

## **BAB 3**

### **METODOLOGI PENELITIAN**

#### **3.1 Objek Penelitian**

Objek penelitian ini data Demam Berdarah *Dengeu* (DBD). Data tersebut selanjutnya diolah menggunakan perangkat lunak RapidMiner sebagai himpunan data (*dataset*) dan data uji (*data testing*) untuk dilakukan klasifikasi menggunakan algoritma data mining

#### **3.2 Alat dan Bahan**

Alat merupakan sesuatu yang digunakan untuk melakukan penelitian berupa benda, sedangkan bahan adalah sesuatu yang diperlukan dan merupakan bagian dari sesuatu yang akan dibuat. Alat yang digunakan dalam penelitian ini adalah:

- a. Perangkat keras (*hardware*) yang terdiri dari seperangkat komputer (laptop) dengan spesifikasi Processore Core i-5, VGA Nvidia GeForce 750m, RAM 8Gb, SSD 500Gb
- b. Perangkat lunak (*software*) yang digunakan (1) RapidMiner studio, (2) Microsoft Excel (3) *Software Reference Manager* Mendeley Desktop (4) Browser Google Chrome
- c. Bahan penelitian menggunakan *dataset* DBD dari Puskesmas Taman Krocok Kabupaten Bondowoso – Jawa Timur tahun 2014-2022 (Jasri et al., 2022).

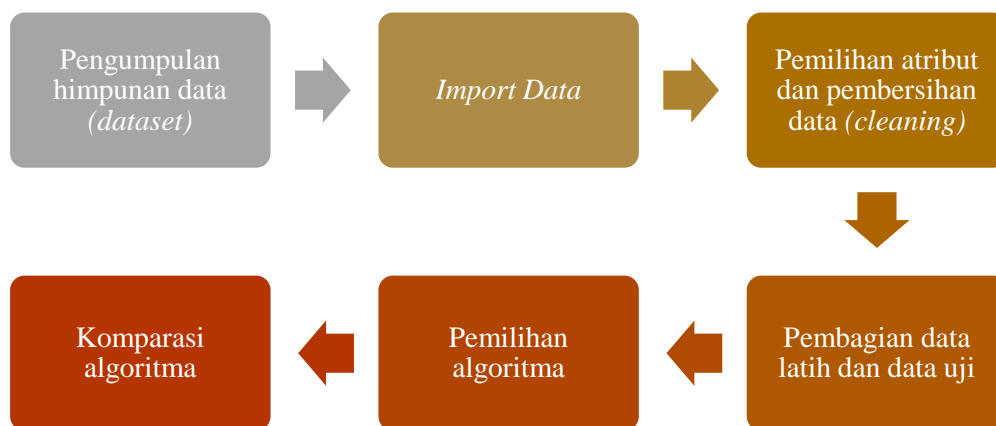
#### **3.3 Metode Pengumpulan Data**

Pengumpulan data dilakukan untuk memperoleh informasi yang dibutuhkan dalam rangka mencapai tujuan penelitian. Tujuan yang diungkapkan dalam bentuk hipotesis merupakan jawaban sementara terhadap pertanyaan penelitian. Jawaban itu masih perlu diuji secara empiris dan untuk maksud inilah dibutuhkan pengumpulan data. Data yang dikumpulkan oleh sampel yang telah ditentukan sebelumnya. Sampel tersebut terdiri atas sekumpulan unit analisis sebagai sasaran penelitian. Beberapa metode pengumpulan data yang dilakukan sebagai berikut :

- a. Tinjauan studi yaitu melakukan pengumpulan data dengan cara membaca, mengutip, beberapa teori dari buku maupun artikel penelitian sebelumnya
- b. Pengambilan data sekunder yaitu menggunakan himpunan data yang dibagi menjadi 2 (dua) bagian yaitu data latih (*data training*) dan data uji (*data testing*) dari penelitian sebelumnya

### 3.4 Tahapan Penelitian

Tahapan penelitian merupakan langkah-langkah yang dilakukan dalam melaksanakan penelitian. Tahapan tersebut terdapat memiliki proses yang dilakukan secara terstruktur, runtut, baku, logis dan sistematis. Alur tahapan penelitian ini dijelaskan pada Gambar 3.1.



**Gambar 3. 1** Tahapan Penelitian

#### 3.4.1 Pengumpulan Dataset

Pada penelitian ini menggunakan data himpunan diagnosis DBD sebanyak 337 data pasien dari Puskesmas Taman Krocok Kabupaten Bondowoso – Jawa Timur yang diambil dari tahun 2014 sampai 2022 (Jasri et al., 2022) yang disajikan pada Tabel 3.1.

