Applied Information System and Management (AISM)
Volume 5, (2) 2022, p. 75-82
P-ISSN: 2621-2536; E-ISSN: 2621-2544; DOI: 10.15408/aism.v5i2.24887

Peningkatan Performa *Decision Tree* dengan *AdaBoost* untuk Klasifikasi Kekurangtransparanan Informasi Anti-Korupsi

Zico Karya Saputra Domas[1], Roby Rakhmadi[2]

***Abstrak*—Di era *big data* saat ini, peran teknik *data mining* sangatlah dibutuhkan terkait kebutuhan pengambilan keputusan yang akurat. Algoritma *decision tree* telah lazim diterapkan untuk menemukan pola klasifikasi karena mudah diinterpretasikan namun harus senantiasa dievaluasi tingkat performanya. *Adaboost* merupakan salah satu metode untuk meningkatkan performa algoritma *decision tree*. Eksperimen dilakukan pada 141 sampel perusahaan yang melantai di Bursa Efek Indonesia sektor konstruksi-infrastruktur, pertambangan-perminyakan, dan sektor perbankan pada periode 2019, dengan menerapkan teknik *adaboost* pada *decision tree* dengan parameter *maximum depth* dan *confidence* yang diuji dalam enam skenario berbeda berdasarkan informasi anti-korupsi pada pengungkapan laporan tahunan perusahaan. Hasil ekperimen *decision tree* untuk akurasi sebesar 69,5%, AUC-optimistis 0,826, dan AUC 0,756, sedangkan rerata hasil dari enam skenario *decision tree* versi *adaBoost* untuk akurasi sebesar 71,16%, AUC-optimistis 0,8905, dan AUC 0,744, sehingga dapat disimpulkan bahwa pengklasifikasian atas prediksi klasifikasi dengan metode *adaBoost* layak diterapkan sebagai upaya alternatif untuk meningkatkan tingkat performa yang lebih baik.**

Kata kunci*—Anti-korupsi, Transparansi, *Data mining,* Klasifikasi, *Decision tree, Adaboost.

I. PENDAHULUAN

Metode klasifikasi dengan algoritma *decision tree* memiliki beragam kelebihan, mulai dari visualisasi pohon keputusan yang mudah diinterprestasikan, tingkat akurasi yang cukup layak, efisien dalam menangani atribut yang bertipe diskret maupun yang bertipe numerik. Menurut Quinlan, *decision tree* dapat menangani *overfitting* atribut yang kontinu, memilih *attribute selection* yang tepat, menangani *training data* meski dengan kondisi nilai atribut yang hilang, hingga menghasilkan efisiensi kalkulasi [1]. Namun, algoritma *decision tree* juga rentan menghasilkan kalkulasi pengklasifikasian yang salah karena *noisy* maupun ketidakseimbangan data sehingga tingkat akurasi menjadi kurang optimal. Pada umumnya, *decision tree* dirancang untuk *dataset* yang relatif seimbang namun memiliki kelemahan pada entropi dan ketika *dataset* memiliki *class imbalance* yang timpang [2]. Distribusi *class imbalance* salah satunya bercorak lebih banyaknya suatu kelas kasus dibandingkan kelas lainnya, misalnya suatu kelas diwakili oleh sampel yang sangat besar sementara kelas lainnya diwakili oleh sampel yang jauh lebih minim [3].

Secara alamiah, kondisi ketidakseimbangan label atau kelas merupakan permasalahan yang lazim terjadi karena kelas atas suatu data tidak dapat dipaksa untuk selalu merata atau berimbang [4] sehingga [5], [6], dan [7] juga menilai adanya potensi risiko yang dapat menurunkan tingkat akurasi pada fungsi klasifikasi. Kondisi *class imbalance* dalam kendala distribusi data ada kalanya dapat ditanggulangi dengan metode *sampling* [8]. Namun, beberapa teknik untuk mengatasi *class imbalance* seperti *oversampling* cenderung mengurangi jumlah pemangkasan sementara *under-sampling* cenderung memperbanyak jumlah pemangkasan yang tidak perlu [9]. Oleh karena itu, algoritma *decision tree* perlu dikombinasikan dengan suatu metode yang dapat meningkatkan performa klasifikasi yang lebih baik. Penelitian [10] mengamati bahwa *adaboost* (*Adaptive Boosting*) merupakan salah satu algoritma yang mampu memperbaiki kinerja pengklasifikasi. Algoritma *boosting* adalah algoritma yang memberikan bobot yang berbeda pada distribusi *training data* di setiap iterasi. Setiap iterasi *boosting* menambah bobot pada ragam klasifikasi yang salah dan menurunkan bobot pada ragam klasifikasi yang tepat sehingga merubah distribusi *data training* secara efektif [11]. Menurut penelitian [12], metode *adaBoost* pada konteks *selective costing ensemble* mampu menghadirkan solusi yang lebih efektif. Kemudian, penelitian [13] menerapkan *decision tree* dengan tingkat akurasi dan AUC (*area under cover*) sebesar 86,59% dan 0,957, yang kemudian dioptimalkan menggunakan a*daBoost* sehingga meningkatkan nilai akurasi dan AUC menjadi 92,24% dan 0,982. Penelitian [14] juga menemukan temuan serupa bahwa metode *boosting* dapat meningkatkan performa klasifikasi *decision tree*. Pada penelitian ini, peneliti akan menerapkan metode *adaBoost* pada

Received: 21 February 2022; Revised: 22 March 2022; Accepted: 13 April 2022.

[1]Z. K. S. Domas, Politeknik Keuangan Negara STAN (email: 1401190046.zicoksd@gmail.com)
[2]R. Rakhmadi, Jurusan Hubungan Internasional Universitas Lampung, Indonesia (e-mail: roby.rakhmadi007@fisip.unila.ac.id)

dataset kekurangtransparanan pengungkapan informasi anti-korupsi sebagaimana *dataset* yang telah digunakan dalam penelitian. Penelitian [15] juga menilai bahwa transparansi anti-korupsi pada korporasi Indonesia merupakan hal yang juga harus dibenahi sebagai bentuk dukungan terhadap budaya anti-korupsi.

