BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Beberapa penelitian terkait penyakit hepatitis, data mining, dan algoritme klasifikasi dijadikan rujukan pada penelitian ini. Beberapa penelitian sebelumnya dapat dilihat pada Tabel 2.1.

Tabel 2.1 Penelitian Terkait Hepatitis dan Algoritme Klasifikasi

No	Judul	Peneliti	Masalah	Metode dan Hasil (Akurasi)
1	Implementasi Algoritma Naïve Bayes Pada Data Set Hepatitis Menggunakan Rapid Miner.	Novianti (2019) [34].	Penyakit hepatitis merupakan penyakit yang tergolong jenis penyakit mematikan di dunia dan hingga saat ini belum ada obat untuk penyembuhan penyakit tersebut. Menurut Kementrian Kesehatan RI tahun 2014 sebanyak 1,5 juta dunia meninggal setiap tahunnya dikarenakan penyakit Hepatitis.	Algoritme Naïve Bayes menghasilkan akurasi 76.77% menggunakan *software* Rapid Miner, namun dalam penelitian ini dapat menggunakan algoritme dua atau lebih sekaligus sebagai pembanding algoritme mana yang lebih lebih akurat dengan nilai akurasi paling tinggi sebagai algoritme terbaik sebagai pengembangan dalam penelitian.

No	Judul	Peneliti	Masalah	Metode dan Hasil (Akurasi)
2	Data Mining Optimization Based on Particle Swarm Optimization For Diagnosis of Inflammatory Liver Disease.	Amrin, Pahlevi (2021) [35]	Peradangan hati merupakan jenis penyakit menular yang menimbulkan masalah kesehatan masyarakat dan dapat mempengaruhi angka kematian, status kesehatan dan angka harapan hidup, dan dampak social masyarakat lainnya.	Pengujian menggunakan metode Cross Validation Confusion Matrix dan kurva ROC sebagai pemecah permasalahan menggunakan Rapid Maner. Akurasi algoritme C4.5 dengan PSO dengan nilai akurasi tertinggi yaitu 79,51 dengan nilai AUC sebesar 0,950.
3	Sistem Diagnosis Penyakit Hati Menggunakan Metode Naïve Bayes.	Prayoga, Hidayat, Dewi (2018) [36]	Kesulitan masyarakat dalam mengenali jenis penyakit hati merupakan sebuah masalah yang serius. Gejala pada penyakit hati mempunyai variasi cukup banyak dan diantara gejala-gejala tersebut memiliki kemiripan diantara satu sama lain sehingga sulit untuk diidentifikasikan.	Sistem berbasis *offline* pada platform android menggunakan metode Naïve Bayes. Pengujian nilai akurasi memperoleh hasil sebesar 87,4%.

No	Judul	Peneliti	Masalah	Metode dan Hasil (Akurasi)
4	Prediksi Harapan Hidup Pasien Penderita Hepatitis B Menggunakan *Classification And Regresion Trees.*	Saparudin, Khaira, Maulidina, Fikri, Sumarlin, Wahyudi (2018) [37]	Dalam dunia kesehatan memprediksi suatu penyakit spesialis bukanlah hal yang mudah terlebih hingga memprediksi harapan hidup pasien yang mengidap penyakit tersebut.	Algoritme CART menghasilkan akurasi dengan presentase yang tinggiyaitu mencapai 93.6% sehingga dapat diimplementasikan sebagai algoritme dasar dalam pengembangan perangkat lunak prediksi harapan hidup prediksi harapan hidup penderita hepatitis B.
5	Perbandingan Akurasi Algoritma *Random Forest* dan Algoritma *Artificial Neural Network* Untuk Klasifikasi Penyakit Diabetes.	Witjaksana, Saedudin, Widartha (2021) [38]	Berdasarkan data *Internasional Diabetes Federation* Indonesia menduduki peringkat ke-7 di dunia negara dengan jumlah pasien penyakit terbanyak di dunia. Indonesia berada pada status waspada sehingga perlu adanya klasifikasi terhadap penyakit diabetes.	Perbandingan Algoritme *Random Forest* dan algoritme *Artificial Neural Network*. Data melewati tahap *preprocessing*, kemudian untuk pencarian nilai akurasi terbaik dengan cara *Confusion Matrix* untuk menemukan hasil dari ROC AUC dan hasil F1-*Score* dari tiap algoritme. Hasil pengujian menunjukan nilai akurasi algoritme *Random Forest* memiliki nilai akurasi tertinggi yaitu 90.62%.

