KLASIFIKASI PADA DATA MINING MENGGUNAKAN BOOSTING BERDASARKAN PENDEKATAN PARTISI

Henricus Nova Yudiawan[1], Kiki Maulana[2], Shaufiah[3]

[1]Teknik Informatika, Fakultas Teknik Informatika, Universitas Telkom

Abstrak

Klasifikasi merupakan salah satu kegiatan populer dalam Data Mining yang selalu bersentuhan dengan data berukuran kecil sampai besar. Dalam klasifikasi dataset biasanya langsung dimuat ke dalam memori dan kadang tanpa perhitungan kapasitas memori, akibatnya sering kali proses klasifikasi berjalan lambat karena terlalu banyak file-swapping. Solusi paling sederhana yang biasa terpikirkan adalah menambah kapasitas memori, namun menambah memori saja tidak akan menyelesaikan masalah apabila data yang dimuat terus bertambah.

Dalam Tugas Akhir ini diperkenalkan solusi alternatif selain menambah kapasitas memori, yaitu dengan memecah dataset menjadi beberapa bagian dan dimuat serta diproses bergantian. Solusi inilah yang disebut sebagai pendekatan partisi terhadap data training.

Kata Kunci : dataset, file-swapping, partisi, klasifikasi, memori

Abstract

Classification as one of the most popular task in Data Mining always in touch with small until large data. In the case of classification, dataset usually be load in memory without calculation of memory capacity, so classification often running slow because of file-swapping. The simpliest solution by adding memory capacity is not enough to solve the problem if the data to be load become bigger.

this TA will introduce the alternative solution except adding memory capacity, by partitioning the dataset to some pieces then loading and processing each pieces one by one. This solution called partition approach to data training.

Keywords : dataset, file-swapping, partition, classification, memory

BAB 1

PENDAHULUAN

1.1 Latar belakang masalah

Perkembangan teknologi yang pesat dalam bidang basis data memungkinkan orang, kelompok, atau organisasi untuk mengumpulkan dan menyimpan data dalam jumlah besar yang berasal dari aktivitas sehari-hari mereka. Data tersebut tidak cukup hanya dikumpulkan dan disimpan saja, agar memberikan manfaat maka masih perlu diolah dengan suatu mekanisme yang disebut data mining. Data mining adalah proses menemukan pengetahuan yang berguna secara otomatis dari tempat penyimpanan data yang besar. Data mining sering juga dianggap sebagai bagian tak terpisahkan dari *Knowledge Discovery in Databases* (KDD). Sedangkan KDD sendiri adalah keseluruhan proses mengubah *raw* data menjadi *knowledge* yang berguna.

Salah satu task data mining yang menjadi inspirasi dari Tugas Akhir ini adalah klasifikasi. Klasifikasi terbagi dalam single classification model (misal: *Decision tree*, *Rule-based Classifier, Nearest-neighbor Classifier, Bayesian Classifier, Artificial Neural Network, Support Vector Machine*) dan *multiple classification model/ensemble method* (misal: *Bagging, Boosting, Random Forest*) [2]. *Ensemble method* mampu memperbaiki akurasi dari *single classification model* yaitu dengan melakukan kombinasi terhadap prediksi yang dibuat oleh beberapa *classifier* [2]. Fokus Tugas Akhir ini adalah *Boosting* dengan menerapkan algoritma Adaboost.

Boosting adalah sebuah prosedur iterative untuk mengubah data training secara adaptive dengan fokus pada data yang sulit diklasifikasikan [2]. Tugas-tugas dalam data mining sering kali melibatkan data dalam jumlah sangat besar dan berdasarkan uraian tentang *boosting* di atas, maka kita harus menyediakan main memory dengan kapasitas memadai untuk menampung data training baru, yang mana ukurannya sama dengan data training asli. Jika kapasitas main memory kurang maka proses *boosting* menjadi lambat oleh karena terjadinya swapping dari main memory ke harddisk. Solusi dengan up-grade main memory bukanlah yang terbaik karena main memory yang tersedia di pasar masih memiliki kapasitas yang terbatas sedangkan data bisa

berlipat-lipat ukurannya seiring waktu, selain itu harga main memory yang masih mahal juga merupakan kendala tersendiri bagi kalangan tertentu. Masalah seperti inilah yang dicoba untuk ditemukan solusinya dalam Tugas Akhir ini dengan menerapkan pendekatan partisi data training pada proses *boosting* [1].

1.2 Perumusan masalah

Berdasarkan uraian sebelumnya maka diperlukan solusi alternatif selain meng-upgrade main memory. Solusi alternatif yang layak untuk dipertimbangkan adalah menerapkan *boosting*. Namun *boosting* di sini berbeda dari *boosting* secara normal karena terlebih dahulu dilakukan partisi terhadap data training asli sehingga menghasilkan beberapa bagian. Kemudian *boosting* diterapkan pada masing-masing bagian tersebut. Hasil dari proses *boosting* pada masing-masing bagian kemudian dikombinasikan untuk memperoleh prediksi final terhadap data training. Dalam rangka mewujudkan solusi alternatif ini maka dapat dirumuskan beberapa masalah sebagai berikut :

1. Algoritma apakah yang akan dipakai dalam proses *boosting*?
2. Teknik apakah yang dipakai untuk tahap pembentukan *classifier* pada setiap iterasi dan partisi (misal : teknik *decision tree* induction, rule-based *classifier*, atau neural network).?
3. Bagaimana cara untuk mempartisi data training asli?
4. Data training sebaiknya dibagi dalam berapa partisi?
5. Bagaimana cara untuk mengkombinasikan semua *classifier* yang diperoleh dari setiap partisi sehingga menghasilkan *classifier* final?
6. Spesifikasi hardware dan data yang seperti apakah yang mampu merepresentasikan kondisi seperti disebutkan pada latar belakang masalah?
7. Bagaimana cara melakukan pengujian dan analisis hasil *boosting* yang seperti ini?
8. Bagaimanakah performansi *boosting* dengan partisi data training jika dibandingkan *boosting* yang biasa?