Penelitian ini disusun menjadi rangkaian bab yang utuh. Bab I menjelaskan pendahuluan dan latar belakang atas urgensi tema peningkatan performa prediksi suatu klasifikasi. Kemudian, bab II menguraikan telaah literatur dari penelitian terdahulu yang relevan, bab III menjelaskan informasi atas *dataset* yang digunakan beserta metodologi penelitian, dan bab IV memaparkan hasil olah data bersamaan dengan inti pembahasan. Selanjutnya, bab V menyatakan kesimpulan, keterbatasan, hingga saran yang dihasilkan dari penelitian ini.

II. KAJIAN LITERATUR

Data mining adalah proses analisis data untuk menggali informasi yang tersembunyi di dalam suatu *database* yang ukurannya sangat besar dengan mengkombinasikan ilmu statistik dan *artificial intelligence*, sehingga ditemukan suatu pola yang sebelumnya belum diketahui agar lebih mudah dipahami [15]. Hal ini bermanfaat untuk kepentingan pengambilan keputusan di masa mendatang karena temuan suatu pola atau hubungan yang sebelumnya kurang disadari atau bahkan tidak terduga.

Penelitian [17] menerapkan algoritma *decision tree* untuk melakukan prediksi atas klasifikasi kelulusan tepat waktu dengan memanfaatkan *dataset* akademik mahasiswa tahun 2012-2014 yang diperoleh dari Pusat Informasi dan Data (PUSTIPANDA) UIN Syarif Hidayatullah Jakarta sebanyak 754 *records*. Penelitian [18] juga menerapkan algoritma klasifikasi untuk melakukan prediksi atas klasifikasi kelayakan kredit rumah dengan memanfaatkan *dataset* seluruh konsumen pembeli properti di PT. Pramtra Perumahan *Queen Residence* di Karawang dari bulan Maret 2019 hingga Februari 2020 sebanyak 50 data transaksi. Kemudian penelitian [19] menerapkan enam algoritma klasifikasi, salah satunya adalah *decision tree*, untuk melakukan prediksi atas klasifikasi pendeteksian kecurangan kartu kredit dengan memanfaatkan *dataset* pemilik kartu kredit di eropa pada periode September 2013 sebanyak 284.807 *record*s.

Penelitian [20] membandingkan teknik *over-sampling*, *under-sampling*, dan *synthetic minority over-sampling* (SMOTE) untuk meningkatkan akurasi prediksi pada kelas minoritas dengan menerapkan empat metode klasifikasi yang populer, yaitu LR, DT, NN, serta SVM. Dengan proses validasi *10-fold cross validation,* hasil penelitian menunjukan bahwa SVM berbasis SMOTE mencapai kinerja terbaik dengan tingkat akurasi sebesar 90,24%. Selanjutnya, penelitian [1] mengamati bahwa algoritma *decision tree* menghasilkan tingkat akurasi dan AUC untuk prediksi kelulusan mahasiswa sebesar 87,18% dan 0,864, yang kemudian dioptimalkan menggunakan *adaBoost* sehingga tingkat akurasi dan AUC mengalami peningkatan menjadi 90,45% dan 0,951. Lebih lanjut penelitian [21], [22], [23], serta penelitian [24] juga menemukan bahwa algoritma klasifikasi dengan menggunakan metode *adaBoost* terbukti mengalami peningkatan atas performa klasifikasi. Temuan ini menjadi dasar argumen peneliti bahwa *adaBoost* terbukti memiliki peran alternatif dalam menyelesaikan masalah ketidakseimbangan kelas karena mampu meningkatkan akurasi atas suatu fungsi klasifikasi, termasuk pula ketika diuji pada *dataset* yang mengalami kendala *class imbalance*.

Penelitian [25] mengamati bahwa algoritma *decision tree* dengan teknik *adaBoost* pada *dataset* SMA Islam Sultan Fatah Wedung Demak sebanyak 230 sampel menghasilkan tingkat akurasi untuk prediksi kelulusan siswa IPA sebesar 99,57% +-1,3% sehingga tingkat akurasi *adaBoost* diamati lebih baik dibandingkan algoritma *decision tree*. Lalu, [14] mengamati bahwa algoritma *decision tree* pada *dataset* German Credit Card menghasilkan tingkat akurasi untuk prediksi risiko kredit sebesar 70,5%, yang kemudian dioptimalkan menggunakan *adaBoost* sehingga tingkat akurasi mengalami peningkatan menjadi 74,2%. Penelitian [13] juga mengamati bahwa algoritma *decision tree* menghasilkan tingkat akurasi dan AUC untuk prediksi penyakit jantung sebesar 86,59% dan 0,957, yang kemudian dioptimalkan menggunakan *adaBoost* sehingga tingkat akurasi dan AUC mengalami peningkatan menjadi 92,24% dan 0,982.