No	Judul	Peneliti	Masalah	Metode dan Hasil (Akurasi)
6	Prediksi Harapan Hidup Penderita Hepatitis Kronik Menggunakan Metode-Metode Klasifikasi.	Siti Khomsah (2018) [25]	Penderita hepatitis kronik berpotensi menjadi kanker hati. Hingga saat ini pengobatan bagi penderita hepatitis kronik tersedianya vaksin yang tepat untuk menangani penyakit tersebut, oleh sebab itu pengobatannya hanya terbatas untuk memperpanjang harapan hidup pasien.	Akurasi dari algoritme klasifikasi untuk memprediksi harapan hidup penderita hepatitis kronik yang dibandingkan yaitu: K-NN (70,31%), *Naive Bayes* (72,90%), *D-Tree* (74,84%), dan *Random Forest* (79,35%). Algoritme *Random Forest* memiliki nilai akurasi tertinggi yaitu 79,35%.
7	Perbandingan K-Nearest Neighbor dan *Random Forest* dengan Seleksi Fitur Information Gain untuk Klasifikasi Lama Studi Mahasiswa.	Hasan, Resmawan, Ibrahim (2018) [39].	Keberhasilan suatu perguruan tinggi adalah dilihat dari ketepatan lama studi mahasiswa. Predikat mahasiswa lulus tepat waktu yaitu apabila mahasiswa mampu menyelesaikan masa studi sarjana dalam waktu maksimal 4 (empat) tahun.	Metode klasifikasi dengan algoritme K-NN dan *Random Forest*. Hasil akurasi, tertinggi yaitu algoritme *Random Forest* dengan akurasi mencapai 100%, lebih unggul daripada algoritme K-NN dengan nilai akurasi sebesar 86.67%.

No	Judul	Peneliti	Masalah	Metode dan Hasil (Akurasi)
8	Perbandingan Metode Data Mining *SVM* Dan NN Untuk Klasifikasi Penyakit Ginjal Kronis.	Hilda Amalia (2018) [40].	Penyakit ginjal kronis tergolong jenis penyakit mematikan. Diketahui pada setiap tahunnya penyakit tersebut selalu mengalami kenaikan. WHO memperkirakan Indonesia mengalami peningkatan 46% dari tahun 1955-2025.	Metode klasifikasi dengan algoritme *SVM* dan *NN*. Hasil akurasi, tertinggi yaitu algoritme *SVM* dengan akurasi sebesar 95,16%, lebih unggul daripada algoritme *NN* dengan nilai akurasi sebesar 93,37%. Penelitian ini menggunakan dataset dari website UCI yang bersifat *public* dengan bentuk *supervised*.
9	Komparasi Algoritma *Support Vector Machine* Dan Naive Bayes Pada Klasifikasi Ras Kucing.	Kusuma, Jinan, Lubis, Rubianto, Rosnelly (2022) [41].	Tercatat terdapat 315 ras kucing pada seluruh dunia dengan ciri-ciri yang berbeda tiap rasnya. Hal tersebut menyebabkan dalam penentuan ras kucing menjadi lebih sulit.	Hasil akurasi, terbaik yaitu algoritme *SVM* dengan nilai akurasi sebesar 88.4%, precision 88.5% dan recall 88.4% Algoritma Naive Bayes (NB) dengan nilai akurasi sebesar 79.5%, precision 79.9% dan recall 79.5%.
10	Perbandingan Tiga Algoritma Classifier Untuk Penentuan	Yulhendri, Juman (2021) [42].	Penerimaan Peserta Didik Baru Baru (PPDB) merupakan program pemerintah untuk	Metode klasifikasi dengan membandingkan pe*RF*orma beberapa algoritme. Hasil akurasi

