BAB II
TINJAUAN PUSTAKA

Dalam penelitian ini, sumber referensi dibutuhkan untuk meninjau teori-teori mengenai pembahasan yang akan dilakukan. Tinjauan pustaka yang terkait dengan penelitian ini diuraikan pada bab ini melalui pembagian sub-bab di bawah ini.

2.1 Pencapaian Terdahulu

Penelitian ini dilakukan berdasarkan pencapaian dari penelitian terdahulu terhadap penggunaan algoritma KNN dan *Random Forest* atau prediksi terhadap diabetes melitus.

Pertama, referensi yang digunakan berupa penelitian mengenai pembuatan sistem prediksi penyakit diabetes dengan cara klasifikasi data terhadap penderita menggunakan algoritma *decision tree,* dimana hasil klasifikasi dievaluasi dengan *confusion matrix* dan kurva *ROC (Receiver Operating Characteristic)* yang bertujuan untuk mengetahui akurasi dari hasil klasifikasi tersebut (Andriani, 2013). Evaluasi pertama menggunakan *confusion matrix*. Hasil yang diperoleh dengan *confusion matrix* yaitu tingkat akurasi data *training* sebesar 87,86% dan data *testing* sebesar 84,29%. Evaluasi kedua menggunakan kurva ROC. Hasil yang diperoleh, keakuratan data *training* sebesar 0,963, sementara pada data *testing* sebesar 0,941. Andriani membuktikan bahwa penyakit diabetes dapat dibuktikan dengan menggunakan algoritma *machine learning* dengan adanya sistem prediksi yang dibuat menggunakan *decision tree.*

Kedua, referensi yang digunakan berupa pengujian performa dari tiga algoritma yaitu, *Naïve Bayes, K-Nearest Neighbors,* dan *Random Forest* dalam klasifikasi penyakit kronis pada ginjal atau *Chronic Kidney Diseases* (CKD). CKD itu sendiri merupakan penyakit dimana fungsi ginjal tidak normal atau kegagalan secara progresif terhadap ginjal dalam jangka pendek selama beberapa bulan atau tahun (Devika et al., 2019.). Hasil yang didapatkan dalam pengujian data untuk mengukur akurasi dari ketiga algoritma tersebut yaitu, *Naïve Bayes* sebesar 99,635%, KNN sebesar 87,78%, dan *Random Forest* sebesar 99.844. Sehingga

didapatkan oleh Devika dkk. bahwa bahwa algoritma *Random Forest* lebih baik dari dua algoritma lainnya.

Ketiga, referensi yang didapatkan berupa pengujian terhadap dua algoritma klasifikasi yaitu *Decision Tree* dan *Naïve Bayes* dalam memprediksi penyakit diabetes (Permana & Dewi, 2021). Hasil yang didapatkan yaitu bahwa *Decision Tree* lebih baik dalam melakukan prediksi dengan nilai akurasi sebesar 95,58% dibandingkan dengan *Naïve Bayes* yang memiliki nilai akurasi sebesar 87,69%.

Keempat, referensi yang digunakan yaitu penelitian dalam melakukan pengujian terhadap algoritma ID3 dan *Naïve Bayes* dalam memprediksi diabetes untuk didapatkan komparasi antara keduanya (Nurdiana & Algifari, 2020). Hasil yang didapatkan dari penelitian tersebut yaitu evaluasi nilai akurasi terhadap algoritma ID3 sebesar 74% sementara *Naive Bayes* sebesar 76%. Pada penelitian ini didapatkan bahwa algoritma *Naïve Bayes* memiliki kinerja yang lebih baik dibandingkan dengan ID3.

Dari beberapa referensi yang didapatkan, didapatkan asumsi bahwa algoritma *machine learning* dapat memprediksi suatu penyakit. Pada penelitian ini digunakan algoritma *K-Nearest Neighbors* dan *Random Forest* dalam menguji penyakit diabetes melitus serta membandingkan akurasi keduanya menggunakan metode *Confusion Matrix* dalam mengevaluasi model *machine learning* serta mengukur tingkat akurasi.

