ANALISA 4 ALGORITMA DALAM KLASIFIKASI PENYAKIT LIVER MENGGUNAKAN RAPIDMINER

Annisa Putri Ayudhitama[1], Utomo Pujianto[2]

[1,2] Teknik Elektro, Fakultas Teknik, Universitas Negeri Malang
[1]ichadhitama@gmail.com, [2] utomo.pujianto.ft@um.ac.id

Abstrak

Hati merupakan salah satu organ penting dalam tubuh manusia yang berfungsi untuk detoksifikasi racun atau penetral racun dari segala sesuatu yang masuk ke dalam tubuh kita, sehingga tubuh menjadi lebih sehat. Hati dapat terserang suatu penyakit yang mampu mengganggu tugasnya, apabila penyakit hati sudah menyerang maka racun akan tersebar ke seluruh tubuh dan membuat tubuh menjadi tidak sehat. Penyakit *liver* merupakan penyakit hati yang disebabkan oleh virus, alkohol, pola hidup dan lainnya. Menurut data WHO *(World Health Organization)* menunjukkan hampir 1,2 juta orang per tahun khususnya di Asia Tenggara dan Afrika mengalami kematian akibat terserang penyakit *liver*. Seseorang sering tidak menyadari atau terlambat mengetahui penyakit liver sehingga ketika diperiksa penyakit liver sudah parah, akan lebih baik apabila dilakukan penanganan lebih awal dengan mengetahui gejala-gejala yang diderita. Data *mining* mampu membantu diagnosa penyakit *liver* dengan lebih mudah terutama untuk membantu para dokter dalam menentukan apakah pasien menderita penyakit *liver* atau tidak, dengan gejala hampir mendekati penyakit *liver*. Proses diagnosa penyakit *liver* dilakukan dengan proses klasifikasi dan hasilnya berupa pasien tersebut menderita *liver* atau tidak. Penelitian ini menggunakan 4 algoritma data *mining* yaitu *Naïve Bayes*, K-*Nearest Neighbor* (KNN), *Decision Tree* dan *Neural Network*. *Dataset* yang digunakan yaitu *Indian Liver Patient Dataset* (ILPD) dari *website* UCI *Machine Learning Repository*. Keempat algoritma tersebut dibandingkan manakah yang lebih baik akurasinya untuk kasus diagnosa penyakit *liver*. Hasilnya menunjukkan bahwa algoritma *Naïve Bayes* memiliki akurasi 55,42%, algoritma K-*Nearest Neigbor* memiliki akurasi 66,03%, algoritma *Decision Tree* memiliki akurasi 72,74%, dan algoritma *Neural Network* memiliki akurasi 69,64%. Akurasi tersebut tergolong rendah karena kelas atau label antara pasien penyakit *liver* dan pasien tidak memiliki *liver* tidaklah seimbang, kelas pasien penyakit *liver* lebih banyak dibandingkan pasien tidak memiliki *liver*, sehingga banyak data yang diklasifikasikan sebagai pasien penyakit *liver*.

Kata kunci : **Data Mining, *Decision Tree*, Klasifikasi, KNN, *Liver, Naïve Bayes, Neural Network***

1. Pendahuluan

Di dalam tubuh kita terdapat beberapa organ penting yang masing-masing fungsinya sangat berguna, salah satunya adalah hati. Hati adalah organ di dalam tubuh kita yang memiliki ukuran paling besar dan memiliki fungsi yang sangat penting [1]. Fungsi organ hati yaitu sebagai tempat untuk penyimpanan glikogen, membantu proses pembentukan dan sekresi empedu, sintesa urea, berperan sebagai metabolisme kolesterol dan lemak serta fungsi utama hati sebagai detoksifikasi racun atau penetral racun [2]. Dengan adanya organ hati maka tubuh kita akan terhindar dari berbagai racun yang mampu mengganggu kesehatan. Organ hati juga dapat terserang penyakit yang mengakibatkan hati tidak mampu berfungsi seperti biasanya bahkan menyebabkan kematian, penyakit liver merupakan penyakit hati yang sudah lama ada dan cukup umum di masyarakat. Menurut data WHO *(World Health Organization)* menunjukkan hampir 1,2 juta orang per tahun khususnya di Asia Tenggara dan Afrika mengalami kematian akibat terserang penyakit liver. Faktor penyebab penyakit liver diantaranya kelainan hati yang sudah aja sejak lahir, adanya infeksi virus atau bakteri, kecanduan alkohol, merokok aktif dan pola hidup yang buruk dan masih banyak lainnya [3]. Apabila penyakit liver menyerang dan merusak organ hati maka kemampuan tubuh perlahan akan menurun tertama kemampuan untuk menetralisir racun yang masuk ke tubuh kita, hal tersebut akan membahayakan tubuh jika tidak segera ditangani.

Selama ini banyak orang yang tidak menyadari apakah ia terkena penyakit liver atau tidak walaupun memiliki gejala liver, bahkan diantara mereka tidak berusaha untuk datang ke dokter dan memeriksa keluhannya. Hampir semua orang mengalami keterlambatan penanganan karena mereka baru memeriksakan ketika penyakit liver sudah parah. Untuk mengatasi permasalahan tersebut diperlukan sebuah sistem yang mampu menentukan apakah seseorang tersebut tergolong sebagai pasien liver

atau tidak sehingga dapat dilakukan pemeriksaan lebih dini dan secara rutin agar penanganan penyakit *liver* dapat dilakukan dengan cepat bagi penderitanya. Sistem tersebut mampu menghasilkan klasifikasi dengan bantuan algoritma data *mining*.

