RULE-BASED CLASSIFIER UNTUK MENDETEKSI PENYAKIT LIVER

Pudji Widodo
Program Studi Manajemen Informatika
AMIK Bina Sarana Informatika Yogyakarta
Jalan Ring Road Barat Ambarketawang Gamping Sleman 55294
Telp. (0294)4342536
Email: pudji.piw@bsi.ac.id

Abstrak

Hati merupakan organ yang paling besar dan penting bagi tubuh kita. Kita tidak bisa hidup tanpa hati. Penyakit hati tidak mudah ditemukan pada stadium awal. Penanganan pasien penyakit hati pada stadium awal akan memperpanjang usia pasien. Untuk mendeteksi penyakit hati, pasien harus melakukan tes darah. Program klasifikasi otomatis bisa mengurangi beban kerja dokter. Makalah ini menyajikan penerapan algoritma berbasis rule untuk mendeteksi penyakit hati berdasarakan hasil tes darah. Dataset yang digunakan pada penelitian ini adalah dataset Indian Liver Patient Dataset (ILPD) yang diambil dari UCI Machine Learning Repository. Algoritma berbasis rule yang digunakan pada penelitian ini adalah ZeroR, OneR, RIPPER dan C4.5. Algoritma tersebut dibandingkan berdasarkan empat kriteria, yaitu accuracy, presition, sensitivity dan spesificity. Berdasarkan hasil perbandingan, diperoleh algoritma berbasis rule terbaik dalam mendeteksi penyakit hati.

Kata Kunci: Liver, ZeroR, OneR, RIPPER dan C4.5

Pendahuluan

Hati merupakan organ yang paling besar dan penting bagi kita (Lumongga, 2009). Hati berfungsi sebagai pembentukan dan sekresi empedu, tempat menyimpan glikogen, sintesa urea, metabolisme kolestrol dan lemak dan detoksifikasi racun. Kita tidak bisa hidup apabila hati mengalami kegagalan (Hepatitis B Fondation). Penyakit hati sulit ditemukan pada stadium awal (Lin, 2009). Hati mampu menjaga fungsinya bahkan ketika ada bagian yang mengalami kerusakan. Penanganan penyakit hati pada stadium awal akan memperpanjang usia pasien. Diagnosa awal merupakan permasalahan yang sangat penting dalam penyakit hati. Untuk mendiagnosa penyakit hati perlu dilakukan tes darah. Berdasarkan hasil tes darah tersebut dapat diketahui seorang pasien menderita penyakit hati atau tidak.

Artificial inteligence sudah lama digunakan pada bidang medis sejak awal tahun 1960an (Praetorius, 2006). Awalnya artificial inteligence in medicine (AIM) digunakan sebagai alat untuk membentuk kembali sistem informasi kesehatan. Selama bertahun-tahun, AIM mengalami perubahan perspektif dari tadinya hanya sebagai sistem informasi kesehatan berubah menjadi *Clinical Decision Support System* (CDSS). Sebagai CDSS, AIM digunakan pada sektor kesehatan yang ber-beda-beda, seperti mendukung obat resep pada lingkungan laboratorium dan pendidikan serta untuk diagnosa penyakit. AIM mampu melakukan learning atau pembelajaran suatu pola penyakit (Praetorius, 2006). Salah satu AIM yang paling banyak digunakan adalah machine learning.

Machine learning telah banyak digunakan dalam bidang medis (Kononenko, 2001) untuk menganalisa dataset medis, khususnya pada penyakit liver (Lin, 2009). Berdasarkan penelitian (Lin, 2009) algoritma CART dan CBR dapat digunakan untuk mendiagnosa penyakit liver dengan tingkat akurasi 92,94% dan 90%, dataset yang digunakan adalah dataset penyakit liver Taiwan. Penelitian lain menunjukkan bahwa algoritma berbasis rule dapat digunakan untuk mengklasifikasi pasien penyakit hati (Ramana, Babu, & Venkateswarlu, Liver Classification Using Modified Rotation Forest, 2012). Beberapa algoritma berbasis rule yang digunakan adalah algoritma C4.5, ZeroR dan PART menunjukkan bahwa algoritma C4.5 memiliki tingkat akurasi yang paling tinggi (Ramana, Babu, & Venkateswarlu, Liver Classification Using Modified Rotation Forest, 2012). Selain itu beberapa algoritma klasifikasi lain telah digunakan dalam mendeteksi penyakit liver yaitu Naïve Bayes, C4.5, Neural Network, K-NN dan Support Vector Machine (SVM) (Ramana, Babu, & Venkateswarlu, A Critical Study Of Selected Classification Algorithms For Liver Disease Diagnosis, 2011)

menunjukkan bahwa neural network memiliki tingkat akurasi dan presisi yang paling tinggi. Algoritma C4.5 memiliki sensitivity paling tinggi dan Naïve bayes memiliki spesificity paling tinggi.