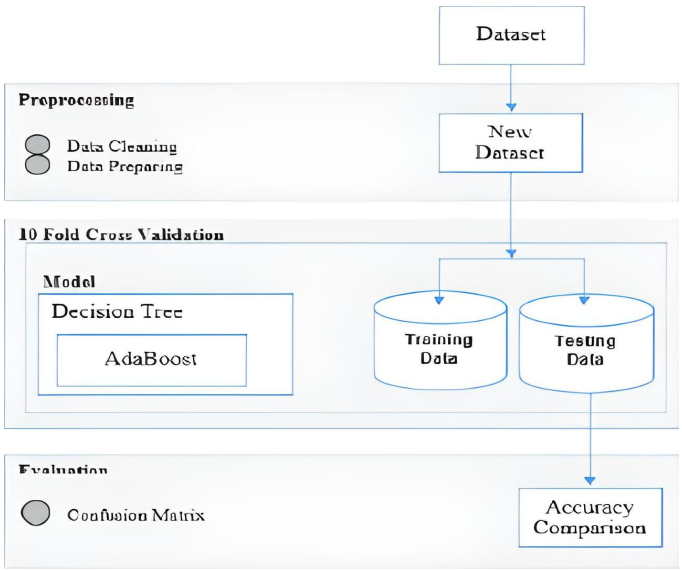
Pada penelitian ini, peneliti menerapkan algoritma klasifikasi *decision tree* yang dikombinasikan dengan metode *adaBoost* untuk dibandingkan dengan algoritma *decision tree* tanpa metode *adaBoost*. *Dataset* yang menjadi basis penelitian adalah *dataset* kekurangtransparanan pengungkapan informasi anti-korupsi pada sektor swasta yang digunakan dalam penelitian [15] meskipun *dataset* ini tidak memenuhi kendala *class imbalance*. Kemudian, peneliti akan membandingkan tingkat performa pada kedua algoritma ini, baik dari aspek akurasi, AUC-optimis, hingga aspek AUC (*area under curve*).

III. METODOLOGI PENELITIAN

A. Kerangka Pemikiran

Metode dalam penelitian ini dirancang untuk meningkatkan kinerja *decision tree* dengan teknik *adaBoost* pada klasifikasi *dataset* kekurangtransparanan pengungkapan informasi anti-korupsi pada sektor swasta yang telah digunakan dalam penelitian [15]. Kemudian, dilakukan validasi menggunakan *10-fold cross validation* dan dilakukan analisa uji beda menggunakan *t-Test*. Keseluruhan model kerangka pemikiran ditunjukkan sebagaimana Gambar 1. Mengacu Gambar 1, data yang sudah melalui proses pembersihan telah siap dipilah menjadi sebuah dataset untuk keperluan *training* dan *testing*. Setelah itu diolah menggunakan dua jenis *classifier*, yaitu: algoritma *decision tree* tanpa *adaBoost* dan algoritma *decision tree* berbasis *adaBoost*.

Applied Information System and Management (AISM)
Volume 5, (2) 2022, p. 75-82
P-ISSN: 2621-2536; E-ISSN: 2621-2544; DOI: 10.15408/aism.v5i2.24887



Gambar 1. Kerangka Pemikiran [1]

B. Decision Tree Tanpa AdaBoost

Algortima *decision tree* dalam penelitian ini merujuk kepada metode C4.5 yang merupakan pengembangan dari algortima *decision tree* ID3, dimana pengembangan dilakukan dengan tujuan mengatasi *missing* data, data kontinu, maupun *pruning* [15]. Corak tahapan yang paling diperlukan pada algoritma *decision tree* C4.5 adalah dengan menentukan akar dari pohon [15]. Pilih akar dari atribut dengan menghitung nilai gain dari semua atribut, dimana yang menjadi akar pertama adalah nilai gain yang tertinggi. Sebelum menentukan nilai gain, hitung nilai *entropy* terlebih dahulu dengan menggunakan persamaan:

$$Entropy\ (s) = -\sum^{n}{}_{i=1}\ (-pi*Log^2\ pi) \quad (1)$$

dengan *S* menyatakan himpunan kasus; *n* adalah jumlah partisi *S*; sedangkan *pi* adalah proporsi dari *Si* terhadap *S*. Selanjutnya hitung nilai gain dengan menggunakan persamaan:

$$Gain\ (S,A) = Entropy\ (S) - \sum^{n}{}_{i=1}\ (|S_i|/|S|) * Entropy(S_i) \quad (2)$$

dengan *S* adalah himpunan kasus; *A* merupakan atribut; *n* adalah jumlah partisi atribut *A*; |Si| merupakan jumlah kasus pada partisi ke-*i*; |S| merupakan jumlah kasus dalam *S*. Setelah itu, ulangilah langkah tersebut hingga semua *record* terpartisi secara sempurna. Proses partisi dari pohon keputusan akan berhenti pada saat semua *record* dalam simpul *n* mendapat kelas yang sama, tidak ada atribut pada *record* yang dipartisi lagi, dan tidak ada *record* didalam cabang yang kosong. Selanjutnya, algoritma *decision tree* tanpa *adaBoost* ini akan dibandingkan performanya dengan algoritma *decision tree* berbasis *adaBoost*.

C. Decision Tree Berbasis AdaBoost

Teknik *adaBoost* menggunakan dua parameter yaitu: pengaturan *maximum depth* dan *confidence* dengan perlakuan sebagaimana Tabel 1.

Tabel 1.
Perlakuan Parameter pada *Decision Tree* Berbasis *Adaboost* [16]

Maximum Depth	*Confidence*	
	0,05	0,15
10	Result	Result
30	Result	Result
50	Result	Result

Pada dasarnya, metode *boosting* dilakukan dengan cara mengatur kombinasi dari suatu model di mana hasil klasifikasi yang terpilih adalah model yang memiliki nilai bobot terbesar. Perubahan pada parameter tertentu, dalam hal ini *maximum depth* dan *confidence,* dilakukan untuk mengukur nilai akurasi tertinggi yang dihasilkan dari setiap skema pengamatan atas penerapan *decision tree* berbasis *adaBoost*. Berdasarkan Tabel 1, penelitian ini akan menerapkan enam skenario pengamatan atas penerapan algoritma *decision tree* yang berbasis *adaBoost,* mulai dari skenario pertama yang menerapkan *maximum depth* dan *confidence* 10 dan 0,05, skenario kedua yang menerapkan *maximum depth* dan *confidence* 10 dan 0,15, skenario ketiga yang menerapkan *maximum depth* dan *confidence* 30 dan 0,05, skenario keempat yang menerapkan *maximum depth* dan *confidence* 30 dan 0,15, skenario kelima yang menerapkan *maximum depth* dan *confidence* 50 dan 0,05, hingga skenario keenam atau terakhir yang menerapkan *maximum depth* dan *confidence* 50 dan 0,15.