Klasifikasi merupakan salah satu metode data *mining supervised learning*, dimana membutuhkan data pelatihan yang sudah diberi kelas label sebagai pembelajaran untuk memperkirakan kelas dari suatu objek yang belum diketahui kelasnya. Pada data *mining* terdiri dari banyak algoritma yang dapat digunakan untuk proses klasifikasi yaitu *Naïve Bayes, Decision Tree, K-Nearest Neighbor* (KNN), *Neural Network, Support Vector Machine* (SVM) dan masih banyak algoritma data *mining* lainnya. Sebelum menentukan algoritma apa yang ingin kita pakai dalam sebuah kasus klasifikasi, akan lebih baik jika kita mencari tahu terlebih dahulu diantara algoritma tersebut manakah algoritma yang paling bagus dan memiliki akurasi yang tinggi. Akurasi yang tinggi sangat berpengaruh bagi suatu algoritma karena jika algoritma memiliki akurasi yang tinggi dalam menyelesaikan sebuah kasus klasifikasi maka dapat dikategorikan klasifikasi tersebut tergolong berhasil dengan hasil akurat dan tepat. Pada kasus ini proses klasifikasi dengan beberapa algoritma data *mining* berfungsi untuk menentukan apakah pasien tersebut terkena penyakit *liver* atau tidak dengan dataset ILPD (*Indian Liver Patient Dataset*), serta membandingkan diantara beberapa algoritma yang digunakan manakah algoritma terbaik untuk menyelesaikan kasus ini.

Sebelumnya telah ada penelitian klasifikasi penyakit *liver* menggunakan berbagai algoritma data *mining*. Penelitian yang dilakukan oleh Kalyan Nagaraj menunjukkan bahwa klasifikasi penyakit *liver* menggunakan algoritma *NeuroSVM* (gabungan SVM dan *artificial neural network*) menghasilkan akurasi yang tinggi sebesar 98,83% [4]. Penelitian yang dilakukan oleh Bedi Venkata Ramana menunjukkan bahwa kasifikasi penyakit *liver* menggunakan algoritma *Naïve Bayes*, C4.5, *backpropagation*, *K-Nearest Neighbor, Support Vector Machine* dapat dilakukan dan memperoleh algoritma terbaik dalam kasus tersebut yaitu *backpropagation, K-Nearest Neighbor, Support Vector Machine* dengan akurasi yang cukup tinggi [5]. Penelitian yang dilakukan oleh Dr.S.Vijayarani menunjukkan bahwa kasifikasi penyakit *liver* menggunakan algoritma SVM dan *Naïve Bayes* dapat dilakukan dan memperoleh algoritma terbaik dalam kasus tersebut yaitu *Support Vector Machine* (SVM) dengan akurasi yang cukup tinggi [6].

Pada penelitian ini menggunakan 5 algoritma data *mining* untuk klasifikasi penyakit *liver* dengan dataset yang diambil dari UCI *Machine Learning Repository* yaitu ILPD (*Indian Liver Patient Dataset*), melakukan perbandingan diantara kelima algoritma tersebut manakah yang cocok digunakan untuk menyelesaikan kasus ini. Adapun 5 algoritma tersebut yaitu *Naïve Bayes, Decision Tree, K-Nearest Neighbor, Neural Network, Support Vector Machine*. Pada penelitian ini menggunakan aplikasi *RapidMiner* 9.1 yang akan menunjukkan akurasi masing – masing algoritma.

2. Metodologi Penelitian

A. *Indian Liver Patient Dataset*

Pada penelitian ini menggunakan dataset yang didapat dari *website* UCI *Machine Learning Repository* yaitu *Indian Liver Patient Dataset* (ILPD). Dataset ini berisi data yang dikumpulkan dari para pasien yang ada di timur laut Andhra Pradesh, India. Dataset berisi 416 pasien penderita liver sedangkan 167 pasien bukan penderita liver, data pasien pria yaitu 441 dan 142 pasien wanita. Dataset ini memiliki 11 atribut dimana 10 atribut merupakan atribut biasa sedangkan 1 atribut sebagai *class* atau label yang ditunjukkan pada tabel 1, dan memiliki 583 *instance* atau isi data. *Instance* pada dataset ini tidak memiliki nilai *missing* sehingga semua atribut berisi nilai dalam kondisi baik dan siap diproses.

Tabel 1. Keterangan Atribut

Atribut	Keterangan
Age	Umur pasien
Gender	Jenis kelamin pasien (Female: 0, Male:1)
TB	Total bilirubin pasien
DB	Direct bilirubin pasien
Alkphos	Alkaline phospotase
Sgpt	Alamine aminotransferase
Sgot	Aspartate aminotransferase
TP	Total protiens
ALB	Albumin
A/G Ratio	Albumin dan Globulin ratio
Dataset	*Selector/Class/Label* untuk menentukan apakah pasien terkena liver atau tidak

Diantara beberapa atribut diatas, terdapat beberapa atribut penting yang mampu menentukan apakah pasien tersebut menderita penyakit hati. Adapun atribut penting yang menentukan pasien terkena penyakit liver yaitu : 1) peningkatan total bilirubin (TB); 2) peningkatan SGPT (*Alamine Aminotransferase*); 3) peningkatan SGOT (*Asparte Aminotransferase*); 4) penurunan albumin (ALB) [7].

B. Klasifikasi

Klasifikasi merupakan proses untuk menemukan suatu kelas data suatu objek yang belum diketahui berdasarkan data sebelumnya [8], klasifikasi termasuk ke dalam metode pembelajaran atau *supervised* karena membutuhkan pembelajaran data sebelumnya untuk menentukan hasil dari data

baru. Klasifikasi memiliki 4 komponen dasar yaitu : 1) class, merupakan variabel yang menjadi label atau hasil suatu objek; 2) *predictor*, merupakan variabel yang menjadi atribut dari data yang akan digunakan pada klasifikasi; 3) *training dataset*, merupakan dara yang telah memiliki label sebelumnya; 4) *testing dataset*, merupakan data baru yang akan dilakukan proses klasifikasi.