**Tabel 3.1.** *Sample* Data Himpunan Diagnosis DBD

No	Nama	Usia	Jenis Kelamin	Suhu Tubuh	Bintik Merah	Sakit Kepala	Tubuh Lemas	Trombosit	Diagnosa (Class)
1	Suparman	35	Laki-laki	39	0	Tidak	Ya	130.000	Negatif
2	Anggita	5	Perempuan	39	1	Ya	Ya	90.000	Positif
3	Silvia	25	Perempuan	40	4	Ya	Ya	45.000	Positif
4	Ulfatun nazihah	21	Perempuan	38	3	Ya	Tidak	48.000	Positif
5	Cindy aulia putri	10	Perempuan	40	2	Tidak	Ya	43.000	Positif

No	Nama	Usia	Jenis Kelamin	Suhu Tubuh	Bintik Merah	Sakit Kepala	Tubuh Lemas	Trombosit	Diagnosa (Class)
6	Belqis aprilianti	8	Perempuan	38	3	Ya	Ya	41.000	Positif
7	Zainal arifin	15	Laki-laki	39	1	Ya	Ya	76.000	Positif
8	Muhammad ali	35	Laki-laki	39	0	Tidak	Tidak	122.000	Negatif
9	Ahmad effendi	30	Laki-laki	39	0	Ya	Tidak	148.000	Negatif
10	Indah nur umami	23	Perempuan	39	0	Tidak	Tidak	138.000	Negatif
11	Rafikatul hasanah	21	Perempuan	37	0	Tidak	Ya	144.000	Negatif
12	Diana putri	22	Perempuan	41	3	Ya	Ya	40.000	Positif
13	Muhammad anshari	18	Laki-laki	39	1	Ya	Ya	90.000	Positif
14	Supriyadi	22	Laki-laki	39	2	Ya	Ya	46.000	Positif
15	Muhammad ghafur	23	Laki-laki	39	2	Tidak	Ya	39.000	Positif
16	Samsul arifin	25	Laki-laki	38	1	Tidak	Ya	67.000	Positif
17	Novita sari	11	Perempuan	39	0	Tidak	Ya	101.000	Negatif
18	Maghfirah	19	Perempuan	39	0	Ya	Tidak	110.000	Negatif
19	Nikmatul husna	5	Perempuan	40	4	Ya	Ya	38.000	Positif
20	Rahmatullah	26	Laki-laki	39	1	Ya	Ya	75.000	Positif
21	Rafi'atul maghfiroh	10	Perempuan	39	0	Ya	Tidak	110.000	Negatif

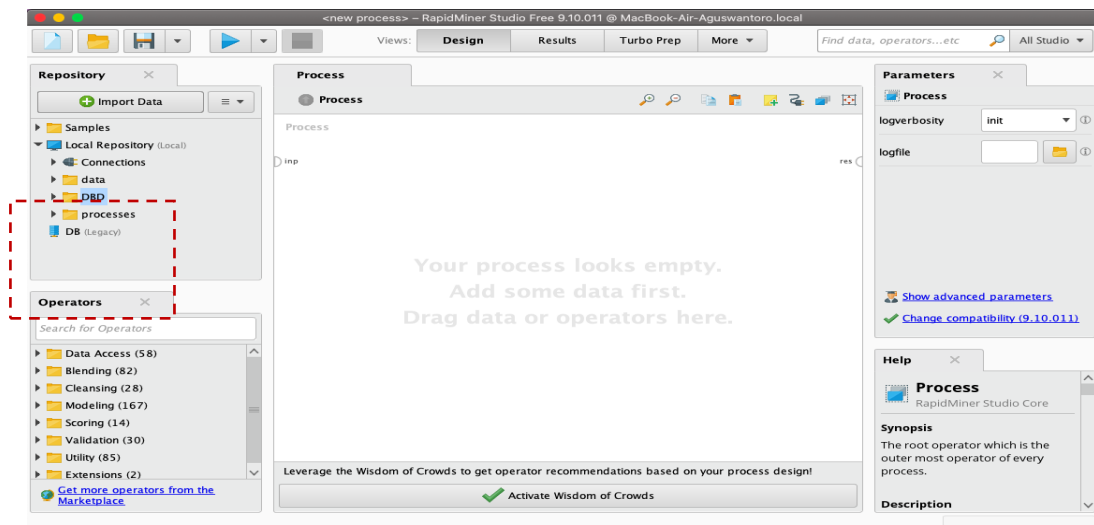
Sumber : <https://garuda.kemdikbud.go.id/documents/detail/3154265>

Keterangan :

- Bintik Merah 0 (Tidak ada)
- Bintik Merah 1 (Sedikit)
- Bintak Merah 2 (Cukup Banyak)
- Bintik Merah 3 (Banyak)
- Bintik Merah 4 (Sangat Banyak)

### 3.4.2 Memasukkan (*import*) Data

Tahap ini memasukkan data kedalam folder yang telah dibuat sebelumnya. Data diambil dari local komputer. Pada dataset terdapat 2 (dua) class atau label yaitu Positif dan Negatif yang telah diverifikasi oleh Pakar. Dari 337 data, jumlah *class* “Positif” sebanyak 252 data dan *class* “Negatif” sebanyak 85 data. Tahapan *import* data dapat dilihat pada Gambar 3.2.