Dalam Tugas Akhir ini ada beberapa batasan masalah yang perlu diperhatikan, antara lain :

1. *Boosting* pada Tugas Akhir ini menggunakan algoritma Adaboost.
2. Teknik *decision tree* induction dipakai dalam tahap pembentukan *classifier* pada setiap partisi.

Hipotesa awal dari tugas akhir yang hendak saya buat ini adalah klasifikasi menggunakan *boosting* dengan pendekatan partisi terhadap data training memiliki hasil akhir sebaik *boosting* yang biasa dengan beberapa keunggulan (diantaranya : tidak boros memori).

1.3 Tujuan

Tujuan penulisan Tugas Akhir ini adalah :

1. Membangun perangkat lunak yang mengimplementasikan teknik *boosting* untuk menyelesaikan masalah klasifikasi dalam data mining.
2. Menganalisis pengaruh partisi data training terhadap proses *boosting*.
3. Menganalisis kelayakan dari pendekatan partisi data training pada proses *boosting* menggunakan algoritma Adaboost.
4. membuktikan apakah keterbatasan *hardware* dalam data mining dapat diatasi dengan teknik *boosting* berdasarkan pendekatan partisi terhadap *data training*.

1.4 Metodologi penyelesaian masalah

Metode penyelesaian masalah dalam penulisan tugas Akhir ini disusun dalam langkah-langkah sebagai berikut:

1. Studi literatur, yang dilakukan dengan membaca dan mempelajari beberapa sumber tertulis (makalah, buku dan jurnal) yang berkaitan dengan data mining, klasifikasi, *ensemble method*, *boosting*, algoritma Adaboost, pendekatan partisi data training terhadap algoritma Adaboost, *decision tree* induction
2. Pengumpulan dan analisis data yang mendukung implementasi dan analisis algoritma Adaboost dalam perangkat, yaitu data dari kompetisi PAKDD 2006
3. Analisis kebutuhan dan perancangan perangkat lunak, untuk menentukan kebutuhan pembangan pembangunan perangkat lunak, serta perancangan struktur data dan aktivitas perangkat lunak yang dibangun.
4. Implementasi, yang merupakan langkah penerapan rancangan yang telah dibuat ke dalam perangkat lunak yang dapat digunakan untuk menyelesaikan masalah klasifikasi dengan algoritma Adaboost menggunakan pendekatan partisi data training.
5. Pengujian dan analisis hasil, yaitu langkah untuk menilai performansi algoritma Adaboost dengan pendekatan partisi data training dalam perangkat lunak terhadap algoritma Adaboost tanpa pendekatan partisi data training.

BAB IV
IMPLEMENTASI DAN PENGUJIAN

4.1 Metode Uji Coba Sistem

Pada bab ini dilakukan pengujian untuk membandingkan *boosting-decision tree* biasa terhadap *boosting-decision tree* yang menggunakan partisi. Parameter yang menjadi pembanding adalah penggunaan memori (*memory usage*) , penggunaan prosesor (processor usage) , waktu training, dan akurasi.

Decision tree yang digunakan dalam sistem ini adalah fungsi yang merupakan paket bawaan MATLAB 7. Sedangkan penulis berfokus kepada penerapan algoritma Adaboost terhadap *decision tree* tersebut dalam dua kategori yaitu dengan partisi dan tanpa partisi.

4.2 Sistem yang Digunakan

Pengujian dilakukan menggunakan perangkat lunak dan perangkat keras sebagai berikut :

1. Prosesor : AthlonXP ~2 GHz
2. RAM : 512 MB
3. Harddisk : 160 GB
4. OS : Windows XP
5. MATLAB 7

4.2 Skenario Pengujian

Dataset yang dipergunakan dalam pengujian adalah data kompetisi yang terdiri dari dua macam, seperti pada tabel berikut ini :

Tabel 4.2 Dataset Pengujian

Nama Data	Data				Keterangan
	Training		Testing		
	Baris	Kolom	Baris	Kolom	
PKADD 2006	18000, terdiri dari : * 15000 3G * 3000 2G	250	6000 terdiri dari : * 5000 3G * 1000 2G	250	Imbalance dengan 2 kelas (3G dan 2G)
					Merupakan data customer sebuah perusahan telekomunikasi seluler
EUNITE	12000, terdiri dari : * 6000 aktif * 6000 pasif	37	12000, terdiri dari : * 6000 aktif * 6000 pasif	37	Balance dengan 2 kelas (aktif dan pasif)
					Merupakan data customer bank yang terdiri dari customer aktif dan customer pasif.

