ANALISIS PEMANFAATAN *SMALL DISJUNCT* PADA *DECISION TREE* DENGAN ALGORITMA GENETIKA

Andhik Budi Cahyono
[1]*Jurusan Teknik Informatika, Fakultas Teknologi Industri,Universitas Islam Indonesia*
Jl. Kaliurang Km. 14 Yogyakarta 55501
E-mail: andhikbudi@gmail.com

ABSTRAKS

Rule classifier merupakan salah satu metode yang digunakan dalam data mining dan bisa diperoleh dari pohon keputusan. *Rule* yang diperoleh dari pohon keputusan bisa dikategorikan berdasarkan jumlah data benar yang di cakup yaitu *small disjunct* dan *large disjunct*. *Large disjunct* adalah rule yang mencakup data benar dalam jumlah yang relatif besar. Sedangkan *Small disjunct* adalah *rule* yang mencakup data benar dalam jumlah yang sedikit, sehingga sering menyebabkan kesalahan klasifikasi pada data testing. Walaupun sebuah *small disjunct* mencakup data yang relatif kecil, sekumpulan *small disjunct* bisa mencakup data dalam jumlah yang besar. Bagaimanapun diperlukan pendekatan yang tepat untuk menangani *small disjunct* ini.

Dalam tugas akhir ini akan dibangun algoritma genetika untuk mengatasi masalah *small disjunct* pada *decision tree* tersebut. Sedangkan algoritma *decision tree* yang digunakan adalah J48 yang merupakan varian dari C45 yang dikembangkan dalam bahasa pemrograman java. Akan dilakukan sedikit modifikasi pada J48 sehingga bisa mengidentifikasi *rule-rule* ke dalam *small disjunct* atau *large disjunct*. Model akhir yang ingin dibangun adalah gabungan *large disjunct* dan *rule* klasifikasi yang dihasilkan oleh algoritma genetika.
Pada proses analisa akan digunakan enam data yang bertipe numerik untuk mengukur performansi akurasi dari solusi yang dibangun dan akan dibandingkan dengan classifier yang lain. Selain itu pada proses analisa juga akan ditunjukkan seberapa besar error klasifikasi yang disebabkan oleh small disjunct*.*

Kata Kunci: rule *classifier,* decision tree*, J48,* small disjunct*, algoritma genetika*

1. PENDAHULUAN

1.1 1.1 Latar belakang masalah

Klasifikasi merupakan salah satu tugas/task dalam data mining yang digunakan untuk meramalkan sebuah nilai dari sekumpulan data. Klasifikasi bekerja dengan cara menemukan model atau fungsi yang menjelaskan atau membedakan konsep atau kelas data, dengan tujuan memperkirakan kelas dari suatu objek yang labelnya tidak diketahui. Penggunaan task klasifikasi sebagai salah satu metode peramalan masih merupakan isu yang terus berkembang seiring dengan kebutuhan akan data mining. Salah satu metode klasifikasi yang sering digunakan adalah *decision tree.*

Klasifikasi dengan *decision tree* menghasilkan sebuah model yang bisa direpresentasikan dalam aturan IF <kondisi>THEN <prediksi (kelas)>. *Small disjunct* adalah aturan (*rule*) yang hanya mencakup sedikit training data. Aturan seperti ini seringkali diabaikan yang bisa menyebabkan berkurang nya akurasi. Ini bisa menjadi masalah yang cukup serius karena walaupun setiap *small disjunct* mencakup data yang sedikit, tapi sekumpulan *small disjunct* bisa saja mencakupi data yang jumlahnya besar.

Keumuman dalam membuat aturan (*rule*) biasanya diperoleh dari large disjuct (aturan yang mencakup data dalam jumlah banyak). Karena aturan yang diperoleh dari *large disjunct* cenderung memberikan tingkat kepercayaan (*confident*) yang lebih tinggi daripada aturan yang diperoleh dari *small disjunct*. Namun, sekali lagi bukan berarti aturan dari *small disjunct* tidak perlu diperhatikan. Karena aturan yang diperoleh dari *small disjunct* bisa saja mempengaruhi/mencakup data yang lebih besar. Karena itu diperlukan sebuah cara untuk menemukan aturan *small disjunct* yang akurat dalam upaya untuk memperoleh akurasi klasifikasi yang bagus.