Kemudian, hasil klasifikasi atau prediksi yang terpilih adalah model yang memiliki nilai bobot paling besar. Jadi, setiap model yang dibangkitkan memiliki atribut berupa nilai bobot. Penelitian [1] menjelaskan teknik pembobotan pada algoritma *adaBoost* sebagai berikut:

Input:
Dataset $(D_t) = (x_1, y_1), \ldots, (x_n,y_n)$;
Weak Lern (L)
T menyatakan jumlah iterasi
Proses:
Inisialisasi nilai bobot
$D_1(i)$ = untuk $i = 1, \ldots, n$
for $t = 1, \ldots, T$;
Pengujian terhadap distribusi D_t
$h_t = L(D_t)$
hitung *error* dari $\epsilon_t = Pr_{x\sim D_t,y}\ I[h_t(x) \neq y]$
if $\epsilon_t > 0.5$ *then break*
menentukan bobot dari h_t
$\alpha_t = \frac{1}{2} \ln \{(1-\varepsilon_t)/\ \varepsilon_t\}$;
$D_{t+1}(i) = (D_t(i)/z_t)\, x\ \{\exp(-\alpha_t)$ if $h_t(x_i) = y_i\}$
$\{\exp(\alpha_t)$ if $h_t(x_i) \neq y_i\}$

Update distribusi, dimana Z_t sebuah faktor normalisasi

yang mengaktifkan D_{t+1} menjadi distribusi

$\{D_t(i)\exp(-\alpha_t y_i h_t(x_i))\}/Z_t$

end for

Output:

$H(x) = sign\ \{ \sum^{T}_{t=1} \alpha_t h_t(x) \}$

D. Ten-Folds Cross Validation

Cross validation merupakan metode yang membagi *dataset* menjadi dua bagian, dimana satu bagian berperan sebagai *data training* sementara bagian lainnya berperan sebagai *data testing*. Beberapa penelitian membagi data menjadi 10 bagian, 90% digunakan sebagai *data training* dan 10% lainnya digunakan sebagai *data testing*. Proses ini dilakukan berulang hingga 10 kali sehingga dikenal juga dengan istilah *ten-folds cross validation*. Teknik ini banyak digunakan oleh peneliti karena terbukti menghasilkan performa algoritma yang lebih stabil [15].

E. Evaluasi Model

Confusion matrix merupakan *data set* yang memiliki dua kelas, yaitu positif dan negative yang terdiri dari *True Positive* (TP), *False Positive* (FP), *True Negative* (TN) dan *False Negative* (FN) [19].

Tingkat akurasi = (TP+TN)/ (TP+FP+TN+FN)

Selanjutnya, penelitian [26] menjelaskan bahwa performa *Area Under Curve* (AUC) dapat diklasifikasikan menjadi lima kelompok yaitu :

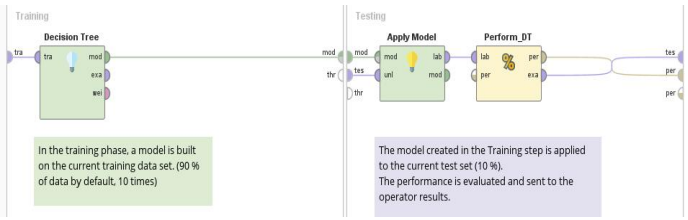
1. 0,90 – 1,00 = *Exellent Clasification*
2. 0,80 – 0,90 = *Good Clasification*
3. 0,70 – 0,80 = *Fair Clasification*
4. 0,60 – 0,70 = *Poor Clasification*
5. 0,50 – 0,60 = *Failure*

IV. HASIL DAN PEMBAHASAN

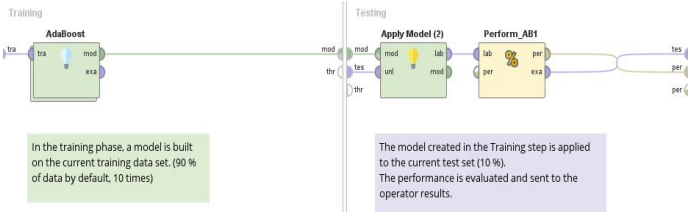
Eksperimen dilakukan dengan menggunakan aplikasi *Rapidmine*r 9.9. Luas pengungkapan informasi anti-korupsi pada sektor swasta yang digunakan sebagai *dataset* yaitu: pengungkapan laporan tahunan perusahaan oleh 141 sampel perusahaan di sektor konstruksi infrastruktur, pertambangan perminyakan, dan perbankan, sepanjang periode 2019. Dari jumlah tersebut, perusahaan yang tergolong mengungkapkan lebih banyak informasi anti-korupsi (terklasifikasi sebagai transparan) sebanyak 67 records (47,52%) dan perusahaan yang tergolong mengungkapkan lebih minim informasi anti-korupsi (terklasifikasi sebagai kurang transparan) sebanyak 74 *records* (52,48%). Kemudian, hasil pemetaan atas atribut yang telah diinput disajikan sebagaimana Tabel 2 dimana penerapan sub-proses seluruh skenario eksperimen disajikan sebagaimana Gambar 2, 3, dan 4.

