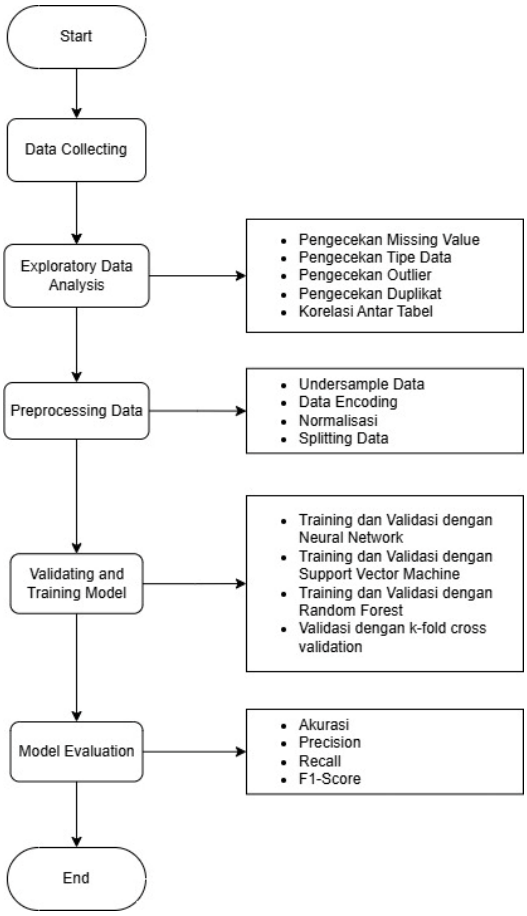
BAB III

METODOLOGI PENELITIAN

3.1 Alur Penelitian

Tahapan yang dilakukan untuk membuat perbandingan tingkat akurasi dari algoritma dengan *machine learning* adalah sebagai berikut:



Gambar 3. 1 Alur penelitian

3.2 Data Collection

Penelitian ini memanfaatkan *dataset* yang bersumber dari situs Kaggle dengan 100.000 *record* data pasien penderita diabetes dengan variabel berisi *gender*, *age*, *hypertension*, *heart disease*, *smoking history*, *bmi*, *HbA1c_level*, *blood glucose level*, dan diabetes sebagai label.

gender	age	hypertensi	heart_dise	smoking_h	bmi	HbA1c_lev	blood_glu	diabetes	diabetes(kategorikal)
Female	80	0	1	never	25.19	6.6	140	0	no
Female	54	0	0	No Info	27.32	6.6	80	0	no
Male	28	0	0	never	27.32	5.7	158	0	no
Female	36	0	0	current	23.45	5	155	0	no
Male	76	1	1	current	20.14	4.8	155	0	no
Female	20	0	0	never	27.32	6.6	85	0	no
Female	44	0	0	never	19.31	6.5	200	1	yes
Female	79	0	0	No Info	23.86	5.7	85	0	no
Male	42	0	0	never	33.64	4.8	145	0	no
Female	32	0	0	never	27.32	5	100	0	no
Female	53	0	0	never	27.32	6.1	85	0	no
Female	54	0	0	former	54.7	6	100	0	no

Gambar 3. 2 Contoh *dataset*

3.3 Exploratory Data Analysis

Exploratory data analysis (*EDA*) adalah metode yang digunakan untuk melakukan analisis data secara eksploratif yang bertujuan untuk memahami karakteristik utama dari *dataset*. *Exploratory data analysis* bertujuan untuk mengetahui anomali data sehingga dapat dilakukan tindakan selanjutnya yang kemudian dapat meningkatkan kualitas data yang ada. Dalam penelitian ini, *exploratory data* yang akan dilakukan antara lain, yaitu:

a. Pengecekan *missing value*. Mengidentifikasi data yang hilang dari setiap variabel.
b. Pengecekan tipe data. Mengecek jenis atau tipe data dari masing-masing variabel.
c. Pengecekan *outlier*. Mengidentifikasi adanya kemungkinan adanya *outlier* dari masing-masing variabel.
d. Pengecekan korelasi antar tabel. Menganalisis dan mengidentifikasi hubungan antar variabel.
e. Pengecekan data yang duplikat. Mengidentifikasi duplikasi data dari setiap variabel.

3.4 Preprocessing Data

Preprocessing data adalah langkah untuk mempersiapkan data agar lebih siap digunakan dalam rangka analisis dan ekstraksi pengetahuan [37]. Proses ini termasuk untuk mengubah data ke dalam format yang lebih sesuai untuk dipahami oleh sistem. Dalam tahapan ini, hal yang akan dilakukan antara lain, yaitu *undersample*, *encoding*, normalisasi dan *splitting*.

a. *Underesampling*. Mengurangi jumlah kelas yang bernilai 1 pada variabel diabetes dikarenakan proporsi yang tidak seimbang antara kelas 1 dan kelas 0.
b. *Encoding*. Mengubah data kategorikal menjadi format numerik dengan menggunakan *one-hot encoding*.
c. Normalisasi. Mengubah data ke dalam skala yang lebih kecil dan seragam dalam rentang 0 hingga 1.
d. *Splitting.* Pembagian *dataset* menjadi 80% untuk dilatih dan 20% untuk diuji.

