

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 *Markov Chain*

Rantai Markov, juga dikenal sebagai Markov chains, adalah alat matematika yang secara luas digunakan untuk memodelkan dinamika sistem dan proses bisnis dengan menganalisis transisi antar keadaan berdasarkan data historis dari variabel-variabel yang dinamis. Dengan kemampuannya untuk memprediksi perubahan di masa depan dari variabel tersebut, rantai Markov tidak hanya menyediakan prediksi tetapi juga memberikan kerangka analisis matematis yang mendalam untuk memahami dan mengelola peristiwa kompleks yang mungkin terjadi.

Sebagai pendekatan krusial dalam riset operasional dan pengambilan keputusan manajerial, rantai Markov digunakan dalam berbagai konteks. Contohnya, dalam pemasaran, alat ini digunakan untuk memahami bagaimana preferensi merek atau perilaku pelanggan berubah dari waktu ke waktu. Di bidang keuangan, rantai Markov diterapkan untuk model perubahan harga saham dan perilaku pasar keuangan. Selain itu, dalam industri layanan seperti perawatan mesin atau manajemen persediaan, rantai Markov membantu dalam meramalkan kebutuhan perawatan atau pengelolaan persediaan berdasarkan pola historis dari variabel yang relevan. Semua aplikasi ini adalah sebagian kecil dari berbagai cara di mana rantai Markov memberikan wawasan yang berharga dan prediksi yang berguna di berbagai bidang ilmu. [3]

Rantai markov merupakan proses stokastik. Proses stokastik adalah sebagai suatu himpunan variabel acak  $X(t)$  dimana  $t$  diambil dari himpunan data yang telah diketahui ( $T$ ) yang umumnya terdiri dari bilangan bulat positif, setiap variabel  $X(t)$  mencerminkan karakteristik yang dapat diukur pada saat  $t$ . Dalam konteksnya, ini merujuk pada proses stokastik di mana perubahan variabel-variabel ini dipantau dan dipahami berdasarkan pengamatan pada titik waktu yang spesifik. Untuk memperdalam pemahaman, perlu dipelajari hukum probabilitas yang mengatur setiap variabel acak yang menjadi bagian dari himpunan tersebut.. dapat dijelaskan secara lengkap bila untuk  $n = 0, 1, 2, \dots$  dan untuk setiap  $t_0 \leq t_1 \leq \dots \leq t_n$  [3]

$$P[X(t_n) = x_n | X(t_{n-1}) = x_{n-1}, \dots, X(t_0) = x_0] \quad (1)$$

Probabilitas ini mengindikasikan bahwa pada waktu atau langkah ke-n+1, pengaruh hanya berasal dari langkah ke-n (sifat Markov), tanpa mempertimbangkan langkah-langkah sebelumnya.

### 2.1.1 Matriks Probabilitas Transisi

Untuk menunjukkan kemungkinan perpindahan antara berbagai merek, termasuk kembali ke merek yang sama, matriks probabilitas transisi dibuat dari elemen matematis. Untuk menghitung nilai-nilai dalam matriks ini, proporsi dari semua kemungkinan perpindahan merek yang terjadi selama periode pengamatan tertentu diperhitungkan. Berdasarkan data yang dikumpulkan, proses ini menunjukkan bagaimana transisi antar-merek terjadi. Dalam formulasi yang tepat, pergeseran dari merek i ke merek j selama periode t disebut  $P_{ij}(t)$ . Ini menunjukkan rasio antara jumlah pergeseran dari merek i ke merek j selama periode t dengan jumlah total produk yang berada di merek i pada awal periode t. Persamaan yang disebutkan di atas dapat digambarkan sebagai berikut: [4]

$$P_{ij} = \frac{n_{ij}(t)}{n_i(t)} \quad (2)$$

Secara matriks, probabilitas transisi satu langkah dan probabilitas transisi m langkah dapat ditulis :at didefinisikan sebagai berikut:

$$P = \begin{bmatrix} P_{00} & P_{01} & P_{02} & \dots & P_{0j} \\ P_{10} & P_{11} & P_{12} & \dots & P_{1j} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ P_{i0} & P_{i1} & P_{i2} & \dots & P_{ij} \end{bmatrix}$$

$$P^{(m)} = \begin{bmatrix} P_{00}^{(m)} & P_{01}^{(m)} & P_{02}^{(m)} & \dots & P_{0j}^{(m)} \\ P_{10}^{(m)} & P_{11}^{(m)} & P_{12}^{(m)} & \dots & P_{1j}^{(m)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ P_{i0}^{(m)} & P_{i1}^{(m)} & P_{i2}^{(m)} & \dots & P_{ij}^{(m)} \end{bmatrix}$$