Tabel 2.
Pemetaan Jenis Data

Role	Nama	Jenis data
Label atau target prediksi	KTPA (kekurang transparanan pengungkapan informasi anti-korupsi)	Binominal (menggunakan proksi yang digunakan oleh penelitian [15]) 0: laporan tahunannya lebih luas dalam mengungkapkan informasi anti-korupsi. 1: laporan tahunannya lebih minim dalam mengungkapkan informasi anti-korupsi.
Reguler	Tekanan	Binominal 0: mengalami profit. 1: mengalami kerugian.
Reguler	Kesempatan	Integer Diukur dengan proksi jumlah anggota komisaris independen (makin banyak anggota komisaris independen berarti makin kecil tingkat kesempatan, dan sebaliknya).
Reguler	Rasionalitas	Binominal 0: sektor usaha perbankan. 1: sektor usaha non-perbankan.
Reguler	Kompetensi	Binominal 0: tidak terjadi pergantian direktur pada tahun berikutnya. 1: terjadi pergantian setidaknya salah seorang dewan direktur pada tahun berikutnya.
Reguler	Arogansi	Binominal 0: pemerintah bertindak sebagai salah satu pemilik saham hak suara. 1: pemerintah tidak bertindak sebagai pemilik saham hak suara (swasta murni).
Reguler	Kolusi	Binominal 0: mengungkapkan informasi transaksi penjualan ke pihak istimewa. 1: tidak mengungkapkan informasi transaksi penjualan ke pihak istimewa.

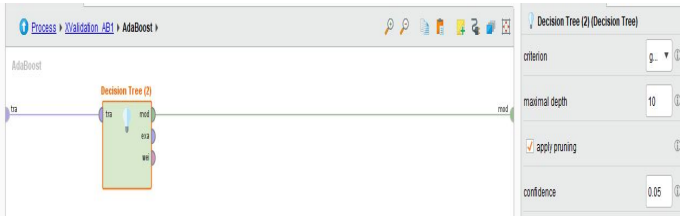


Gambar 2. Sub-proses Penerapan Algoritma *Decision Tree* Tanpa *Adaboost*



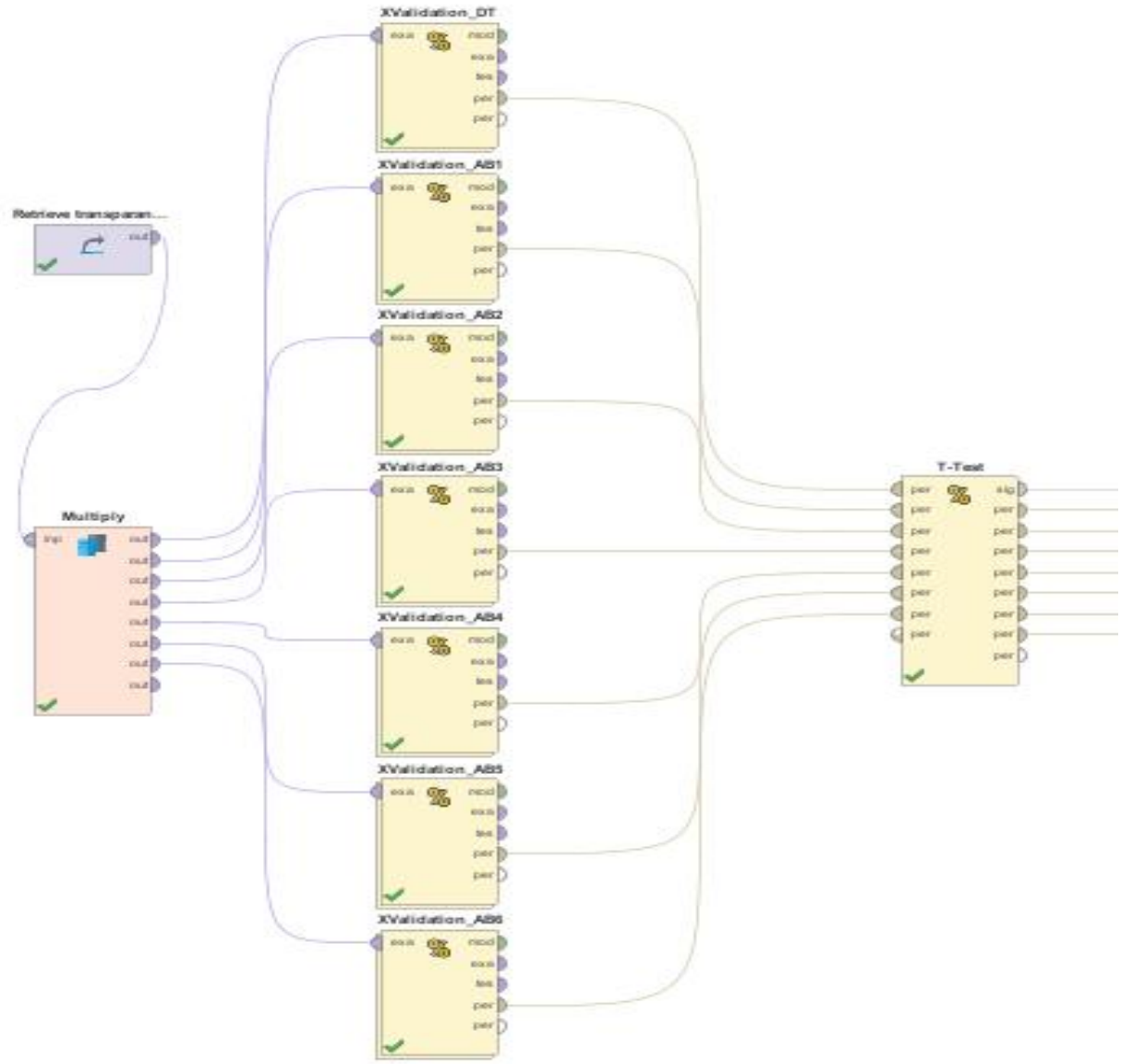
Gambar 3. Sub-Proses Penerapan Algoritma *Decision Tree* Berbasis *Adaboost* Skenario Uji 1

Applied Information System and Management (AISM)
Volume 5, (2) 2022, p. 75-82
P-ISSN: 2621-2536; E-ISSN: 2621-2544; DOI: 10.15408/aism.v5i2.24887



Gambar 4. Detail Sub-Proses Penerapan Algoritma *Decision Tree* Berbasis *Adaboost* Skenario Uji 1

Penelitian ini akan terus mengulangi proses dan sub-proses sebagaimana disajikan pada Gambar 3 dan 4 dengan cara mengatur parameter *maximal depth* dan *confidence* sesuai Tabel 1 hingga skenario pengujian keenam tuntas diselesaikan sebagaimana Gambar 5.