No	Judul	Peneliti	Masalah	Metode dan Hasil (Akurasi)
	Penerimaan Peserta Didik Baru Pada Sekolah Menengah Atas.		mengupayakan pemerataan pendidikan. Sampai saat ini masalah yang kerap dihadapi calon siswa adalah prediksi untuk pemilihan calon sekolah hanya melihat dari passing grade tahun-tahun sebelumnya.	terbaik yaitu pada algoritme *SVM* RBF didapatkan akurasi sebesar 64.78%. Kedua yaitu pada algoritma Decesion Tree didapatkan akurasi sebesar 61.7%. Ketiga yaitu pada algoritma Naïve Bayes, didapatkan akurasi sebesar 52.94%.

Menurut penelitian [34][35][36][37] penyakit hepatitis merupakan jenis penyakit menular yang dapat berakibat fatal yaitu kematian. Penyakit hati memiliki gejala-gejala yang bervariasi cukup banyak dan diantara gejala-gejala tersebut memiliki kemiripan diantara satu sama lain. Oleh sebab itu, dibutuhkan langkah untuk dapat mengklasifikasi apakah pasien berisiko meninggal dunia karena penyakit tersebut sehingga dapat meminimalisir keterlambatan proses diagnosis serta dapat sebagai acuan dalam memberikan perlakuan yang tepat pada pasien maka kenaikan angka kematian dapat dihindari. Menurut penelitian [38][25][39][40][41][42] algoritme *Random Forest* dan *SVM* memiliki kemampuan baik dilihat dari tinggi nilai akurasi lebih unggul dibandingkan dengan algoritme lain. Maka pada penelitian ini penulis memilih kedua algoritme tersebut yaitu *Random Forest* dan *SVM* untuk dibandingkan agar mendapatkan algoritme dengan kemampuan terbaik diantara kedua algoritme tersebut. Pada penelitian [38][25][39][40][41][42] tidak dilakukan pengujian berdasarkan waktu komputasi, pada penelitian ini akan dilakukan pengujian waktu komputasi terhadap algoritme yang digunakan yaitu *Random Forest* dan *SVM* sehingga membuat proses validasi kinerja semakin baik.

2.2 Dasar Teori

2.2.1 Hepatitis

Hati adalah salah satu organ vital yang memainkan peran penting dalam tubuh diantaranya fungsi metabolisme, detoksifikasi terhadap racun dan sintesis protein [43]. Kerusakan organ hati dapat mengakibatkan gangguan fungsi hati [44]. Hepatitis adalah peradangan pada sel hati, penyebab utama penyakit hepatitis ialah infeksi (virus, bakteri, parasite), namun beberapa hal lain juga dapat menyebabkan kondisi tersebut seperti: konsumsi obat-obatan, lemak berlebih dan autoimun [45]. Terdapat jenis-jenis virus yang dapat menyebabkan penyakit hepatitis yaitu : Virus Hepatitis A (VHA) adalah penyebab penyakit Hepatitis A, Virus Hepatitis B (VHB) adalah penyebab penyakit Hepatitis B, Virus Hepatitis C (VHC) adalah penyebab penyakit Hepatitis C, Virus Hepatitis D (VHD) adalah penyebab penyakit Hepatitis D, dan Virus Hepatitis E (VHE) adalah penyebab penyakit Hepatitis E [46]. Pada penelitian [47] menandai hepatitis dapat dilihat *indicator* atau pemeriksaan kadar albumin dan bilirubin sebagai *indicator*:

- *Albumin*, merupakan protein plasma darah dengan jumlah 55-60% di dalam tubuh manusia. Pemeriksaan albumin dilakukan untuk menilai kapasitas hati untuk memproduksi protein [47].
- *Bilirubin*, merupakan komponen hasil pemecahan sel darah merah yang sudah tua pemeriksaan bilirubin dilakukan untuk mengukur jumlah total *bilirubin* yang ada di dalam tubuh [48].

