Analisis Perbandingan Algoritma *Decision Tree* untuk Prediksi Karyawan dengan Potensi Atrisi di PT. XYZ

Ilyas Jayanto[1], Benisius*[2]
[1,2] Program Studi Informatika, Fakultas Teknik dan Ilmu Komputer, Universitas Kristen Krida Wacana
e-mail: [1]ilyas.2017tin020@civitas.ukrida.ac.id, [2]ehba@ukrida.ac.id
*Correspondence author email: *ehba@ukrida.ac.id*

Abstrak

Sumber daya manusia (SDM) merupakan salah satu aset terpenting bagi perusahaan. Setiap perusahaan tentunya menginginkan SDM yang handal agar dapat bertumbuh untuk mencapai tujuannya. Untuk itu dibutuhkan sebuah strategi dalam mengatur SDM perusahaan yaitu manajemen SDM. Salah satu tujuan dari manajemen SDM adalah mempertahankan karyawan yang berkompeten dalam pekerjaannya, akan tetapi perusahaan sering kali diperhadapkan pada permasalahan atrisi karyawan (kehilangan karyawan karena mengundurkan diri atau pensiun). Dampak dari atrisi karyawan antara lain dapat membuat pendapatan suatu perusahaan menjadi berkurang, dan bahkan memengaruhi produktivitas organisasi. Penelitian ini bermaksud membandingkan beberapa algoritma decision tree dalam memprediksi kasus atrisi karyawan di PT. XYZ. Kelima algoritma dimaksud adalah algoritma C4.5, CART, Random Forest, Gradient Boost dan Adaboost. Data yang digunakan adalah data primer (data perusahaan PT. XYZ) dan data sekunder (data website Kaggle). Hasil dari penelitian ini menyimpulkan bahwa algoritma C4.5 dan random forest memiliki hasil yang lebih baik dimana nilai akurasi yang diperoleh adalah 82.35%, presisi 86.76 persen, recall 83.35 persen, dan f1-score 81.57 persen.

Kata kunci*— Atrisi Karyawan, Klasifikasi, Machine Learning, Decision Tree*

1. PENDAHULUAN

Setiap perusahaan berusaha untuk dapat mencapai tujuannya. Untuk mencapai hal tersebut perusahaan haruslah mempunyai strategi agar dapat tetap kompetitif. Salah satu hal yang memengaruhi performa perusahaan adalah komponen sumber daya manusia (SDM) [1]. Setiap perusahaan tentunya menginginkan SDM yang handal sehingga manajemen SDM yang baik diperlukan. Kualitas dari kinerja SDM merupakan salah satu faktor yang mendukung peningkatan produktifitas dari suatu perusahaan [2]. Salah satu tujuan dari manajemen SDM adalah mempertahankan karyawan yang berkompeten dalam pekerjaannya, akan tetapi untuk melakukan manajemen SDM terdapat satu masalah yaitu atrisi karyawan [3].

Atrisi karyawan adalah Perusahaan kehilangan karyawannya melalui sejumlah keadaan, seperti pengunduran diri atau pensiun [4]. Penyebab atrisi dapat berupa sukarela atau tidak sukarela [3]. Dampak dari atrisi karyawan yaitu dapat membuat kinerja suatu perusahaan menjadi berkurang, dan mempengaruhi produktivitas organisasi karena kehilangan SDM terbaiknya. Hal ini diperparah dengan proses perekrutan karyawan baru yang membutuhkan pelatihan, dan pengembangan karena karyawan baru membutuhkan waktu untuk dapat mencapai tingkat keahlian yang sama dengan karyawan sebelumnya [5].

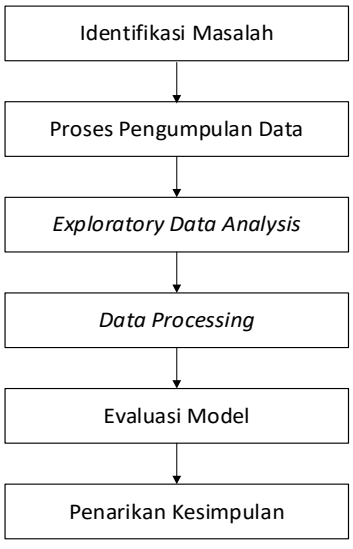
PT. XYZ adalah sebuah perusahaan yang bergerak di bidang teknologi. Perusahaan ini dulunya adalah *startup* dan baru resmi berubah menjadi PT. di tahun 2020. Sebuah *startup* tentunya perlu untuk terus berkembang dan memperbesar diri sehingga komponen SDM perlu mendapatkan perhatian khusus [6]. Atrisi rupanya menjadi salah satu tantangan terbesar perusahaan ini. Sebagai gambaran saat ini PT. XYZ mempunyai hanya 9 karyawan aktif dari sebelumnya 17 orang. Jumlah karyawan yang kian menurun menyebabkan beberapa karyawan harus menanggung beban lebih sehingga efektifitas kerja menjadi tidak terjaga. Berangkat dari permasalah tersebut, maka penelitian ini bermaksud mencari pendekatan terbaik yang dapat membantu PT. XYZ untuk memprediksi karyawan yang berpotensi atrisi. Dengan mengetahui