Sebelum dipergunakan dalam proses training, terlebih dahulu dilakukan *cleaning* terhadap data training. *Cleaning* dilakukan secara terurut, dimulai dengan menghilangkan atribut yang mayoritas berisi NULL, kemudian dilanjutkan dengan menghilangkan record yang masih mengandung NULL. Jika terjadi *cleaning* yang mengakibatkan pembuangan atribut pada data traning maka harus dilakukan pembuangan atribut yang sama pada data testing. Cleaning tidak wajib dilakukan, akan tetapi sebagai bagian preprosesing maka sebaiknya dilakukan. Pembentukan data partisi dilakukan di luar sistem dengan melakukan partisi secara horizontal dan vertikal. Masing-masing teknik partisi(horizontal, vertikal) akan membagi data training menjadi beberapa bagian. Dengan pertimbangan bahwa terlalu banyak partisi cenderung akan menurunkan akurasi karena kegagalan data partisi dalam merepresentasikan keseluruhan data training. Pembagian atau pemartisian data training dilakukan secara random. Pembagian secara horizontal berbeda dengan yang vertikal, yaitu pada pembagian secara horizontal dilakukan pembagian secara proporsional antara satu kelas dengan kelas lainnya, sedangkan pembagian secara vertikal dilakukan secara random biasa.

Semua pengujian dilakukan menggunakan iterasi boosting sebanyak 4, 7, dan 10 kali, berikut ini detail pengujian yang dilakukan :

Tabel 4.3 Skenario Pengujian

Nama Data	Jenis Pengujian	*Imbalance*			*Balance*
		Non-sampling	*Under-sampling*	*Over-sampling*	*Non-sampling*
PAKDD 2006	Tanpa Partisi	x	x	x	
	Partisi Horizontal	x	x	x	
	Partisi Vertikal	x	x	x	
EUNITE	Tanpa Partisi				x
	Partisi Horizontal				x
	Partisi Vertikal				x

4.3 Hasil Pengujian

4.3.1 PAKDD 2006

4.3.1.1 Teknik Tanpa Partisi - Teknik Partisi Horizontal

- Kasus data imbalance tidak dilakukan proses *balancing*

Tabel 4.4.1 Perbandingan Teknik Tanpa Partisi-Teknik Partisi Horizontal pada Kasus *Imbalance*
4 iterasi

Partisi	Akurasi (%)			Memori (Kb)	Waktu (detik)	Prosesor (%)
	3G	2G	R			
1	16	84.02	50.01	30320	568	99
2	19.3	79.26	49.28	23005	150	99
3	25.2	72.54	48.87	12624	70	99
4	19.6	79.29	49.45	9231	95	99
5	9.96	92.82	51.39	5336	112	99
6	8.45	92.93	50.69	4923	126	99
7	6.9	94.84	50.87	3960	130	99
8	5.21	94.98	50.01	3560	123	99
9	4.57	95.06	49.82	3200	111	99
10	3.62	96.80	50.21	2836	105	99

Tabel 4.4.2 Perbandingan Teknik Tanpa Partisi-Teknik Partisi Horizontal pada Kasus *Imbalance*
7 iterasi

Partisi	Akurasi (%)			Memori (Kb)	Waktu (detik)	Prosesor (%)
	3G	2G	R			
1	20.8	79.90	50.35	30320	1421	99
2	26.5	70.89	48.7	23005	365	99
3	31.92	63.15	47.54	12624	117	99
4	22.61	78.32	50.47	9231	156	99
5	12.6	88.44	50.52	5336	195	99
6	10.03	91.89	50.96	4923	218	99
7	8.78	92.02	50.4	3960	240	99
8	7.36	93.58	50.47	3560	226	99
9	6.78	94.24	50.51	3200	198	99
10	5.9	95.00	50.45	2836	180	99

Tabel 4.4.3 Perbandingan Teknik Tanpa Partisi-Teknik Partisi Horizontal pada Kasus *Imbalance*
10 iterasi

Partisi	Akurasi (%)			Memori (Kb)	Waktu (detik)	Prosesor (%)
	3G	2G	R			
1	26.5	75.96	51.23	30320	1847	99
2	34.59	67.32	50.96	23005	458	99
3	42	59.58	50.79	12624	168	99
4	31.5	70.01	50.75	9231	236	99
5	16.6	84.38	50.49	5336	280	99
6	14.06	85.23	49.65	4923	304	99
7	11.5	88.04	49.77	3960	321	99
8	10.23	88.65	49.44	3560	273	99
9	8.97	89.42	49.2	3200	260	99
10	7.7	90.98	49.34	2836	250	99

Keterangan :

- Jumlah partisi = 1 menyatakan bahwa data training tetap utuh, kondisi ini menyatakan bahwa data training digunakan pada teknik tanpa partisi. Jumlah partisi selain 1 menyatakan bahwa data training dipakai pada teknik partisi horizontal.
- 3G : prosentase yang menyatakan banyaknya kelas 3G yang diprediksi benar sebagai kelas 3G.
- 2G : prosentase yang menyatakan banyaknya kelas 2G yang diprediksi benar sebagai kelas 2G
- R : akurasi rata-rata yang diperoleh dengan menjumlahkan akurasi 3G dan 2G, kemudian dibagi dua.
- Memori : penggunaan memori saat training.
- Waktu : lama proses training.
- Prosesor : penggunaan prosesor selama training.

Dari tabel 4, 7, 10 iterasi di atas, oleh karena kelas minoritas pada imbalance begitu penting maka dipilih tabel yang menggunakan 10 iterasi sebagai tabel terbaik berdasarkan kemampuan mengenali kelas minoritas yang lebih baik.

Beberapa fakta yang diperoleh dari tabel 10 iterasi (tabel 4.4.3) di atas adalah sebagai berikut :

1. Jumlah partisi berbanding terbalik dengan penggunaan memori.
2. Akurasi rata-rata (R) stabil di sekitar 50%, nilai R terbaik adalah 51,23% menjadi milik teknik tanpa partisi, akurasi 3G terbaik dari teknik tanpa partisi adalah 26,5%, akurasi terbaik dari teknik partisi horizontal adalah 42% yaitu saat jumlah partisi = 3, akurasi 3G menurun terus saat jumlah partisi lebih dari 3, dan selalu akurasi 2G > 3G.
3. Waktu training tak menentu, kadang naik kadang turun saat jumlah partisi bertambah, dan terjadi penurunan drastis dari teknik tanpa partisi terhadap teknik partisi horizontal.
4. Penggunaan prosesor setara untuk semua jumlah partisi, yaitu 99%.