2.2.2 *Data Mining*

Data mining merupakan teknik pengumpulan data sebagai penentuan keteraturan pola dalam suatu dataset dengan tujuan sebagai penentuan pengambilan keputusan [49]. *Data mining* adalah proses yang menggabungkan metode statistik dengan matematika, kecerdasan buatan, dan pembelajaran mesin untuk menemukan pola berulang dalam sejumlah besar data yang selanjutnya dapat digunakan untuk mengekstraksi informasi penting [50].

Pengelompokan Data Mining terbagi kedalam beberapa kelompok yaitu [51]:

1. Deskripsi

   Deskripsi merupakan teknik atau cara yang digunakan untuk menggambarkan pola dan kecenderungan yang dimiliki oleh data.
2. Estimasi

   Estimasi memiliki kemiripan dengan klasifikasi, namun pembeda diantara keduanya adalah variable estimasi lebih kearah numerik sedangkan klasifikasi kearah kategorial.
3. Prediksi

   Berfungsi untuk melakukan prediksi nilai yang akan dicapai di masa yang akan datang berdasarkan nilai pada periode sebelumnya sebagai dasar prediksi.
4. Klasifikasi

   Klasifikasi merupakan proses untuk menemukan kesamaan karakteristik dari sekumpulan model sebagai pembeda atau pemisah kelas-kelas data.
5. Klastering

   Klastering yaitu pengelompokan suatu data yang belum diketahui labelnya sebagai proses identifikasi kelompok-kelompok data yang mempunyai karakteristik khusus.
6. Asosiasi

   Asosiasi bertugas untuk menemukan pengelompokan hal-hal yang dapat terjadi secara bersamaan.

2.2.3 *Data Preprosessing*

Preprosessing adalah langkah awal sebelum proses data mining dilakukan yaitu melakukan pengolahan data mentah agar sesuai dengan proses *data mining* yang dilakukan [52]. Tujuan dari *preprosessing* data adalah untuk mengolah data mentah menjadi data yang siap digunakan pada proses analisis selanjutnya, sehingga dapat meningkatkan kualitas data serta meningkatkan akurasi dan efisiensi proses *mining* [53].

Preprocessing data dalam penelitian ini dilakukan dengan melakukan *feature selection* dan imputasi data. *Feature selection* yaitu proses untuk mengidentifikasi subset atribut yang optimal dengan tahap pertama yaitu

melakukan pelatihan dan pengujian kemudian tahap kedua penilaian atribut yaitu mengurutkan seberapa penting atribut tersebut terhadap proses klasifikasi sehingga dapat mengurangi dimensi data dan mengurangi lama waktu pelatihan [54]. Imputasi data yaitu metode alternatif pengisian informasi suatu data yang hilang [55].

2.2.4 Algoritme *Random Forest*

Algoritme *Random Forest* merupakan algoritme dengan konsep membuat pohon keputusan [56]. Algoritme *Random Forest* terdiri atas banyak pohon keputusan dimana semakin banyak penggunaan pohon akan mempengaruhi nilai akurasi yang didapatkan menjadi lebih baik [57].

Pohon keputusan pada algoritme *Random Forest* terbentuk dimulai dengan pencarian nilai *entropy* untuk menentukan cabang pohon keputusan lalu pencarian nilai *information gain* untuk menentukan node akar. Rumus persamaan 2.1 untuk perhitungan nilai *entropy* dan rumus persamaan 2.2 untuk perhitungan nilai *information gain* [58].

$$Entropy\ (Y) = -\sum_i p(c|Y) Log_2 p(c|Y) \tag{2.1}$$

Keterangan :

Y = himpunan kasus.

p(c|Y) = proposisi nilai Y terhadap c.

$$InformationGain\ (Y, a) = Entropy\ (Y) - \sum_{v \in Values(a)} \frac{Yv}{Ya} Entropy(Yv) \tag{2.2}$$

Keterangan :

Values(a) = semua nilai yang mungkin dalam himpunan kasus a.

Yv = subkelas dari Y dengan kelas v yang berkaitan dengan kelas a.

Ya = semua nilai yang sesuai dengan a.