Decision tree merupakan salah satu metode klasifikasi yang menghasilkan model dan sering merepresentasikan model tersebut dalam bentuk sekumpulan *rule*. Oleh karena itu pada *rule* yang di induksi dari *decision tree* sering dijumpai *small disjunct*. Dalam tugas akhir ini akan dianalisa seberapa jauh *small disjunct* dapat dimanfaatkan untuk membuat aturan (*rule*) baru yang dapat meningkatkan akurasi atau sebaliknya, dapat mengurangi akurasi dari klasifikasi *decision tree*. Untuk men-*generate rule* baru dari *small disjunct* pada *decision tree* akan digunakan algoritma genetika.

Penggunaan algoritma genetika untuk mengolah small disjunct didasarkan karena algoritma genetika cenderung lebih baik, *robust*, dan fleksibel menangani interaksi atribut daripada kebanyakan algortima pencarian *rule* yang sederhana lainnya. Hal itu bisa dijelaskan sebagai berikut. Pertama, algoritma genetika bekerja dengan sekumpulan kandidat solusi (populasi). Kedua, kandidat solusi dievaluasi semuanya dengan menggunakan fungsi fitness. Karakteristik ini berbeda dengan kebanyakan algoritma pencarian

rule lainnya yang bekerja hanya dengan satu kandidat solusi pada satu waktu, dan dievalusi hanya berdasar informasi lokal. Kemampuan algoritma genetika untuk menangani interaksi atribut inilah yang menjadikannya berpotensi sebagai solusi dari masalah small disjunct karena interaksi atribut merupakan salah satu penyebab adanya small disjunct (Carvalho).

1.2 1.2 Tujuan

Tujuan yang ingin dicapai dalam pembuatan tugas akhir ini adalah

1. Memanfaatkan *small disjunct* pada induksi aturan yang dihasilkan dari algoritma C45 dengan menggunakan Algoritma Genetika.
2. Melihat seberapa jauh pengaruh *small disjunct* pada *decision tree*, apakah dapat memperbaiki keakurasian klasifikasi atau justru sebaliknya, mengurangi keakurasian klasifikasi.
3. Menganalisa performa dari gabungan *decision tree* dan algoritma genetika berdasarkan parameter keakurasian proses klasifikasi.

2. DECISION TREE

Decision tree merupakan salah satu metode klasifikasi yang menggunakan representasi struktur pohon (tree) dimana setiap node merepresentasikan atribut, cabang nya merepresentasikan nilai dari atribut, dan daun merepresentasikan kelas. Node yang paling atas dari decision tree disebut sebagai root (Han, 2001).

Decision tree merupakan metode klasifikasi yang paling populer digunakan. Selain karena pembangunannya relatif cepat, hasil dari model yang dibangun mudah untuk dipahami.

Pada *decision tree* terdapat 3 jenis node , yaitu :

a. Root Node
   Merupakan node paling atas, pada node ini tidak ada input dan bisa tidak mempunyai output atau mempunyai output lebih dari satu.
b. Internal Node
   Merupakan node percabangan, pada node ini hanya terdapat satu input dan mempunyai output minimal dua.
c. Leaf node atau terminal node
   Merupakan node akhir, pada node ini hanya terdapat satu input dan tidak mempunyai output (Han, 2001).

