c. Algoritma *support vector machine* (*support vector machine*)

Untuk file dengan format *ipynb*, dapat dilihat melalui link berikut ini (*file .ipynb*)