Gambar 3. 2 Import dataset

### 3.4.3 Pemilihan (*selection*) Atribut

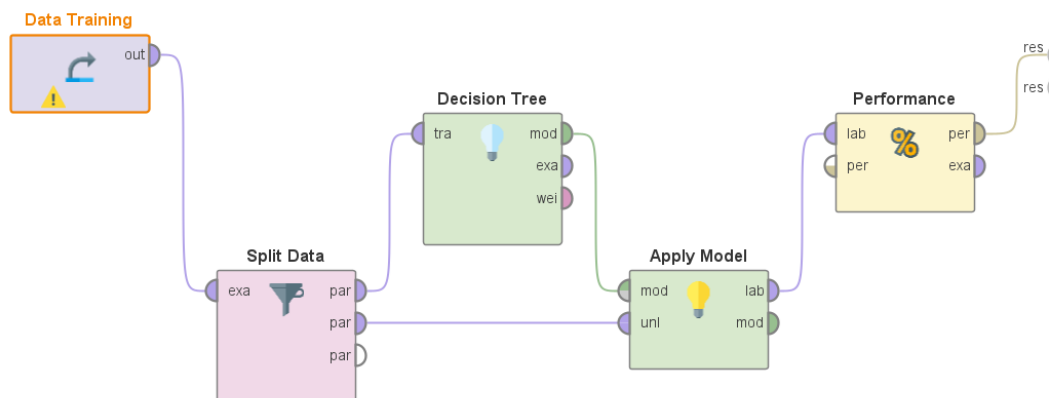
Pemilihan atribut dilakukan untuk menentukan atribut yang akan digunakan dalam proses klasifikasi. Beberapa atribut yang tidak digunakan seperti Nama Pasien, Usia, dan Jenis Kelamin. Selanjutnya, tahap penentuan atribut identitas (*id*) dan atribut *class (label)* serta penentuan tipe data pada masing-masing atribut. Hasil atribut yang telah dipilih dan proses penentuan atribut (*id*) dan tipe data ditampilkan pada Gambar 3.3.

	Nama polinomial id	Suhu Tubuh integer	Bintik Merah integer	Sakit Kepala polinomial	Tubuh Le... polinomial	Trombosit integer	Nyeri Otot polinomial	Diagnosa polinomial
1	P-001	39	0	Tidak	Ya	130000	Tidak	Negatif
2	P-002	39	1	Ya	Ya	90000	Ya	Positif
3	P-003	40	4	Ya	Ya	45000	Ya	Positif
4	P-004	38	3	Ya	Tidak	48000	Ya	Positif
5	P-005	40	2	Tidak	Ya	43000	Ya	Positif
6	P-006	38	3	Ya	Ya	41000	Ya	Positif
7	P-007	39	1	Ya	Ya	76000	Ya	Positif
8	P-008	39	0	Tidak	Tidak	122000	Ya	Negatif
9	P-009	39	0	Ya	Tidak	148000	Tidak	Negatif
10	P-010	39	0	Tidak	Tidak	138000	Tidak	Negatif
11	P-011	37	0	Tidak	Ya	144000	Ya	Negatif
12	P-012	41	3	Ya	Ya	40000	Tidak	Positif
13	P-013	39	1	Ya	Ya	90000	Ya	Positif
14	P-014	39	2	Ya	Ya	46000	Ya	Positif
15	P-015	39	2	Tidak	Ya	39000	Ya	Positif
16	P-016	38	1	Tidak	Ya	67000	Ya	Positif
17	P-017	39	0	Tidak	Ya	101000	Tidak	Negatif
18	P-018	39	0	Ya	Tidak	110000	Tidak	Negatif
19	P-019	40	4	Ya	Ya	38000	Ya	Positif

Gambar 3. 3 Hasil pemilihan atribut, penentuan atribut (*id*) dan label serta penentuan tipe data

### 3.4.4 Pembagian Data *Training* dan Data *Testing*

Pembagian data menggunakan fungsi Split Data dengan membagi persentase (%) data training 70% dan data testing 30%. Data *training* digunakan sebagai data latih sistem untuk membuat pengetahuan. Data testing digunakan untuk menguji pengetahuan sistem. Pembagian data training dan testing ditampilkan pada Gambar 3.4.



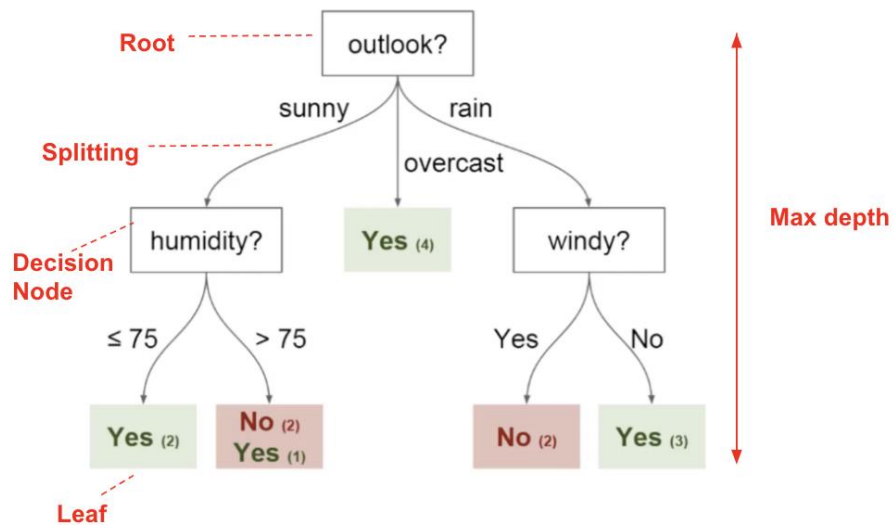
**Gambar 3. 4** Pembagian data training dan data testing

### 3.4.5 Pemilihan Algoritma

Pemilihan algoritma sebagai penentuan algoritma yang akan dipilih untuk melakukan pembelajaran dari dataset. Berikut beberapa algoritma yang akan digunakan untuk klasifikasi data DBD

#### 3.4.5.1 Algoritma Decision Tree (D3)

Tujuan penggunaan Decision tree adalah untuk membuat training model yang dapat digunakan untuk memprediksi kelas atau nilai variabel target dengan mempelajari aturan keputusan sederhana yang disimpulkan dari data sebelumnya (*data training*). Contoh diagram pohon keputusan ditampilkan pada Gambar 3.5.



