BAB 2
LANDASAN TEORI

Dalam melakukan penelitian, telaah literatur diperlukan untuk memahami lebih dalam mengenai konsep yang akan dipakai terkait topik penelitian.

2.1 Klasifikasi

Klasifikasi merupakan salah satu teknik dalam data mining yang digunakan untuk mengelompokkan data ke dalam kelompok atau *class* yang telah ditentukan sebelumnya. Metode ini termasuk dalam *supervised learning*, yang memerlukan data berlabel untuk menghasilkan aturan yang dapat mengklasifikasikan data ke dalam kelompok atau kelas yang telah ditetapkan [25].

2.2 Machine Learning

Machine learning merupakan pengembangan dari kecerdasan buatan yang dapat melakukan pembelajaran secara mandiri tanpa harus di program berulang kali oleh manusia [11]. Penerapan *machine learning* melibatkan berbagai bidang seperti visi komputer, pengenalan ucapan, kontrol robot, deteksi *spam email*, dan kedokteran [26]. Terdapat tiga jenis pembelajaran *machine learning*, yaitu *unsupervised learning, supervised learning*, dan *reinforcement learning* [26].

- *Unsupervised learning* merupakan salah satu cabang dari *machine learning* yang berfokus pada mencari pola tertentu di dalam *dataset* yang besar dan berfokus untuk mengklasifikasikan data dari *dataset* menjadi beberapa kategori tanpa adanya pelatihan secara eksplisit yang berpotensi sangat besar untuk menyelesaikan permasalahan *clustering* [27].
- *Supervised learning* digunakan untuk membentuk model prediktif yang memproyeksikan nilai yang tidak diketahui dalam menggunakan nilai lain yang ada pada *dataset*. *Supervised learning* memiliki kumpulan data masukan dan keluaran, dan membangun model untuk membuat prediksi terhadap tanggapan terhadap *dataset* baru [28]. *Supervised learning* dapat dibagi menjadi metode klasifikasi dan regresi. *K-nearest Neighbor* (KNN), *Decision Tree* (DT), *Support Vector Machine* (SVM), *Logistic Regression*

(LR), *Artificial Neural Network* (ANN), *Naïve Bayes* (NB), adalah beberapa algoritma dalam *supervised learning* [29].

- *Reinforcement learning* merupakan metode *machine learning* yang mengoptimalkan pengambilan keputusan yang didasarkan pada tahapan. *Reinforcement learning* sementara tidak bisa dianggap sebagai metode *supervised learning* maupun *unsupervised learning* karena hasil akhir (*output*) yang diinginkan tidak diketahui untuk melatih model dan tidak ada batasan dalam melatih model yang disebabkan oleh kurangnya label [30].

2.3 Logistic Regression

Logistic Regression adalah sebuah algoritma berfungsi untuk mencari sebuah peluang atau probabilitas dari variabel dependen (target) yang dipengaruhi oleh variabel independen (prediktor) [16]. *Logistic Regression* sebenarnya merupakan pengembangan dari regresi linear [17]. Persamaan regresi linear [31].

$$Y = \beta_0 + \beta_1 X + e \tag{2.1}$$

Keterangan:

Y merupakan variabel target.
β_0 merupakan konstanta.
β_1 merupakan koefisien regresi untuk variabel prediktor.
X merupakan variabel prediktor.
e merupakan *error term*.

Dalam regresi linear terdapat pelanggaran yang disebut sebagai *Gauss-Markov*, yang dapat terjadi seperti di mana variabel dependennya memiliki tipe kategori, tetapi variabel independennya memiliki tipe interval [32]. Oleh karena itu, digunakan *Logistic Regression* yang perlu dilakukan transformasi terlebih dahulu dengan transformasi logit untuk membuat peluang sukses selalu berada pada rentang nol dan satu [33]. Dalam algoritma *Logistic Regression* variabel dependen Y bersifat dikotomi yang artinya hanya dapat memiliki dua kategori, yaitu nilai 0 (misalnya, ”tidak”) atau 1 (misalnya, ”ya”) [17]. Persamaan *Logistic Regression* sebagai berikut [17].