C. Data *Mining*

Data *mining* (*Knowledge Discovery in Database* (KDD) adalah suatu cara yang dilakukan untuk mengolah data dalam jumlah yang besar agar mendapatkan pengetahuan baru [8]. Data mining memiliki 4 fungsi dasar yang biasa kita temui yaitu:

1) Fungsi Klasifikasi, fungsi ini merupakan fungsi untuk menemukan model/fungsi untuk menggambarkan suatu class dari data. Dengan klasifikasi mampu membelajari data sehingga mampu meramalkan kecenderungan data
2) Fungsi Prediksi, fungsi ini digunakan untuk menemukan pola data menggunakan variabel dalam memprediksi variabel lain yang belum diketahui jenisnya
3) Fungsi Deskripsi, fungsi ini digunakan untuk menemukan karakteristik data
4) Fungsi Asosiasi, fungsi ini digunakan untuk menemukan hubungan yang ada pada nilai atribut dari banyaknya data

Dalam data *mining* terdapat beberapa tahapan yang perlu kita lakukan yaitu:

1) Pembersihan Data, tahap ini merupakan tahap untuk menghilangkan suatu *noise* pada data. Sebagian besar data yang kita peroleh memiliki nilai yang hilang atau *missing*, dengan data *mining* maka data *missing* mampu dihilangkan
2) Integrasi Data, tahap ini merupakan tahap untuk menggabungkan data dari beberapa *database* ke satu *database* yang baru
3) Seleksi Data, tahap ini merupakan tahap untuk menyeleksi data pada *database* yang tidak semuanya dipakai. Data yang sesuai saja yang akan diambil dari *database* untuk diproses
4) Transformasi Data, tahap ini merupakan tahap untuk mengubah dan menggabung data ke dalam format yang sesuai
5) Proses *mining*, tahap ini merupakan tahap untuk menemukan ilmu pengetahuan dari sebuah data
6) Evaluasi Pola, tahap ini merupakan tahap untuk melakukan identifikasi pola atau model prediksi ke dalam data mining
7) Presentasi pengetahuan, tahap ini merupakan tahap untuk memvisualisasikan dan menjadikan pengetahuan tentang metode yang digunakan

D. *Cross Validation*

Cross validation adalah salah satu metode dalam statistika untuk mengevaluasi dan membandingkan algoritma, dengan metode ini maka tingkat akurasi sebuah algoritma dapat ditampilkan dalam melakukan proses klasifikasi terutama kasus identifikasi liver [9]. Pada metode *cross validation* terdapat 2 *segmen* yaitu: 1) data latih, berfungsi membentuk model regresi; 2) data uji, berfungsi memvalidasi terhadap model dari data latih.

E. *Compare ROCs*

Kurva ROC merupakan visualisasi untuk menilai hasil prediksi data dengan menggunakan 2 *class* sebagai keputusan, TP (*True Positif*) berada di sumbu Y sedangkan FP (*False Positif*) berada di sumbu X [10]. Ukuran kualitas *classifier* dengan ROC ada pada tabel 1.

Tabel 2. Kualitas *Classifier*

Rentang Akurasi	Kualitas *Classifier*
0.90 – 1.00	*Excellent*
0.80 – 0.90	*Good*
0.70 – 0.80	*Fair*
0.60 – 0.70	*Poor*
0.50 – 0.60	*Failure*

Compare ROCs merupakan salah satu *tools* dalam RapidMiner yang digunakan untuk perbandingan performa beberapa algoritma dalam klasifikasi kasus penelitian ini, pada *compare ROCs* memiliki rentang nilai mulai 0 sampai 1. Pada *compare ROCs* terdapat 2 parameter yaitu *number of folds* dan *split ratio. Number of folds* berfungsi untuk pembagian data untuk diuji dengan kurva ROC, nilai default untuk *number of folds* yaitu 10 dengan perbandingan 9 sebagai data latih dan 1 sebagai data uji. Sedangkan parameter *split ratio* berfungsi untuk pembagian data uji dan data latih, nilai default untuk *split ratio* yaitu 0,7 sebagai data latih dan 0,3 sebagai data uji.

Hasil dari *Compare ROCs* yaitu menampilkan kurva masing–masing algoritma, dari kurva tersebut kita akan mengetahui manakah kurva yang memiliki performa terbaik dalam klasifikasi kasus identifikasi *liver*. Apabila salah satu kurva algoritma mendekati angka 1 pada sumbu Y (*True Positif*), maka algoritma tersebut tergolong baik dalam menghasilkan klasifikasi kasus *liver*.

F. *Confussion Matrix*

Confussion matrix merupakan metode yang digunakan dalam perhitungan akurasi data mining, bentuk metode ini berupa tabel. Pada tabel terdapat 4 *record* yaitu : 1) *true positif* (TP), merupakan banyak data nilai sebenarnya adalah positif dengan kelas prediksi sebagai nilai positif; 2) *false positif* (FP), merupakan banyak data nilai sebenarnya adalah negatif dengan kelas prediksi sebagai nilai

positif; 3) *false negatives* (FN), merupakan banyak data nilai sebenarnya adalah positif dengan kelas prediksi sebagai nilai negatif; 4) *true negatives* (TN), merupakan *record* data negatif dengan klasifikasi sebagai nilai negatif.