**Gambar 3. 5** Diagram Pohon Keputusan

Tahapan-tahapan atau langkah algoritma D3 sebagai berikut :

- a) Langkah 1 : Mulai dari simpul akar, hitung jumlah total kasus dan jumlah keputusan “Positif” atau “Negatif”
- b) Langkah 2 : Gunakan atribut dalam dataset menggunakan *Attribute Selection Measure* (ASM) seperti Usia, Suhu Tubuh, Bintik Merah, Sakit Kepala, Tubuh Lemas, Trombosit, dan Nyeri Otot. ASM yang bisa digunakan di antaranya Information Gain dan Gini Index
- c) Hitung :  $gain(S, A) = Entropy(S) - \sum_{i=1}^n Entropy(S_i)$

Keterangan :

- a. (S) Himpunan
- b. (A) Atribut
- c. (n) Jumlah Partisi Atribut A
- d. (Si) Jumlah Kasus pada Partisi ke-1
- e. (S) Jumlah kasus pada S

- d) Hitung :  $Entropy = \sum_{n=1}^n -p_i * \log_2 p_i$

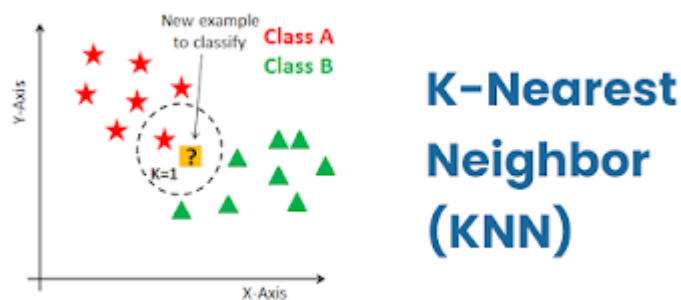
Keterangan:

- a. (S) Himpunan kasus
- b. (A) Fitur
- c. (N) Jumlah partisi S
- d. (pi) Proporsi dari Si terhadap S

- e) Langkah 3 : Pisahkan himpunan S menjadi himpunan bagian yang berisi kemungkinan nilai untuk atribut terbaik.
- f) Langkah 4 : Buat simpul decision tree baru secara *recursive* menggunakan himpunan bagian dari kumpulan data yang dibuat pada langkah 3.
- g) Langkah 5 : Lanjutkan proses ini sampai tahap terakhir di mana kita tidak dapat mengklasifikasikan simpul lebih lanjut. Simpul ini yang menjadi simpul akhir atau disebut sebagai simpul daun (*leaf node*)

### 3.4.5.2 Algoritma K-NN

Algoritma K-Nearest Neighbor (KNN) adalah algoritma machine learning yang bersifat non-parametric dan lazy learning. Cara kerja algoritma KNN ditampilkan pada Gambar 3.6.



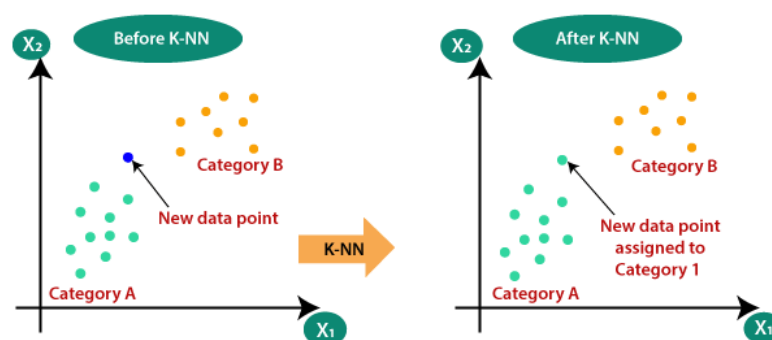
**Gambar 3. 6** Cara Kerja Algoritma k-NN

KNN bekerja berdasarkan prinsip bahwa setiap titik data yang berdekatan satu sama lain akan berada di kelas yang sama. Dengan kata lain, KNN mengklasifikasikan titik data baru berdasarkan kemiripan seperti yang ditampilkan pada Gambar 3.7:



**Gambar 3. 7** Contoh *Input* dan *Output* Algoritma k-NN

Contoh lain misalkan ada 2 (dua) kategori, yaitu Kategori A dan Kategori B, dan kita memiliki titik data baru  $x_1$ , jadi jika ingin mengetahui titik data ini akan terletak di kategori mana, dengan bantuan KNN, kita dapat mengidentifikasi kategori atau kelas dari *dataset* tertentu seperti Gambar 3.8.



**Gambar 3. 8** Diagram proses Algoritma k-NN (before-after)

Adapun alur atau Langkah-langkah cara kerja algoritma k-NN sebagai berikut:

- a) Langkah 1: Pilih nilai K, sedangkan nilai k minimal adalah 1 dan maksimal adalah jumlah set data latih
- b) Langkah 2: Melakukan normalisasi untuk semua pola set data latih ataupun set data uji. Hal ini bertujuan supaya rentang keseluruhan nilai pola memiliki rentang nilai yang sama, antara 0 sampai 1. Perhitungan normalisasi yang digunakan adalah metode *min-max* dengan persamaan :

$$\text{Normalisasi} = \frac{\text{data}(x) - \text{data}(\min)}{\text{data}(\max) - \text{data}(\min)}$$

Keterangan :

- a. Data (x) adalah data yang akan dihitung normalisasi yang diambil berdasarkan kolom datanya.