Pada makalah ini menyajikan penerapan algoritma berbasis rule dalam mendeteksi penyakit liver. Algoritma berbasis rule yang digunakan adalah ZeroR, OneR, RIPPER dan C4.5. Keempat algoritma tersebut dievaluasi *accuracy, precision, sensitivity* dan *spesificity*. Algoritma berbasis rule dipilih karena menghasilkan rule atau aturan yang sederhana tetapi memiliki akurasi yang cukup tinggi (Holte, 1993).

Tinjauan Pustaka

a. Algoritma ZeroR

Algoritma ZeroR secara sederhana memprediksi mayoritas kelas dalam training data (Andreeva, Dimitrova, & Radeva, 2004). Meskipun algoritma ZeroR memiliki sedikit akal untuk digunakan sebagai prediktor, algoritma ZeroR bermanfaat untuk menentukan performance dasar sebagai benchmark untuk skema pembelajaran yang lain.

b. Algoritma OneR

OneR adalah algoritma klasifikasi sederhana yang membangun pohon keputusan satu level (tingkat). Algoritma OneR pertama kali dibuat oleh Robert C. Holte pada tahun 1993. Algoritma OneR membuat ranking atribut rerata eror pada data training seperti pengukuran entropi pada C4.5 (Holte, 1993). OneR memperlakukan nilai numerik atribut sebagai nilai *continous* dan menggunakan metode sederhana untuk membagi cakupan nilai ke dalam beberapa interval terpisah. OneR membuat satu aturan untuk masing-masing atribut dalam data training kemudian memilih rule dengan eror terkecil yang disebut satu aturan (*one rule*) (Buddhinath & Derry, 2005).

1. *Pada data training hitung nilai kelas C yang mempunyai nilai C untuk atribut A: simpan informasi ini dalam matrik 3 dimensi, COUNT(C,V,A).*
2. *Kelas default adalah kelas yang memiliki nilai terbanyak pada training. Akurasi kelas default adalah jumlah training pada kelas default dibagi jumlah seluruh data training.*
3. *Untuk masing-masing atribut numerik,A, buat versi nominal dari A dnegan menetukan interval nilai yang tetap. Interval ini menjadi nilai versi nominal A.*
4. *Untuk masing-masing atribut A,*
   a. *Buat sebuah hipotesis yang melibatkan atribut A dengan memilih, untuk masing-masing nilai V dari A, sebuah kelas optimal untuk V.*
   b. *Tambahkan hipotesis yang dibangun ke sebuah aturan yang disebut hipoteses. Aturan ini berisi satu hipotesis untuk masing-masing atribut.*
5. *Pilih aturan yang dari hipoteses memiliki akurasi tertinggi pada data training (jika ada beberapa pilih secara acak)*

Gambar 2 Pseudocode algoritma OneR

Algoritma OneR yang menghasilkan rule yang sederhana menghasilkan akurasi yang sedikit lebih rendah (3,1%) dibandingkan dengan akurasi algoritma klasifikasi yang terkenal, C4.5. Algoritma C4.5 menghasilkan yang lebih besar dibandingkan dengan OneR. Rule sederhana seringkali dijadikan alternatif dari sistem rule yang lebih kompleks (Holte, 1993).

c. Algoritma RIPPER

RIPPER, *Repeated Incremental Pruning to Produce Error Reduction* adalah algoritma machine leraning berbasis rule yang merupakan penyempurnaan *dari Incremental Reduce Error Prunning* (IREP) (Cohen, 1995). RIPPER dimulai dengan kasus dua kelas dimana kita menyebutnya contoh positif dan negatif, yang kemudian digeneralisasi K>2. Rule ditambahkan untuk menjelaskan contoh positif sedemikian hingga jika sebuah instance tidak dicakup oleh rule lain, yang kemudian diklasifikasi sebagai negatif (Alpaydin, 2010). Pada RIPPER kondisi ditambahkan pada rule untuk memaksimalkan pengukuran information gain yang digunakan pada algoritma Foil Quinlan (1990). Misal kita mempunyai R dan R' adalah kandidat rule setelah penambahan sebuah kondisi. Perubahan gain ditentukan oleh:

$$Gain(R', R) = s.\left(log_2 \frac{N'_+}{N'} - log_2 \frac{N_+}{N}\right)$$

Dimana N adalah jumlah instance yang dicakup oleh D dan N+ adalah jumlah true positifnya. N' dan N'+ demikian juga menentukan R'. s adalah jumlah true positif dalam R, yang masih true positif dalam R', setelah penambahan kondisi.