3.5 Validating dan *training model*

3.5.1 *Neural Network*

Tahapan proses perhitungan algoritma *neural network*, yaitu:

a. Menginisialisasikan 8 bobot awal yang dipilih secara acak, karena dalam penelitian ini hanya terdapat 8 fitur independen.
b. Kemudian setiap fitur dikalikan dengan bobot awal dan ditambah dengan bias, sehingga menghasilkan rumus, yaitu $Z = (X_i . W_i) + b$
c. Kemudian *output* dari Z akan diteruskan ke fungsi aktivasi *ReLU* untuk menambahkan elemen *non linearitas* dalam jaringan dengan rumus, yaitu $f(x) = \max(0, x)$. Jika hasilnya negatif, maka akan menjadi 0, jika positif maka akan sesuai dengan aslinya.
d. Setelah itu, akan dikirim ke *output layer* dengan bobot yang baru dengan rumus dan akan melewati fungsi aktivasi *sigmoid*. Rumus untuk menghitung fungsi *sigmoid*, yaitu $Y_{pred} = \frac{1}{1+e^{-z^n}}$. Hasil dari perhitungan rumus tersebut yaitu angka yang berkisar 0-1 yang kemudian jika angkanya berada di atas 0,5 akan menjadi nilai 1 dan begitupun sebaliknya.
e. Karena menggunakan kode (*MLPClassifier*) yang merupakan *stochastic Gradient Descent* (*SGD*), maka rumus yang digunakan adalah $W_{baru} = W_{lama} - \alpha . gradien$ di mana α adalah *learning rate* atau nilai kecil agar tidak terjadi *overshooting* dan proses ini akan berlanjut sampai model mencapai titik *konvergensi* atau hingga mencapai 1000 iterasi.

3.5.2 *Random Forest*

Tahapan proses perhitungan algoritma *random forest* dalam penelitian ini adalah sebagai berikut:

a. Model akan mengambil 100 *subset* data secara acak dan mungkin terdapat sampel yang bisa terpilih lebih dari sekali dan bisa saja ada yang tidak terpilih.
b. Pada setiap *subset*, model akan membangun 1 *decision tree* dengan mencapai kedalaman maksimal 10.
c. Kemudian *internal node* akan ditemukan dengan mencari nilai *weighted gini index* terkecil. Rumus untuk mendapatkan *weighted gini index* terkecil, yaitu $gini_{weighted} = \frac{N_1}{N}\ gini_1 + \frac{N_2}{N}\ gini\ _2$. untuk mendapatkan *gini* dari rumus tersebut yaitu dengan menggunakan rumus $gini = 1 - \sum_{i-1}^{k} p2_i$.
d. Untuk mendapatkan variabel y prediksi, maka akan dilakukan *voting* dari setiap pohon. Jika mayoritas pohon memprediksi 1, maka *random forest* akan menghasilkan angka 1.

3.5.3 *Support Vector Machine*

Tahapan proses perhitungan algoritma *support vector machine* adalah sebagai berikut:

a. Menghitung bobot dan bias untuk digunakan dalam *decision function*. Rumus untuk mencari bobot, yaitu $w = \sum_{i=i}^{N} \alpha_i . y_i . x_i$. Sementara untuk menghitung bias dilakukan dengan rumus $b = y_i - w . x_i$
b. Mencari *decision function* dengan rumus $f(x) = w . x + b = 0$. Di mana w adalah bobot dan b adalah bias. Jika $f(x)$ kurang dari 0, maka akan diklasifikan sebagai -1.

Setelah model dilatih, maka model akan dievaluasi dengan menggunakan teknik *10-fold cross validation.* Tujuan validasi ini adalah memastikan bahwa model tidak sekedar mengingat data pelatihan, tetapi juga bisa memprediksi data yang tidak pernah ditemui. Selain itu, diharapkan dengan teknik validasi ini kinerja model secara keseluruhan lebih akurat. *10-fold cross validation* dapat dihitung dengan menggunakan pendekatan matematika dengan menggunakan rumus, yaitu:

$Metrik_i = \frac{Jumlah\ prediksi\ benar\ pada\ fold-i}{Total\ data\ pada\ fold-i}$, kemudian hasil dari semua fold tersebut akan dijadikan sebagai metrik akhir dengan rumus: $Metrik\ akhir = \frac{\sum_{i=1}^{10} Metrik_i}{10}$.

3.6 Model Evaluation

Model Evaluation yang digunakan dalam penelitian ini yaitu, *akurasi, precision, recall, dan f1-score* yang kemudian akan divisualisasikan agar lebih mudah dipahami.