Tabel 3. *Confussion Matrix*

		Nilai Sebenarnya	
		TRUE (Liver)	FALSE (NonLiver)
Nilai Prediksi	TRUE (Liver)	TP	FP
	FALSE (NonLiver)	FN	TN

Didalam *confussion* matrix umumnya ditampilkan 3 metode pengujian yaitu:

1. Accuracy

Accuracy atau akurasi adalah salah satu metode pengujian suatu algoritma berdasarkan tingkat kedekatan antara nilai prediksi dengan nilai sebenarnya. Rumus menghitung akurasi yaitu:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} * 100\% \qquad (1)$$

2. Precision

Precision atau presisi adalah salah satu metode pengujian suatu algoritma dengan melakukan suatu perbandingan data benar yang diperoleh sistem dengan jumlah seluruh data yang diambil oleh sistem yang benar maupun salah. Rumus menghitung presisi yaitu:

$$Precision = \frac{TP}{TP+FP} * 100\% \qquad (2)$$

3. Recall

Recall adalah salah satu metode pengujian suatu algoritma dengan melakukan perbandingan jumlah data benar yang diperoleh sistem dengan jumlah seluruh data benar yang diambil atau tidak diambil oleh sistem. Rumus menghitung *recall* yaitu:

$$Recall = \frac{TP}{TP+FN} * 100\% \qquad (3)$$

G. *Naïve Bayes*

Naïve bayes merupakan salah satu algoritma *supervised learning* dalam data mining yang paling populer, algoritma ini menggunakan konsep teorema bayes atau probabilitas untuk memprediksi kelas data [11]. *Naïve bayes* dikenal sebagai algoritma dengan perhitungan yang sederhana, jika digunakan untuk klasifikasi maka algoritma ini lebih dikenal dengan sebutan *naïve bayes classifier*. Kelebihan algoritma *naive bayes* yaitu : 1) mudah dalam pengimplementasiannya karena tidak memerlukan optimasi numerik, matriks dan lainnya; 2) tergolong efisien dalam pelatihan serta penggunaannya; 3) data *binary* atau polinom dapat digunakan; 4) sifat independen sehingga mampu diimplementasikan dengan bermacam – macam dataset; 5) hasil akurasi relatif tinggi. Sedangkan kekurangan *naive bayes* yaitu tidak akuratnya perkiraan kemungkinan kelas dan harus menentukan batasan/*threshold* secara manual.

Adapun rumus dari algoritma *naïve bayes* yaitu:

$$p\ (H|E) = \frac{p(E|H)x\ p(H)}{p(E)} \qquad (4)$$

Dimana penjabaran rumus diatas yaitu : 1) p(H|E) menunjukkan probabilitas hipotesis H dapat terjadi apabila *evidence* E terjadi; 2) p(E|H) menunjukkan probabilitas kemunculan *evidence* E apabila hipotesis H terjadi; 3) p(H) menunjukkan probabilitas dari hipotesis H tanpa memandang *evidence* apapun; 4) p(E) menunjukkan probabilitas dari *evidence* E tanpa memandang apapun.

Alur pada algoritma *naïve bayes* yaitu [12] :

1. Membaca data latih (*data training*)
2. Menghitung jumlah dan probabilitas tetapi apabila data berupa numerik maka:
   a) Mencari nilai mean dan standar deviasi pada masing – masing parameter data numerik
   b) Mencari nilai probabilistik, caranya hitung jumlah datayang sesuai dari kategori yang sama dibagi dengan jumlah data pada kategori tersebut
3. Mendapatkan nilai dalam tabel *mean, standart deviasi* dan probabilitas

H. *K-Nearest Neighbor (k-NN)*

K-Nearest Neighbor (k-NN) merupakan salah satu algoritma *supervised learning* dalam data mining yang tidak kalah populer *dengan naïve bayes*, algoritma ini memiliki konsep dengan melihat jarak tetangga terdekat dengan data baru [13]. Data latih diletakkan pada kelas dimana ruangan ini merepresentasikan fitur dari data, ruang tersebut terbagi menjadi banyak bagian sesuai dengan klasifikasi data latih. K-Nearest Neigbor akan mencari jarak yang paling dekat antara data uji dengan k tetangga terdekat pada data latih. Kelebihan dari algoritma ini yaitu pelatihan dari data latihnya sangat cepat, sederhana, mudah dipelajari, tahan akan data pelatihan yang mengandung derau serta tetap efektif walaupun data latihnya besar. Sedangkan kekurangan dari algoritma ini yaitu nilai k nya bias, komputasi yang dibutuhkan kompleks, memori terbatas dan mudah tertipu apabila ada atribut yang tidak relevan.

Alur pada algoritma *K-Nearest Neigbor* :

1. Mencari nilai bobot dari setiap atribut pada kelas yang ada dan menghitung bobot dengan keanggotaan kelas yang ada pada data latih. Rumus mencari rata – rata bobot untuk setiap atribut yaitu :

   $$w = \frac{x1+x2+x3+\cdots+xn}{n} \qquad (5)$$

   Dimana penjabaran rumus diatas yaitu : 1) w, menunjukkan rata – rata bobot tiap atribut; 2) xn, menunjukkan data masukan ke-n tiap atribut; 3) n, menunjukkan jumlah data
2. Hitung jarak *euclidean* dengan rumus :

$$di = \sum_{i=1}^{p} w\,(x_{2i} - x_{1i})^2 \quad (6)$$

Dimana penjabaran rumus diatas yaitu : 1) w, menunjukkan nilai bobot dari setiap inputan; 2) x1, menunjukkan nilai data latih; 3) x2, menunjukkan nilai data uji; 4) i, menunjukkan variabel data; 5) d, menunjukkan nilai jarak; 6) p, menunjukkan dimensi data

3. Untuk menentukan label dari data baru maka ambil sebanyak k tetangga terdekatdari label kelas tetangga sebelumnya

I. *Decision Tree*

Decision tree (pohon keputusan) merupakan salah satu algoritma *supervised learning* dalam data *mining*, algoritma ini berisi berbagai faktor yang dapat digunakan sebagai pemecah masalah [14]. Penggunaan algoritma *decision tree* sebagai pemecah masalah klasifikasi tergolong sangat baik karena kita dapat mengetahui hanya berdasarkan pola bentuk pohonnya. Kelebihan dari algoritma ini yaitu mudah untuk dimengerti, fleksibel dan memiliki tampilan menarik karena digambarkan dalam bentuk pohon. Sedangkan kekurangan algoritma ini yaitu sering terjadi overlap apabila jumlah datanya sangat banyak, menentukan desain pohon keputusan yang optimal dirasa masih sulit, hasil kualitas bergantung pada desain pohon keputusan.