$$\ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X \tag{2.2}$$

Logistic Regression menggunakan fungsi *sigmoid* untuk merepresentasikan probabilitas hasil biner yaitu 0 dan 1. Persamaan fungsi *sigmoid* dapat ditulis sebagai berikut [34].

$$S(x) = \frac{1}{1+e^{-x}} \tag{2.3}$$

Dengan kedua rumus tersebut, rumus akhir dari *Logistic Regression* dapat ditulis dengan persamaan berikut [16].

$$P(Y) = \frac{1}{1+e^{-(\beta_0+\beta_1 X)}} = \frac{e^{(\beta_0+\beta_1 X)}}{1+e^{(\beta_0+\beta_1 X)}} \tag{2.4}$$

Keterangan:

ln merupakan logaritma natural.

p merupakan probabilitas atau peluang.

e merupakan eksponen = 2,71828182845904.

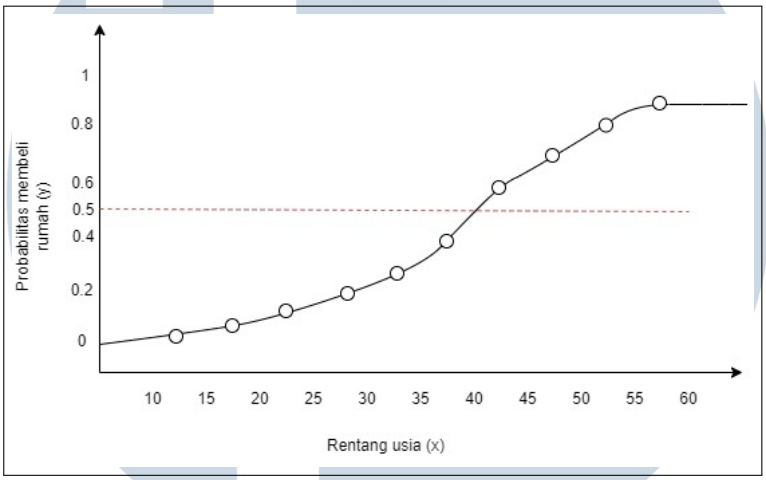
Model dari *Logistic Regression* dapat di deskripsikan dengan *pseudocode* dibawah ini [35].

Algorithm 1 *Logistic Regression*

Input: Data yang sudah di *preprocess*
Output: Klasifikasi terbaik
Algorithm:
for $i \leftarrow 1$ **to** k **do**
    for *instance* data training & testing d_i **do**
        Tetapkan nilai target untuk regresi sebagai berikut:
        $Z \leftarrow \frac{y_j - P(1-d_j)}{[P(1-d_j)\cdot(1-p(1-d_j))]}$
        Inisialisasi bobot *instance* d_i menjadi $P(1|d_i)\cdot(1-P)\cdot(1|d_i)$
        Finalisasi nilai $f(i)$ untuk data dengan nilai kelas z_i & bobot w_i
        Penentuan Label Klasifikasi
        if $P(1|d_i) > 0.5$ **then**
            Tetapkan (label kelas:1)
        else
            Tetapkan (label kelas:0)
        end if
    end for
end for

Sumber: [35]

Kurva dalam *Logistic Regression* mempunyai bentuk S yang di sebut dengan *sigmoid* yang digunakan untuk menghasilkan probabilitas dari nol ke satu [36]. Berikut merupakan kurva dari *Logistic Regression* berdasarkan contoh dari suatu *dataset*.