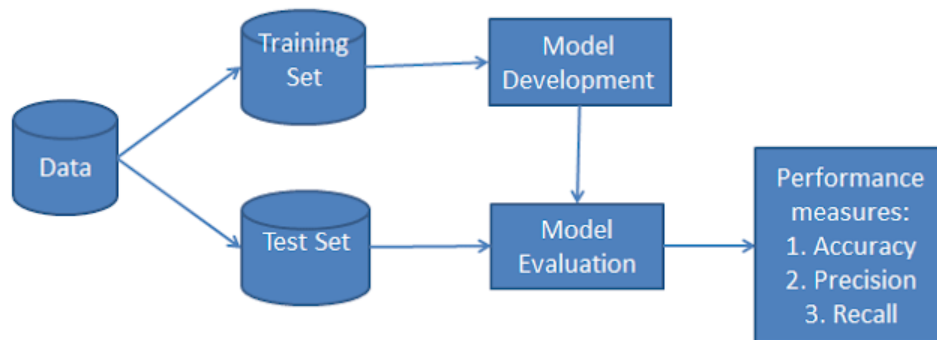


- b. Data (min) adalah data terkecil di kolom yang sama
- c. Data (max) adalah data terbesar di kolom yang sama dengan data yang akan dinormalisasi
- c) Langkah 3 : Menghitung jarak *Euclidean* menggunakan persamaan  $Deuclidean(x, y) = \sqrt{\sum i(x_i - y_i)^2}$  dimana (i) adalah banyaknya set data dan x adalah jumlah set data uji, sedangkan y adalah set data latih.
- d) Langkah 4 : Ambil tetangga terdekat K sesuai jarak yang dihitung
- e) Langkah 5 : Di antara tetangga k ini, hitung jumlah titik data di setiap kategori.
- f) Langkah 6 : Tetapkan titik data baru ke kategori yang jumlah tetangganya paling banyak.
- g) Langkah 7 : Langkah terakhir adalah menentukan hasil klasifikasi berdasarkan tahap keempat yang terbanyak

### 3.4.5.3 Algoritma Naive Bayes

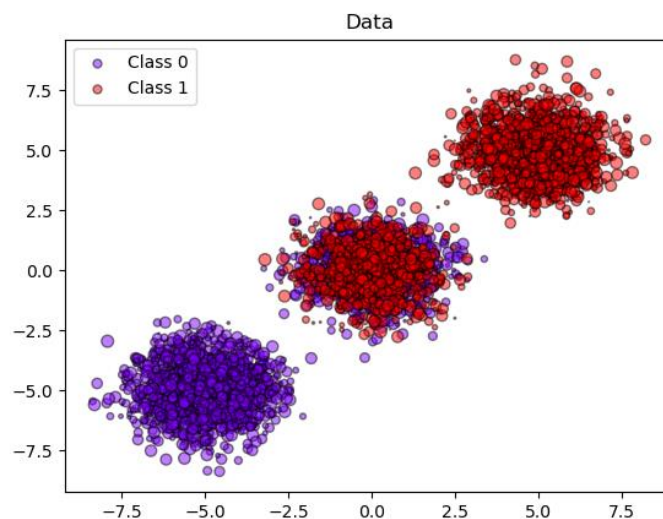
Setiap kali melakukan klasifikasi, langkah pertama adalah memahami masalah dan mengidentifikasi fitur dan label potensial. Fitur adalah karakteristik atau atribut yang mempengaruhi hasil label. Misalnya, dalam hal distribusi pinjaman, manajer bank mengidentifikasi pekerjaan, pendapatan, usia, lokasi, riwayat pinjaman sebelumnya, riwayat transaksi, dan skor kredit nasabah. Karakteristik ini dikenal sebagai fitur yang membantu model mengklasifikasikan pelanggan.

Klasifikasi memiliki 2 (dua) fase yaitu fase pembelajaran, dan fase evaluasi. Pada fase pembelajaran, *classifier* melatih modelnya pada *dataset* yang diberikan dan pada fase evaluasi. Kinerja dievaluasi berdasarkan berbagai parameter seperti akurasi, presisi, dan recall seperti Gambar 3.9.



**Gambar 3.9** Alur kerja naïve bayes classifier

Cara menerapkan Algoritma Naive Bayes yaitu apabila fitur kontinu tidak memiliki distribusi normal, maka harus menggunakan transformasi atau metode yang berbeda untuk mengubahnya menjadi distribusi normal, hapus beberapa fitur yang berkorelasi dikarenakan fitur-fitur yang sangat berkorelasi dipilih 2 (dua) kali dalam model dan dapat menyebabkan kepentingan yang berlebihan, dan jangan menggabungkan teknik kombinasi pengklasifikasian lain seperti *boosting* karena tidak membantu dikarenakan naive bayes tidak memiliki varians untuk diminimalkan seperti pada Gambar 3.10.