lebih dini maka diharapkan pihak manajemen dapat melakukan tindakan antisipatif yang dirasa perlu.

Salah satu cara melakukan prediksi adalah dengan melakukan pendekatan *machine learning*. *Machine Learning* merupakan cabang dari *Artificial Intelligence* yang memiliki fokus pada pengembangan sistem yang dapat belajar sendiri. Salah satu metode dalam *machine learning* adalah klasifikasi. Metode yang akan digunakan pada penelitian ini adalah *Decision Tree* bagian klasifikasi, dimana metode *Decision Tree* ini memiliki beberapa algoritma tersendiri di dalamnya. Algoritma yang akan digunakan pada penelitian ini adalah algoritma C4.5, CART, *Random Forest Classifier, Gradient Boosting Classifier* dan *Adaboost Classifier*. Masing masing algoritma mempunyai beberapa keunggulan tersendiri, yaitu Algoritma C4.5 dapat melakukan olah data numerik dan diskrit, menghasilkan aturan yang mudah dipahami dan performanya lebih cepat dari algoritma lain, alasan dibutuhkan kecepatan dalam algoritma adalah seiring bertambahnya data, waktu yang dibutuhkan untuk menjalankan fungsi di dalamnya juga akan bertambah maka itu dibutuhkan kecepatan dalam mengolah data tersebut [7]. Algoritma CART memiliki beberapa keunggulan dibandingkan algoritma lainnya, misalnya program CART dapat digunakan pada data dengan jumlah variabel yang banyak karena algoritma CART mengidentifikasi variabel yang paling signifikan dan menghilangkan variabel yang tidak signifikan [8]. Algoritma *Random Forest Classifier* bekerja dengan membuat sebanyak mungkin *tree* pada subset data, dengan cara ini algoritma tersebut dapat mengurangi masalah overfitting sehingga dapat meningkatkan akurasi hasil klasifikasi [9]. Algoritma *Gradient Boosting Classifier* memiliki keunggulan dalam kecepatan dan efisien dalam penggunaan memori [10]. Algoritma *Adaboost Classifier* menghitung bobot untuk data pelatihan. Klasifikasi yang berbeda dilatih secara berurutan selama pelatihan terhadap data. Setiap pengklasifikasi baru dilatih berdasarkan keefektifan pengklasifikasi terlatih yang ada sehingga dapat meningkatkan akurasi akhir untuk klasifikasi [11].

2. METODE PENELITIAN

Penelitian ini dilakukan dengan menggunakan metode *decision tree* bagian klasifikasi dan melalui beberapa tahapan seperti yang ditunjukkan **Gambar 1**.



Gambar 1. Tahapan penelitian

Langkah pertama adalah identifikasi masalah. Pada tahap ini dilakukan analisa terkait kondisi PT. XYZ yang mempunyai tingkat atrisi karyawan yang tinggi dan olehnya diperlukan sebuah pendekatan yang mampu memprediksi karyawan PT. XYZ yang berpotensi atrisi, sehingga dapat membantu pihak manajemen dapat mengantisipasi hal tersebut.

Langkah kedua adalah mengumpulkan data yang dibutuhkan untuk model pembelajaran mesin**.** Data diperoleh dari dua sumber yaitu data sekunder dan juga data primer, dengan data sekunder diperoleh dari *website* Kaggle yang dimana data tersebut juga digunakan oleh penelitian [6] dengan nama dataset "*HR data for analytics*". Data primer merupakan data perusahaan PT. XYZ. Yang dimana data tersebut didapatkan dari hasil wawancara ke karyawan aktif PT. XYZ. **Gambar 2** dan **Gambar 3** memperlihatkan data primer dan data sekunder yang digunakan dalam penelitian ini

No	Nama	Umur	Jabatan	Jenis Kela	Status Perkawinan	Project	Total Jam Kerja	Lama Bekerja (Tahun)	Resign	Performa	Satisfaction
1	MA	30	Managing	L	Kawin	23	46	7	No	95	100
2	HR	28	Develope	L	Kawin	23	46	7	No	95	95
3	YD	29	Develope	L	Kawin	23	46	7	No	95	95
4	RW	44	Advisor	L	Kawin	18	46	6	No	95	95
5	AA	23	Develope	L	Belum Kawin	8	46	2	Yes	90	