**Gambar 2.1** Matriks Probabilitas Transisi m

### 2.1.2 Vektor Keadaan (State Vector)

Vektor state dalam konteks rantai Markov merupakan representasi matematis dari kondisi sistem pada waktu tertentu, yang terdiri dari beberapa state atau keadaan yang mungkin ada. Vektor ini menggambarkan probabilitas distribusi

sistem berada dalam setiap state pada waktu  $t$ , diwakili oleh sejumlah elemen dalam vektor tersebut, sesuai dengan jumlah total state yang ada. Hal ini memberikan gambaran yang jelas mengenai kondisi sistem pada saat tertentu, dengan memperhitungkan proses transisi antar-state yang terjadi sesuai dengan aturan Markov.

$$x = [x_1, x_2, x_3, \dots, x_n] \quad (3)$$

dimana,

$x_1$  adalah peluang sistem tersebut berada pada state 1.

$x_2$  adalah peluang sistem tersebut berada pada state 2.

$x_n$  adalah peluang sistem tersebut berada pada state  $n$ .

### 2.1.3 Peluang Transisi n-langkah

Probabilitas transisi n-langkah  $P_{ij}^n$  merujuk pada kemungkinan bahwa sistem, setelah melakukan  $n$  transisi tambahan, akan berpindah dari state  $i$  ke state  $j$  dalam sebuah rantai Markov. Konsep ini didasarkan pada sifat probabilitas dari proses Markov, di mana perubahan state pada langkah selanjutnya hanya bergantung pada state saat ini dan tidak dipengaruhi oleh sejarah state sebelumnya. Dengan memahami  $P_{ij}^n$  analisis dapat memprediksi dengan lebih akurat bagaimana perubahan jangka panjang sistem akan memengaruhi distribusi probabilistik dari berbagai state dalam jangka waktu tertentu.  $P_{ij}^n$  disebut peluang transisi n-langkah dari state  $i$  ke state  $j$ .

Probabilitas transisi n-langkah  $P_{ij}^n$  merujuk pada kemungkinan sistem berpindah dari state  $i$  ke state  $j$  setelah melakukan  $n$ -langkah, dimulai dari state  $i$  pada waktu  $t$ . Ini dijelaskan dalam bentuk matriks  $P^n$  di mana setiap elemennya  $P_{ij}^n$  mewakili nilai probabilitas transisi tersebut. Yang dapat ditulis sebagai berikut:

$$P^n = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ m \end{bmatrix} \times \begin{bmatrix} P_{00}^n & P_{01}^n & \dots & P_{0j}^n \\ P_{10}^n & P_{11}^n & \dots & P_{1j}^n \\ \vdots & \vdots & \ddots & \vdots \\ P_{m0}^n & P_{m1}^n & P_{m2}^n & P_{mn}^n \end{bmatrix}$$

**Gambar 2.2** Matriks Peluang Transisi n Langkah

Dalam konteks rantai Markov, ketika  $n = 1$ , kita tahu bahwa  $P_{ij}^n$  yang merupakan peluang transisi satu-langkah dari state  $i$  ke state  $j$  sama dengan  $P_{ij}$  yakni peluang transisi dari state  $i$  ke state  $j$ . Karakteristik probabilistik dari  $P_{ij}^n$  memastikan bahwa nilainya selalu non-negatif, menggambarkan kemungkinan transisi antar state dalam proses rantai Markov. Karena sifat dasar rantai Markov mengharuskan adanya kemungkinan transisi antar state, maka  $P_{ij}^n \geq 0$  untuk semua  $i$  dan  $j$ . Aturan normalisasi menyatakan bahwa total probabilitas transisi dari state  $i$  harus sama dengan 1, yaitu  $\sum_{j=1}^m P_{ij}^n = 1$ . Ini menunjukkan bahwa rantai Markov mempertahankan konsistensi dalam distribusi probabilitas transisi antar state, memastikan bahwa semua peluang terdistribusi dengan benar sesuai dengan prinsip probabilitas.