3. SMALL DISJUNCT

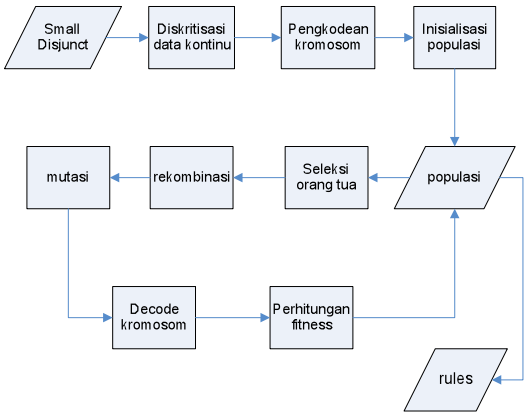
Small disjunct adalah aturan yang mencakup data training yang jumlahnya sedikit. Dengan kata lain aturan hanya diperoleh dari data yang jumlahnya sedikit sehingga seringkali menyebabkan error dalam proses klasifikasi (Weiss, 2003). *Small disjunct* ini juga merupakan problem yang dilematis. Dalam beberapa kasus (Weiss, 2000) tentang *small disjunct* menunjukkan bahwa ketika *small disjunct* dihilangkan atau tidak dihilangkan bisa menyebakan pengaruh yang negatif dalam model klasifikasi yang dibuat. Ketika *small disjunct* ini dihilangkan masalah yang timbul adalah berkurangnya akurasi *rule* yang dihasilkan oleh *large disjunct*. Namun, ketika tidak dihilangkan sering menyebabkan misclasification pada *rule* yang diperoleh dari *small disjunct* itu sendiri.

Berbagai cara digunakan untuk mengatasi problem *small disjunct* ini. Diantaranya adalah memperbaiki algoritma pruning pada decesion tree yaitu dengan tidak melakukan pruning pada semua *small disjunct*. Hanya beberapa *small disjunct* yang sering menyebabkan misclasification yang di-prunning. Cara yang lain adalah melakukan perbaikan pada pemilihan data sample (data training), sehingga dari sejak awal sistem sudah berusaha meminimalkan terbentuknya *small disjunct* (Weiss, 2000). Berbeda dengan dua cara yang sebelumnya cara yang ketiga menggunakan pendekatan lain. Yaitu bagaimana memanfaatkan *small disjunct* dari *decision tree* yang terbentuk dari proses pembelajaran (training) sehingga bisa menghasilkan clasifier yang tetap akurat atau bahkan lebih baik.

4. ALGORTIMA GENETIKA UNTUK SMALL DISJUNCT

Seperti yang sudah dijelaskan sebelumnya, algoritma genetika (**AG**) akan digunakan untuk mencari *rule* baru dari sekumpulan data training yang tergolong (tercakup) dalam *small disjunct*. *Rule* baru hasil AG akan dievaluasi dengan menggunakan data *small disjunct* dan setelah ditemukan sekumpulan *rule* terbaik, kemudian akan ditambahkan pada large disjunct yang sudah dihasilkan oleh decision tree (C45). Jadi model akhir dari classifier adalah large disjunct ditambah *rule* AG.

Berikut adalah gambar alur penginduksian *rule* dengan menggunakan Algoritma Genetika.



Gambar 1. Alur penginduksian *rule* dengan AG

Keterangan :

a. System mengambil data small disjunct dari database oracle.

b. Untuk setiap atribut data kontinu akan dibagi menjadi dua nilai diskrit berdasar *cut point* yang dicari menggunakan entropy.
c. Data yang sudah terdiskritisasi akan dikodekan ke dalam kromosom. Representasi kromosom untuk tugas akhir ini menggunakan representasi biner.
d. Inisialisasi individu dilakukan dengan cara memilihi secara acak kromosom hasil pengkodean data masukan. Jumlah individu dalam populasi merupakan variabel yang bisa berubah tergantung *input*an dari pengguna system.
e. Dari populasi yang ada akan dipilih dua individu terbaik sebagai calon orang tua. Pemilihan orang tua didasarkan pada perhitungan fitness masing-masing individu yang ada dalam populasi dan berdasar metode pemilihan orang tua tertentu. Pada tugas akhir ini seleksi orang tua dilakukan dengan menggunakan roulette wheel.
f. Kedua individu yang terpilih akan melakukan pindah silang pada gen-gen tertentu sehingga gen-gen tersebut akan bertukar antar individu orang tua. Individu hasil rekombinasi inilah yang kemudian disebut anak.
g. Individu anak hasil pindah silang kemungkinan akan termutasi pada gen tertentu. Kemungkinan mutasi ini tergantung pada besar kecilnya probabilitas mutasi yang ditetapkan. Semakin besar probabilitas mutasi semakin besar kemungkinan suatu individu termutasi.
h. Setelah pindah silang dan mutasi, individu anak akan dievaluasi lagi dengan nilai fitness yang sudah ditentukan sebelumnya. Karena pada kasus ini satu individu sama dengan satu buah *rule*, maka individu harus didekodekan dulu dalam bentuk *rule* kemudian baru dihitung nilai fitnesnya.
i. Jika individu anak memiliki nilai fitness yang lebih baik dari populasi sebelumnya maka individu anak akan mengganti dua individu terjelek dari populasi sebelumnya.
j. Proses nomor 5 sampai nomor 9 akan berulang terus sampai kondisi berhenti dicapai. Kondisi berhenti ini bisa berupa jumlah iterasi (genersi) atau fitness tertinggi sama untuk beberapa generasi yang berurutan.