2.2.5 *Support Vector Machine*s (*SVM*)

Algoritme *Support Vector Machine*s (*SVM*) adalah algoritme yang pertama kali diperkenalkan oleh Vapnik pada tahun 1992 [59]. *Support Vector Machine*s (*SVM*) bekerja dengan tujuan untuk menemukan *hyperplane* yaitu suatu fungsi yang dapat digunakan sebagai pemisah dua buah kelas [60]. Algoritme *SVM* memiliki kelebihan yaitu dapat memisahkan antara model linier maupun *non*-linier yang diperoleh dengan fungsi *kernel* yang diperlukan [61]. Model linier merupakan data yang tersebar membentuk garis linier sedangkan model *non*-linier yaitu data yang tersebar tidak membentuk suatu garis linier [62]. Tingkat akurasi model yang akan dihasilkan sangat bergantung pada fungsi kernel yang digunakan, beberapa tipe *kernel* yang digunakan dapat dilihat persamaannya pada Tabel 2.2 dibawah ini [63].

Tabel 2.2 Tabel Persamaan Umum *Kernel* pada *SVM*

Nama Kernel	**Rumus Persamaan**
Linear *(linear)*	$K(xi.x) = xi^T x$
Polinomial *(non-linier)*	$K(xi.x) = (\gamma.xi^T x + r)p\ , \gamma > 0$
RBF Gaussian *(non-linier)*	$K(xi.x) = exp(-\gamma\|xi - x\|^2)\ , \gamma > 0$
Siqmoid *(non-linier)*	$K(xi.x) = tanh(\gamma xi^T + r)$

Berdasarkan penelitian [42] *kernel* RBF merupakan *kernel* terbaik untuk menangani kasus data *non-linear* dikarenakan *kernel* mampu memetakan sampel ke ruang dimensi yang lebih tinggi sehingga dapat menangani kasus *non-linear*.

Tahapan yang dilakukan dalam algoritme SVM adalah sebagai berikut [64][65]:

1. Menentukan variable yang akan dijadikan kelas atau kategori.
2. Inisialisasi pada σ_1, α, C, λ, γ, dan ε.

   Keterangan:

   σ_1 = Sigma.

   α = *Alfa*, berfungsi untuk inisialisasi awal *support vector*.

   C = Untuk memberi batas nilai *alfa*.

λ = Turunan fungsi.

γ = *Gamma* untuk mencari nilai kecepatan.

ε = *Epsilon* berfungsi untuk nilai *error*.

3. Menentukan nilai matriks menggunakan persamaan (2.3).

$$D_{ij} = y_i y_j (K(x_i \cdot x_j) + \lambda^2) \tag{2.3}$$

Dengan *i*,*j* : 1,…,*n*.

Keterangan :

D_{ij} = Elemen dari matriks data ke-*ij*.

y_i = Kelas atau kategori data ke-*i*.

y_j = Kelas atau kategori data ke-*j*.

λ = Turunan fungsi.

$K(x_i \cdot x_j)$ = *Karnel* nilai yang dipakai.

4. Menentukan nilai *error*, delta alpha, dan alpha baru dapat dihitung mengguakan persamaan (2.4), (2.5), (2.6).

a. Nilai *error*

$$E_i = \sum_{j=1}^{n} a_i D_{ij} \tag{2.4}$$

Keterangan :

E_i = Nilai error (kesalahan) pada data ke-*i*.

D_{ij} = Elemen dari matriks data ke-*ij*.

a_i = Bobot nilai setiap titik pada data.

b. Delta alpha

$$\delta a_i = \min \{\max[\gamma(1 - E_i), -a_i], C - a_i\} \tag{2.5}$$

Keterangan :

a_i = Bobot nilai setiap titik pada data.

γ = Tingkat pembelajaran yang sudah dilakukan.

$max_{(i)} D_{ij}$ = Nilai maksimum pada diagonal matriks hesse.

E_i = Nilai error (kesalahan) pada data ke-*i*.

C = Nilai konstanta

c. Alpha baru

$$a_i = a_i + \delta a_i \tag{2.6}$$

Keterangan :

a_i = Bobot nilai setiap titik pada data.