Kondisi ditambahkan kepada rule sampai tidak mencakup contoh negatif. Setelah rule dibangkitkan, rule dilakukan prune dengan menghapus kondisi dalam urutan terbalik, untuk menentukan rule yang nilai *rule value metric* maksimal:

$$rmv(R) = \frac{p - n}{p + n}$$

Dimana p dan n adalah jumlah true dan false positif, yang masing-masing pada aturan pruning, yang 1/3 data, 2/3 sebagai growing set.

```
Ripper(Pos,Neg,k)
   RuleSet ← LearnRuleSet(Pos,Neg)
   For k times
      RuleSet ← OptimizeRuleSet(RuleSet,Pos,Neg)
LearnRuleSet(Pos,Neg)
   RuleSet ← ∅
   DL ← DescLen(RuleSet,Pos,Neg)
   Repeat
      Rule ← LearnRule(Pos,Neg)
      Add Rule to RuleSet
      DL' ← DescLen(RuleSet,Pos,Neg)
      If DL'>DL+64
        PruneRuleSet(RuleSet,Pos,Neg)
        Return RuleSet
      If DL'<DL DL ← DL'
        Delete instances covered by Rule from Pos and Neg
   Until Pos = ∅
   Return RuleSet
PruneRuleSet(RuleSet,Pos,Neg)
   For each Rule ∈ RuleSet in reverse order
      DL ← DescLen(RuleSet,Pos,Neg)
      DL' ← DescLen(RuleSet-Rule,Pos,Neg)
      IF DL'<DL Delete Rule from RuleSet
   Return RuleSet
OptimizeRuleSet(RuleSet,Pos,Neg)
   For each Rule ∈ RuleSet
      DL0 ← DescLen(RuleSet,Pos,Neg)
      DL1 ← DescLen(RuleSet-Rule+
            ReplaceRule(RuleSet,Pos,Neg),Pos,Neg)
      DL2 ← DescLen(RuleSet-Rule+
            ReviseRule(RuleSet,Rule,Pos,Neg),Pos,Neg)
      If DL1=min(DL0,DL1,DL2)
            Delete Rule from RuleSet and
                  add ReplaceRule(RuleSet,Pos,Neg)
      Else If DL2=min(DL0,DL1,DL2)
            Delete Rule from RuleSet and
                  add ReviseRule(RuleSet,Rule,Pos,Neg)
   Return RuleSet
```

Sumber: (Alpaydin, 2010)
Gambar 3. Pseudocode algoritma RIPPER

Setelah rule ditumbuhkan dan di-prune, contoh positif dan negatif dalam training yang dicakup oleh rule dihapus dari data training. Jika masih ada sisa contoh positif, induksi rule terus berlangsung. Pada kasus noise, bisa dihentikan awal, yaitu ketika sebuah rule tidak cukup menjelaskan contoh. Untuk mengukur nilai sebuah rule digunakan *minimum description length*. Khusunya, induksi dihentikan jika deskripsi rule tidak lebih pendek dari instance yang dijelaskanya. *Description length* dari sebuah rule adalah jumlah *description length* seluruh *rule* pada *rule base* ditambah *description instance* yang tidak dicakup oleh *rule base*. Ripper menghentikan penambahan rule ketika description length rule base lebih dari pada 64 bit lebih besar dari pada description length terbaik sejauh ini. Ketika rule base dipelajari, kita melewatkan melalui rule dalam urutan terbalik untuk melihat jika mereka dapat dihapus tanpa meningkatkan description length.

Rule pada rule base juga dioptimisasi setelah dipelajari. Ripper mempertimbangkan dua alternetif pada rule: satu disebut penggantian rule, mulai dari rule yang kosong, ditumbuhkan dan di-prune. Kedua disebut revisi rule, mulai dengan rule itu sendiri, ditumbuhkan dan kemudian di-prune. Dua metode ini dibandingkan dengan rule original, dan tiga terpendek ditambahkan pada rule base. Optimisasi rule base dapat dilakukan sebanyak *k* kali, biasanya dua kali (Alpaydin, 2010).