- Kasus data *imbalance* yang di-*balancing* dengan cara *undersampling*.

Tabel 4.5.1 Perbandingan Teknik Tanpa Partisi-Teknik Partisi Horizontal pada Kasus *Undersampling* 4 iterasi

Partisi	Akurasi (%)			Memori (Kb)	Waktu (detik)	Prosesor (%)
	3G	2G	R			
1	30.48	56.23	43.38	8700	66	99
2	30.01	57.11	43.56	5025	63	99
3	28.68	57.95	41.32	3236	57	99
4	26.56	67.23	46.89	2563	39	99
5	21.48	72.97	47.23	1587	23	99
6	21.73	73.51	47.62	1359	25	99
7	21.24	73..24	47.24	1298	29	99
8	19.08	76.32	47.7	1078	29	99
9	17.87	80.25	49.06	903	25	99
10	15.24	81.99	48.62	859	24	99

Tabel 4.5.2 Perbandingan Teknik Tanpa Partisi-Teknik Partisi Horizontal pada Kasus *Undersampling* 7 iterasi

Partisi	Akurasi (%)			Memori (Kb)	Waktu (detik)	Prosesor (%)
	3G	2G	R			
1	40.4	53.74	47.07	8700	115	99
2	39.21	54.65	46.93	5025	106	99
3	37.28	55.32	46.3	3236	100	99
4	31.83	64.12	47.98	2563	83	99
5	27.92	69.65	48.78	1587	41	99
5	27.69	69.23	48.46	1359	47	99
7	27.61	69.9	48.76	1298	52	99
8	24.45	72.98	48.72	1078	50	99
9	20.65	76.11	48.38	903	47	99
10	19.81	79.32	49.65	859	45	99

Tabel 4.5.3 Perbandingan Teknik Tanpa Partisi-Teknik Partisi Horizontal pada Kasus *Undersampling* 10 iterasi

Partisi	Akurasi (%)			Memori (Kb)	Waktu (detik)	Prosesor (%)
	3G	2G	R			
1	50.8	51.18	50.99	8700	161	99
2	50.2	51.36	50.78	5025	156	99
3	47.8	52.68	50.24	3236	143	99
4	42.3	58.59	50.45	2563	89	99
5	35.8	66.34	51.07	1587	56	99
6	35.7	66.39	51.05	1359	62	99
7	35.4	66.58	50.99	1298	71	99
8	32.8	67.23	50.02	1078	68	99
9	29.31	71.51	50.41	903	63	99
10	25.4	74.54	49.97	859	60	99

Dari tabel 4, 7, 10 iterasi di atas, oleh karena kelas minoritas pada imbalance begitu penting maka dipilih tabel yang menggunakan 10 iterasi sebagai tabel terbaik berdasarkan kemampuan mengenali kelas minoritas yang lebih baik.

Beberapa fakta yang diperoleh dari tabel 10 iterasi (tabel 4.5.3) di atas adalah sebagai berikut

1. Jumlah partisi berbanding terbalik dengan penggunaan memori.
2. Akurasi R stabil di sekitar 50%, akurasi R terbaik 51,07% pada saat jumlah partisi = 5, akurasi 3G terbaik dari teknik tanpa partisi adalah 50,8% dan akurasi 3G terbaik dari teknik partisi adalah 50,2% yaitu saat jumlah partisi = 2, akurasi 3G berbanding terbalik dengan jumlah partisi, akurasi 2G selalu lebih besar daripada akurasi 3G.
3. Waktu training tak menentu, perubahan naik-turunnya waktu training tidak terlalu mencolok antara teknik tanpa partisi terhadap teknik partisi horizontal.
4. Penggunaan prosesor untuk semua jumlah partisi adalah setara yaitu 99%.

- Kasus data imbalance yang di-*balancing* dengan cara *oversampling*

Tabel 4.6.1 Perbandingan Teknik Tanpa Partisi-Teknik Partisi Horizontal pada Kasus *Oversampling* 4 iterasi

Partisi	Akurasi (%)			Memori (Kb)	Waktu (detik)	Prosesor (%)
	3G	2G	R			
1	14	88.22	51.11	43798	4388	99
2	1.32	100	50.66	25362	504	99
3	0.72	100	50.36	14234	300	99
4	0.54	100	50.27	12540	614	99
5	0.12	100	50.06	8954	1020	99
6	0	100	50	7456	210	99
7	0	100	50	6103	156	99
8	0	100	50	5127	150	99
9	0	100	50	4825	135	99
10	0	100	50	4289	120	99

Tabel 4.6.2 Perbandingan Teknik Tanpa Partisi-Teknik Partisi Horizontal pada Kasus *Oversampling* 7 iterasi

Partisi	Akurasi (%)			Memori (Kb)	Waktu (detik)	Prosesor (%)
	3G	2G	R			
1	18.17	84.68	51.43	43798	3070	99
2	1.57	100	50.79	25362	421	99
3	0.95	100	50.48	14234	220	99
4	0.72	100	50.36	12540	489	99
5	0.16	100	50.08	8954	720	99
6	0	100	50	7456	159	99
7	0	100	50	6103	109	99
8	0	100	50	5127	98	99
9	0	100	50	4825	92	99
10	0	100	50	4289	86	99