Gambar 2.1. Kurva *Logistic Regression*

Sumber: [16]

Kurva tersebut merupakan contoh dari kasus sederhana untuk memprediksi seseorang akan membeli rumah atau tidak berdasarkan usia. Sumbu horizontal (x) melambangkan variabel independen (prediktor) yaitu rentang usia dan sumbu vertikal (y) merupakan probabilitas atau peluang seseorang membeli rumah atau tidak berdasarkan rentang usia. Angka 0.5 merupakan garis batas pada kurva yang berguna sebagai tolak ukur bahwa jika garis atau probabilitas yang di bawah 0.5 maka akan masuk ke dalam kategori nol (tidak) dan jika garis atau probabilitasnya di atas 0.5, maka akan masuk ke dalam kategori satu (ya) [37].

2.4 Diabetes

Diabetes dapat dijelaskan sebagai suatu kondisi medis yang bersifat kronis dan memiliki penyebab yang bervariasi, yang dimulai dengan tingginya kadar glukosa dalam darah serta gangguan metabolisme karbohidrat, lipid, dan protein yang disebabkan oleh kurangnya fungsi insulin. Insulin yang kurang dapat disebabkan oleh kurangnya produksi insulin oleh sel-sel di pankreas atau karena respons sel-sel tubuh terhadap insulin rendah [38].

Terdapat tiga tipe diabetes yang ada yaitu, diabetes tipe 1 merupakan diabetes yang terjadi karena tubuh hanya memproduksi sedikit insulin, terjadi pada remaja, atau anak-anak. Lalu ada diabetes tipe 2 yang ditandai dengan tubuh tidak merespons insulin secara maksimal, yang disebabkan karena pola makan yang buruk, gaya hidup yang kurang gerak (berolahraga) dan tingkat obesitas. Lalu ada diabetes *mellitus gestasional* (GDM) yang terjadi karena gula darah tinggi selama kehamilan dan setelah kelahiran diabetes tipe ini biasanya hilang [1].

Untuk diagnosa diabetes terdapat beberapa tes yang dilakukan [39].

Tabel 2.1. Tes diagnosa diabetes

Tes	**Normal**	**Prediabetes**	**Diabetes**
A1C	Di bawah 5.7%	5.7% sampai 6.4%	6.5% atau lebih tinggi.
Fasting Blood Sugar	Di bawah 100 mg/dL	100 mg/dL sampai 125 mg/dL	126 mg/dL atau lebih tinggi.
Glucose Tolerance	Di bawah 140 mg/dL	140 mg/dL sampai 199 mg/dL	200 mg/dL atau lebih tinggi.
Random Blood Sugar	-	-	200 mg/dL atau lebih tinggi.

Sumber: [39]

Tes A1C dilakukan dengan mengukur rata-rata kadar gula darah dalam rentan dua sampai tiga bulan terakhir. Lalu pada tes *fasting blood sugar* atau tes gula darah puasa dilakukan dengan mengukur tingkat gula darah setelah melakukan puasa semalaman. Pada tes *glucose tolerance* dilakukan dengan mengukur gula darah sebelum dan sesudah meminum cairan yang mengandung glukosa. Tes *random blood sugar* dilakukan dengan kapan saja dengan mengukur tingkat gula darah [40].

Tekanan darah merupakan salah satu faktor penting dalam diabetes, karena jika tekanan darah tinggi dapat membuat sel dalam tubuh tidak sensitif terhadap insulin, yang di mana insulin itu sendiri berperan dalam meningkatkan glukosa di banyak sel, sehingga jika terjadi resistensi insulin oleh sel, maka kadar gula di dalam darah juga dapat mengalami gangguan [41]. Tekanan darah yang normal pada anak-anak pada umur tiga tahun sekitar 91–120 mmHg (sistolik) and 46–80 mmHg (diastolik), tetapi jika sudah berumur 6-12 tahun, tekanan darah yang normal sekitar 96–131 mmHg (sistolik) and 55–62 mmHg (diastolik). Untuk orang dewasa

tekanan darah yang normal sekitar 90–120 mmHg (sistolik) and 60–80 mmHg (diastolik) [42].