2.2 Tinjauan Teoritis

Referensi teoritis dimana sub-bab yang menjelaskan teori-teori yang diambil sebagai referensi dari penelitian yang dilakukan. Teori yang diambil terbagi menjadi beberapa bagian sub-sub-bab yang diangkat dari berbagai sumber berbentuk jurnal, atau artikel ilmiah terkait topik yang dibahas.

2.2.1 Diabetes Melitus

Diabetes Melitus merupakan sekumpulan dari penyakit yang menyerang metabolism ditandai dengan terjadinya hiperglikemia kronis dikarenakan gangguan pada sekresi insulin, kerja insulin, atau keduanya (Kharroubi, 2015).

Jumlah sebanyak 57 juta orang dengan pra-diabetes, terjadi peningkatan kadar glukosa darah yang meningkatkan risiko terhadap berbagai macam penyakit salah satunya diabetes (Williams et al., 2017). Banyak skor dalam menilai risiko dan dilakukan persamaan untuk memperoleh prediksi yang telah dikembangkan dalam mengidentifikasi orang yang memiliki risiko tinggi terjangkit diabetes atau pra-diabetes berdasarkan faktor umum seperti indeks massa tubuh (IMT) dan riwayat keluarga diabetes (Fusar-Poli et al., 2015). Pada penelitian yang dilakukan oleh Brown dkk. bahwa adanya perilaku terkait dukungan yang dinilai di seluruh aspek dari perawatan diri yaitu antara lain pola makan, aktivitas fisik, pemantauan glukosa darah, dan konsumsi obat dan/atau insulin (Mayberry, 2012). Kejadian tingginya angka diabetes juga meningkat seiring bertambahnya usia hingga sekitar usia 65 tahun. Akibatnya, orang tua dengan diabetes memiliki kemungkinan terjangkit diabetes atau memang sudah di usia pertengahan atau bahkan usia awal (Kirkman et al., 2012). Berdasarkan sumber-sumber yang ada, fitur-fitur yang digunakan cukup mewakili dalam memprediksi apakah seseorang terindikasi diabetes atau tidak.

2.2.2 *Machine Learning*

Machine Learning pada dasarnya merupakan sebuah pendekatan secara teknik terhadap suatu fokus maksimal untuk setiap teknik yang memiliki kecenderungan dalam meningkatnya perubahan secara adaptif (Bonaccorso, 2017). Bonaccorso (2017) juga menjelaskan tujuan utama dari *Machine Learning* sendiri untuk mempelajari, merekayasa, serta meningkatkan model matematika yang dapat dilatih berdasarkan konteks data untuk mendapatkan suatu kesimpulan dan keputusan tanpa membutuhkan seluruh faktor yang berpengaruh. *Machine Learning* sendiri terbagi menjadi beberapa metode, salah satunya yaitu *supervised learning. Supervised learning* pada dasarnya memperkirakan fungsi dimana data pelatihan *(training)* yang berupa kumpulan pasangan *x* dan *y* dan bertujuan untuk menghasilkan hasil yang berupa prediksi *y,* dengan kata lain metode ini menempatkan *y* sebagai keluaran prediksi untuk setiap *input x* (Jordan & Mitchell, 2015). Salah satu model *supervised learning* yaitu klasifikasi, dimana dilakukan prediksi model untuk mengkategorikan kelas label (Data Mining - Classification & Prediction, 2022). Adapun proses pembelajaran yang dimaksud merupakan suatu

usaha untuk memperoleh kecerdasan melalui dua tahap yaitu pelatihan *(training)* dan pengujian *(testing)* (Huang et al., 2015). Salah satu *platform* yang dapat mengaplikasikan *machine learning* yaitu Google Colab yang merupakan salah satu layanan dari Google. *K-Nearest Neighbors* dan *Random Forest* juga merupakan algoritma yang diterapkan ke pemodelan *machine learning.* Dalam penerapannya, *machine learning* memerlukan kumpulan data, atribut atau fitur, serta model yang merupakan hasil dari *machine learning.*

2.2.3 Data *Pre-processing*

Pada *pre-processing* dilakukan beberapa tahapan yaitu data *cleansing* dan normalisasi data. Data *cleansing* atau data *cleaning* digunakan dalam meningkatkan kualitas data yang digunakan dengan mengidentifikasi serta menghapus *error* atau ketidak konsistenan pada data yang digunakan (Fakhitah Ridzuan, 2019).