Alur pada algoritma *decision tree* yaitu:

1. Siapkan data latih dan data uji
2. Menghitung *entropy* untuk menentukan akar pohon dengan rumus:

$$Entropi\,(S) = \sum_{j=1}^{k} - pj\ \log_2 pj \quad (7)$$

Dimana penjabaran rumus diatas yaitu : 1) S, menunjukkan himpunan kasus; 2) k, menunjukkan jumlah partisi S; 3) pj, menunjukkan probabilitas yang didapat dari hasil jumlah Ya/Tidak dibagi dengan total kasus

3. Menghitung nilai gain, nilai gain yang paling tinggi akan menjadi akar pohon pertama, rumus gain:

$$Gain\,(S,A) = Entrophy\,(s) - \sum_{i=1}^{n} - \frac{|si|}{|s|} * Entrophy\,(si) \quad (8)$$

Dimana penjabaran rumus diatas yaitu : 1) S, menunjukkan himpunan kasus; 2) A, menunjukkan atribut; 3) n, menunjukkan jumlah partisi atribut A; 4) |Si|, menunjukkan jumlah kasus pada partisi ke –I; 5) |S|, menunjukkan jumlah kasus dalam S

4. Mengulangi langkah ke 2 sampai semua *record* terpartisi
5. Partisi akan berhenti apabila :
   a) Semua *record* yang ada pada simpul N mendapatkan kelas yang sama
   b) Tidak adanya atribut pada *record* yang dipartisi
   c) Tidak adanya *record* pada cabang yang kosong

J. *Neural Network*

Neural network (jaringan syaraf tiruan) merupakan salah satu algoritma data *mining* dengan konsep menirukan fungsi otak manusia [15]. Pada algoritma ini memiliki *neuron* yaitu berbagai unit pengolah berukuran kecil yang ada di otak, *neuron* saling berhubungan satu sama lain dengan koneksi *neuron*. Satu *neuron* mengambil input dari 1 set *neuron*, kemudian *output* akan dikumpulkan oleh *neuron* lain untuk diproses lebih lanjut. Kelebihan algoritma ini yaitu mampu melakukan pekerjaan berdasarkan data yang telah diberikan, mampu membuat representasi dari informasi yang telah diterima, dapat melakukan perhitungan secara paralel. Sedangkan kekurangan algoritma ini yaitu tidak efektif dalam operasi numerik dengan presisi tinggi, tidak efisien dalam operasi algoritma aritmatik serta logika dan simbolik, waktu pelatihan akan sangat lama apabila jumlah datanya besar.

Alur pada algoritma *neural network*:

1. Menginsialisasi bobot jaringan secara acak
2. Menghitung input untuk simpul, berdasarkan nilai input dan bobot jaringan, rumus:

$$input_j = \sum_{i=1}^{n} O_i W_{ij} + \theta_j \quad (9)$$

Dimana penjabaran rumus diatas yaitu: 1) O_i, menunjukkan output simpul I dari layer sebelumnya; 2) W_{ij}j, menunjukkan bobot relasi dari simpul i pada layer sebelumnya ke simpul j; 3) Θ_j, menunjukkan bias untuk pembatas

3. Bangkitkan *output* untuk simpul dengan fungsi aktifasi *sigmoid*, rumus:

$$Output = \frac{1}{1+ e^{-input}} \quad (10)$$

4. Menghitung nilai *error* antara nilai prediksi dengan nilai sesungguhnya, rumus:

$$Error_j = Output_j\big(1 - Output_j Target_j - Output_j\big) \quad (11)$$

Dimana penjabaran rumus diatas yaitu: 1) $Output_j$, menunjukkan aktual dari simpul j; 2) $Target_j$, menunjukkan nilai target yang diketahui pada data latih

5. Menghitung nilai *error* pada *hidden* layer, rumus :

$$Error_j = Output_j(1 - Output_{jk} = 1nErrorkW_{jk}) \quad (12)$$

Dimana penjabaran rumus diatas yaitu : 1) $Output_j$, menunjukkan output aktual dari simpul j; 2) $Error_k$, menunjukkan error simpul k; 3) W_{jk}, menunjukkan bobot relasi dari simpul j ke simpul k pada layer selanjutnya

6. Memperbarui nilai bobot relasi, rumus :

$$W_{ij} = W_{ij} + l.Error_j.Output_j \quad .(13)$$

Dimana penjabaran rumus diatas yaitu : 1) W_{ij}, menunjukkan bobot relasi dari unit i pada layer sebelumnya ke unit j; 2) l, menunjukkan *learning rate* (konstanta, memiliki nilai antara

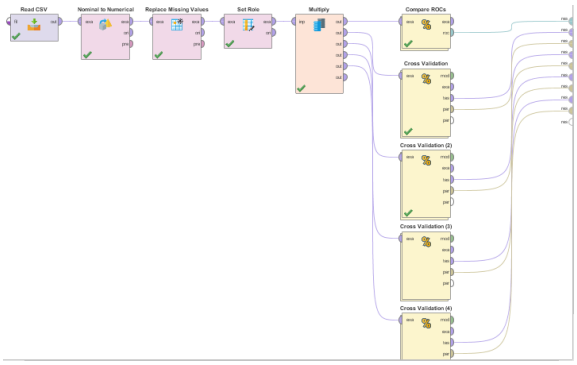
0 sampai 1); 3) $Error_j$, menunjukkan *error* pada *output layer* simpul j; 4) $Output_j$, menunjukkan output dari simpul i.