δa_i = Delta bobot nilai setiap titik pada data.

5. Mencari nilai bobot vektor (w) dan nilai bias (b) dapat dihitung menggunakan persamaan (2.7), (2.8).

$$w = \sum_{i=1}^{m} a_i y_i K(x_i\,, x_j) + b \tag{2.7}$$

Keterangan :

w = Bobot dari vector.

m = Jumlah data yang diuji.

a_i = *Alfa* nilai ke-*i*.

y_i = Data nilai dari kelas ke-*i*.

$K(x_i, x_j)$ = *Karnel* nilai yang dipakai.

b = nilai *bias*.

Dibawah ini merupakan rumus untuk mencari nilai *bias* :

$$b = -\frac{1}{2}[w \cdot x^+ + w \cdot x^-] \tag{2.8}$$

Keterangan :

w = Bobot dari vector.

$y_i \in \{-1, +1\}$ = Kelas atau kategori data.

6. Mencari nilai fungsi kernel atau $K(x_i, x_{test})$ seperti pada persamaan (2.9).

$$K(x_i, x_{test}) = \sum_{i=1}^{m} y_i y_j K(x_i\,, x) \tag{2.9}$$

Keterangan .

m = Jumlah data yang diuji.

y_i = Data kelas ke-*i*.

y_j = Data kelas ke-*j*.

$K(x_i, x)$ = *Karnel* nilai yang dipakai.

7. Langkah terakhir mencari nilai keputusan atau $f(xtest)$ seperti pada persamaan (2.10).

$$f(xtest) = \sum_{i=1}^{m} a_i y_j K(x_i, x) + b \qquad (2.10$$

Keterangan :

m = Jumlah data yang diuji.

y_i = Data kelas ke-*i*.

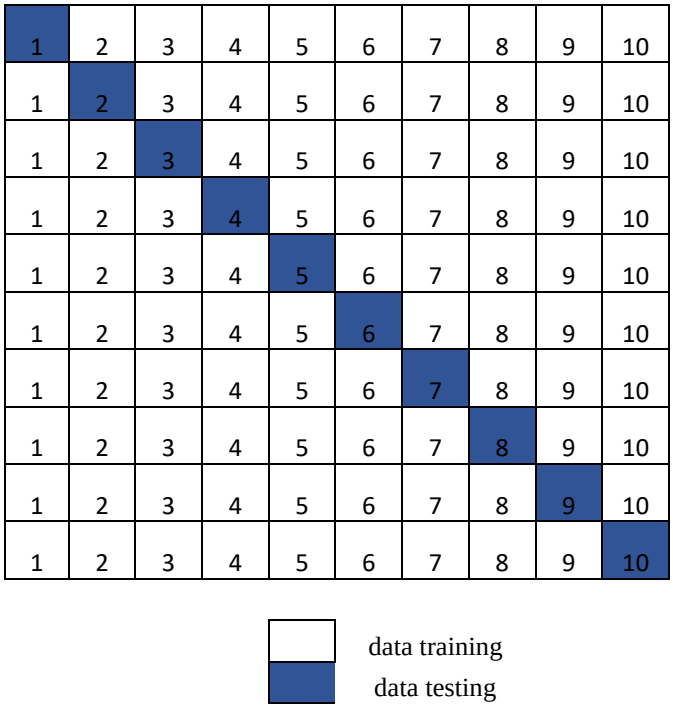
y_j = Data kelas ke-*j*.

$K(x_i, x)$ = *Karnel* nilai yang dipakai.

b = nilai *bias*.

2.2.6 *K-Fold Cross Validation*

K-Fold Cross Validation adalah sebuah teknik validasi model untuk menilai, memprediksi dan memperkirakan keakuratan sebuah model yang dipecah menjadi k bagian set data dengan ukuran yang sama [66]. Tujuan dari penggunaan *K-Fold Cross Validation* adalah untuk menghilangkan bias pada data [67]. Gambar 2.1 adalah contoh penggunaan *K-fold Cross Validation* dengan nilai k=10.