Normalisasi yang digunakan berupa *Min Max Scaler (MMS)* sebagai model normalisasi. MMS pada dasarnya menurunkan skala dari data pada rentang [0, 1] atau [-1, 1]. Rumus matematika pada *Min Max Scaling* adalah sebagai berikut :

$$x' = \frac{x - min\,(x)}{(x)\ - min\,(x)}$$

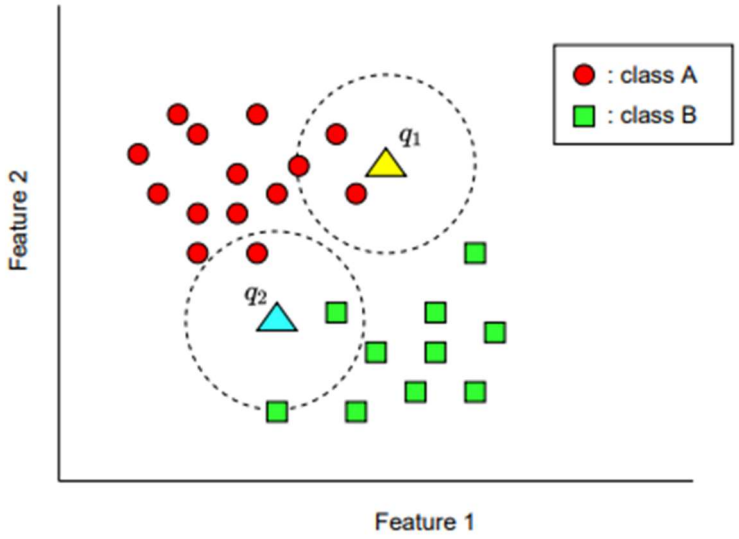
Dengan menggunakan metode *pre-processing Min Max Scaler* dalam setiap model yang digunakan, akan mendapatkan tingkat prediksi terhadap akurasi dari perhitungan sebelumnya (Samrat Kumar Dey, 2018)

2.2.4 *K-Nearest Neighbors* (KNN)

K-Nearest Neighbors (KNN) merupakan salah satu metode klasifikasi yang paling mendasar dan sederhana juga sering digunakan untuk studi klasifikasi dimana klasifikasi KNN dikembangkan berdasarkan kebutuhan untuk melakukan analisis terhadap diskriminan ketika perkiraan parameter kepadatan probabilitas tidak diketahui atau sulit ditentukan. Algoritma KNN termasuk kelompok *instance-based learning* dimana algoritma ini juga merupakan salah satu teknik *lazy learning*. Proses KNN pada dasarnya dilakukan dengan mencari kelompok terhadap objek K dari data *training* yang terdekat (mirip) terhadap objek pada data *testing* (Leidiana, 2013).

Secara umum dalam mendefinisikan jarak antara kedua objek x dan y menggunakan rumus jarak *Euclidean* sebagai berikut :