d. Algoritma C4.5

Algoritma C4.5 disebut juga decision tree atau pohon keputusan karena algoritma ini menghasilkan pohon keputusan. Algoritma C4.5 merupakan algoritma berbasis rule yang diperoleh secara tidak langsung, karena rule diperoleh dari pohon keputusan yang dihasilkan oleh algoritma C4.5. algoritma C4.5 mempunyai dua tipe node, node internal dan node daun. Node internal berhubungan dengan tes untuk sampel pada atribut individu atau kelompok dan node daun menugaskan kelas label pada sampel berdasarkan distribusi kelas rekamanya. Algoritma C4.5 mengklasifikasi sampel dengan cara top-down, mulai dari node akar dan menjaga pergerakan sesuai dengan hasil tes pada node internal, sampai node daun dicapai dan label kelas ditugaskan (Yu, Chen, Koronios, Zu, & Guo, 2007).

Konstruksi pohon keputusan pada C4.5 berdasarkan pemisahan node internal secara rekursif. Pemilihan atribut yang dipisah pada node internal sangat penting selama proses konstruksi dan menentukan jangkauan luas struktur akhir pohon keputusan. Banyak usaha telah dilakukan pada aspek ini dan serangkaian kriteria pemisahan, seperti gini index, information gain dan test chi square. Teori entropi diadopsi untuk memilih pemisahan

atribut yang tepat oleh algoritma C4.5. Misal N adalah ukuran dataset D dan Nj adalah jumlah sampel pada kelas j. Asumsi ada K label kelas, teori entropi menyatakan bahwa rata-rata jumlah informasi yang dibutuhkan untuk mengklasifikasi sebuah sampel adalah sebagai berikut:

$$Info(D) = -\sum_{j=1}^{k} \frac{N_j}{N} log_2\left(\frac{N_j}{N}\right)$$

Ketika dataset D dipisah ke dalam beberapa subset D1,D2,D3,….,Dn sesuai hasil dari atribut X, information gain didefinisikan sebagai:

$$Gain(X, D) = Info(D) - \sum_{i=1}^{n} \frac{N^i}{N} Info(D_i)$$

Dimana N^i adalah jumlah sampel pada subset D_i. C4.5 mengaplikasikan Gain_ratio, dari pada Gain, sebagai kriteria:

$$Gain\_ratio(X, D) = \frac{Gain(X, D)}{\left(-\sum_{i=1}^{n} \frac{N^i}{N} log_2\left(\frac{N^i}{N}\right)\right)}$$

C4.5 secara greedy mempartisi node sampai nilai trivial Gain_ratio dicapai. Sebuah prosedur prune kemudian dijalankan untuk menghindari pohon yang kompleks yang overfit data (Yu, Chen, Koronios, Zu, & Guo, 2007).

```
GenerateTree(X)
    If NodeEntropy(X)< θ1 /* equation 9.3 */
      Create leaf labelled by majority class in X
      Return
    i ← SplitAttribute(X)
    For each branch of xi
      Find Xi falling in branch
      GenerateTree(Xi)

SplitAttribute(X)
    MinEnt← MAX
    For all attributes i = 1,...,d
      If xi is discrete with n values
        Split X into X1,...,Xn by xi
        e ← SplitEntropy(X1,...,Xn) /* equation 9.8 */
        If e<MinEnt MinEnt ← e; bestf ← i
      Else /* xi is numeric */
        For all possible splits
            Split X into X1, X2 on xi
            e←SplitEntropy(X1,X2)
            If e<MinEnt MinEnt ← e; bestf ← i
    Return bestf
```

Sumber: (Alpaydin, 2010)

Gambar 4. Pseudocode algoritma C4.5

Metode Penelitian

Penelitian ini adalah penelitian eksperimen, tool yang digunakan adalah *Waikato Environment Knowledge Analysis* (WEKA). Dataset yang digunakan adalah data *Indian Liver Patient Dataset* (ILPD) yang diambil dari UCI *Machine Learning Repository*. Dataset terdiri dari 584 instance dan 11 atribut dan 1 label kelas. Label kelas bertipe binominal yang terdiri dari dua nilai, yaitu penderita liver dan bukan penderita liver.

Tabel 1. Atribut pada indian liver patient dataset

Atribut	Tipe
Usia	Integer
Jenis kelamin	Binominal
Total Bilirubin (TB)	Integer
Direct Bilirubin (DB)	Integer
Alkaline Phosphotase (Alkphos)	Integer
Sgpt Alamine (Sgpt)	Integer
Sgot Aspartate (Sgot)	Integer
Total Protiens (TP)	Integer
Albumin (ALB)	Integer
Ratio Albumin and Globulin (Rasio A/G)	Integer
Penderita Liver	Binominal

Sumber: (Ramana, Prabu, & Venkateswarlu, ILPD (Indian Liver Patient Dataset), 2012)

Atribut pada dataset dilakukan perankingan menggunakan *ChiSquareAttributeEval* yang tersedia pada WEKA. Selanjutnya algoritma berbasis rule diterapakan pada data set untuk menggali informasi pada dataset. Algoritma yang digunakan algoritma ZeroR, OneR, RIPPER dan C4.5. Hasil penerapan algoritma terbut dibuat dalam confusion matrix (Witten, Frank, & Hall, 2011).