3. Hasil dan Pembahasan

Performa suatu algoritma dalam menyelesaikan masalah klasifikasi dapat diketahui dengan melakukan pengukuran, salah satu cara yang paling umum dilakukan yaitu menghitung akurasi dari algoritma. Apabila akurasi suatu algoritma dikatakan tinggi bukan berarti algoritma tersebut dikatakan bagus untuk penyelesaian klasifikasi, perlu diketahui terlebih dahulu apakah algoritma tersebut mampu mendeteksi pasien yang menderita penyakit *liver* dalam jumlah banyak atau malah lebih banyak mendeteksi pasien yang tidak terkena *liver*. Akan lebih baik apabila akurasi algoritma terhitung rendah namun algoritma tersebut mampu mendeteksi lebih banyak pasien yang menderita penyakit liver.

Algoritma *Naïve Bayes* dan *Decision Tree* mampu mendukung klasifikasi data dengan perolehan akurasi yang baik apabila tipe datanya nominal atau huruf. Sedangkan algoritma *K-Nearest Neighbor* dan *Neural Network* mampu mendukung klasifikasi data dengan perolehan akurasi yang baik apabila tipe datanya numerik atau angka. Hal tersebut membuktikan bahwa tipe data sangatlah berpengaruh dalam menyelesaikan masalah klasifikasi menggunakan algoritma data *mining*.

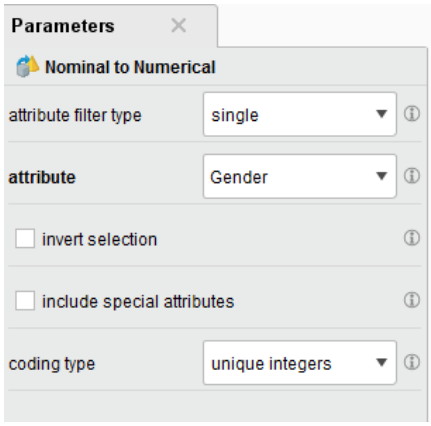
Proses implementasi klasifikasi *liver* menggunakan 4 algoritma yaitu *Naïve Bayes, K-Nearest Neighbor, Decision Tree* dan *Neural Network* berupa bentuk skema pada *RapidMiner Studio* 9.1 ditunjukkan pada gambar 1. Ketika skema tersebut dijalankan atau di *running* perlu waktu sekitar 59 detik untuk memunculkan hasil klasifikasi, waktu yang diperlukan tergantung dengan spesifikasi laptop yang dimiliki.



Gambar 1. Skema Klasifikasi *Liver* pada *RapidMiner* 9.1

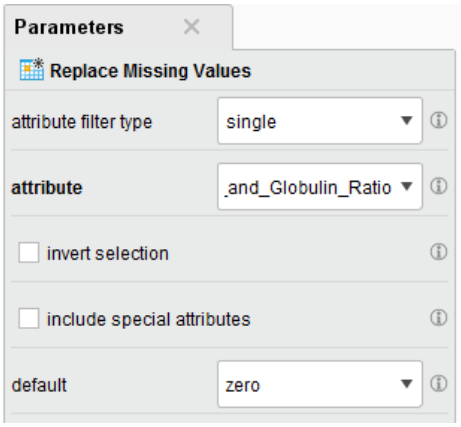
Implementasi klasifikasi penyakit *liver* dengan 4 algoritma data mining pada *RapidMiner* 9.1 menggunakan 7 operator yang telah disediakan di *RapidMiner*. Adapun operator dan penjelasannya yaitu:

1. *Read* CSV, operator yang digunakan untuk membaca *file* CSV yang telah diimport. Data *Indian Liver Patient Dataset* (ILPD) diambil dari UCI *Machine Learning Repository* dan sudah dalam format CSV
2. *Nominal to Numerical*, operator yang digunakan untuk mengubah tipe data nominal menjadi tipe data numerik. Pada operator ini terdapat parameter "*attribute filter type*" lalu pilih *single*, pada parameter ini digunakan untuk memilih tipe *filter* atribut dan pada data *liver* hanya perlu mengubah 1 atribut yaitu *Gender*. Lalu parameter "*attribute*" pilih atribut yang akan diubah tipe datanya. Pada data *liver*, atribut *Gender* memiliki tipe nominal dan harus diubah menjadi data numerik karena algoritma kNN dan *Neural Network* tidak mau menerima data nominal. Jadi *gender* "*Female*" diubah menjadi angka 0, *gender* "*Male*" diubah menjadi angka 1. Seperti pada gambar 2



Gambar 2. Parameter Nominal to *Numerical*

3. *Replace Missing Value*, operator yang digunakan untuk memberikan atau mengisi nilai apabila ada data pada atribut yang kosong sehingga tidak menyebabkan suatu kesalahan atau *error*. *Replace Missing Value* dipilih apabila jumlah data yang kosong tidak lebih dari 1/3 total data. Pada operator ini terdapat parameter *attribut filter type* dan pilih *single* apabila atribut yang akan diisi nilainya hanyalah satu atribut saja. Lalu *attribute* digunakan untuk memilih atribut mana yang akan diisi nilainya, kemudian *default* yang digunakan untuk memilih nilai apakah yang akan diisi pada data atribut yang kosong, pada penelitian ini dipilih *zero*. Sehingga data yang kosong pada atribut "*Albumin* and *Gobulin Ratio*" akan diisi dengan nilai 0.