Gambar 5. Perancangan Evaluasi atas Keseluruhan Skenario Eksperimen

Untuk memvalidasi model algoritma, maka diterapkanlah *ten-fold stratified cross-validation*, yaitu pengulangan sepuluh kali pada seluruh *dataset* dimana setiap pengulangannya menggunakan data acak yang berbeda [27].

Setelah *ten-fold stratified cross-validation* selesai dilakukan, hasil dari sepuluh lipatan uji untuk 90% *data training* digabungkan, kemudian pola atas hasil *data training* ini secara otomatis diterapkan pada 10% *data testing* sehingga hasil evaluasi performa atas tujuh skenario algoritma *decision tree* berbasis *adaBoost* dapat terukur secara objektif sebagaimana disajikan Tabel 3, 4, dan 5.

Tabel 3.
Rekapitulasi Evaluasi Perbandingan Tingkat Akurasi

D-tree	Skema *decision tree* berbasis *adaBoost*							Pengaruh *adaBoost* pada akurasi
	1	2	3	4	5	6	Rerata	
69,5%	74,47%	74,47%	66,67%	70,92%	68,09%	72,34%	**71,16%**	**Meningkat**

Berdasarkan Tabel 3, rerata algoritma *decision tree* berbasis *adaBoost* terbukti meningkatkan tingkat akurasi yang lebih baik dibandingkan algoritma *decision tree* versi biasa.

Tabel 4.
Rekapitulasi Evaluasi Perbandingan Tingkat AUC-Optimis

D-tree	Skema *decision tree* berbasis *adaBoost*							Pengaruh *adaBoost* pada AUC-optimis
	1	2	3	4	5	6	Rerata	
0,826	0,901	0,915	0,868	0,864	0,884	0,911	**0,8905**	**Meningkat**

Berdasarkan Tabel 4, rerata algoritma *decision tree* berbasis *adaBoost* terbukti meningkatkan AUC-optimis yang lebih baik dibandingkan algoritma *decision tree* versi biasa.

Berdasarkan Tabel 5, rerata algoritma *decision tree* berbasis *adaBoost* belum terbukti mampu meningkatkan AUC yang lebih baik dibandingkan algoritma *decision tree* versi biasa karena AUC pada algoritma *decision tree* tanpa *adaBoost* justru lebih tinggi dibandingkan AUC pada algoritma *decision tree* berbasis *adaBoost*. Jika yang dibandingkan antara *decision tree* versi biasa dengan algoritma *decision tree* berbasis *adaBoost* skenario uji 1, uji 2, dan uji 6, maka dapat disimpulkan secara mutlak bahwa algoritma *adaBoost* mampu konsisten meningkatkan tingkat performa, baik dari aspek akurasi, AUC-optimis, maupun AUC. *Decision tree* versi biasa secara statistik belum dapat disimpulkan berada pada klaster kelayakan yang berbeda dengan algoritma *decision tree* yang berbasis *adaBoost* meskipun *adaBoost* secara ringkas terlihat memiliki tingkat performa yang lebih baik berdasarkan keenam skenario pengujian ini. Untuk itulah perlunya dilakukan uji beda *T-test* agar dapat diketahui tingkat perbedaan secara statistik sebagaimana disajikan Tabel 6.

Tabel 6.
Hasil uji *T-test*

Probabilities for random values with the same result						
-----	0,362	0,299	0,639	0,821	0,788	0,579
-----	-----	0,982	0,205	0,49	0,279	0,627
-----	-----	-----	0,164	0,425	0,232	0,548
-----	-----	-----	-----	0,503	0,851	0,324
-----	-----	-----	-----	-----	0,636	0,762
-----	-----	-----	-----	-----	-----	0,435
-----	-----	-----	-----	-----	-----	-----

Tabel 6 menunjukkan di antara ketujuh skenario pengujian secara statistik tidak memiliki perbedaan yang signifikan karena tidak ada nilai *alpha* kurang dari 0,05 sehingga dapat disimpulkan bahwa antara *decision tree* versi biasa dan *decision tree* berbasis *adaBoost* berada pada klaster kelayakan yang sepadan meskipun secara ringkas terjadi perbedaan dalam hal akurasi, AUC-optimis, dan AUC *(area under curve)*. Dengan kata lain, algoritma *decision tree* berbasis *adaBoost* mampu meningkatkan tingkat performa *decision tree* versi biasa dalam perspektif keakuratan prediksi klasifikasi

Tabel 5
Rekapitulasi Evaluasi Perbandingan Tingkat AUC

D-tree	*Decision tree* Berbasis *adaBoost*							Pengaruh *adaBoost* (rerata 6 skema) pada AUC	Rerata *adaBoost* skema 1, 2, dan 6	Pengaruh *adaBoost* (rerata skema 1, 2, dan 6) pada AUC
	Skema 1	Skema 2	Skema 3	Skema 4	Skema 5	Skema 6	Rerata 6 skema			
0,756	0,769	0,778	0,715	0,727	0,716	0,758	**0,744**	**Menurun**	**0,768**	**Meningkat**

perusahaan yang bersikap kurang transparan atau bersikap transparan terkait konteks pengungkapan informasi anti-korupsi, namun secara statistik algoritma *adaBoost* belum memberikan perbedaan yang signifikan dengan *decision tree* versi biasa. Hal ini dapat dipahami karena *dataset* pada penelitian ini tidak berada pada kondisi *class imbalance* sementara algoritma *adaBoost* lazim diterapkan untuk mengatasi kendala *class imbalance* pada suatu *dataset*.