Tabel 2 Bentuk confusion matrix (Witten, Frank, & Hall, 2011)

		Predicted class	
		Yes	No
Actual class	Yes	True Positive	False negative
	No	False Positive	True Negative

Algoritma tersebut dievaluasi berdasarkan accuracy, sensitivity, precision dan spesificity, yang dapat diukur sebagai berikut (Witten, Frank, & Hall, 2011):

Accuracy= $\frac{number\ true\ positive + number\ true\ negative}{number\ true\ positive + false\ positive + false\ negative + true\ negative}$

Sensitivity= $\frac{number\ true\ positive}{number\ true\ positive + false\ negative}$

Precision= $\frac{number\ true\ positive}{number\ true\ positive + false\ positive}$

Specificity= $\frac{number\ true\ negative}{number\ true\ negative + false\ positive}$

Pembahasan

Atribut dataset dirnaking berdasarkan prioritas menggunakan algoritma ranking yang tersedia di WEKA. Atribut pada dataset dilakukan perankingan menggunakan metode *Chi Square*. Urutan atribut hasil perankingan adalah Sgpt, TB, DB, Alkphos, Sgot, Rasio A/G, Usia, Albumin, Jenis Kelamin, TP dan Pendita Liver. Hasil perankingan ditunjukkan pada tabel 3.

Tabel 3. Hasil perankingan atribut menggunakan algortima perankingan

Ranking	Atribut
1	Sgpt
2	TB
3	DB
4	Alkphos
5	Sgot
6	Rasio A/G
7	Usia
8	Albumin
9	Jenis Kelamin
10	TP
11	Penderita Liver

Sumber: Dokumentasi Peneliti

Performance algoritma berbasis rule pada kombinasi atribut yang berbeda ditunjukkan pada tabel 4 – 7. Berdasarkan hasil eksperimen terlihat bahwa algoritma ZeroR memiliki nilai accuracy, sensitivity, precision dan spesificy yang konstan mulai dari 8 atribut pertama sampai 11 atribut pertama. Artinya penambahan atribut tidak berpengaruh terhadap accuracy, sensitivity, precision dan spesificty algoritma ZeroR. Algoritma ZeroR membuat rule berdasarkan jumlah instance terbanyak. Algoritma ZeroR memberikan nilai accuracy dan precision yang paling besar diantara algoritma berbasis rule lainya. Algoritma OneR memiliki tingkat accuracy, sensitivity, precision dan spesificity yang konstan mulai pada dataset dengan jumlah atribut 8 sampai dataset dengan jumlah atribut 11. Hal ini menunjukkan bahwa penambahan atribut tidak mempengaruhi accuracy, sensitivity, precision dan spesificity algoritma OneR. Penambahan atribut juga tidak mempengaruhi jumlah true positive dan true negative. Algoritma RIPPER memiliki nilai sensitivity yang tersebesar diantara algoritma lain. Hal ini menunjukkan algoritma RIPPER mempunyai true positive yang paling besar. Nilai sensitivity meningkat pada dataset dengan atribut 8 ke dataset dengan atribut 9. Hal ini menunjukkan bahwa penambahan atribut pada dataset atribut 9 meningkatkan jumlah true positive. Nilai sensitivity tertinggi pada jumlah atribut 11, penambahan jumlah atribut mempengaruhi jumlah true positive. Nilai spesificity mengalami penurunan pada dataset atribut 9 ke dataset atribut 10. Penurunan nilai spesificity ini menunjukkan jumlah true negatif menjadi menurun. Algoritma C4.5 mengalami peningkatan nilai sensitivity pada dataset atribut 9 dan 10, mengalami penurunan sensitivity pada atribut 11. Hal ini menunjukkan pada dataset atribut 9 dan 10 mengalami peningkatan true positive, mengalami penurunan true negative pada dataset atribut 11. Algoritma C4.5 mengalami peningkatan nilai spesificity pada dataset atribut 9 dan 10, mengalami penurunan spesificity pada dataset atribut 11. Hal ini menunjukkan peningkatan true negative pada dataset atribut 9 dan 10, mengalami penurunan true negative pada dataset 11. Nilai akurasi terbesar dicapai oleh algoritma ZeroR sebesar 71.36%. hal ini dukarenakan algoritma zeroR mengklasifikasi berdasarkan jumlah instance terbanyak yaitu kelas Penderita Liver sebanyak 416 instance.