Tabel 4.6.3 Perbandingan Teknik Tanpa Partisi-Teknik Partisi Horizontal pada Kasus *Oversampling* 10 iterasi

Partisi	Akurasi (%)			Memori (Kb)	Waktu (detik)	Prosesor (%)
	3G	2G	R			
1	23	80.2	51.6	43798	4388	99
2	1.81	99.11	50.46	25362	511	99
3	1.2	98.9	50.05	14234	300	99
4	0.93	99.23	50.08	12540	583	99
5	0.2	99.74	49.97	8954	1020	99
6	0	99.87	49.94	7456	193	99
7	0	99.94	49.97	6103	156	99
8	0	100	50	5127	148	99
9	0	100	50	4825	129	99
10	0	100	50	4289	120	99

Dari tabel 4, 7, 10 iterasi di atas, oleh karena kelas minoritas pada imbalance begitu penting maka dipilih tabel yang menggunakan 10 iterasi sebagai tabel terbaik berdasarkan kemampuan mengenali kelas minoritas yang lebih baik.

Beberapa fakta yang diperoleh dari tabel 10 iterasi (tabel 4.6.3) di atas adalah sebagai berikut:

1. Jumlah partisi berbanding terbalik dengan penggunaan memori.
2. Akurasi R stabil di sekitar 50%, akurasi R terbaik adalah 51,6% yang menjadi milik teknik tanpa partisi, akurasi 3G terbaik dari teknik tanpa partisi adalah 23% dan akurasi 3G terbaik dari teknik partisi adalah 1,81% saat jumlah partisi = 2, akurasi 3G hasil teknik partisi horizontal jauh lebih buruk daripada teknik tanpa partisi, akurasi 2G selalu lebih besar daripada akurasi 3G.
3. Waktu training tak menentu, naik-turunnya waktu training sangat mencolok antara teknik tanpa partisi dengan teknik partisi horizontal.

4. Penggunaan prosesor setara untuk semua jumlah partisi, yaitu 99%.

Kesimpulan yang dapat ditarik dari teknik partisi horizontal adalah :

1. Jumlah partisi berbanding terbalik dengan penggunaan memori.
2. akurasi rata-rata stabil di sekitar 50%, teknik partisi horizontal menurunkan akurasi 3G dan meningkatkan akurasi 2G seiring pertambahan jumlah partisi, akurasi 2G selalu lebih besar daripada akurasi 3G.
3. Urutan akurasi 3G pada partisi horizontal dimulai dari yang terbaik ke yang terjelek : partisi horizontal - *undersampling* > partisi horizontal - *imbalance* > partisi horizontal - *oversampling*.
4. waktu training tak menentu, waktu training paling cepat adalah pada kasus *undersampling* dimana waktu training teknik tanpa partisi dan teknik partisi tidak berbeda terlalu jauh, perbedaan waktu training sangat jauh antara teknik tanpa partisi dan teknik partisi terjadi pada kasus *imbalance* dan *oversampling.*
5. Penggunaan prosesor selalu 99%.

4.3.1.2 Teknik Tanpa Partisi - Teknik Partisi Vertikal

- Kasus data *imbalance* tidak dilakukan proses *balancing*

Tabel 4.7.1 Perbandingan Teknik Tanpa Partisi – Teknik Partisi Vertikal pada Kasus *Imbalance* 4 iterasi

Partisi	Akurasi (%)			Memori (Kb)	Waktu (detik)	Prosesor (%)
	3G	2G	R			
1	16	83.56	49.78	30320	740	99
2	15.2	83.60	49.4	21005	652	99
3	15.6	84.5	50.05	13084	490	99
4	16.01	84.92	50.47	9563	512	99
5	17.4	82.7	50.5	5632	970	99
6	16.23	82.59	49.41	4821	824	99
7	14.8	82.69	48.75	4122	645	99
8	13.82	83.72	48.77	3951	839	99
9	11.35	85.16	48.26	3420	1027	99
10	10.7	87.69	49.2	2968	1500	99

Tabel 4.7.2 Perbandingan Teknik Tanpa Partisi – Teknik Partisi Vertikal pada Kasus *Imbalance* 7 iterasi

Partisi	Akurasi (%)			Memori (Kb)	Waktu (detik)	Prosesor (%)
	3G	2G	R			
1	20.94	79.76	50.35	30320	1290	99
2	20.38	80.23	50.31	21005	1068	99
3	20.54	80.66	50.6	13084	840	99
4	21.41	79.77	50.59	9563	1132	99
5	22.91	78.94	50.93	5632	1679	99
6	20.25	78.05	49.15	4821	1209	99
7	19.6	77.5	48.55	4122	1046	99
8	17.65	79.97	48.81	3951	1195	99
9	15.2	81.59	48.39	3420	1205	99
10	14	83.8	48.9	2968	1420	99

Tabel 4.7.3 Perbandingan Teknik Tanpa Partisi – Teknik Partisi Vertikal pada Kasus *Imbalance* 10 iterasi

Partisi	Akurasi (%)			Memori (Kb)	Waktu (detik)	Prosesor (%)
	3G	2G	R			
1	26.5	75.96	51.23	30320	1847	99
2	26	76.29	51.15	21005	1465	99
3	26.2	76.82	51.51	13084	1200	99
4	27.84	76.05	51.95	9563	1800	99
5	29	75.18	52.09	5632	2392	99
6	26.94	74	50.47	4821	1745	99
7	24.8	73.8	49.3	4122	1500	99
8	20.01	75.6	47.8	3951	1832	99
9	18.97	77.12	48.05	3420	1905	99
10	17.8	79.72	48.76	2968	2040	99

Dari tabel 4, 7, 10 iterasi di atas, oleh karena kelas minoritas pada *imbalance* begitu penting maka dipilih tabel yang menggunakan 10 iterasi sebagai tabel terbaik berdasarkan kemampuan mengenali kelas minoritas yang lebih baik.