Berat badan juga merupakan salah satu faktor yang bisa menyebabkan diabetes. Ketika seseorang terkena obesitas atau berat badan berlebih, tubuhnya akan menghasilkan lebih banyak lemak dan hormon yang dapat menurunkan sensitivitas insulin, yang menyebabkan sel-sel tubuh tidak dapat menggunakan insulin secara efektif untuk mengontrol kadar glukosa darah [43]. Indeks masa tubuh yang menentukan seseorang obesitas atau tidak sebagai berikut [44].

1. Kurang dari 18,5: Berat badan kurang (*underweight*).
2. 18,5–24,9: Berat badan normal.
3. 25–29,9: Berat badan berlebih (*overweight*).
4. 30–34,9: Obesitas tahap 1.
5. 35–39,9: Obesitas tahap 2.
6. Di atas 40: Obesitas tahap 3.

Seiring bertambahnya usia, khususnya pada usia lebih dari 40 tahun mempunyai risiko meningkatnya terkena diabetes karena penuaan yang menyebabkan kurangnya kemampuan sel pancreas dalam memproduksi insulin sehingga dapat berdampak pada kadar glukosa dalam darah [45].

Selama masa kehamilan, perubahan hormon seperti meningkatkan hormon estrogen yang berpotensi menghambat kerja insulin, maka tubuh akan lebih sulit dalam mengelola gula darah dan menyebabkan terjadinya resistensi insulin pada ibu hamil [46].

2.5 Holdout Data

Holdout data merupakan salah satu metode yang membagi *dataset* menjadi beberapa bagian, yaitu data *training*, dan data *test*. Data *training* digunakan untuk melatih model, sedangkan data *test* digunakan untuk evaluasi akhir model setelah proses pelatihan selesai [47].

2.6 Synthetic Minority Oversampling Technique

Synthetic Minority Oversampling Technique (SMOTE) adalah sebuah teknik atau metode yang berfungsi untuk mengatasi masalah data yang tidak seimbang. Cara kerja SMOTE yaitu dengan menambahkan jumlah *class* yang minoritas sampai setara dengan *class* mayoritas dengan cara membentuk data buatan [48].

2.7 Standar Scaller

Standar Scaller adalah sebuah metode dari teknik *preprocessing* yang melakukan standardisasi terhadap fitur dengan cara menghapus rata-rata dan menskalakan unit varian. Standardisasi dilakukan agar mencegah adanya nilai yang terlalu besar dibanding dengan nilai yang lain antar data. Persamaan dari *standar scaller* sebagai berikut [49].

$$z = \frac{x - \mu}{\sigma} \tag{2.5}$$

Keterangan:

z merupakan *z-score*.

x merupakan nilai yang akan di standardisasi.

μ merupakan *mean* dari x.

σ merupakan standar deviasi.

Mean didapatkan dari persamaan berikut ini [50].

$$\bar{x} = \frac{x_1 + x_2 + \ldots + x_i}{n} \tag{2.6}$$

Sedangkan untuk persamaan standar deviasi sebagai berikut [50].

$$S = \sqrt{\frac{\sum (x_i - \bar{x})^2}{n}} \tag{2.7}$$

Keterangan:

S merupakan standar deviasi.

x_i merupakan nilai x ke i.

$\bar{x}$ merupakan nilai rata-rata data.

n merupakan jumlah data.

2.8 Principal Component Analysis

Principal Component Analysis (PCA) adalah salah satu teknik dari metode *feature extraction* yang dapat digunakan untuk mencari pola pada sebuah data dan melakukan reduksi dimensi dari atribut data tersebut tanpa menghilangkan informasi penting pada data tersebut [51]. *Principal Component Analysis* (PCA) dapat melakukan reduksi dimensi dari suatu objek, sehingga objek tersebut menjadi lebih ringkas dan tetap mampu mempertahankan karakteristik penting yang dimilikinya, serta dapat digunakan sebagai metode untuk menguji apakah setiap variabel dalam *dataset* saling terkait atau tidak terkait sama sekali [52].