$P_{ik}^n P_{kj}^m$  menggambarkan probabilitas bahwa, ketika memulai dari state  $i$ , proses akan berada di state  $j$  setelah melalui tepat  $n + m$  dengan proses pertama-tama melakukan  $n$  langkah awal diikuti  $m$  langkah. Dengan memanfaatkan sifat komutatif dalam perkalian probabilitas, kita dapat menyimpulkan bahwa untuk setiap state  $k$  ini menjelaskan probabilitas bahwa proses akan mencapai state  $j$  setelah  $n + m$  langkah.

Jika kita menggunakan  $P^n$  untuk menunjukkan matriks peluang transisi  $n$ -langkah  $P_{ik}^n$ , maka persamaan  $P^{(n+m)} = P^n \cdot P^m$  menjelaskan bahwa matriks peluang transisi untuk  $n+m$  langkah yaitu  $P^{(n+m)}$  dapat diperoleh dengan mengalikan matriks peluang transisi  $n$ -langkah  $P^n$  dengan matriks peluang transisi  $m$ -langkah  $P^m$ . Dengan demikian, kita dapat menghitung matriks peluang transisi  $n$ -langkah  $P^n$  dengan melakukan perkalian berulang kali dari matriks peluang transisi satu langkah  $P$ . Proses ini sangat penting karena memungkinkan kita untuk memahami bagaimana sistem bertransisi dari satu keadaan ke keadaan lain dalam jangka waktu yang lebih panjang, hanya dengan menggunakan informasi dari matriks peluang transisi satu langkah. Oleh karena itu, perkalian berulang ini memberikan cara yang efisien untuk memodelkan dan menganalisis perilaku sistem dalam periode waktu yang lebih panjang, membantu kita dalam memprediksi perubahan keadaan sistem dengan lebih akurat. [5]

#### 2.1.4 Probabilitas Steady-State

Dalam situasi ini, proses Markov pada akhirnya akan mencapai keadaan keseimbangan. Setelah beberapa waktu, probabilitas setiap status akan tetap konstan. Keadaan ini disebut sebagai kemungkinan keadaan tetap. Matriks transisi stokastik ganda disebut jika jumlah dari setiap kolomnya sama dengan satu. Untuk matriks transisi stokastik ganda dengan jumlah status  $m$ , probabilitas steady state untuk masing-masing status adalah  $1/m$ . Dalam analisis sistem Markov, distribusi probabilitas steady state sangat penting karena menunjukkan bagaimana sistem akan berperilaku dalam jangka panjang dan memberikan informasi tentang proporsi. Ini juga memberikan gambaran tentang probabilitas jangka panjang dari setiap status dalam sistem Markov, yang menunjukkan berapa lama atau seberapa sering sistem akan berada dalam setiap status setelah pengaruh kondisi awal hilang.

$$\begin{aligned}\pi_j &= \lim_{n \rightarrow \infty} p_{ij}(n) \\ &= \lim_{n \rightarrow \infty} P(X_n = j)\end{aligned}\tag{4}$$

Dalam konteks rantai Markov,  $\pi_j$  merujuk pada probabilitas steady-state. Probabilitas steady-state mengindikasikan kemungkinan bahwa sistem akan berada pada status  $j$  dalam jangka panjang setelah mencapai keseimbangan. Matriks transisi dianggap stasioner jika peluang untuk status  $j$  yaitu  $\pi_j$ , tetap konstan seiring waktu berlalu. Dengan kata lain,  $P(X_0 = j) = \pi_j$ , untuk semua  $j$  dan peluang bahwa suatu proses ditemukan dalam status  $j$  pada waktu  $n = 1, 2, 3, \dots$ , juga sama dengan  $\pi_j$  atau  $P(X_0 = j) = \pi_j$ . Ini menunjukkan bahwa setelah sistem mencapai kondisi steady-state, distribusi probabilitasnya tidak berubah meskipun proses terus berlanjut, dan probabilitas steady-state untuk setiap status  $j$  tetap  $\pi_j$  secara konsisten. [6]