Gambar 2. Gambar tabel hasil wawancara PT. XYZ

satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_company	Work_accident	left	promotion	sales	salary
0.38	0.53	2	157	3	0	1	0	sales	low
0.8	0.86	5	262	6	0	1	0	sales	medium
0.11	0.88	7	272	4	0	1	0	sales	medium
0.72	0.87	5	223	5	0	1	0	sales	low

Gambar 3. Gambar tabel data sekunder

Tools yang digunakan adalah bahasa pemrograman *python* dengan *library pandas*, *numpy* dan *sklearn* yang mendukung algoritma yang digunakan dalam penelitian ini. Di dalam *library* sklearn juga terdapat *grid search* yang digunakan untuk mendapatkan parameter terbaik untuk setiap algoritma, serta *confusion matrix, accuracy score* dan *classification report* untuk mendapat skor hasil klasifikasi setiap algoritma yang nanti akan menjadi acuan perbandingan setiap algoritma yang digunakan untuk mendapat algoritma terbaik dari kelima algoritma yang digunakan.

Langkah ketiga adalah *exploratory data analysis.* Data yang telah didapat nantinya akan dianalisa dengan melihat informasi dalam data yang telah didapat, yaitu dengan melihat kesamaan tipe data dalam suatu kolom, dan juga mengenai kelengkapan data/tidak ada nilai hilang dalam suatu kolom

Langkah keempat adalah *data processing*. Dalam tahapan ini data yang telah dianalisis akan diolah lebih lanjut agar nantinya data model dapat belajar dari data dengan lebih baik.

Beberapa hal yang nanti dilakukan dalam proses ini yaitu, 1) Penghapusan kolom yang tidak digunakan. Tahapan pertama yang akan dilakukan adalah menghapus kolom pada dataset yang tidak digunakan. Tujuan penghapusan kolom ini adalah agar model menggunakan kolom yang lebih informatif yang sesuai dengan hasil yang didapat dari penelitian oleh [12]. 2) Mengisi nilai kolom yang hilang. Dikarenakan terdapat nilai yang hilang pada data primer, maka itu akan dilakukan pengisian nilai pada kolom tersebut, cara melakukan pengisian nilai ada beberapa metode yaitu dengan mengisinya dengan nilai *mean/median* dari data prediktif lainnya. Pada penelitian ini langkah yang akan digunakan adalah mengisi nilai kolom yang hilang tersebut dengan nilai *mean* dari data prediksi lain yaitu data sekunder. 3) Penyamaan nama kolom dan perubahan nilai. Data sekunder dan data primer masing masing mempunyai penamaan yang berbeda, oleh karena itu agar model dapat belajar dengan lebih baik maka akan dilakukan penamaan ulang pada data sekunder dan primer agar nama kolomnya tidak ada yang berbeda. Selain penamaan terdapat 1 nilai yang definisinya serupa tapi agak berbeda dalam perhitungannya, yaitu kolom *average monthly hour* pada data sekunder dan total jam kerja per minggu pada data primer, nilai kolom total jam kerja nantinya akan diubah sehingga dapat dihasilkan rata–rata jam kerja per bulan seperti kolom *average monthly hour* pada data sekunder. Selain kolom *average monthly hour,* kolom performa dan *satisfaction* pada data

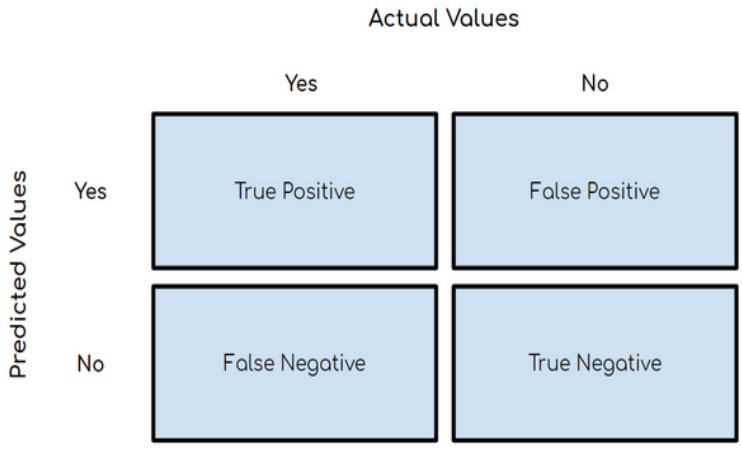
primer akan dihitung ulang nilainya sehingga tipe data nilainya sama dengan data sekunder. 4) Normalisasi nilai pada setiap kolom. Dikarenakan pada data sekunder dan data primer terdapat kolom yang memiliki rentang nilai yang berbeda maka akan dilakukan normalisasi pada kedua data sehingga rentang nilai pada kedua data sama. 5) Konversi label kategori menjadi numerik. Pada data primer terdapat data kolom *resign* yang merupakan data kategorik yang berisi (*Yes,No*), nantinya data ini akan diubah menjadi data numerik , dimana *Yes* diganti dengan 1, dan *No* diganti dengan 0. 6) Membagi data menjadi data latih dan data uji. Setelah dilakukan pengonversian data, langkah selanjutnya adalah membagi data menjadi data latih dan data uji. Untuk data latih data yang digunakan adalah data sekunder sedangkan untuk data uji data yang digunakan adalah data primer. **Tabel 1** merupakan kolom-kolom hasil dari tahapan *data processing*, yang akan digunakan menjadi data latih dan data uji:

Tabel 1. Tabel kolom hasil *data processing*

Nama Kolom
averageHours
lamaBekerja
performa
project
resign
satisfaction

Langkah kelima adalah evaluasi model. Setelah data diproses, langkah selanjutnya adalah menilai model dengan mengevaluasi kinerja model pembelajaran. Tingkat evaluasi model dibuat sebagai berikut: 1) Pemilihan parameter terbaik dengan *Grid Search*. Proses pertama sebelum mengevaluasi model adalah melakukan pemilihan parameter terbaik untuk setiap algoritma dengan menggunakan *Grid Search*, tahapan ini dilakukan agar setiap algoritma mendapatkan parameter terbaik sehingga hasil klasifikasi yang didapat lebih optimal. 2) Melihat hasil Akurasi, *Confusion Matrix* dan *Classification Report*. Setelah didapat parameter terbaik untuk setiap algoritma langkah selanjutnya melihat hasil klasifikasi setiap algoritma dengan data uji yaitu dengan menggunakan c*onfusion matrix* dan *classification report. Confusion matrix* adalah tabel yang dipakai untuk melakukan penilaian dalam kinerja model *machine learning*.

Gambar 4Gambar 4 merupakan *confusion matrix* dengan *True Positive* (TP): model memprediksi data positif dengan benar; *True Negative* (TN): model memprediksi data negatif dengan benar; *False Positive* (FP): model memprediksi data sebagai data positif tetapi seharusnya negative; *False Negative* (FN): model memprediksi data sebagai data negatif tetapi seharusnya positif.



Gambar 4. *Confusion Matrix*

Dari matrix tersebut dapat disimpulkan hasilnya menggunakan fungsi *Classification Report.* Laporan Klasifikasi digunakan untuk mengukur kualitas prediksi dari algoritma klasifikasi untuk setiap algoritma yang digunakan.

Gambar 5 Merupakan contoh hasil laporan klasifikasi yang nantinya akan digunakan setelah menghitung hasil yang didapat dari *confusion matrix*. *Precision* adalah rasio prediksi yang benar atrisi/tidak atrisi dibandingkan dengan keseluruhan hasil yang diprediksi atrisi/tidak atrisi, untuk penelitian ini berarti berapa persen karyawan yang benar atrisi dari keseluruhan karyawan yang diprediksi atrisi oleh model *machine learning*. *Recall* adalah rasio prediksi benar atrisi/tidak atrisi dibandingkan dengan keseluruhan data yang benar atrisi/tidak atrisi, pada penelitian ini recall berarti berapa persen karyawan yang diprediksi atrisi dibandingkan keseluruhan data karyawan yang atrisi. *F1-score* adalah metrik yang serupa dengan metrik akurasi akan tetapi untuk perbedaanya yaitu *f1-score* memperhitungkan pentingnya nilai *false positive* dan *false negative* yang dimana metrik ini merupakan nilai rata–rata harmonis dari presisi dan *recall*.

	precision	recall	f1-score	support
0	0.5833	1.0000	0.7368	7
1	1.0000	0.3750	0.5455	8
accuracy			0.6667	15
macro avg	0.7917	0.6875	0.6411	15
weighted avg	0.8056	0.6667	0.6348	15

Gambar 5. Contoh *Classification Report*

3. HASIL DAN PEMBAHASAN

Bagian ini membahas hasil dari implementasi kelima algoritma yang digunakan.