Principal Component Analysis (PCA) mempunyai langkah-langkah yang harus dilakukan yaitu:

1. Standardisasi data

   standardisasi data merupakan proses untuk mengubah atau mengkonversi nilai pada data agar setiap nilai yang ada pada data mempunyai rentang nilai yang sama [53].

2. Matriks kovarian

   Matriks kovarian digunakan untuk melihat hubungan antara dua fitur yang akan di masukan nilainya untuk mencari nilai dari *eigen value* dan *eigen vector*. Persamaannya sebagai berikut [53].

   - Varian atribut

$$\text{var}(A_1) = \frac{\sum_{i-1}^{n} (x_i - \bar{x})^2}{n-1} \tag{2.8}$$

Keterangan:

x_i merupakan data ke i.

$\bar{x}$ merupakan nilai rata-rata dari seluruh nilai x

n merupakan jumlah data

$$\text{var}(A_2) = \frac{\sum_{i-1}^{n}(y_i - \bar{y})^2}{n-1} \tag{2.9}$$

Keterangan:

y_i merupakan data ke i.

$\bar{y}$ merupakan nilai rata-rata dari seluruh nilai y.

- Kovarian dua atribut

$$\text{cov}(A_1, A_2) = \frac{\sum_{i-1}^{n}(x_i - \bar{x}) - (y_i - \bar{y})}{n-1} \tag{2.10}$$

- Matriks kovarian

$$= \begin{pmatrix} \text{cov}(A_1, A_1) & \text{cov}(A_1, A_2) \\ \text{cov}(A_2, A_1) & \text{cov}(A_2, A_2) \end{pmatrix} \tag{2.11}$$

Keterangan:

A_1, A_2 merupakan varian atribut.

3. Menghitung *eigenvalue*

Eigenvalue merupakan nilai untuk menunjukkan besar keragaman yang dapat dijelaskan oleh variabel *principal component* [53]. Persamaannya sebagai berikut:

$$Mv = \lambda v \tag{2.12}$$

Setiap nilai dari *eigenvalue* harus memenuhi persamaan determinan berikut:

$$|M - \lambda I| = 0 \tag{2.13}$$

Keterangan:

M merupakan kovarian.

v merupakan eigenvector.

λ merupakan eigenvalue.

I merupakan matriks identitas.

4. Menghitung *principal component* (PC)

   Principal component (PC) dapat dihitung jika nilai dari *eigenvalue* dan *eigenvector* sudah didapat, nilai dari PC dihitung dengan mengurutkan nilai *eigen* dari yang terbesar ke terkecil [53].

5. Reduksi Dimensi

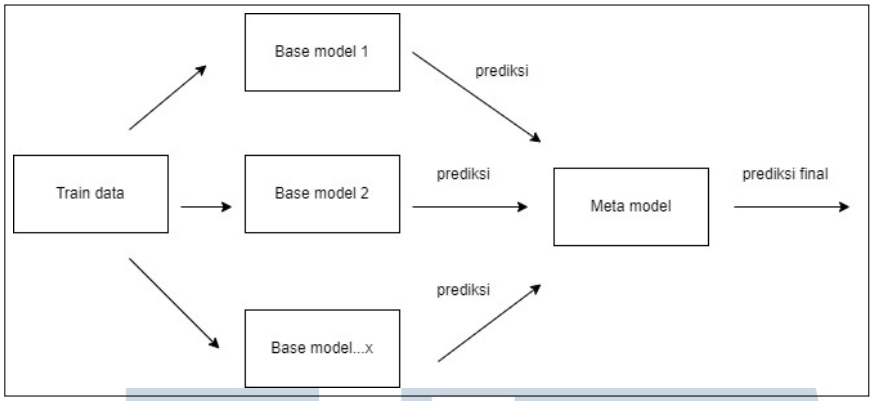
   Pada tahap reduksi tidak semua nilai dapat dipilih. Nilai PC yang dapat diambil harus mempunyai nilai *eigen* >1 atau nilai PC yang memiliki kumulatif lebih dari 75% [53].