Beberapa fakta yang diperoleh dari tabel 10 iterasi (tabel 4.7.3) di atas adalah sebagai berikut

1. Jumlah partisi berbanding terbalik dengan penggunaan memori.
2. Akurasi R stabil di sekitar 50%, akurasi rata-rata terbaik adalah 52,09% menjadi milik teknik partisi saat partisi = 5, akurasi terhadap 3G terbaik dari teknik tanpa partisi adalah 26,5% dan akurasi 3G terbaik dari teknik partisi adalah 29% saat jumlah partisi = 5, dan akurasi 2G selalu lebih tinggi daripada akurasi 3G.
3. waktu training tak menentu

4. Penggunaan prosesor untuk semua jumlah partisi adalah setara yaitu 99%.

- Kasus data *imbalance* yang di-*balancing* dengan cara *undersampling*.

Tabel 4.8.1 Perbandingan Teknik Tanpa Partisi – Teknik Partisi Vertikal pada Kasus *Undersampling* 4 iterasi

Partisi	Akurasi (%)			Memori (Kb)	Waktu (detik)	Prosesor (%)
	3G	2G	R			
1	30.48	65.2	47.84	8700	65	99
2	25.87	68.42	47.15	3500	58	99
3	26.1	68.53	47.32	2976	56	99
4	26.84	69.51	48.18	2253	63	99
5	27.5	70.14	48.82	1409	75	99
6	28.04	67.88	47.96	1356	76	99
7	28.92	66.36	47.64	1156	80	99
8	25.78	69.55	47.67	1058	98	99
9	23.13	72.46	47.79	925	113	99
10	20.28	75.8	48.04	803	130	99

Tabel 4.8.2 Perbandingan Teknik Tanpa Partisi – Teknik Partisi Vertikal pada Kasus *Undersampling* 7 iterasi

Partisi	Akurasi (%)			Memori (Kb)	Waktu (detik)	Prosesor (%)
	3G	2G	R			
1	40.13	53.74	46.94	8700	110	99
2	33.85	58.54	46.19	3500	103	99
3	34.37	58.74	46.56	2976	100	99
4	35.02	57.21	46.12	2253	115	99
5	36.18	56.45	46.32	1409	125	99
6	36.98	56	46.49	1356	131	99
7	38.08	55.8	46.94	1156	140	99
8	32.25	65	48.63	1058	214	99
9	28.49	68.56	48.53	925	289	99
10	26.7	72.37	49.54	803	320	99

Tabel 4.8.3 Perbandingan Teknik Tanpa Partisi – Teknik Partisi Vertikal pada Kasus *Undersampling* 10 iterasi

Partisi	Akurasi (%)			Memori (Kb)	Waktu (detik)	Prosesor (%)
	3G	2G	R			
1	50.8	51.18	50.99	8700	161	99
2	42.9	56.23	49.57	3500	156	99
3	43.5	55.94	49.72	2976	140	99
4	44.25	54.02	49.14	2253	164	99
5	45.8	53.76	49.78	1409	180	99
6	46.51	52	49.26	1356	193	99
7	48.2	51.24	49.69	1156	200	99
8	43	57	50	1058	354	99
9	38	62.85	50.43	925	421	99
10	33.8	68.92	51.36	803	450	99

Dari tabel 4, 7, 10 iterasi di atas, oleh karena kelas minoritas pada *imbalance* begitu penting maka dipilih tabel yang menggunakan 10 iterasi sebagai tabel terbaik berdasarkan kemampuan mengenali kelas minoritas yang lebih baik.

Beberapa fakta yang diperoleh dari tabel 10 iterasi (tabel 4.8.3) di atas adalah sebagai berikut :

1. Jumlah partisi berbanding terbalik dengan penggunaan memori.
2. Akurasi R stabil di sekitar 50%, akurasi R terbaik adalah 51,36% menjadi milik teknik partisi dengan 10 buah partisi,akurasi 3G terbaik dari teknik tanpa partisi adalah 50,8% dan akurasi 3G terbaik teknik partisi adalah 48,2% saat jumlah partisi = 7, akurasi 2G selalu lebih tinggi daripada akurasi 3G.
3. Waktu berbanding lurus dengan jumlah partisi untuk jumlah partisi > 2, tetapi relatif lebih cepat daripada waktu training pada kasus imbalance.
4. Penggunaan prosesor untuk semua jumlah partisi adalah setara yaitu 99%.

- Kasus data *imbalance* yang di-*balancing* dengan cara *oversampling*

Tabel 4.9.1 Perbandingan Teknik Tanpa Partisi – Teknik Partisi Vertikal pada Kasus *Oversampling* 4 iterasi

Partisi	Akurasi (%)			Memori (Kb)	Waktu (detik)	Prosesor (%)
	3G	2G	R			
1	13.8	88.22	51.01	43798	1755	99
2	6.3	94.68	50.49	35214	1500	99
3	6.6	95.7	51.15	23056	1270	99
4	5.7	96	50.85	16450	1480	99
5	5.2	96.12	50.66	9230	1765	99
6	5.87	96.35	51.19	8723	1744	99
7	6.5	96.5	51.5	8109	1728	99
8	3.25	97.5	50.38	7520	1980	99
9	1.2	99.25	50.23	6824	2080	99
10	0.78	99.89	50.34	6253	2150	99