**Gambar 3.10** Algoritma naïve bayes dalam menentukan klasifikasi

Klasifikasi Naive Bayes menghitung probabilitas suatu peristiwa dalam langkah-langkah berikut:

- a) Langkah 1: Hitung probabilitas sebelumnya untuk label (class) yang

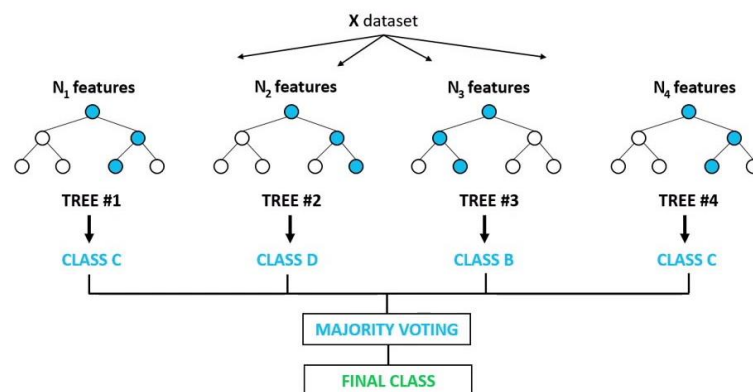
menggunakan persamaan :  $P(h|D) = \frac{P(X|H)P(h)}{P(X)}$  Dimana X merupakan data

tuple hasil pengujian dari suatu set data yang telah ditentukan masuk ke dalam kelas tertentu.  $H$  merupakan suatu hipotesis yang akan menentukan  $X$  masuk ke dalam kelas  $C$ .  $P(H|X)$  merupakan peluang atau probabilitas  $X$  yang merupakan data tuple atau bukti yang diperoleh pada saat observasi masuk ke dalam kelas  $C$ , dengan kata lain mencari probabilitas  $X$  dimiliki oleh kelas  $C$ .  $P(H|X)$  merupakan probabilitas posterior,  $H$  dikondisikan pada  $X$ . Sebaliknya  $P(H)$  merupakan probabilitas prior, atau probabilitas sebelumnya. Kemudian  $P(X|H)$  merupakan probabilitas posterior dimana  $X$  dikondisikan pada  $H$ . Sedangkan  $P(X)$  merupakan probabilitas sebelumnya dari  $X$

- b) Langkah 2: Temukan probabilitas Peluang dengan setiap atribut untuk setiap kelas menggunakan persamaan  $P(c_j|x) = \frac{p(X|c_j)p(c_j)}{p(x)}$  dimana  $(c_j)$  adalah kategori yang akan diklasifikasikan, dan  $p(c_j)$  merupakan probabilitas prioritas dari kategori  $(c_j)$ . Variabel  $(d)$  merupakan atribut yang direpresentasikan sebagai himpunan
- c) Langkah 3: Masukkan nilai ini dalam Formula Bayes dan hitung probabilitas posterior.
- d) Langkah 4: Lihat kelas mana yang memiliki probabilitas lebih tinggi, mengingat input milik kelas probabilitas lebih tinggi

#### 3.4.5.4 Algoritma Random Forest

Algoritma ini kumpulan dari beberapa jumlah *decision tree* jadi prinsip kerjanya sama dengan *decision tree* dimana masing-masing bisa menghasilkan *output* yang berbeda beda. Bedanya *random forest* ini akan melakukan voting terlebih dahulu dan akan memberikan *output* berupa mayoritas hasil dari *decision tree* seperti yang ditampilkan Gambar 3.11.



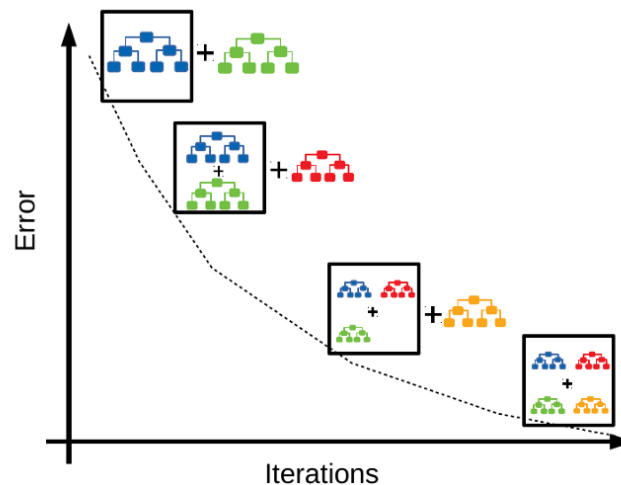
**Gambar 3. 11** Alur Kerja Algoritma Random Forest

Tahapan atau langkah-langkah klasifikasi menggunakan algoritma Random Forest sebagai berikut :

- a) Langkah 1 : Tentukan data sample ( $z$ ) dari suatu ukuran  $N$  kumpulan data (dataset)
- b) Langkah 2 : Tentukan  $m$  variable secara random dari  $p$  variable, dimana  $m \leq p$ . Untuk ukuran  $m$  terbaik akan dipilih melalui aproksimasi dari akar kuadrat berdasarkan total jumlah  $p$  variable yaitu  $\sqrt{p}$ . Nilai  $m$  diperoleh dari persamaan ( $m = 2|\sqrt{p}|$ ), selanjutnya hitung nilai setengah dari nilai akar menggunakan persamaan ( $m = \frac{1}{2}|\sqrt{p}|$ )
- c) Langkah 3 : Setelah dilakukan pemilihan  $m$  secara random, maka pohon ditumbuhkan tanpa *pruning* (pemangkasan). Pemecahan simpul terbaik dari suatu pohon dilakukan dengan menggunakan index gain
- d) Langkah 4 : Setelah dilakukan Langkah 1-3 sebanyak  $n$  kali sehingga didapatkan suatu *forest (classifier)* sebanyak  $n$  pohon
- e) Langkah 5 : Setelah terbentuk forest kemudian dicari nilai klasifikasi *error* untuk mendapatkan nilai optimal dan diperoleh tingkat kepentingan variable (*variable important*) yang lebih stabil
- f) Langkah 6 : Terakhir penentuan prediksi suatu kelas dilakukan dengan *majority vote* (suara terbanyak)