Gambar 2. 1 Model K-*Fold Cross Validation* dengan K=10

Pada Gambar 2.1 merupakan penggunaan 10-*Fold Cross Validation*, model pengujian dapat diasumsikan dengan nama data D1, D2, D3, D4, D5, D6, D7, D8, D9, dan D10.

a. Percobaan ke-1 (satu), data D1 digunakan untuk data testing. Sebagai data pelatihan, menggunakan data berikut: D2, D3, D4, D5, D6, D7, D8, D9, dan D10.
b. Percobaan ke-2 (dua), data D2 digunakan untuk data testing. Sebagai data pelatihan, menggunakan data berikut: D1, D3, D4, D5, D6, D7, D8, D9, dan D10.
c. Percobaan ke-3 (tiga), data D3 digunakan untuk data testing. Sebagai data pelatihan, menggunakan data berikut: D1, D2, D4, D5, D6, D7, D8, D9, dan D10.
d. Percobaan ke-4 (empat), data D4 digunakan untuk data testing. Sebagai data pelatihan, menggunakan data berikut: D1, D2, D3, D5, D6, D7, D8, D9, dan D10.
e. Percobaan ke-5 (lima), data D5 digunakan untuk data testing. Sebagai data pelatihan, menggunakan data berikut: D1, D2, D3, D4, D6, D7, D8, D9, dan D10.
f. Percobaan ke-6 (enam), data D6 digunakan untuk data testing. Sebagai data pelatihan, menggunakan data berikut: D1, D2, D3, D4, D5, D7, D8, D9, dan D10.
g. Percobaan ke-7 (tujuh), data D7 digunakan untuk data testing. Sebagai data pelatihan, menggunakan data berikut: D1, D2, D3, D4, D5, D6, D8, D9, dan D10.
h. Percobaan ke-8 (delapan), data D8 digunakan untuk data testing. Sebagai data pelatihan, menggunakan data berikut: D1, D2, D3, D4, D5, D6, D7, D9, dan D10.

i. Percobaan ke-9 (sembilan), data D9 digunakan untuk data testing. Sebagai data pelatihan, menggunakan data berikut: D1, D2, D3, D4, D5, D6, D7, D8, dan D10.
j. Percobaan ke-10 (sepuluh), data D10 digunakan untuk data testing. Sebagai data pelatihan, menggunakan data berikut: D1, D2, D3, D4, D5, D6, D7, D8, dan D9.

2.2.7 *Confussion Matrix*

Confussion Matrix adalah metode untuk mengukur keefektifan suatu metode klasifikasi dan diilustrasikan dengan table yang melaporkan jumlah data uji yang diklasifikan dengan benar dan jumlah data uji yang salah diklasifikasikan [68]. Tabel *Confusion matrix* dapat dilihat pada Table 2.17 berikut :

Tabel 2.3 *Confussion Matrix*

Actual Label	**Predicted Label**	
	Positive (+)	Negative (-)
Positive (+)	True Positives (TP)	False Negatives (FN)
Negative (-)	False Positives (FP)	True Negatives (TN)

Penjelasan mengenai nilai table *Confussion Matrix* beserta persamaan yang digunakan adalah sebagai berikut [68][69]:

a. Nilai *True Posetive* (TP) merupakan jumlah data positif yang terklasifikasi dengan benar oleh system sebagai data positif. Persamaan *True Posetivies* (TP) dapat dilihat oleh persamaan (2.11):

$$TPR = \frac{TP}{TP + FN} \tag{2.11}$$

Keterangan :

TPR = *True Positive Rate*

TP = *True Positive*

FN = *False Negative*

b. Nilai *True Negative* (TN) merupakan jumlah data negatif yang terklasifikasi dengan benar oleh system sebagai data negatif. Persamaan *True Negative* (TN) dapat dilihat oleh persamaan (2.12):

$$TNR = \frac{TN}{TN + FP} \tag{2.12}$$

Keterangan :