Tabel 4.9.2 Perbandingan Teknik Tanpa Partisi – Teknik Partisi Vertikal pada Kasus *Oversampling* 7 iterasi

Partisi	Akurasi (%)			Memori (Kb)	Waktu (detik)	Prosesor (%)
	3G	2G	R			
1	18.17	84.21	51.19	43798	3080	99
2	7.82	94.55	51.18	35214	2506	99
3	8.69	93.26	50.98	23056	2225	99
4	7.54	94.62	51.08	16450	2891	99
5	6.79	96.43	51.61	9230	3090	99
6	7.01	96.15	51.58	8723	3062	99
7	8.53	96.08	52.3	8109	3025	99
8	5.23	98.2	51.72	7520	3256	99
9	3.12	99.97	51.55	6824	3589	99
10	1.03	100	50.52	6253	3800	99

Tabel 4.9.3 Perbandingan Teknik Tanpa Partisi – Teknik Partisi Vertikal pada Kasus *Oversampling* 10 iterasi

Partisi	Akurasi (%)			Memori (Kb)	Waktu (detik)	Prosesor (%)
	3G	2G	R			
1	23	80.2	51.6	43798	4388	99
2	10.58	87.59	49.09	35214	3850	99
3	11	88.82	49.91	23056	3180	99
4	9.5	90.03	49.77	16450	3862	99
5	8.6	91.84	50.22	9230	4413	99
5	9.13	91.6	50.37	8723	4398	99
7	10.8	91.5	51.15	8109	4320	99
8	6.2	95.5	50.85	7520	4800	99
9	4.89	96	50.45	6824	5109	99
10	1.3	99.38	50.34	6253	5400	99

Dari tabel 4, 7, 10 iterasi di atas, oleh karena kelas minoritas pada *imbalance* begitu penting maka dipilih tabel yang menggunakan 10 iterasi sebagai tabel terbaik berdasarkan kemampuan mengenali kelas minoritas yang lebih baik.

Beberapa fakta yang diperoleh dari tabel 10 iterasi (tabel 4.9.3) di atas adalah sebagai berikut :

1. Jumlah partisi berbanding terbalik dengan penggunaan memori.
2. Akurasi R stabil yaitu sekitar 50%, akurasi R terbaik adalah 51,6% menjadi milik teknik tanpa partisi, akurasi 3G terbaik dari teknik tanpa partisi adalah 23% dan akurasi terbaik dari teknik partisi adalah 11% saat jumlah partisi = 3, dan akurasi 3G selalu lebih kecil daripada akurasi 2G.
3. Waktu training tak menentu , tetapi relatif lebih lama daripada waktu training pada kasus imbalance dan undersampling.

4. Penggunaan prosesor untuk semua jumlah partisi adalah setara yaitu 99%.

Kesimpulan yang dapat ditarik dari teknik partisi vertikal adalah :

1. Jumlah partisi berbanding terbalik dengan penggunaan memori.
2. Akurasi rata-rata stabil di sekitar 50%, akurasi 2G selalu lebih besar daripada akurasi 3G.
3. Urutan akurasi 3G pada partisi vertikal dimulai dari yang terbaik ke yang terjelek : partisi vertikal-*undersampling* > partisi vertikal-*imbalance* > partisi vertikal-*oversampling*.
4. Waktu training tak menentu, waktu training partisi vertikal pada kasus *undersampling* berbanding lurus dengan jumlah partisi saat jumlah partisi >3, waktu training kasus undersampling jauh lebih cepat daripada kasus imbalance maupun *oversampling*.
5. Penggunaan prosesor selalu 99%.

4.3.2 EUNITE

4.3.2.1 Teknik Tanpa Partisi - Teknik Partisi Horizontal

- Kasus data *balance*

Tabel 4.10.1 Perbandingan Teknik Tanpa Partisi-Teknik Partisi Horizontal pada Kasus *Balance* 4 iterasi

Partisi	Akurasi (%)			Memori (Kb)	Waktu (detik)	Prosesor (%)
	A	B	R			
1	40.89	31.64	36.26	3604	100	99
2	32.51	39.25	35.88	1832	38	99
3	30.96	36.79	33.88	1540	25	99
4	25.45	40.26	32.86	986	23	99
5	20.03	45.30	32.67	696	20	99
6	21.52	41.38	31.45	300	17	99
7	20.48	42.56	31.52	270	17	99
8	19.36	43.29	31.33	225	16	99
9	18.49	44.50	31.49	210	15	99
10	17.36	45.21	31.29	192	15	99

Tabel 4.10.2 Perbandingan Teknik Tanpa Partisi-Teknik Partisi Horizontal pada Kasus *Balance* 7 iterasi

Partisi	Akurasi (%)			Memori (Kb)	Waktu (detik)	Prosesor (%)
	A	B	R			
1	43.57	34.18	38.88	3604	175	99
2	35.20	36.52	35.86	1832	68	99
3	32.65	38.37	35.51	1540	42	99
4	27.23	42.68	34.96	986	41	99
5	21.91	46.43	34.17	696	39	99
6	23.97	42.65	33.31	300	28	99
7	22.53	43.68	33.11	270	28	99
8	21.29	44.89	33.09	225	26	99
9	20.50	46.52	33.51	210	26	99
10	19.89	47.56	33.73	192	26	99

Tabel 4.10.3 Perbandingan Teknik Tanpa Partisi-Teknik Partisi Horizontal pada Kasus *Balance* 10 iterasi

Partisi	Akurasi (%)			Memori (Kb)	Waktu (detik)	Prosesor (%)
	A	B	R			
1	45.48	38.95	42.22	3604	250	99
2	46.50	36.10	41.3	1832	85	99
3	47.08	35.52	41.3	1540	59	99
4	38.23	39.54	38.89	986	57	99
5	32.83	44.93	38.88	696	55	99
6	33.88	41.23	37.52	300	40	99
7	31.02	42.56	36.79	270	40	99
8	28.13	43.85	35.99	225	39	99
9	25.56	44	34.78	210	38	99
10	22.07	45.62	33.85	192	37	99

Keterangan :

- A : prosentase yang menyatakan banyaknya kelas A yang diprediksi benar sebagai kelas A.
- B : prosentase yang menyatakan banyaknya kelas B yang diprediksi benar sebagai kelas B.