4.1 Representasi kromosom

Kromosom merepresentasikan individu yang akan tumbuh dan berkembang dalam ekosistem *AG*. Setiap kromosom terdiri dari gen. Gen pada representasi kromosom mewakili informasi atribut pada *data set*. Untuk atribut yang bersifat kontinu akan dilakukan diskritisasi dulu, yaitu membagi data menjadi dua atau lebih bagian berdasar *cut point* (nilai tengah). Proses diskritisasi dilakukan dengan cara menghitung nilai information gain (berdasar *entropy*) pada setiap hasil dari partisi. *Cut point* yang terbaik adalah *cut point* yang memberikan nilai information gain yang terbesar.

Setelah itu atribut akan dikodekan dalam representasi biner. Panjang kromosom sama dengan banyaknya atribut dalam *data set*. Misal jumlah atribut n buah maka panjang kromosom adalah n gen. Dengan demikian setiap gen merupakan hasil encode dari atribut dalam *data set*. Pada permasalahan ini setiap gen dibagi ke dalam dua field yaitu : *Flag*(F_i), dan *value*(V_i). Dimana atribut kelas prediksi tidak akan dimasukan ke dalam kromosom.

$Gene_1$		…	…	$Gene_n$	
F_1	V_1	…	…	F_n	V_n

Keterangan:

- *Flag Field*(F_i) : variable nilai biner di dalam range, variable ini mengindikasikan apakah atribut tersebut termasuk ke dalam *rule.* Jika termasuk dalam aturan maka akan diisi oleh bit biner 1, sedangkan 0 jika sebaliknya.
- *Value*(V_i): menggambarkan nilai cut point (titik diskritisasi data). Bernilai 1 jika lebih besar daripada (>) cut point dan bernilai 0 jika lebih kecil dan sama dengan (<=) *cut point*.

4.2 Fungsi Fitness

Fungsi *fitness* mengkombinasikan dua indikator secara umum yang digunakan untuk mengklasifikasikan *dataset* (Saggar, 2004). Indikator tersebut adalah *confidence factor* dan *completeness*. Berikut adalah table akurasi untuk nilai fitness yang merepresentasikan bobot setiap individu sebagai kandidat *kromosom* yang digunakan untuk operator genetika. *Confidence* adalah rasio antara jumlah data yang meliputi semua item pada *anticendent* dan *consequent* dengan jumlah data yang meliputi semua item pada *anticendent*. Sedangkan *completeness* adalah rasio antara jumlah data yang meliputi semua item pada *anticendent* dan *consequent* dengan jumlah data yang meliputi semua item pada *consequent.*

		Actual Class		
		Positive	Negative	
Predict Class	Positive	(TP)	(FP)	TP + FP
	Negative	(FN)	(TN)	FN + TN
		TP + FN	FP + TN	

Keterangan:

- TP = Aturan klasifikasi yang memprediksi kelas kasus dengan benar, dimana aturan memprediksi suatu kasus memiliki kelas prediksi dan memang benar kasus tersebut memiliki kelas yang diprediksi oleh aturan.
- FP = Aturan klasifikasi yang memprediksi kelas kasus dengan salah, dimana aturan memprediksi suatu kasus memiliki kelas prediksi, tetapi kasus tersebut tidak memiliki kelas yang diprediksi oleh aturan.
- FN = Aturan klasifikasi yang memprediksi kelas kasus dengan salah, dimana aturan memprediksi suatu kasus tidak memiliki kelas prediksi,tetapi kenyataannya kasus tersebut memiliki kelas yang diprediksi oleh aturan.
- TN = Aturan klasifikasi yang memprediksi kelas kasus dengan benar, dimana aturan memprediksi suatu kasus tidak

 ISSN: 1907-5022

memiliki kelas prediksi,dan memang benar kelas kasus tersebut sesuai dengan prediksi.

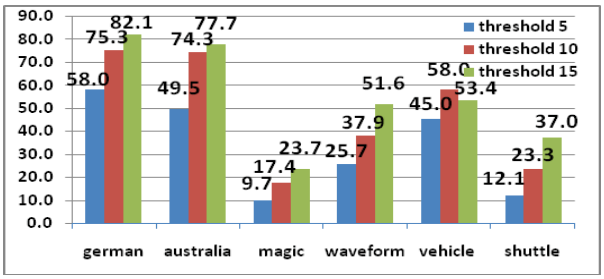
Confidence Factor = TP/(TP + FP)
Completeness = TP/(TP +FN)
*Fungsi Fitness = Confidence Factor * Completeness*

5. PENGUJIAN DAN ANALISA

Ada enam *data set* yang diambil dari repository data UCI untuk menguji sistem yang dibuat. Keenam dataset tersebut adalah data magic, bupaliver, australia credit card, german, shuttle, dan vehicle. Keenam data tersebut merupakan data yang semua atributnya bertipe numeric kecuali atribut prediksinya yang bertipe nominal (diskrit). Berikut keterangan lebih detail mengenai data yang digunakan untuk menguji system.

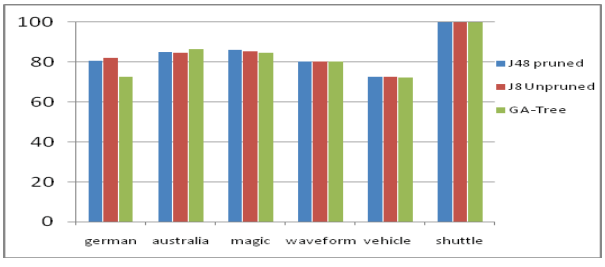
Sedangkan untuk pengujiannya adalah sebagai berikut:

1. Pengujian pengaruh *small disjunct* terhadap total kesalahan dari model klasifikasi yang terbentuk. Pengujian dilakukan dengan mengubah-ubah jumlah threshold *small disjunct* sehingga bisa dibandingkan pengaruh *small disjunct* jika mencakup data yang berbeda-beda pula.
2. Pengujian akurasi *rule* yang dihasilkan oleh algoritma genetika dan *decision tree*. Pengujian dilakukan pada data training yang berbeda dan data testing yang berbeda pula. Selain itu variabel lain yang juga akan diubah-ubah dalam pengujian ini adalah threshold *small disjunct*.



Gambar 2. Grafik kontribusi (*contribution*) kesalahan small disjunct pada total kesalahan klasifikasi

Grafik di atas menunjukkan bahwa kontribusi kesalahan terbesar ada pada data german, kemudian data australia, dan data vehicle. Dari percobaan yang dilakukan diperoleh data bahwa kontribusi small disjunct dipengaruhi juga oleh coverage small disjunct pada saat training.



Gambar 3. Grafik perbandingan akurasi J48 dengan GA-Tree pada *threshold* 5

Dari grafik di atas dapat dilihat bahwa akuarasi J48 masih lebih baik daripada GA-tree. Namun, secara keseluruhan akurasi dari GA-tree masih cukup baik dengan penurunan akurasi hanya berkisar antara satu persen (±1%) sampai dengan tiga persen (± 3%) kecuali untuk data german. Penurunan terbesar ada pada data german, yaitu bisa mencapai kurang lebih sembilan persen (9%). Coverage *small disjunct* pada data german cukup besar. Demikian juga dengan contribution-nya (kontribusi pada kesalahan total klasifikasi), sehingga pengaruh akuarsi dari *rule* AG yang dihasilkan sangat besar terhadap akurasi *rule* classifier secara keseluruhan.