#### 2.1.5 Perkalian Matriks

Perkalian matriks adalah sebuah operasi fundamental dalam matematika yang melibatkan penggunaan dua atau lebih matriks. Matriks, yang merupakan suatu struktur data terdiri dari baris dan kolom, diidentifikasi dan dinotasikan dengan simbol-simbol khusus. Proses perkalian matriks  $m$  dengan  $n$  dilambangkan sebagai  $m \times n$ , atau sering disingkat menjadi  $mn$ . Pentingnya operasi ini terletak

pada persyaratan bahwa jumlah kolom dalam matriks pertama (m) harus sama dengan jumlah baris dalam matriks kedua (n) untuk dapat dilakukan. Dalam hasil perkalian matriks  $c = mn$ , setiap elemen  $c_{ij}$ , yaitu elemen yang terletak pada baris ke-i dan kolom ke-j dari matriks hasil c, diperoleh dengan menjumlahkan hasil dari perkalian setiap elemen pada baris ke-i dari matriks m dengan elemen-elemen yang sesuai pada kolom ke-j dari matriks n.

$$c_{ij} = \sum_k m_{ik}n_{kj} \quad (5)$$

Keterangan:

$c_{ij}$  = adalah elemen hasil dari perkalian matriks, yang merupakan jumlah dari perkalian elemen-elemen pada baris ke-i dari matriks pertama dengan elemen-elemen pada kolom ke-j dari matriks kedua

m = matriks yang pertama diberi label m

n = Matriks yang kedua diberi label

i = mengacu pada baris dari matriks.

j = mengacu pada kolom dari matriks

Beberapa kelebihan dan kekurangan dari algoritma Markov Chain adalah:

Kelebihan:

1. Lebih mudah digunakan
2. Lebih simpel dibandingkan data mining yang lain
3. Pengolahan data yang besar
4. Memberikan output hasil optimal

Kekurangan :

1. State Independen Sepanjang waktu
2. Penggunaan *stochastic* yang hanya mempunyai karakter bahwa terjadinya suatu state pada suatu saat bergantung pada dan hanya pada state yang bisa digunakan
3. Terbatas pada beberapa kasus tertentu

## 2.2 Naive Bayes

Metode klasifikasi yang dikenal sebagai Algoritma Naive Bayes didasarkan pada teorema probabilitas Bayes dan diberi nama oleh ilmuwan Inggris Thomas Bayes. Algoritma ini berfungsi dengan menggunakan pengalaman masa lalu untuk memprediksi kemungkinan apa yang akan terjadi di masa depan. Naive Bayes berpendapat dalam konteks klasifikasi bahwa setiap karakteristik dari sebuah kelas secara kondisional berbeda dari kelas lainnya. Dengan kata lain, keberadaan atau tidaknya suatu karakteristik dalam sebuah kelas diasumsikan tidak memiliki korelasi dengan karakteristik lain dalam kelas yang sama.

Dengan pendekatan ini, Naive Bayes menggunakan informasi probabilitas dari atribut yang ada untuk menghitung kemungkinan kelas yang paling mungkin terjadi. Meskipun asumsi bahwa atribut-atribut tersebut saling bebas tidak selalu terpenuhi di dunia nyata, Naive Bayes sering memberikan hasil yang efisien dan efektif dalam berbagai tugas klasifikasi, terutama dalam analisis teks dan klasifikasi dokumen. [7]

Berikut adalah rumus persamaan teorema Bayes:

$$P(H|X) = \frac{P(X|H) \times P(H)}{P(X)} \quad (6)$$

Penjelasan dari persamaan (1) sebagai berikut:

X : Data yang memiliki kelas atau label yang belum diketahui

H : Hipotesis yang menggambarkan kelas atau label tertentu untuk data X

P(H|X) : Probabilitas bahwa hipotesis H benar, diberikan data X. Ini merupakan nilai probabilitas posterior yang ingin diprediksi.

P(H) : Probabilitas prior dari hipotesis H, tanpa mempertimbangkan data X.

P(X|H) : Probabilitas dari data X diberikan hipotesis H. Ini merupakan nilai likelihood yang menggambarkan seberapa mungkin data X akan muncul jika hipotesis H benar

P(X) : Probabilitas dari data X secara keseluruhan, tanpa mempertimbangkan hipotesis tertentu.