Dari tabel 4, 7, 10 iterasi di atas, oleh karena pentingnya keseimbangan akurasi kelas A dan B pada data yang asli balance dari semula maka dipilih tabel yang menggunakan 10 iterasi sebagai tabel terbaik berdasarkan kesimbangan akurasi kelas A dan B yang lebih baik.

Beberapa fakta yang diperoleh dari tabel 10 iterasi (tabel 4.10.3) di atas adalah sebagai berikut :

1. Jumlah partisi berbanding terbalik dengan penggunaan memori.

2. Jumlah partisi berbanding terbalik dengan akurasi rata-rata.
3. Akurasi rata-rata (R) terbaik adalah 42,22% yang menjadi milik teknik tanpa partisi.
4. Perbedaan nilai akurasi A dan akurasi B pada teknik partisi cukup jauh, padahal seharusnya keduanya memiliki nilai setara karena kedua kelas (A dan B) adalah sama pentingnya. Hal ini berbeda dengan yang terjadi pada teknik tanpa partisi, dimana perbedaan akurasi kelas A dan kelas B tidak terlalu jauh.
5. Jumlah partisi berbanding terbalik dengan waktu training.
6. Penggunaan prosesor untuk semua jumlah partisi adalah setara yaitu 99%.

4.3.2.2 Teknik Tanpa Partisi - Teknik Partisi Vertikal

- Kasus data *balance*

Tabel 4.11.1 Perbandingan Teknik Tanpa Partisi-Teknik Partisi Vertikal pada Kasus *Balance* 4 iterasi

Partisi	Akurasi (%)			Memori (Kb)	Waktu (detik)	Prosesor (%)
	A	B	R			
1	41.20	32.57	36.89	3604	110	99
2	41.95	34.25	38.1	1953	150	99
3	42.83	41.62	42.23	1556	185	99
4	51.22	36.19	43.71	1208	230	99
5	61.56	31.52	46.54	900	260	99
6	57.71	39.84	48.78	725	359	99
7	52.46	46.21	49.34	532	390	99
8	43.20	51.26	47.23	501	463	99
9	37.63	58.45	48.04	489	510	99
10	31.10	63.06	47.08	456	580	99

Tabel 4.11.2 Perbandingan Teknik Tanpa Partisi-Teknik Partisi Vertikal pada Kasus *Balance* 7 iterasi

Partisi	Akurasi (%)			Memori (Kb)	Waktu (detik)	Prosesor (%)
	A	B	R			
1	43.56	34.23	38.89	3604	180	99
2	44.15	41.20	42.68	1953	256	99
3	45.89	42.02	43.96	1556	325	99
4	55.17	38.5	46.84	1208	413	99
5	63.25	33.10	48.18	900	470	99
6	59.73	41.03	50.38	725	551	99
7	54.38	48.56	51.47	532	675	99
8	48.21	54.55	51.38	501	739	99
9	42.16	59.18	50.67	489	843	99
10	32.69	64.85	48.77	456	990	99

Tabel 4.11.3 Perbandingan Teknik Tanpa Partisi-Teknik Partisi Vertikal pada Kasus *Balance* 10 iterasi

Partisi	Akurasi (%)			Memori (Kb)	Waktu (detik)	Prosesor (%)
	A	B	R			
1	45.48	38.95	42.22	3604	250	99
2	44.92	42.14	43.53	1953	370	99
3	46.17	45.12	45.67	1556	452	99
4	58.10	43.3	50.7	1208	548	99
5	65.6	37.65	51.63	900	656	99
6	60.02	44.88	52.45	725	705	99
7	55.22	53.5	54.36	532	960	99
8	52.31	56.21	54.26	501	1032	99
9	46.69	61.89	54.29	489	1198	99
10	36.85	72.08	54.47	456	1402	99

Dari tabel 4, 7, 10 iterasi di atas, oleh karena pentingnya keseimbangan akurasi kelas A dan B pada data yang asli balance dari semula maka dipilih tabel yang menggunakan 10 iterasi sebagai tabel terbaik berdasarkan kesimbangan akurasi kelas A dan B yang lebih baik.

Beberapa fakta yang diperoleh dari tabel 10 iterasi (tabel 4.11.3) di atas adalah sebagai berikut :

1. Jumlah partisi berbanding terbalik dengan penggunaan memori.
2. Jumlah partisi berbanding lurus dengan akurasi rata-rata (R).
3. Akurasi rata-rata yang lebih besar tidak selalu diikuti oleh keseimbangan nilai akurasi A dan nilai akurasi B. Sebagai contoh akurasi rata-rata saat menggunakan 10 partisi lebih baik daripada saat menggunakan 7 partisi, namun perbedaan nilai akurasi A dan B justru lebih mencolok pada saat menggunakan 10 partisi.
4. Jumlah partisi berbanding lurus dengan waktu training.
5. Penggunaan prosesor untuk semua jumlah partisi adalah setara yaitu 99%.