Algoritma C4.5

Tahapan pertama pada algoritma C4.5 adalah mencari parameter terbaik dengan *grid search* yang hasilnya dapat dilihat pada **Gambar 6**. Parameter yang akan digunakan yaitu:

1. *Max_depth* : kedalaman maksimum pohon.
2. *Max_features* : jumlah fitur yang perlu dipertimbangkan saat mencari pemisahan terbaik
3. *Random_state* : mengontrol pengacakan data saat melatih model
4. *Criterion :* fungsi untuk mengukur kualitas pembagian pohon
5. *Splitter :* fungsi yang digunakan untuk memilih pembagian pada setiap cabang dalam pohon

```
Best Score: 0.982065844170279
Best params: {'criterion': 'entropy', 'max_depth': 17, 'max_features': 'sqrt', 'random_state': 21, 'splitter': 'best'}
```

Gambar 6. Hasil *Grid Search* Algoritma C4.5

Tabel 2 merupakan tabel hasil matriks kebingungan dari algoritma C4.5, dari tabel tersebut hasil yang didapat untuk algoritma C4.5 adalah sebagai berikut: 1) Klasifikasi untuk karyawan yang tidak atrisi sebanyak 9 orang dari total 9 karyawan yang benar tidak atrisi. 2) Klasifikasi untuk karyawan yang atrisi sebanyak 5 orang dari total 8 karyawan yang benar tidak atrisi. 3) Klasifikasi karyawan yang seharusnya tidak atrisi tetapi dinyatakan atrisi oleh

algoritma ini sebanyak 0 orang. 4) Klasifikasi karyawan yang seharusnya atrisi tetapi dinyatakan tidak atrisi oleh algoritma ini sebanyak 3 orang.

Tabel 2 Tabel *Confusion Matrix Algoritma C4.5*

True Positive	*False Positive*	*True Negative*	*False Negative*
9	0	5	3

Algoritma C4.5 mendapat akurasi sebesar 82.35 persen yang dapat dilihat pada **Gambar *77***, dan untuk presisi, *recall* dan *f1-score* nilai yang digunakan adalah *weighted average*, dikarenakan nilai tersebut menghitung rata-rata dengan memperhitungkan bobot pada setiap datanya. Nilai dari presisi, *recall* dan *f1-score* untuk algoritma C4.5 adalah 86.76 persen, 82.35 persen, 81.57 persen yang dapat dilihat pada **Gambar *88***.

```
Akurasi C.45 :  0.8235294117647058
```

Gambar 7. Hasil Akurasi Algoritma C4.5

	precision	recall	f1-score	support
0	0.7500	1.0000	0.8571	9
1	1.0000	0.6250	0.7692	8
accuracy			0.8235	17
macro avg	0.8750	0.8125	0.8132	17
weighted avg	0.8676	0.8235	0.8158	17

Gambar 8. Hasil *Classification Report* Algoritma C4.5

Algoritma CART

Tahapan pertama pada algoritma CART adalah mencari parameter terbaik dengan *grid search* yang hasilnya dapat dilihat pada **Gambar *99***. Parameter yang digunakan yakni:

1. *Max_depth* : kedalaman maksimum pohon.
2. *Max_features* : jumlah fitur yang perlu dipertimbangkan saat mencari pemisahan terbaik
3. *Random_state* : mengontrol pengacakan data saat melatih model
4. *Criterion :* fungsi untuk mengukur kualitas pembagian pohon
5. *Splitter :* fungsi yang digunakan untuk memilih pembagian pada setiap cabang dalam pohon.

```
Best Score: 0.9823991775036124
Best params: {'criterion': 'gini', 'max_depth': 26, 'max_features': 'sqrt', 'random_state': 2, 'splitter': 'best'}
```

Gambar 9. Hasil *Grid Search* Algoritma CART

Tabel *33* merupakan tabel matriks kebingungan untuk algoritma CART. Terlihat bahwa hasil yang didapat untuk algoritma CART adalah sebagai berikut: 1) Klasifikasi untuk karyawan yang tidak atrisi sebanyak 8 orang dari total 9 karyawan yang benar tidak atrisi. 2) Klasifikasi untuk karyawan yang atrisi sebanyak 5 orang dari total 8 karyawan yang benar tidak atrisi. 3) Klasifikasi karyawan yang seharusnya tidak atrisi tetapi dinyatakan atrisi oleh algoritma ini sebanyak 1 orang. 4) Klasifikasi karyawan yang seharusnya atrisi tetapi dinyatakan tidak atrisi oleh algoritma ini sebanyak 3 orang.