Gambar 3. Parameter *Replace Missing Value*

4. *Set Role*, operator yang digunakan untuk menentukan atrribut kelas atau label dari data. Pada operator ini terdapat parameter "*attribute name*" digunakan untuk menentukan atribut data yang akan dijadikan kelas atau label, pada data liver ini kelas atau labelnya adalah atribut Dataset. Lalu parameter "*target role*" pilih label yang digunakan untuk menentukan bahwa atribut tersebut adalah kelas label
5. *Multiply*, operator yang digunakan untuk menghubungkan banyak operator agar bisa dijalankan secara bersamaan
6. *Compare* ROCs, operator yang digunakan untuk menampilkan kurva ROCs kinerja dari masing-masing algoritma. Didalam *Compare* ROCs diberikan operator 4 algoritma yang digunakan untuk penelitian ini
7. *Cross Validation*, operator yang digunakan untuk menampikan seberapa akurat kinerja dari algoritma. Pada implementasi klasifikasi *liver* terdapat 4 operator *cross validation* karena hanya menggunakan 4 algoritma data *mining*, jadi jumlah *cross validation* tergantung dengan jumlah algoritma yang digunakan. Didalam operator ini terdapat operator algoritma, operator *apply* model yang digunakan untuk mengaplikasikan model data *training* ke data *testing*, dan operator *performance* digunakan untuk melakukan evaluasi algoritma. Hasil performance yaitu *accuracy*, *precision*, *recall* dan berbentuk *confussion matrix*.

Ketika skema pada *RapidMiner* selesai di *running* maka akan menampilkan hasil akurasi dari keempat algoritma yang digunakan untuk klasifikasi liver. Adapun hasil pengukuran akurasi, presisi dan *recall* masing – masing algoritma yaitu :

1. *Naive Bayes*

   Akurasi :

accuracy: 55.44% +/- 3.90% (micro average: 55.44%)

	true 1	true 2	class precision
pred. 1	165	9	94.83%
pred. 2	249	156	38.52%
class recall	39.86%	94.55%	

Perhitungan *akurasi menggunakan rumus :*

Akurasi $= \frac{165+156}{165+156+9+249}$

$= 321 / 579$

$= 55{,}44\%$

Presisi :

Presisi $= \frac{165}{165+9}$

$= 165 / 174$

$= 94{,}83\%$

Recall :

Recall $= \frac{165}{165+249}$

$= 164 / 414$

$= 39{,}86\%$

2. *k-Nearest Neighbor*

   a) Akurasi :

accuracy: 66.32% +/- 5.66% (micro average: 66.32%)

	true 1	true 2	class precision
pred. 1	328	109	75.06%
pred. 2	86	56	39.44%
class recall	79.23%	33.94%	

Perhitungan akurasi menggunakan rumus :

Akurasi $= \frac{328+56}{331+56+109+86}$

$= 384 / 579$

$= 66{,}32\%$

b) Presisi :

Presisi $= \frac{328}{328+109}$

$= 328 / 437$

$= 75{,}06\%$

c) *Recall :*

Recall $= \frac{328}{328+86}$

$= 328 / 414$

$= 79{,}23\%$

3. *Decision Tree*

   a) Akurasi :

accuracy: 72.89% +/- 1.83% (micro average: 72.88%)

	true 1	true 2	class precision
pred. 1	410	153	72.82%
pred. 2	4	12	75.00%
class recall	99.03%	7.27%	

Perhitungan akurasi menggunakan rumus :

Akurasi $= \frac{410+12}{410+12+153+4}$

$= 422 / 579$

$= 72{,}89\%$

b) Presisi :

Presisi $= \frac{410}{410+153}$

$= 410 / 563$

$= 72{,}82\%$

c) *Recall :*

$$Recall = \frac{410}{410+4}$$
$$= 410 / 414$$
$$= 99{,}03\%$$

4. *Neural Network*

a) Akurasi :

accuracy: 70.81% +/- 2.70% (micro average: 70.81%)

	true 1	true 2	class precision
pred. 1	382	137	73.60%
pred. 2	32	28	46.67%
class recall	92.27%	16.97%	

Perhitungan akurasi menggunakan rumus :

$$Akurasi = \frac{382+28}{382+28+137+32}$$
$$= 410 / 579$$
$$= 70{,}81\%$$

b) Presisi :

$$Presisi = \frac{382}{382+137}$$
$$= 382 / 519$$
$$= 73{,}60\%$$

c) *Recall :*

$$Recall = \frac{382}{382+32}$$
$$= 382 / 414$$
$$= 92{,}27\%$$

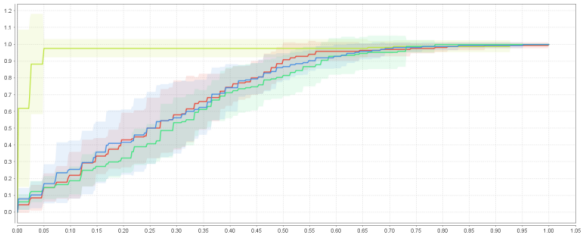
Bentuk tabel dari hasil pengukuran akurasi, presisi dan *recall* dari keempat algoritma diatas yaitu:

Tabel 4. Hasil Pengukuran Algoritma

Algoritma	Akurasi	Presisi	*Recall*
Naïve Bayes	55,44%	94,83%	39,86%
k-Nearest Neighbor	66,32%	75,06%	79,23%
Decision Tree	72,89%	72,82%	99,03%
Neural Network	70,81%	73,60%	92,27%

Berdasarkan tabel 4, dapat dilihat bahwa hasil keempat algoritma untuk menyelesaikan masalah klasifikasi penyakit *liver* terhitung belum begitu baik. Algoritma *Decision Tree* menghasilkan akurasi yang lebih baik dari algoritma *Naive Bayes*, *k-Nearest Neighbor* dan *Neural Network*. *Decision Tree* memiliki akurasi sebesar 72,89%, selain akurasi *decision tree* lebih tinggi ternyata algoritma *decision tree* juga mampu mengklasifikasikan pasien yang menderita *liver* dengan jumlah yang lebih banyak dibanding algoritma lainnya.