$$euc = \sqrt{(\sum_{i=1}^{n} (x_i - y_i)^2)}$$



Gambar 2.1 KNN

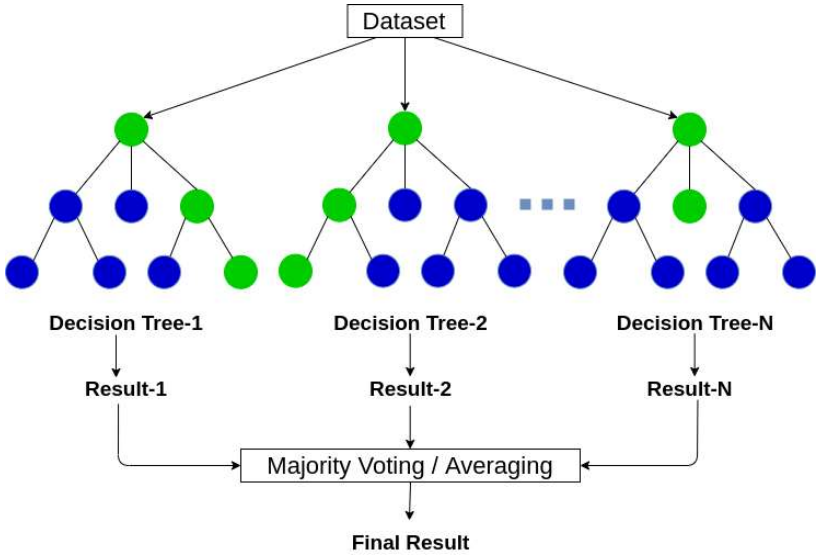
KNN mengaplikasikan kelas label secara mayoritas dari pola nilai K yang terdekat pada ruang data (Kramer, 2013). Dalam menentukan nilai K pada algoritma ini, digunakan salah satu metode untuk menentukan nilai K-Optimal, yaitu Metode *Elbow* atau *Elbow Method. Elbow Method* adalah sebuah metode yang digunakan untuk menghasilkan informasi dalam menentukan nilai *cluster* terbaik dengan melihat persentase antara nilai *cluster* satu dengan yang lainnya, dan akan membentuk sebuah siku atau *elbow* (Rena Nainggolan, 2018). Pada penelitian ini digunakan *Elbow Method* dengan menentukan nilai *error* antara satu nilai K dengan lainnya dalam menentukan K-Optimal sebagai *cluster* perbandingan satu dengan lainnya, kemudian K-Optimal yang didapatkan dari perbandingan tersebut digunakan dalam pemodelan KNN yang nantinya akan dilakukan evaluasi dan perbandingan dengan algoritma *Random Forest.*

2.2.5 *Random Forest*

Algoritma klasifikasi *Random Forest* merupakan salah satu teknik pembelajaran ensemble tersukses yang terbukti menjadi teknik yang sangat populer dan kuat terhadap pengenalan pola dan *machine learning* untuk *high-dimensional classification* dan permasalahan yang mencondong (K. Fernandes, 2015).

Random forest classifier terdiri dari kombinasi *tree classifiers* dimana masing-masing pengklasifikasi dihasilkan dari *random vector* yang didapatkan secara independen dari *input vector.* Masing-masing *tree* memberikan pengaruh terhadap kelas paling banyak dari *input vector. Random forest classifier* digunakan dalam pemilihan fitur secara random atau kombinasi dari fitur yang dapat menghasilkan *tree* (Athey et al., 2019). Sebuah kelemahan dari *tree classifier* yaitu tingginya angka *variance*. Hal ini menyebabkan terjadinya perubahan pada *training dataset* untuk menghasilkan *tree* yang sangat berbeda. *Error* yang didapatkan *nodes* atau simpul pada akar *tree* juga akan merambat sampai ke *leaves* atau daun. Dalam hal membuat sebuah klasifikasi *tree* yang lebih stabil, sebah algoritma *decision forest* diciptakan oleh Ho, Amit, dan Geman, yang kemudian oleh Breiman diintegrasikan dalam *random forest. Decision forest* terdiri dari gabungan *decision trees.* Hal ini dapat dilihat pada sebuah pengklasifikasi dimana berisi beberapa metode klasifikasi, atau hanya satu metode namun dengan beberapa paremeter yang berbeda.