Tabel 3 Tabel *Confusion Matrix* Algoritma CART

True Positive	*False Positive*	*True Negative*	*False Negative*
8	1	5	3

Untuk hasil akurasi dari algoritma CART didapat sebesar 76.47 persen , serta untuk hasil presisi, *recall* , dan *f1-score* didapat sebesar adalah 77.71 persen, 76.47 persen, 75.96 persen. yang dapat dilihat pada **Gambar *10*10** dan **Gambar *11*11**.

```
Akurasi CART :  0.7647058823529411
```

Gambar 10. Hasil Akurasi Algoritma CART

	precision	recall	f1-score	support
0	0.7273	0.8889	0.8000	9
1	0.8333	0.6250	0.7143	8
accuracy			0.7647	17
macro avg	0.7803	0.7569	0.7571	17
weighted avg	0.7772	0.7647	0.7597	17

Gambar 11. Hasil *Classification Report* Algoritma CART

Algoritma *Random Forest Classifier*

Tahapan pertama pada algoritma *random forest* adalah mencari parameter terbaik dengan *grid search* yang hasilnya dapat dilihat pada **Gambar 12.** Hasil Grid Search Algoritma Random Forest Classifier**2**, parameter yang akan digunakan yaitu:

1. *Max_depth* : kedalaman maksimum pohon.
2. *Max_features* : umlah fitur yang perlu dipertimbangkan saat mencari pemisahan terbaik
3. *Random_state* : mengontrol pengacakan data saat melatih model
4. *Criterion :* fungsi untuk mengukur kualitas pembagian pohon

```
Best Score: 0.9919329554295878
Best params: {'criterion': 'gini', 'max_depth': 20, 'max_features': 'sqrt', 'random_state': 49}
```

Gambar 12. Hasil *Grid Search* Algoritma *Random Forest Classifier*

Error! Not a valid bookmark self-reference.**4** merupakan tabel matriks kebingungan untuk algoritma *random forest*, , dari tabel tersebut hasil yang didapat untuk algoritma *random forest* adalah sebagai berikut: 1) Klasifikasi untuk karyawan yang tidak atrisi sebanyak 9 orang dari total 9 karyawan yang benar tidak atrisi. 2) Klasifikasi untuk karyawan yang atrisi sebanyak 5 orang dari total 8 karyawan yang benar tidak atrisi. 3) Klasifikasi karyawan yang seharusnya tidak atrisi tetapi dinyatakan atrisi oleh algoritma ini sebanyak 0 orang . 4) Klasifikasi karyawan yang seharusnya atrisi tetapi dinyatakan tidak atrisi oleh algoritma ini sebanyak 3 orang.

Tabel 4 Tabel *Confusion Matrix* Algoritma *Random Forest Classifier*

True Positive	*False Positive*	*True Negative*	*False Negative*
9	0	5	3

Untuk hasil akurasi dari algoritma *Random Forest* didapat sebesar 82.35 persen, serta untuk hasil presisi, *recall*, dan *f1-score* didapat sebesar 86.76 persen, 82.35 persen, 81.57 persen yang dapat dilihat pada **Gambar 13.** Hasil Akurasi Algoritma Random Forest Classifierdan **Gambar 14.** Hasil Classification Report Algoritma Random Forest Classifier

```
Akurasi Random Forest :  0.8235294117647058
```

Gambar 13. Hasil Akurasi Algoritma *Random Forest Classifier*

	precision	recall	f1-score	support
0	0.7500	1.0000	0.8571	9
1	1.0000	0.6250	0.7692	8
accuracy			0.8235	17
macro avg	0.8750	0.8125	0.8132	17
weighted avg	0.8676	0.8235	0.8158	17

Gambar 14. Hasil *Classification Report* Algoritma *Random Forest Classifier*

Algoritma Gradient Boost Classifier

Tahapan pertama pada algoritma *gradient boost* adalah mencari parameter terbaik dengan *grid search* yang hasilnya dapat dilihat pada **Gambar 155**. Parameter yang akan digunakan yaitu:

1. *Max_depth* : kedalaman maksimum pohon.
2. *Max_features* : jumlah fitur yang perlu dipertimbangkan saat mencari pemisahan terbaik
3. *Random_state* : mengontrol pengacakan data saat melatih model
4. *Criterion :* fungsi untuk mengukur kualitas pembagian pohon

```
Best Score: 0.9918663332221852
Best params: {'criterion': 'friedman_mse', 'max_depth': 10, 'max_features': 'sqrt', 'random_state': 8}
```

Gambar 15 Hasil *Grid Search* Algoritma *Gradient Boost Classifier*

Error! Not a valid bookmark self-reference.**5** merupakan tabel matriks kebingungan untuk algoritma *gradient boost*, dari tabel tersebut hasil yang didapat untuk algoritma *gradient boost* adalah sebagai berikut: 1) Klasifikasi untuk karyawan yang tidak atrisi sebanyak 9 orang dari total 9 karyawan yang benar tidak atrisi; 2) Klasifikasi untuk karyawan yang atrisi sebanyak 3 orang dari total 8 karyawan yang benar tidak atrisi; 3)Klasifikasi karyawan yang seharusnya tidak atrisi tetapi dinyatakan atrisi oleh algoritma ini sebanyak 0 orang; 4) Klasifikasi karyawan yang seharusnya atrisi tetapi dinyatakan tidak atrisi oleh algoritma ini sebanyak 5 orang.