Kemudian, hasil evaluasi performa atas tingkat AUC untuk rerata algoritma *adaBoost* sebesar 0,744 sehingga dapat disimpulkan ada pada kategori *classifier* yang cukup baik [30]. Adapun untuk hasil yang lebih objektif, jika algoritma *decision tree* berbasis *adaBoost* hanya menggunakan skenario uji 1, uji 2, dan uji 6, maka dapat disimpulkan bahwa performa AUC *adaBoost* meningkat menjadi 0,768 meskipun masih belum beranjak dari kategori *classifier* yang cukup baik [26].

V. KESIMPULAN

Berdasarkan hasil pengujian, penelitian ini menghasilkan kesimpulan bahwa algoritma *decision tree* berbasis *adaBoost* mampu meningkatkan tingkat performa *decision tree* versi biasa dalam perspektif keakuratan prediksi atas klasifikasi perusahaan yang bersikap kurang transparan atau yang bersikap transparan terkait konteks pengungkapan informasi anti-korupsi. Namun berdasarkan perspektif statistik, performa algoritma *adaBoost* tidak memiliki perbedaan yang signifikan dengan *decision tree* versi biasa sehingga dapat disimpulkan bahwa kedua algoritma ini ada pada klaster yang sepadan. Dengan kata lain, algoritma *adaBoost* merupakan upaya alternatif yang layak diterapkan jika algoritma *decision tree* dirasa kurang memuaskan terutama jika menemui kondisi distribusi *class imbalance* pada *dataset*.

Kemudian, peneliti juga menyatakan beberapa keterbatasan dalam penelitian. Pertama, sampel penelitian tidak terlalu besar karena terbatas pada tiga sektor industri dan hanya menggunakan satu tahun pengamatan. Kedua, penelitian ini sama sekali tidak membahas keterjadian praktik korupsi di sektor pihak perusahaan swasta, namun hanya berfokus pada transparansi pengungkapan informasi anti-korupsi karena ketidaktersediaan data dalam membuktikan praktik korupsi yang telah dilakukan oleh oknum perusahaan yang melantai di Bursa Efek Indonesia.

Penulis juga merekomendasikan beberapa saran untuk penelitian selanjutnya terkait topik antikorupsi dalam konteks *data mining*. Pertama, penelitian selanjutnya dapat menambah rentang periode yang rentan terpapar risiko korupsi karena berdekatan dengan agenda kontestasi pemilihan umum, misalnya periode pengamatan 2017-2019, periode pengamatan 2022-2024 di kemudian hari agar volume *dataset* menjadi jauh lebih besar. Kedua, penelitian selanjutnya dapat menambah jumlah sampel pengamatan dengan cara memperbanyak jumlah sektor yang tidak hanya terbatas pada tiga sektor.

Terakhir, penulis mencoba memberi masukan kepada pihak otoritas yang berwenang, seperti KPK, OJK, BI, Badan Pengawas Pasar Modal, Kepolisian, serta segenap regulator lainnya, agar lebih mengeksploitasi teknik prediksi klasifikasi status transparansi rutin setiap tahun menggunakan pendekatan *data mining* dengan menyesuaikan *big data* internal masing-masing instansi untuk memetakan sektor dan faktor kunci apa saja yang harus menjadi bahan evaluasi untuk dilakukan sosialisasi dalam rangka upaya mewujudkan budaya anti-korupsi bagi korporasi swasta.

REFERENSI

[1] A. Bisri and R. S. Wahono, "Penerapan AdaBoost untuk Penyelesaian Ketidakseimbangan Kelas pada Penentuan Kelulusan Mahasiswa dengan Metode Decision Tree," *J. Intell. Syst.*, vol. 1, no. 1, pp. 27-32, Feb. 2015. [Online]. Available: http://www.journal.ilmukomputer.org/index.php?journal=jis&page=article&op=view&path%5B%5D=29

[2] D. A. Cieslak, T. R. Hoens, N. V. Chawla, and W. P. Kegelmeyer, "Hellinger distance decision trees are robust and skew-insensitive," *Data Min Knowl Disc*, vol. 24, pp. 136–158, 2011, doi: 10.1007/s10618-011-0222-1.

[3] Y. Sun, M. S. Kamel, A. K. C. Wong, and Y. Wang, "Cost-sensitive boosting for classification of imbalanced data," *ScienceDirect*, vol. 40, no. 12, pp. 3358-3378, Dec. 2007.

[4] F. D. Astuti and F. N. Lenti, "Implementasi SMOTE untuk mengatasi Imbalance Class pada Klasifikasi Car Evolution menggunakan K-NN," *JUPITER (Jurnal Penelit. Ilmu dan Teknol. Komputer)*, vol. 13, pp. 89–98, 2021.

[5] A. N. Kasanah, Muladi, and U. Pujianto, "Penerapan Teknik SMOTE untuk Mengatasi Imbalance Class dalam Klasifikasi Objektivitas Berita Online Menggunakan Algoritma KNN," *RESTI*, vol. 3, no. 2, pp. 196–201, 2019.

[6] R. Siringoringo, "Klasifikasi Data Tidak Seimbang Menggunakan Algoritma SMOTE dan k-Nearest Neighbor," *J. ISD*, vol. 3, no. 1, pp. 44-49, Jan-Jun 2018.

[7] Y. Pristyanto, "Penerapan Metode Ensemble Untuk Meningkatkan Kinerja Algoritme Klasifikasi Pada Imbalanced Dataset," *J. Teknoinfo*, vol. 13, no. 1, pp. 11-16, 2019, doi: 10.33365/jti.v13i1.184.