2.9 Recursive Feature Elimination

Recursive Feature Elimination (RFE) adalah salah satu teknik seleksi fitur yang digunakan untuk meningkatkan kinerja model prediksi dengan memilih fitur-fitur yang paling relevan. RFE bekerja dengan secara berulang-ulang membangun model dan menghapus fitur yang memiliki kontribusi paling sedikit terhadap kinerja model. Proses ini dilakukan sampai sejumlah fitur yang diinginkan tercapai. Keunggulan utama dari RFE adalah kemampuannya untuk meningkatkan akurasi prediksi dan mengurangi *overfitting* dengan mengeliminasi fitur-fitur yang kurang informatif [54].

2.10 Stacking

Stacking adalah teknik *ensemble learning* yang menggabungkan prediksi dari beberapa model dasar untuk menghasilkan prediksi akhir yang lebih akurat. Proses stacking melibatkan dua lapisan model, model dasar pada lapisan pertama yang memberikan prediksi awal, dan model *meta*-prediktor pada lapisan kedua yang mengambil prediksi dari model dasar sebagai *input* dan memberikan prediksi akhir [55].



Gambar 2.2. Tahap metode *stacking*

Sumber: [56]

Pada Gambar 2.2 model awal (*base* model) akan digunakan untuk mempelajari *dataset*, lalu *base* model tersebut akan digabungkan sehingga membentuk *meta* model yang akan melakukan prediksi akhir [56].

2.11 Gradient Boosting

Gradient Boosting adalah algoritma kuat yang menggabungkan beberapa *weak learner* menjadi *strong learner*, di mana setiap model baru dilatih untuk meminimalkan fungsi kerugian dari model sebelumnya dengan menggunakan *gradient descent* [57]. *Gradient Boosting* adalah salah satu algoritma *ensemble* yang kuat dan sering digunakan untuk tugas regresi dan klasifikasi. Gradient Boosting dapat menangani data yang kompleks dan non-linear serta menghasilkan prediksi yang akurat [58].

2.12 K-Nearest Neighbors

K-Nearest Neighbors (KNN) adalah salah satu algoritma *machine learning* yang sederhana dan banyak digunakan dalam masalah klasifikasi dan regresi. Prinsip dasar dari KNN adalah bahwa suatu objek diklasifikasikan berdasarkan mayoritas suara dari titik data terdekatnya, dengan objek ditugaskan ke kelas yang paling umum di antara k titik data terdekat [59]. K merupakan bilangan bulat positif dalam jumlah kecil. Nilai pada k ini biasanya ganjil, agar tidak ada hasil seri pada

voting titik data terdekat. Rumus *K-Nearest Neighbors* sebagai berikut [60].

$$d_{(a,b)} = \sqrt{\sum_{g=1}^{p} \left(x_{ag} - x_{bg}\right)^2} \tag{2.14}$$

Keterangan:

$d_{(a,b)}$ merupakan kovarian.
x_{ag} merupakan eigenvector.
x_{bg} merupakan eigenvalue.
p merupakan matriks identitas.

2.13 Grid Search

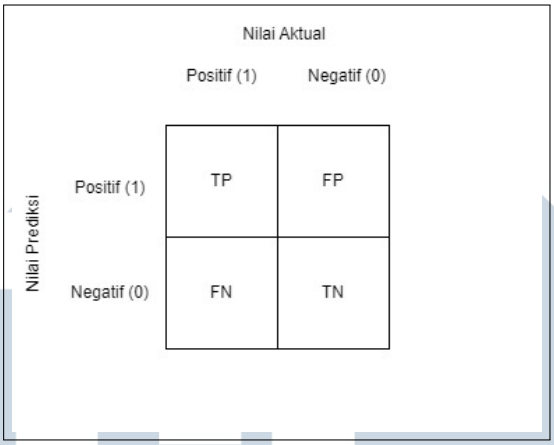
Grid search merupakan sebuah teknik untuk melakukan optimasi *hyperparameter* (*hyperparameter tunning*) untuk meningkatkan performa dari kinerja model [61]. *Grid search* mempunyai fitur yang terdapat pada sklearn yang digunakan untuk melakukan *cross validation* yang disebut sebagai *GridSearchCV* [62]. *Grid search* menguji setiap kombinasi dari *hyperparameter* yang telah ditentukan. Setelah seluruh kombinasi telah di uji coba maka *grid search* akan menghasilkan *best parameter* yang merupakan hasil terbaik dari tahap uji coba kombinasi *hyperparameter* yang telah dilakukan [63].