Hasil dari kurva ROCs untuk klasifikasi *liver* menggunakan 4 algoritma ada pada gambar 6 di bawah ini.



Gambar 6. Kurva ROC

Berdasarkan hasil kurva ROC pada gambar 6, dapat dilihat bahwa algoritma *Decision Tree* memiliki grafik ROC yang mendekati angka 1.0 pada sumbu Y yaitu *class True Positif* (TP). Hal tersebut membuktikan bahwa *Decision Tree* merupakan algoritma yang memiliki kualitas klasifikasi "*Excellent*" dengan rentang akurasi sebesar 0.90 – 1.00, yang berarti bahwa algoritma *Decision Tree* memiliki hasil klasifikasi yang bagus dalam menyelesaikan masalah klasifikasi liver.

4. Kesimpulan

Berdasarkan hasil pengujian algoritma *Naïve Bayes*, *k-Nearest Neighbor*, *Decision Tree*, dan *Neural Network* untuk menyelesaikan masalah klasifikasi pasien penderita penyakit *liver* atau bukan menggunakan aplikasi *RapidMiner studio* 9.1. Data diambil dari UCI *Machine Learning Repository* yaitu *Indian Liver Patient Dataset* (ILPD). Menunjukkan hasil bahwa dari keempat algoritma tersebut, algoritma yang paling baik dan cocok untuk klasifikasi pasien liver yaitu *Decision Tree* dengan hasil akurasi sebesar 72,89%. Selain akurasinya paling tinggi, *Decision Tree* juga mampu mengklasifikasi para pasien yang menderita penyakit liver dengan jumlah lebih banyak sehingga terhitung akurat. Selama ini banyak algoritma yang memiliki nilai akurasi tinggi tetapi tidak mampu untuk mengklasifikasi dengan benar bahkan banyak yang mendeteksi pasien tidak mengalami liver padahal data aslinya terkena liver. Pada kurva ROC hanya algoritma *Decision Tree* yang memiliki grafik sumbu Y mendekati nilai 1.00 yang dikategorikan sebagai "*Excellent*" klasifikasi..

Daftar Pustaka:

N. Musyaffa and B. Rifai, "Model Support Vector Machine Berbasis Particle Swarm Optimization Untuk Prediksi Penyakit Liver," *JITK (Jurnal Ilmu Pengetah. Dan Teknol. Komputer)*, vol. 3, no. 2, pp. 189–194, 2018.

P. Widodo, "Rule-Based Classifier untuk Mendeteksi Penyakit Liver," *Bianglala Inform.*, vol. II, no. 1, pp. 71–80, 2014.

C. Y. Gobel, "Sistem Pakar Penyakit Liver Menggunakan K-Nearest Neighbors Algoritm Berbasis Website," vol. 10, pp. 152–159, 2018.

K. Nagaraj and A. Sridhar, "NeuroSVM: A Graphical User Interface for Identification of

Liver Patients Kalyan Nagaraj 1* and Amulyashree Sridhar 2 1*."

B. Venkata Ramana, M. S. P. Babu, and N. . Venkateswarlu, "A Critical Study of Selected Classification Algorithms for Liver Disease Diagnosis," *Int. J. Database Manag. Syst.*, vol. 3, no. 2, pp. 101–114, 2011.

M. S. D. Dr. S. Vijayarani1, "Liver Disease Prediction using SVM and Naïve Bayes Algorithms," *Int. J. Sci. Eng. Technol. Res.*, vol. 4, no. 4, pp. 816–820, 2015.

A. Rosida, "Pemeriksaan Laboratorium Penyakit Hati," *Berk. Kedokt.*, vol. 12, no. 1, p. 123, 2018.

S. Defiyanti, "Integrasi Metode Klasifikasi Dan Clustering dalam Data Mining," *Konf. Nas. Inform.*, no. March 2017, pp. 39–44, 2015.

R. Spasial, "1st ISCO : Konferensi Nasional Statistika 1st ISCO : Konferensi Nasional Statistika," no. September, pp. 8–10, 2015.

N. Frastian, S. Hendrian, and V. H. Valentino, "Komparasi Algoritma Klasifikasi Menentukan Kelulusan Mata Kuliah Pada Universitas," *Fakt. Exacta*, vol. 11, no. 1, p. 66, 2018.

R. G. Rafsanjani, N. Hidayat, and R. K. Dewi, "Diagnosis Penyakit Hati Menggunakan Metode Naive Bayes Dan Certainty Factor," *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 2, no. 11, pp. 4478–4482, 2018.

T. Informatika, U. Malikussaleh, and A. Utara, "PENERAPAN ALGORITMA NAIVE BAYES," vol. 8, no. 1, pp. 884–898, 2014.

H. T. Wijaya and M. T. Furqon, "Penerapan Fuzzy K-Nearest Neighbor (Fknn) Untuk Diagnosa Penderita Liver Berdasarkan Indian Liver Patient Dataset (Ilpd)," 2012.

D. Nilai, F. Yang, and T. Pasti, "IMPLEMENTASI METODE POHON KEPUTUSAN UNTUK KLASIFIKASI DATA," no. June, 2015.

W. Erawati, "Vol. XII No. 2, September 2015 Jurnal Techno Nusa Mandiri," *Techno Nusa Mandiri*, vol. XII, no. 2, pp. 21–26, 2015.