Dalam menjelaskan teorema Naive Bayes, perlu dipahami bahwa proses klasifikasi menggunakan karakteristik atau petunjuk untuk menentukan kelas yang

tepat untuk sampel yang dianalisis. Dalam hal ini, teorema Bayes, dasar dari Naive Bayes, sangat penting. diubah menjadi seperti berikut:

$$P(C|F_1 \dots F_n) = \frac{P(C)P(F_1 \dots F_n|C)}{P(F_1 \dots F_n)} \quad (7)$$

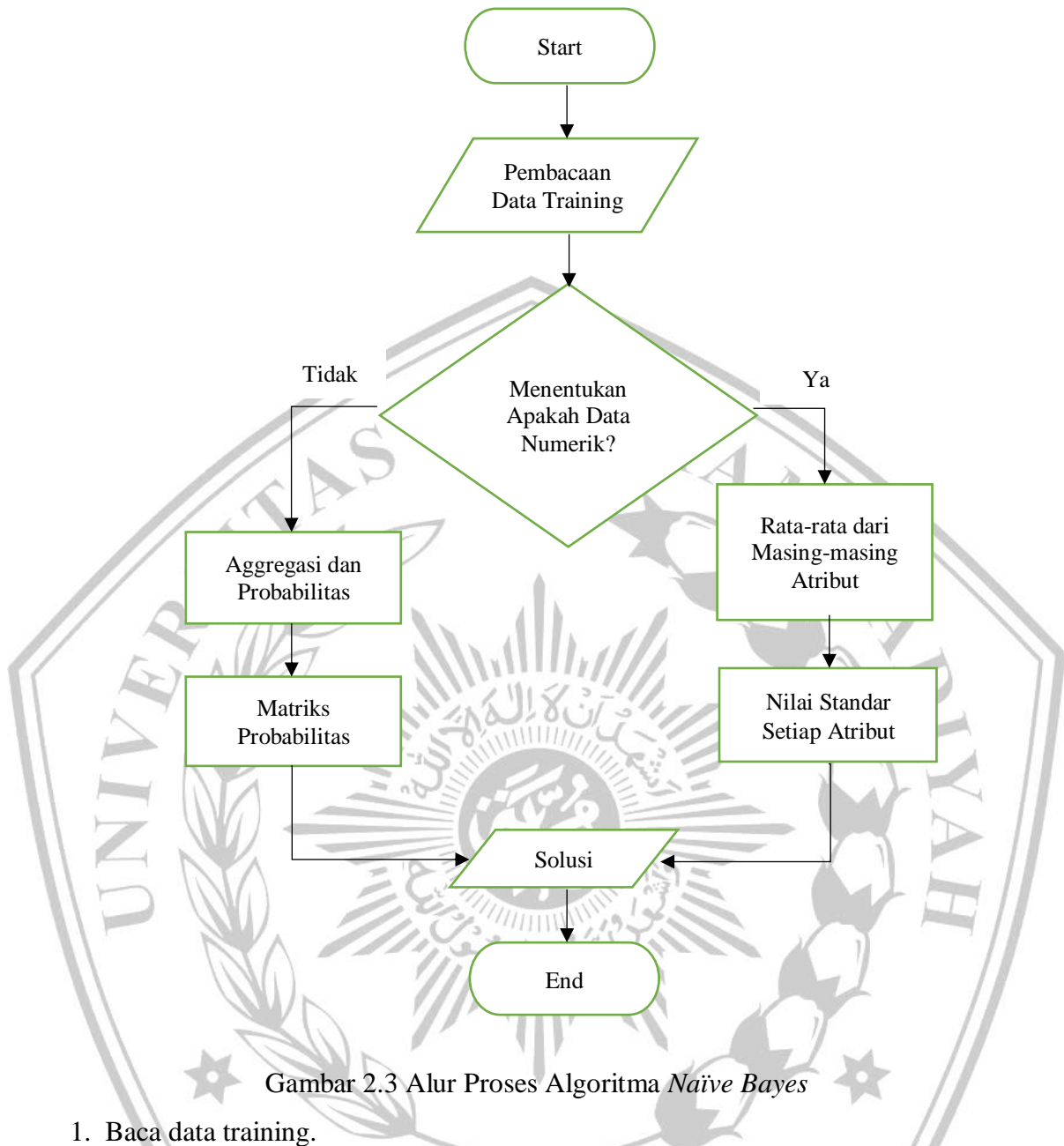
Pada persamaan (2) variabel C merepresentasikan kelas, sementara variabel  $F_1 \dots F_n$  merepresentasikan karakteristik petunjuk yang dibutuhkan untuk melakukan klasifikasi.