Tabel 5 Tabel *Confusion Matrix Algoritma Gradient Boost Classifier*

True Positive	*False Positive*	*True Negative*	*False Negative*
9	0	3	5

Untuk hasil akurasi dari algoritma *Gradient Boost* didapat sebesar 70.58 persen, serta untuk hasil presisi, *recall* , dan *f1-score* didapat sebesar 81.09 persen, 70.58 persen, 67.10 persen yang dapat dilihat pada **Gambar 166** dan **Gambar 177**.

```
Akurasi Gradient Boost :  0.7058823529411765
```

Gambar 16. Hasil Akurasi Algoritma *Gradient Boost Classifier*

	precision	recall	f1-score	support
0	0.6429	1.0000	0.7826	9
1	1.0000	0.3750	0.5455	8
accuracy			0.7059	17
macro avg	0.8214	0.6875	0.6640	17
weighted avg	0.8109	0.7059	0.6710	17

Gambar 17. Hasil *Classification Report* Algoritma *Gradient Boost Classifier*

Algoritma *Adaboost Classifier*

Tahapan pertama pada algoritma adaboost adalah mencari parameter terbaik dengan *grid search* yang hasilnya dapat dilihat pada **Gambar 188.** parameter yang akan digunakan yaitu:

1. *Random_state* : mengontrol pengacakan data saat melatih model
2. *N_estomators :* jumlah maksimum di mana pohon tidak akan lagi melakukan peningkatan.
3. *Algorithm :* algoritma yang digunakan untuk mengukur peningkatan pohon

```
Best Score: 0.9584631988440592
Best params: {'algorithm': 'SAMME.R', 'n_estimators': 26, 'random_state': 1}
```

Gambar 18. Hasil *Grid Search* Algoritma *Adaboost Classifier*

Tabel *66* merupakan tabel matriks kebingungan untuk *adaboost*. Dari tabel tersebut hasil yang didapat untuk algoritma *adaboost* adalah sebagai berikut: 1) Klasifikasi untuk karyawan yang tidak atrisi sebanyak 9 orang dari total 9 karyawan yang benar tidak atrisi; 2) Klasifikasi untuk karyawan yang atrisi sebanyak 0 orang dari total 8 karyawan yang benar tidak atrisi; 3) Klasifikasi karyawan yang seharusnya tidak atrisi tetapi dinyatakan atrisi oleh algoritma ini sebanyak 0 orang; 4) Klasifikasi karyawan yang seharusnya atrisi tetapi dinyatakan tidak atrisi oleh algoritma ini sebanyak 8 orang.

Tabel 6 Tabel *Confusion Matrix Algoritma Adaboost Classifier*

True Positive	***False Positive***	***True Negative***	***False Negative***
9	0	0	8

Untuk hasil akurasi dari algoritma *Adaboost* didapat sebesar 52.94 persen, serta untuk hasil presisi, *recall* , dan *f1-score* didapat sebesar 28.02 persen, 52.94 persen, dan 36.65 persen yang dapat dilihat pada **Gambar 199** dan **Gambar 2020**.

```
Akurasi Adaboost :  0.5294117647058824
```

Gambar 19. Hasil Akurasi Algoritma *Adaboost Classifier*

	precision	recall	f1-score	support
0	0.5294	1.0000	0.6923	9
1	0.0000	0.0000	0.0000	8
accuracy			0.5294	17
macro avg	0.2647	0.5000	0.3462	17
weighted avg	0.2803	0.5294	0.3665	17

Gambar 20. Hasil *Classification Report* Algoritma *Adaboost Classifier*

Tabel *77* menyandingkan semua hasil pengujian dari kelima algoritma untuk akurasi, presisi, recall dan F1-score. Darinya didapati algoritma C4.5, dan *random forest* memiliki hasil akurasi, presisi, *recall* dan *f1-score* yang lebih baik dibandingkan CART, *Gradient Boost* dan *Adaboost*.