### 3.4.5.5 Algoritma Gradient Boosted Tree

Gradient boosting termasuk *supervised learning* berbasis *decision tree* yang dapat digunakan untuk klasifikasi. Gradient boosting dimulai dengan menghasilkan pohon klasifikasi awal dan terus menyesuaikan pohon baru melalui minimalisasi fungsi kerugian seperti Gambar 3.14.



**Gambar 3. 12** Klasifikasi algoritma Gradient Boosted Tree

Berikut merupakan tahapan atau langkah-langkah pada klasifikasi menggunakan Algoritma Gradient Boosting Tree:

- a) Langkah 1 : Inisialisasi Prediksi. Pada tahap ini kita melihat data yang mewakili prediksi awal untuk setiap individu. Untuk metode klasifikasi, inisialisasi akan sama dengan log (odds) dari variabel dependen. Setelah itu, dilanjutkan dengan memasukan hasil log (odds) ke dalam rumus logistic function  $x = \frac{e^{\text{Log}(\text{odds})}}{1 + e^{\text{Log}(\text{odds})}}$
- b) Langkah 2 : Menghitung Nilai Residual. Setelah mengetahui predicted value, selanjutnya adalah menghitung nilai residual menggunakan rumus residual dengan actual value bernilai 0 untuk No dan 1 untuk Yes. Residual = Actual value — Predicted value
- c) Langkah 3 : Memprediksi Nilai Residual. Dalam tahap ini akan dibuat decision tree yang berisi angka prediksi nilai residual. Diperlukan proses transformasi untuk menghitung prediksi nilai residual di setiap datanya. Berikut merupakan rumus transformasi yang umum digunakan :

$$x = \frac{\sum Residual(i)}{\sum Previous Probability(i) x (1 - Previous Probability(i))}$$

- d) Langkah 4 : Mendapatkan nilai probabilitas baru untuk data yang dependen Sampel dataset akan dijalankan dengan melalui decision tree yang berisi nilai residual yang baru. Residu yang diperoleh untuk setiap pengamatan akan ditambahkan ke prediksi sebelumnya untuk mengetahui keakuratan dari data tersebut :

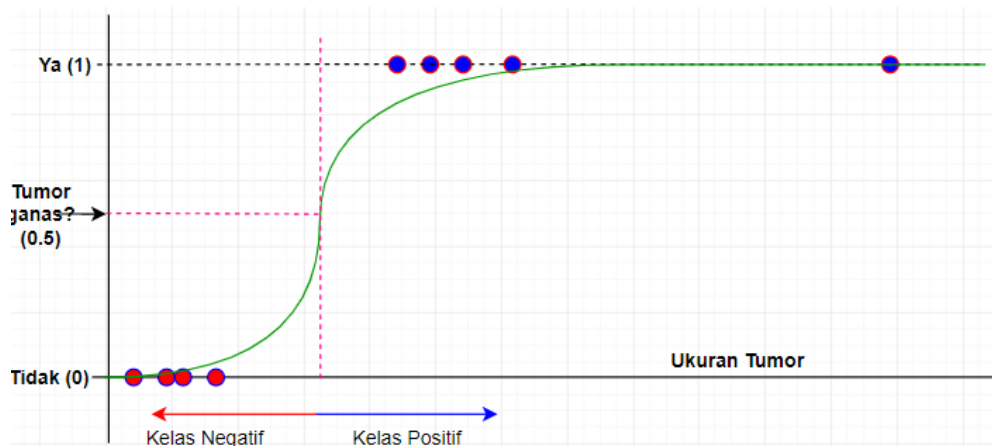
$$x = initial\ prediction + learning\ rate\ x\ predicted\ residual$$

- e) Langkah 5 : Mendapatkan Nilai Residual yang baru setelah mengetahui nilai probabilitas dari seluruh data, selanjutnya adalah menghitung nilai residual dari masing-masing data dengan mengurangi nilai prediksi baru dari actual value
- f) Langkah 6 : Ulangi langkah tiga hingga langkah lima sampai nilai residual mendekati nol atau jumlah iterasi sesuai dengan hyperparameter ketika menjalankan algoritma
- g) Langkah 7 : Komputasi Final. Setelah mengetahui semua nilai probabilitas dan residual yang baru, maka nilai final dari prediksi suatu data didapatkan dari rumus berikut:

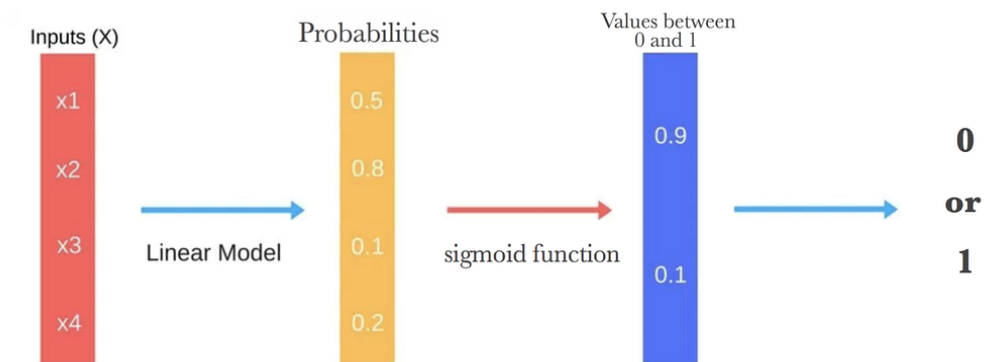
$$x = initial\ prediction + learning\ rate + predicted\ residual(1) \\ + learning\ rate\ x\ predicted\ residual\ (2)$$

#### 3.4.5.6 Algoritma Logistic Regression

Logistic Regression adalah sebuah algoritma klasifikasi untuk mencari hubungan antara fitur (*input*) diskrit/kontinu dengan probabilitas hasil *output* diskrit tertentu. Analisis statistik yang sering digunakan data analyst untuk Pemodelan predictive. Dalam pendekatan analitis ini, variabel dependennya terbatas atau kategoris berupa A atau B (*regresi biner*). Binary Logistic Regression hanya memiliki 2 *output* (mengklasifikasi ke dalam 2 kelas berbeda) contohnya class Positif-Negatif. Gambar 3.15. menunjukkan proses klasifikasi menggunakan algoritma Logistic Regression



**Gambar 3. 13** Klasifikasi algoritma Logistic Regression



**Gambar 3. 14** Tahapan proses algoritma Logistic Regression

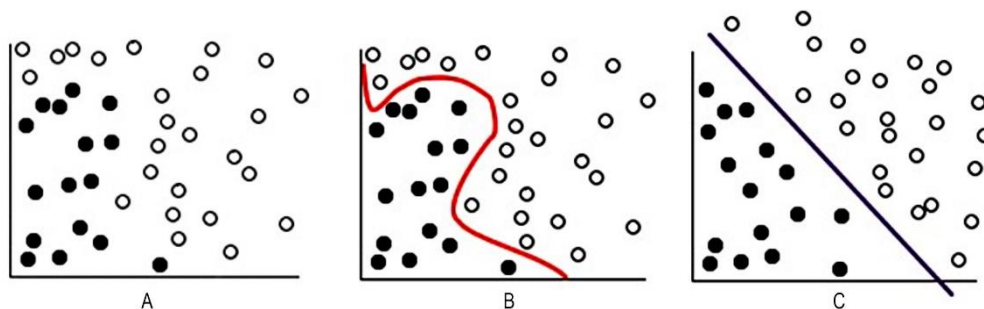
Adapun langkah-langkah yang perlu dilakukan untuk melakukan analisis logistic regression adalah sebagai berikut :

1. Menentukan tujuan dari Analisis Logistic Regression menggunakan persamaan:  $Y = X_1 + X_2 + X_3 \dots X_n$
2. Mengidentifikasi variabel predictor dan variabel response
3. Melakukan pengumpulan data dalam bentuk tabel
4. Menghitung  $X^2$ ,  $XY$  dan total dari masing-masing
5. Menghitung a dan b menggunakan rumus yang telah ditentukan
6. Membuat model Persamaan Garis Regresi
7. Melakukan prediksi terhadap variabel predictor atau response
8. Uji signifikansi menggunakan Uji-t dan menentukan Taraf Signifikan

### 3.4.5.7 Support Vector Machine (SVM)

SVM bekerja untuk mencari hyperplane atau fungsi pemisah (decision boundary) terbaik untuk memisahkan dua buah kelas atau lebih pada ruang input. Hiperplane dapat berupa line atau garis pada dua dimensi dan dapat berupa flat plane pada multiple plane.

Algoritma SVM bekerja dengan cara memetakan data ke ruang fitur berdimensi tinggi sehingga titik data dapat dikategorikan, bahkan ketika data tersebut tidak dapat dipisahkan secara linier. Apabila pemisah antar kategori berhasil ditemukan, data dapat ditransformasikan sedemikian rupa sehingga pemisah tersebut dapat digambarkan sebagai hyperplane. Kemudian, karakteristik data baru dapat digunakan untuk memprediksi pada kelompok mana record baru seharusnya berada.



**Gambar 3. 15** Tahapan proses algoritma Logistic Regression

Sebagai ilustrasi, perhatikan **Gambar 3.15 A** dimana terdapat titik-titik data yang terpisah dalam 2 kategori, yakni bulatan hitam dan bulatan putih. Kedua kategori tersebut kemudian dipisahkan dengan kurva, seperti yang terlihat pada gambar B. Setelah dilakukan transformasi, batas antara dua kategori dapat ditentukan oleh hyperplane, seperti yang ditunjukkan gambar C.

Fungsi matematis yang digunakan pada proses transformasi algoritma SVM dikenal dengan istilah fungsi kernel (kernel function). Fungsi kernel digunakan untuk mengklasifikasikan data non-linier. Caranya adalah dengan mengubah data non-linear menjadi data linier dan kemudian membentuk hyperplane. Fungsi kernel linier direkomendasikan ketika pemisahan linier data dilakukan secara langsung. Dalam kasus yang berbeda, penggunaan fungsi lain mungkin diperlukan.