2.14 Evaluasi Model

Evaluasi model merupakan teknik atau metode yang berfungsi untuk mengukur kinerja model yang dipakai dalam memberikan hasil yang tepat [64]. Macam-macam metode pengukuran yang dipakai dalam evaluasi model adalah.

2.14.1 Confusion Matrix

Confusion Matrix merupakan matriks yang digunakan untuk mengetahui jumlah dari nilai yang diprediksi dan nilai aktual [65]. *Confusion Matrix* digambarkan seperti berikut.



Gambar 2.3. *Confusion matrix*

Sumber: [64]

Pada contoh kasus prediksi diabetes nilai yang diprediksi dan nilai aktual dalam empat bagian dari *confusion matrix* yaitu.

- TN merupakan *True Negative* yang menunjukkan jumlah nilai prediksi negatif dan jumlah nilai aktual negatif [65]. Pada contoh kasus prediksi diabetes, model memprediksi seseorang tidak terkena diabetes, dan ternyata hasil tes benar menunjukkan tidak terkena diabetes.
- TP merupakan *True Positif* yang menunjukkan jumlah nilai prediksi positif dan jumlah nilai aktual positif [65]. Pada contoh kasus prediksi diabetes, model memprediksi seseorang terkena diabetes, dan ternyata hasil tes benar menunjukkan terkena diabetes.
- FP merupakan *False Positive* yang menunjukkan jumlah nilai prediksi positif dan jumlah nilai aktual negatif [65]. Pada contoh kasus prediksi diabetes, model memprediksi seseorang terkena diabetes, dan ternyata hasil tes menunjukkan tidak terkena diabetes.
- FN merupakan *False Negatif* yang merupakan jumlah nilai prediksi negatif dan jumlah nilai aktual positif [65]. Pada contoh kasus prediksi diabetes, model memprediksi seseorang tidak terkena diabetes, dan ternyata hasil tes menunjukkan terkena diabetes.

2.14.2 Akurasi

Akurasi merupakan suatu teknik yang berfungsi untuk mengukur seberapa akurat model yang dipakai saat menghasilkan prediksi dari data [65]. Persamaan akurasi sebagai berikut.

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.15}$$

2.14.3 Recall

Recall merupakan suatu teknik yang digunakan untuk mengukur seberapa baik kinerja model dalam hasil prediksi positif [65]. Persamaan *Recall* sebagai berikut.

$$Recall = \frac{TP}{TP + FN} \tag{2.16}$$

2.14.4 Precission

Precission merupakan suatu teknik yang digunakan untuk mengukur keakuratan model dalam prediksi nilai positif (*positive value*) [65]. Persamaan *Precission* sebagai berikut.

$$Precission = \frac{TP}{TP + FP} \tag{2.17}$$

2.14.5 F1-Score

F1-*Score* merupakan teknik yang digunakan untuk mengambarkan keseimbangan antara *Precission* dan *Recall* [65]. Persamaan F1 *Score* sebagai berikut.

$$F1 - Score = 2 \times \frac{\text{Recall} \times \text{Precission}}{\text{Recall} + \text{Precission}} \tag{2.18}$$

2.14.6 Receiver Operating Characteristic Curve

Receiver Operating Characteristic (ROC) *Curve* adalah kurva yang digunakan untuk mengevaluasi kinerja model klasifikasi. Kurva ROC

menggambarkan hubungan antara *True Positive Rate* (TPR) dan *False Positive Rate* (FPR) pada berbagai ambang batas klasifikasi [66].