Tabel 4. Performance algoritma rule pada 8 atribut pertama dataset

	accurracy	sensitivity	precision	spesificity
zeroR	71.36	71.36	100.00	0.00
oneR	65.87	72.46	72.46	34.00
RIPPER	69.13	74.08	87.26	43.01
C4.5	68.27	71.59	92.07	31.25

Sumber: Dokumentasi Peneliti

Tabel 5. Performance algoritma rule pada 9 atribut pertama pada dataset

	accurracy	sensitivity	precision	spesificity
zeroR	71.36	71.36	100.00	0.00
oneR	65.87	72.46	72.46	34.00
RIPPER	68.44	75.22	83.17	43.09
C4.5	66.72	71.85	87.74	32.00

Sumber: Dokumentasi Peneliti

Tabel 6. Performance algoritma rule pada 10 atribut pertama pada dataset

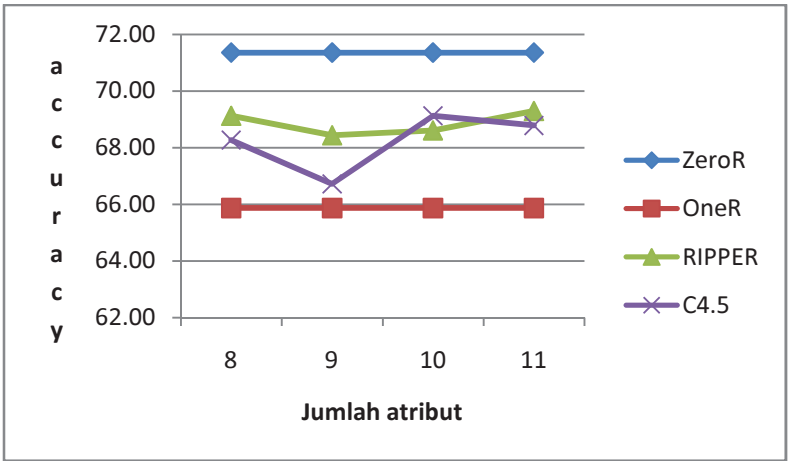
	accurracy	sensitivity	precision	spesificity
zeroR	71.36	71.36	100.00	0.00
oneR	65.87	72.46	72.46	34.00
RIPPER	68.61	74.63	84.86	42.73
C4.5	69.13	75.88	83.17	44.88

Sumber: Dokumentasi Peneliti

Tabel 7. Performance algoritma rule pada 11 atribut permata pada dataset

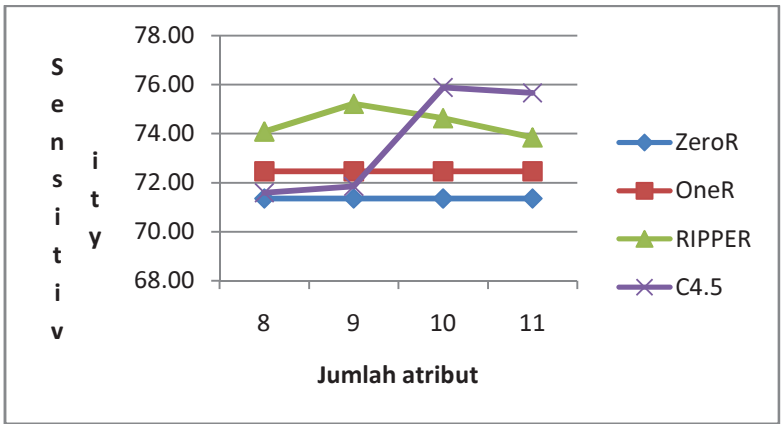
	accurracy	sensitivity	precision	spesificity
zeroR	71.36	71.36	100.00	0.00
oneR	65.87	72.46	72.46	34.00
RIPPER	69.30	73.84	88.22	43.02
C4.5	68.78	75.66	82.93	44.09