Tabel 7 Tabel Perbandingan Hasil Pengukuran

Algoritma	Akurasi	Presisi	Recall	F1-Score
C4.5	82.35%	86.76%	82.35%	81.57%
CART	76.47%	77.71%	76.47%	75.96%
Random Forest	82.35%	86.76%	82.35%	81.57%
Gradient Boost	70.58%	81.09%	70.58%	67.10%
Adaboost	52.94%	28.02%	52.94%	36.65%

4. KESIMPULAN

Berdasarkan pengujian yang dilakukan untuk membandingkan beberapa algoritma klasifikasi *decision tree* dalam memprediksi karyawan yang berpotensi atrisi di PT. XYZ menunjukkan bahwa algoritma C4.5 dan *random forest* memiliki hasil yang lebih baik dimana nilai akurasi yang diperoleh adalah 82.35%, presisi 86.76 persen, *recall* 83.35 persen, dan *f1-score* 81.57 persen.

DAFTAR PUSTAKA

[1] E. Novitasari, *PENGANTAR MANAJEMEN: Panduan Menguasai Ilmu Manajemen*. Anak Hebat Indonesia, 2017.

[2] U. Subagyo and F. Santoso, "Sistem Pendukung Keputusan Penilaian Kinerja Pegawai Pada FIFGROUP dengan Metode Simple Additive Weighting," *Jurnal Informatika Komputer, Bisnis dan Manajemen*, vol. 20, no. 2, pp. 75–86, 2022, doi: https://doi.org/10.61805/fahma.v20i2.31.

[3] T. H. Lamramot, A. I. Hadiana, and I. Santikarama, "Sistem Prediksi Awal Terhadap Atrisi Karyawan Menggunakan Algoritma C4.5 INFORMASI ARTIKEL A B S T R A K," 2022. [Online]. Available: https://e-journal.unper.ac.id/index.php/informatics

[4] F. Fallucchi, M. Coladangelo, R. Giuliano, and E. W. De Luca, "Predicting employee attrition using machine learning techniques," *Computers*, vol. 9, no. 4, pp. 1–17, 2020, doi: 10.3390/computers9040086.

[5] V. Nowotny, “HUMAN RESOURCE PROFESSIONALIZATION IN STARTUPS General Management,” Johannes Kepler University Linz, Altenberger Strabe, 2020.

[6] N. Indah Prabawati, Widodo, and H. Ajie, “Kinerja Algoritma Classification And Regression Tree (Cart) dalam Mengklasifikasikan Lama Masa Studi Mahasiswa yang Mengikuti Organisasi di Universitas Negeri Jakarta,” *PINTER : Jurnal Pendidikan Teknik Informatika dan Komputer*, vol. 3, no. 2, pp. 139–145, Dec. 2019, doi: 10.21009/pinter.3.2.9.

[7] A. Raza, K. Munir, M. Almutairi, F. Younas, and M. M. S. Fareed, “Predicting Employee Attrition Using Machine Learning Approaches,” *Applied Sciences (Switzerland)*, vol. 12, no. 13, Jul. 2022, doi: 10.3390/app12136424.

[8] A. Qutub, A. Al-Mehmadi, M. Al-Hssan, R. Aljohani, and H. S. Alghamdi, “Prediction of Employee Attrition Using Machine Learning and Ensemble Methods,” *Int J Mach Learn Comput*, vol. 11, no. 2, pp. 110–114, Mar. 2021, doi: 10.18178/ijmlc.2021.11.2.1022.

[9] N. J. Apao, L. S. Feliscuzo, C. C. Lyn Sta Romana, and J. S. Aurea Tagaro, “Multiclass Classification Using Random Forest Algorithm To Prognosticate The Level Of Activity Of Patients With Stroke,” *International Journal of Scientific & Technology Research*, vol. 9, no. 04, pp. 1233–1240, 2020, [Online]. Available: www.ijstr.org

[10] O. Wisesa, A. Adriansyah, and O. I. Khalaf, “Prediction Analysis Sales for Corporate Services Telecommunications Company using Gradient Boost Algorithm,” in *2020 2nd International Conference on Broadband Communications, Wireless Sensors and Powering, BCWSP 2020*, Institute of Electrical and Electronics Engineers Inc., Sep. 2020, pp. 101–106. doi: 10.1109/BCWSP50066.2020.9249397.

[11] A. Rehman Javed, Z. Jalil, S. Atif Moqurrab, S. Abbas, and X. Liu, “Ensemble Adaboost classifier for accurate and fast detection of botnet attacks in connected vehicles,” *Transactions on Emerging Telecommunications Technologies*, vol. 33, no. 10, Oct. 2022, doi: 10.1002/ett.4088.

[12] W. Bangun, “Manajemen Sumber Daya Manusia,” 2012.