Didapatkan sebuah *learning set* $L = ((M_1, N_1),\ldots,(M_n, N_n))$ terdiri dari n *vectors,* $M \in X$ dimana X terdiri dari sebuah set angka atau symbol, dan $N \in Y$ dimana Y terdiri dari sebuah set kelas penglabelan. Dalam kasus klasifikasi, pengklasifikasi dilakukan $X \rightarrow Y$. Sebuah *input vector* yang baru diklasifikasikan dari setiap *tree.* Prinsip dari *random forest* untuk membuat *binary sub-trees* menggunakan data *sample training* dari L dan menyeleksinya secara acak pada masing-masing *node* pada *X.* Kemudian dilakukan pemilihan klasifikasi secara menyeluruh dari setiap *tree* yang ada (Azar, 2013).

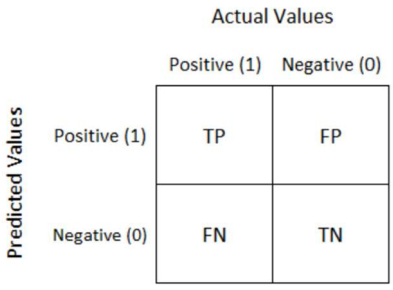


Gambar 2.2 *Random Forest*

Berdasarkan gambar 2.1, *random forest* terdiri dari beberapa *decision tree* dalam menentukan label kelas melalui *majority voting* atau kelas terbanyak dari hasil masing-masing *decision tree* akan menjadi hasil akhir dari sebuah *random forest.*

2.2.6 *Confusion Matrix*

Pada jurnal yang dibuat oleh Xinyang Deng, dkk., *Confusion Matrix* merupakan sebuah konsep *machine learning,* dimana berisi informasi tentang data aktual dan data prediksi klasifikasi yang dilakukan oleh sistem klasifikasi. *Confusion Matrix* memiliki dua dimensi, satu dimensi memiliki indeks terhadap kelas objek aktual dan lainnya indeks dari kelas prediksi (Deng, Liu, Deng, & Mahadevan, 2016).



Gambar 2.3 *Confusion Matrix*

Pada gambar 2.3 terdapat istilah TP, FP, FN, dan TN. TP yaitu *True Positive* yang berarti prediksi yang dilakukan positif dan hasilnya *true,* FP atau *False Positive (Type 1 Error)* dimana prediksi positif dan hasilnya *false,* FN atau *False Negatif (Type 2 Error)* dimana prediksi negative dan hasil juga *false,* dan terakhir yaitu TN atau *True Negative* yaitu prediksi negative dan hasilnya *true* (Narkhede, 2018). Sebagai contoh terdapat sebuah kasus sebagai berikut :

1. TP : Dimana seseorang diprediksi memiliki diabetes, dan memang benar ternyata seseorang tersebut memiliki diabetes
2. TN : Dimana seseorang diprediksi tidak memiliki diabetes dan memang benar ternyata orang tersebut tidak memiliki diabetes
3. FP : Seseorang diprediksi memiliki diabetes, ternyata prediksi tersebut salah, orang tersebut tidak memiliki diabetes
4. FN : Seseorang diprediksi tidak memiliki diabetes, ternyata prediksi tersebut salah, orang tersebut memiliki diabetes

Angka atas pengukuran dari performa klasifikasi dapat didefinisikan berdasarkan *Confusion Matrix.* Beberapa pengukuran yang biasa dilakukan adalah sebagai berikut :

Akurasi yaitu proporsi dari angka total prediksi yang benar, akurasi harus memiliki hasil yang tinggi :

$$Accuracy = \frac{TP + TN}{Total}$$

Presisi yaitu pengukuran dari kelas yang diprediksi sebagai positif ada berapa banyak yang benar positif, *precision* harus memiliki hasil yang tinggi :

$$Precision = \frac{TP}{TP + FP}$$

Recall yaitu pengukuran terhadap kemampuan model prediksi dari kelas yang diprediksi sebagai positif ada berapa banyak yang benar diprediksi secara benar, *recall* harus memiliki hasil yang tinggi :

$$Recall = \frac{TP}{TP + FN}$$