Sumber: Dokumentasi Peneliti



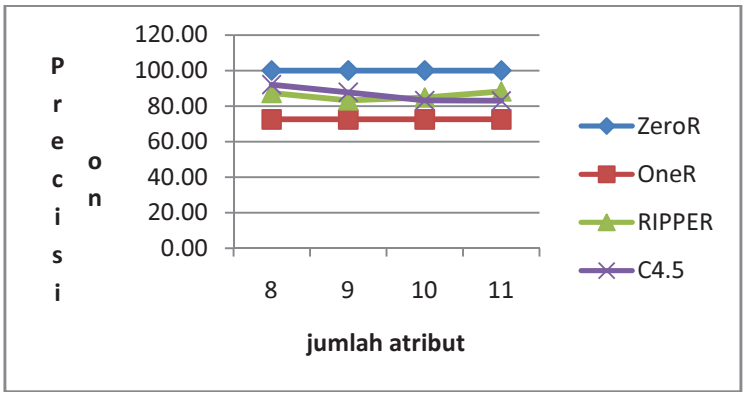
Sumber: Dokumentasi Peneliti

Gambar 5 Akurasi algoritma rule pada jumlah atribut yang berbeda



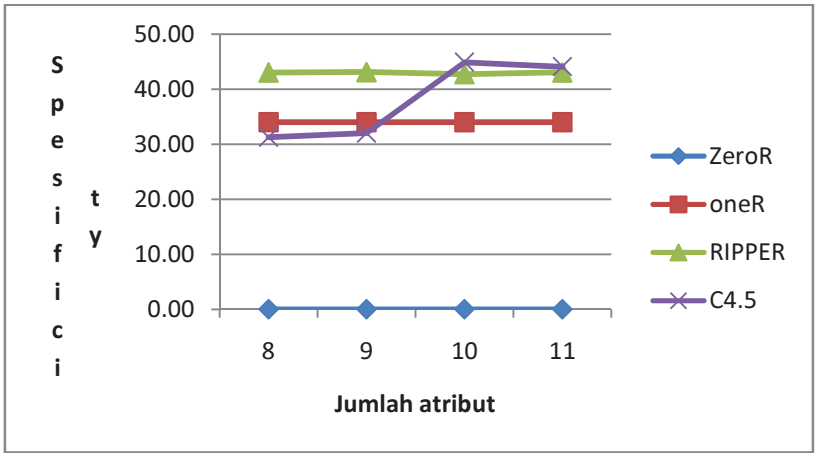
Sumber: Dokumentasi Peneliti

Gambar 6 Sensitivity algoritma rule pada jumlah atribut yang berbeda



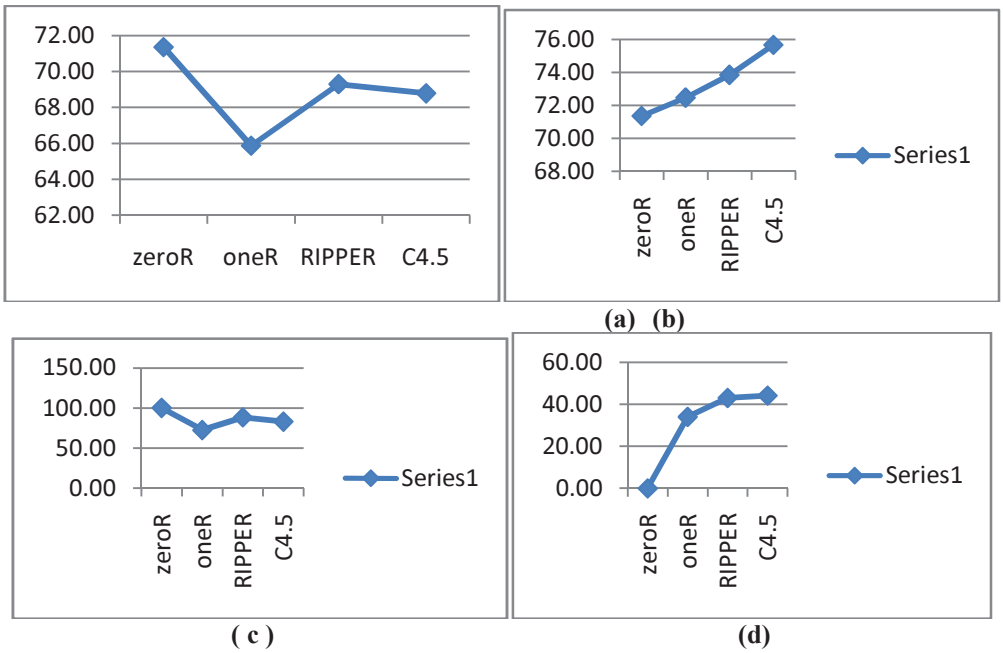
Sumber: Dokumentasi Peneliti

Gambar 7. Precision algoritma rule pada jumlah atribut yang berbeda



Sumber: Dokumentasi Peneliti

Gambar 8 Spesificity algoritma rule pada jumlah atribut yang berbeda



Sumber: Dokumentasi Peneliti

Gambar 9 Performance algoritma rule (a) accuracy (b) sensitivity (c) precision (d) spesificity