[8] J. Van Hulse and T. Khoshgoftaar, "Knowledge discovery from imbalanced and noisy data," *Data Knowl. Eng.*, vol. 68, no. 12, pp. 1513–1542, Dec. 2009, doi: 10.1016/J.DATAK.2009.08.005.

[9] C. Drummond and R. C. Holte, "C4.5, Class Imbalance, and Cost Sensitivity: Why Under-Sampling beats Over-Sampling.," *Phys. Rev. Lett.*, vol. 91, no. 3, 2003.

[10] A. Saifudin and R. S. Wahono, "Pendekatan Level Data untuk Menangani Ketidakseimbangan Kelas pada Prediksi Cacat Software," *J. Softw. Eng.*, vol. 1, no. 2, 2015.

[11] S. Kotsiantis, D. Kanellopoulos, and P. Pintelas, "Handling imbalanced datasets: A review," *GESTS Int. Trans. Comput. Sci. Eng.*, vol. 30, 2006.

[12] S. Kotsiantis and P. E. Pintelas, "Selective costing ensemble for handling imbalanced data sets," *Int. J. Hybrid Intell. Syst.*, vol. 6, no. 3, pp. 123–133, 2009, doi: 10.3233/HIS-2009-0084.

[13] A. Rohman, V. Suhartono, and C. Supriyanto, "Penerapan Agoritma C4.5 Berbasis AdaBoost Untuk Prediksi Penyakit Jantung," *J. Teknol. Inf.*, vol. 13, no. 1, pp. 13-19, Jan. 2017.

[14] A. Nurzahputra and M. A. Muslim, "Peningkatan Akurasi Pada Algoritma C4.5 Menggunakan AdaBoost Untuk Meminimalkan Resiko Kredit," in *Proc. SNATIF ke-4*, 2017. [Online]. https://media.neliti.com/media/publications/173704-ID-none.pdf

[15] Transparency International Indonesia, "Transparency in Corporate Reporting." Ti.or.id. https://ti.or.id/transparency-in-corporate-reporting/ (accessed Apr. 17, 2022).

[16] E. Pradana, "Analisis Penerapan Adaptive Boosting (AdaBoost) Dalam Meningkatkan Performasi Algoritma C4.5," Skripsi, Prodi Teknik Informatika, STT Pelita Bangsa, Bekasi, 2018.

[17] C. Wirawan, "Teknik Data Mining Menggunakan Algoritma Decision Tree C4.5 untuk Memprediksi Tingkat Kelulusan Tepat Waktu," *Appl. Inf. Syst. Manag.*, vol. 3, no. 1, pp. 47–52, 2020, doi: 10.15408/aism.v3i1.13033.

[18] S. Sumanto, L. S. Marita, L. Mazia, and T. W. Ratnasari, "Analisis Kelayakan Kredit Rumah Menggunakan Metode Naïve Bayes untuk Mengurangi Kredit Macet," *Appl. Inf. Syst. Manag.*, vol. 4, no. 1, pp. 17–22, 2021, doi: 10.15408/aism.v4i1.20274.

[19] K. B. Soni; M. Chopade; and R. Vaghela, "Credit Card Fraud Detection Using Machine Learning Approach," *Appl. Inf. Syst. Manage.*, vol. 4, no. 2, pp. 71–76, 2021.

[20] D. Thammasiri, D. Delen, P. Meesad, and N. Kasap, "A critical assessment of imbalanced class distribution problem: The case of predicting freshmen student attrition," *Expert Syst. Appl.*, vol. 41, no. 2, pp. 321–330, 2014.

[21] Y. H. Agustin, Kusrini, and E. T. Luthfi, "Klasifikasi Penerimaan Mahasiswa Baru Menggunakan Algortima C4.5 dan AdabBoost (Studi Kasus : STMIK XYZ)," *CSRID (Computer Sci. Res. Its Dev.) Journal*, vol. 9, no. 1, pp. 1-11, Feb. 2017, doi: 10.22303/csrid.9.1.2017.1-11.

[22] L. Eka and M. Much Aziz, "Penerapan Adaboost untuk Klasifikasi Support Vector Machine Guna Meningkatkan Akurasi pada Diagnosa Chronic Kidney Disease," presented at the Seminar Nasional Teknologi dan Informatika, Indonesia, Juli 2017.

[23] S. Mulyati, Yulianti, and A. Saifudin, "Ketidakseimbangan kelas berbasis naïve bayes pada prediksi," *J. Inform. Univ. PAMULANG*, vol. 2, no. 4, 2017.

[24] P. A. Jusia, "Analisis Komparasi Pemodelan Algoritma Decision Tree Menggunakan Metode Particle Swarm Optimization Dan Metode Adaboost Untuk Prediksi Awal Penyakit Jantung," in *Seminar Nasional Sistem Informasi (SENASIF)*, pp. 1048–1056, 2018.

[25] M. S. Mauludin, and L. Hermawanti, "Merger C4.5 Algorithm and Adaboost for Determining the Department Ipa Students Graduation in Sma Islam Sultan Fatah Wedung Demak," in *Proc. the 2nd International Seminar and Conference on Global Issues (ISCoGI),* European and Asian in the Age of Globalization: Cooperation and Challenge, Nov. 25-26, 2016. [Online]. Available: https://publikasiilmiah.unwahas.ac.id/index.php/ISC/article/view/1664/1739

[26] F. Gorunescu, *Data Mining: Concepts, Models, and Techniques*, 12th ed. Springer, 2011.

[27] J. Perols, "Financial Statement Fraud Detection: An Analysis of Statistical and Machine Learning Algorithms," *Audit. A J. Pract. Theory*, vol. 30, no. 2, pp. 19-50, 2011. [Online]. Available: https://ssrn.com/abstract=2206572