6. KESIMPULAN

1. Pengaruh *small disjunct* pada total kesalahan klasifikasi cukup signifikan. Hal ini disebabkan karena masing-masing *rule small disjunct* mencakup data dalam jumlah yang sedikit, walaupun kolektivitas data yang dicakup oleh seluruh *small disjunct* cukup besar. Hal ini terlihat jelas pada data german, australia, dan shuttle. Sedangkan pengaruh *small disjunct* pada total klasifikasi benar (akurasi) sebaliknya, relatif lebih kecil jika di banding dengan *large disjunct*.
2. Coverege small disjunct ditentukan oleh besarnya threshold yang ditetapkan. Semakin besar threshold, cenderung semakin besar pula coverege-nya. Dari hasil percobaan menunjukkan bahwa coverege sangat berpengaruh pada kontribusi kesalahan klasifikasi. Dengan demikian secara tidak langung threshold small disjunct juga berpengaruh terhadap kontribusi kesalahan klasifikasi.
3. Pemanfaatan *small disjunct* dengan Algoritma Genetika untuk membentuk *rule* baru ternyata relatif tidak meningkatkan akurasi, tetapi mampu mengurangi ukuran *rule* classifier sampai dengan 71 %. Dengan ukuran *rule* yang lebih kecil efisiensi dalam mengklasifikasikan data baru tentu akan semakin baik walau akurasi turun.
4. Akurasi total dari GA-Tree sangat dipengaruhi oleh akurasi dari *rule* yang dihasilkan oleh AG dan coverage data yang dicakup oleh *small disjunct*. Jika akurasi *rule* AG bagus, semakin bagus pula akurasi total dari klasifikasi GA-Tree.

PUSTAKA

Carvalho, Deborah R and Alex Freitas. *A Hybrid Decision tree/Genetic Algorithm Method for Data Mining*.

Holte, Robbert C and Bruce W Porter. *Concept Learning and The Problem of Small disjunct*.

Weiss, Gary M. and Haym Hirsh. (2000). *A Quantitative Study of Small disjuncts*. Department of Computer Science Rutgers University : New Jersey

Weiss, Garry Mitcel. (2003). *The Effect of Small disjuncts and Class Distribution on Decision tree Learning*. New Brunswick. New Jersey

Han, Jiwei and Michelin Kamber. (2001). *Data Mining : Concepts and Techniques*. Intelegent Databases Systems Research Lab, School of Computing Science, Simon Fraser University.

Korkut Koray Gundongan and Bilal Alatas. *Mining Classification Rules by Using Genetic Algorithms with Non-random Initial Population and Uniform Operator*

Witten, Ian H and Eibe Frank. (2000). *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann Publisher

Moertini, Veronica Sri. (2007). *Pengembangan Skalabilitas Algoritma Klasifikasi Dengan Pendekatan Konsep Operator Relasi*. Institut Teknologi Bandung. Bandung.

Saggar, Manish and Abhimanyu Lad. (2004). *Optimization of Association Rule Mining using Improved Genetic Algorithms*. IEEE

Santosa, Budi. (2007). *Data Mining : Teknik Pemanfaatan Data untuk Keperluan Bisnis, Teori dan Klasifikasi.* Graha Ilmu. Yogyakarta

Sumantrika, Ngurah Putu. (2005). *Data Mining Task Klasifikasi Menggunakan Algoritma Genetika*. Teknik Informatika STT TELKOM. Bandung

Suyanto. (2007). *Artificial Intelegence*. Informatika: Bandung

Suyanto. (2008). *Evolutionary Computation : Komputasi Berbasis Evolusi dan Genetika*. Informatika : Bandung

Fowler, Martin. (2004). *UML Distilled: Panduan Singkat Bahasa Pemodelan Standar*. Terjemahan. Penerbit Andi : Yogyakarta