Tabel 8 Performance algoritma rule dengan seluruh atribut pada dataset ILPD

	accurracy	sensitivity	precision	spesificity
zeroR	71.36	71.36	100.00	0.00
oneR	65.87	72.46	72.46	34.00
RIPPER	69.30	73.84	88.22	43.02
C4.5	68.78	75.66	82.93	44.09

Sumber: Dokumentasi Peneliti

Kesimpulan

Algoritma berbasis rule dapat digunakan dan diimplementasikan untuk membuat sistem otomatis dalam mendeteksi penyakit liver. Algoritma ZeroR mempunyai nilai accuracy dan precision yang tertinggi. Algoritma C4.5 memiliki nilai sensitivity paling besar diantara algoritma lainya, yang artinya algoritma C4.5 mempunyai jumlah true positive paling besar. Algoritma C4.5 memiliki spesificity yang paling besar, yang artinya mempunyai jumlah true negative yang paling besar. Penelitian selanjutnya bisa menggunakan yang lain dalam mendeteksi penyakit liver, misal algoritma berbasis pohon keputusan (*tree classifier*) dibandingkan dengan hasil penelitian ini.

Daftar Pustaka

Alpaydin, E. 2010. *Introduction To Machine Learning* (Second Edition ed.). Massachusetts : Massachusetts Institutes of Technology.

Andreeva, P., Dimitrova, M., & Radeva, P. 2004. Data Mining Learning Models And Algorithms For Medical Application. *18 Conference Systems for Automation of Engineering and Research* . Varna, Bulgaria.

Buddhinath, G., & Derry, D. 2005, Januari. Retrieved from Gaya Buddhinath: http://www.buddhinath.net/OtherLinks/Documents/Improved%20OneR%20Algorithm.pdf

Cohen, W. W. 1995. Fast Effective Rule Induction. *Machine Learning: Proceeding of The Twelfth International Conference*, (pp. 115-123).

Hepatitis B Fondation. (n.d.). *Your Liver and How It Works - Hepatitis B Foundation.* Retrieved Maret 1, 2013, from Hepatitis B Fondation: http://www.hepb.org/pdf/the_liver.pdf

Holte, R. C. 1993. Very Simple Classification Rules Perform Well On Most Commonly Used Datasets . *Machine Learning 11*, (pp. 63-91). Ottawa.

Kononenko, I. 2001. Machine Learning For Medical Diagnosis: History, State Of The Art and Perspective. *Artificial Inteligence In Medicine , 23*, 89-109.

Lin, R. H. 2009. An Inteligence Model For Liver Disease Diagnosis. *Artificial Inteligence In Medicine , 47* (1), 53-62.

Lumongga, F. (2009, Juli). *Struktur Liver*. Retrieved from USU Institutional Repository: http://repository.usu.ac.id/handle/123456789/2052

Praetorius, G. 2006. Retrieved from Linkoping University Department Of Computer and Information Science: http://www.ida.liu.se/~729G50/studentpapper-06/gesa-praetorius.pdf

Ramana, B. V., Babu, S. P., & Venkateswarlu, N. B. 2011. A Critical Study Of Selected Classification Algorithms For Liver Disease Diagnosis. *International Journal Of Database Management Systems , 3* (2), 101-114.

Ramana, B. V., Babu, S. P., & Venkateswarlu, N. B. 2012. Liver Classification Using Modified Rotation Forest. *International Journal Of Engineering Research And Development , 1* (6), 17-24.

Ramana, B. V., Prabu, S. P., & Venkateswarlu, N. B. 2012, Mei 21. *ILPD (Indian Liver Patient Dataset)*. Retrieved Juni 17, 2013, from Machine Learning Repository: https://archive.ics.uci.edu/ml/datasets/ILPD+%28Indian+Liver+Patient+Dataset%29

Witten, I. H., Frank, E., & Hall, M. A. 2011. *Data Mining Practical Machine Learning Tools and Technique Third Edition*. New York: Morgan Kaufmann.

Yu, L., Chen, G., Koronios, A., Zu, S., & Guo, X. (2007). Application and Comparison of Classification Techniques in Controlling Credit Risk. *Recent Advances in Data Mining of Enterprise Data: Algorithms and Applications , 6*, pp. 111-146.