FPR = *False Positive Rate*

TN = *True Negative*

FP = *False Positif*

c. Nilai *False Positive* (FP) merupakan jumlah data negatif namun terklasifikasi oleh system sebagai data positif. Persamaan *False Posetivies* (FP) dapat dilihat oleh persamaan (2.13):

$$FPR = \frac{FP}{FP+TN} \quad (2.13)$$

Keterangan :

FPR = *False Positive Rate*

FP = *False Positive*

TN = *True Negative*

d. Nilai *False Negative* (FN) merupakan jumlah data positif yang terklasifikasi oleh system sebagai data negatif. Persamaan *False Negatives* (FN) dapat dilihat oleh persamaan (2.14):

$$FNR = \frac{FN}{FN + TP} \tag{2.14}$$

Keterangan :

FNR = *False Negative Rate*

FN = *False Negative*

TP = *True Positive*

Berdasarkan nilai dari *Confussion Matrix* dapat digunakan untuk perhitungan akurasi, presisi, dan *recall* [70].

a. Akurasi

Akurasi merupakan tingkat kebenaran data yang terklasifikasi menggunakan suatu metode algoritme. Perhitungan nilai akurasi dapat diperoleh dengan persamaan (2.15) [70].

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \tag{2.15}$$

b. *Precision*

Precision merupakan pengukuran tinggkat keakuratan dari suatu model algoritme yang dapat dihitung dengan membandingkan jumlah data kategori *True Positive* dengan jumlah data yang diklasifikasi *positive*. Perhitungan nilai *precision* dapat diperoleh dengan persamaan (2.16) [70].

$$Presisi = \frac{TP}{TP + FP} \tag{2.16}$$

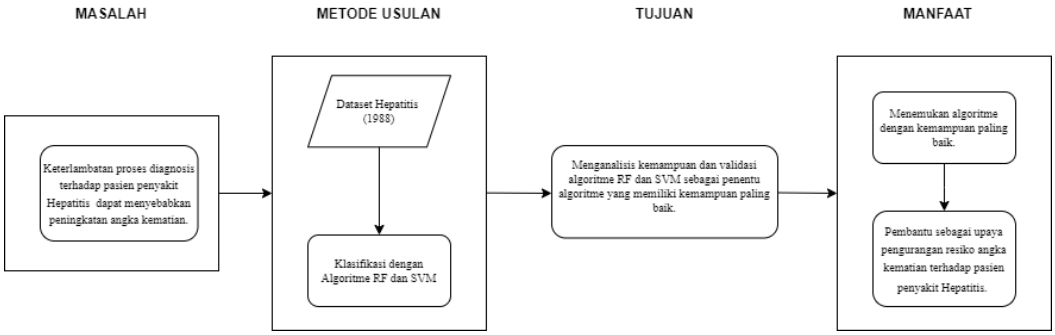
c. *Recall*

Recall merupakan pengukuran yang digunakan untuk membandingkan total data yang benar-benar bernilai *die* atau *live*. Perhitungan nilai *recall* dapat diperoleh dengan persamaan (2.17) [70].

$$Recall = \frac{TP}{TP + FN} \tag{2.17}$$

2.3 Kerangka Berpikir

Gambar 2.2 menunjukkan kerangka berpikir dari penelitian ini yang meliputi masalah, metode usulan, tujuan, dan manfaat. Masalah dari penelitian ini adalah hepatitis merupakan jenis penyakit menular yang dapat menyebabkan kematian. Proses diagnosi yang mengalami keterlambatan dapat berakibat terhadap angka kematian yang relatif tinggi dalam waktu yang relatif singkat. Usulan pada penelitian ini adalah melakukan pengklasifikasian dengan algoritme *RF* dan *SVM* terhadap dataset hepatitis dengan tujuan menganalisis kemampuan dan validasi algoritme *RF* dan *SVM* sebagai penentu algoritme yang memiliki kemampuan paling baik. Manfaat yang diperoleh dari penelitian ini adalah sebagai pencari algoritme terbaik guna mempercepat proses diagnosis yang diharapkan dapat meminimalisir angka kematian terhadap pasien penyakit hepatitis.



Gambar 2. 2 Kerangka Berpikir