Variabel C menunjukkan kelas dalam Persamaan (2), sedangkan variabel  $F_1 \dots F_n$  menunjukkan fitur atau atribut yang digunakan dalam proses klasifikasi. Rumus tersebut menjelaskan bahwa kemungkinan masuknya sampel dengan karakteristik tertentu ke dalam kelas C (posterior) dapat dihitung dengan mengalikan kemungkinan awal kelas C sebelum sampel muncul (prior), mengalikan kemungkinan karakteristik dalam kelas C muncul (likelihood), dan kemudian dibagi dengan kemungkinan total kemunculan karakteristik sampel (evidence). Ini adalah ide utama Naive Bayes untuk mengestimasi probabilitas kelas yang paling mungkin berdasarkan karakteristik yang diamati dari sampel.

$$Posterior = \frac{Prior \times likelihood}{evidence} \quad (8)$$

Nilai-nilai bukti di atas selalu tetap untuk setiap kelas dalam satu sampel. Nilai-nilai posterior kemudian akan dibandingkan dengan nilai-nilai bukti kelas lainnya untuk menentukan ke kelas apa suatu sampel akan diklasifikasikan. Adapun alur proses algoritma Naive Bayes sebagai berikut:





Gambar 2.3 Alur Proses Algoritma *Naïve Bayes*

1. Baca data training.
2. Hitung Jumlah dan probabilitas, namun apabila data numerik maka:
  - a. Cari nilai mean dan standar deviasi dari masing-masing parameter yang merupakan data numerik.
  - b. Cari nilai probabilistik dengan cara menghitung jumlah data yang sesuai dari kategori yang sama dibagi dengan jumlah data pada kategori tersebut.
3. Mendapatkan nilai dalam tabel mean, standart deviasi dan probabilitas.
4. Didapatkan solusi.

Beberapa kelebihan dan kekurangan dari algoritma Naive Bayes adalah:

Kelebihan:

1. Model yang sederhana
2. Bisa digunakan untuk model kualitatif dan kuantitatif
3. Mempunyai akurasi tinggi
4. Kecepatan pelatihan yang tinggi

Kekurangan:

1. Menerapkan asumsi independensi bersyarat
2. Masalah probabilitas nol
3. Estimator yang buruk karena semua fitur yang ada diasumsikan sebagai fitur independen

### 2.3 Pengujian Akurasi Klasifikasi

Sebuah sistem klasifikasi diharapkan dapat mengklasifikasikan semua data secara akurat, meskipun mencapai tingkat kebenaran yang absolut sering tidak memungkinkan. Oleh karena itu, penting untuk mengevaluasi kinerja setiap sistem klasifikasi menggunakan matriks konfusi (confusion matrix) [8]

Matriks konfusi merekam hasil dari proses klasifikasi, memberikan gambaran tentang berapa banyak data yang diklasifikasikan dengan benar (true positive dan true negative) serta berapa banyak yang salah diklasifikasikan (false positive dan false negative). Informasi ini krusial untuk menilai seberapa baik sistem dapat membedakan kelas-kelas yang berbeda dan seberapa sering kesalahan terjadi dalam proses klasifikasi.

$$Akurasi = \frac{\text{Jumlah data yang diprediksi benar}}{\text{Jumlah data yang diprediksi}} \quad (9)$$

atau,

$$Accuracy = \frac{tp+tn}{(tp+tn+fp+fn)} \quad (10)$$

Untuk mengklarifikasi tingkat diagnosis:

tp = Perkiraan positif yang benar.

tn = Perkiraan negatif yang benar.

fp = Perkiraan positif yang salah.

fn = Perkiraan negatif yang salah.

Untuk mengekspresikan *confusion matrix* maka memiliki tingkat nilai diagnosa.[9]

Tingkat diagnosa berdasarkan akurasi adalah sebagai berikut:

- a. Klasifikasi yang baik menunjukkan sangat baik: Akurasi antara 0,90 - 1,00.
- b. Klasifikasi yang baik menunjukkan sangat baik: Akurasi antara 0,80 - 0,90.
- c. Klasifikasi yang baik menunjukkan wajar: Akurasi antara 0,70 - 0,80.
- d. Klasifikasi yang baik menunjukkan buruk: Akurasi antara 0,60 - 0,70.

