

Perbandingan Algoritma C4.5 dan *Adaptive Boosting* dalam Klasifikasi Penyakit Alzheimer

Comparison of C4.5 and Adaptive Boosting Algorithms in Alzheimer's Disease Classification

Alda Amalia Mortara¹, Mitta Permatasari², Anita Desiani³, Yuli Andriani⁴, Muhammad Arhami⁵

Program Studi Matematika, Universitas Sriwijaya, Indonesia^{1,2,3,4}

Program Studi Teknologi Informasi dan Komputer, Politeknik Negeri Lhokseumawe, Indonesia⁵

08011282126069@student.unsri.ac.id¹, 08011282126071@student.unsri.ac.id², anita_desiani@unsri.ac.id³, yuliandriani@unsri.ac.id⁴, muhammad.arhami@pnl.ac.id⁵

Abstrak

Penyakit alzheimer adalah penyakit yang menyerang sistem saraf di dalam otak. Penyakit ini dapat menyebabkan terganggunya aktivitas sehari-hari, ingatan yang tidak terorganisir, dan berkurangnya daya ingat. Deteksi dini penyakit alzheimer dapat memanfaatkan pendekatan matematis menggunakan *data mining*. *Data mining* memiliki model-model klasifikasi yang dapat digunakan untuk mendeteksi dini penyakit alzheimer. Beberapa algoritma yang dapat digunakan untuk klasifikasi diantaranya adalah C4.5 dan *Adaptive Boosting (AdaBoost)* yang diterapkan pada penelitian ini untuk mengklasifikasikan penyakit alzheimer. Perbandingan kedua algoritma ini bertujuan untuk memperoleh algoritma mana yang paling tepat dalam klasifikasi penyakit alzheimer. Untuk menguji kedua algoritma ini digunakan dua teknik pengujian yaitu *percentage split* dan *k-fold cross validation*. Pada *percentage split* dipilih ukuran split sebesar 80% untuk data latih dan 20% sebagai data uji dan *k-fold cross validation* dipilih nilai k sebesar 10. Hasil penerapan dari kedua algoritma diperoleh bahwa untuk *k-fold cross validation* bekerja lebih baik dibandingkan dengan *percentage split*. Hal ini dikarenakan *k-fold cross validation* meningkatkan persentase nilai presisi, *recall*, dan akurasi dari masing-masing algoritma. Untuk kinerja masing-masing algoritma, *AdaBoost* dalam penggunaannya bekerja lebih baik dibandingkan dengan C4.5 dengan nilai presisi, *recall* dan akurasi secara berturut-turut, yaitu 91.5%, 91% dan 91.15%. Dari hasil yang diperoleh dapat disimpulkan bahwa algoritma *AdaBoost* dengan teknik *k-fold cross validation* memiliki performa yang paling baik dalam melakukan klasifikasi penyakit alzheimer dibandingkan algoritma dan teknik pengujian lainnya.

Kata kunci: *Data Mining*; Klasifikasi; Alzheimer; C4.5; *AdaBoost*; *Percentage Split*; *K-fold Cross Validation*.

Abstract

Alzheimer's is a disease that attacks the nervous system of the brain. It can result in disruptions of everyday activities, disorganized memory, and memory loss. Early detection of alzheimer's disease can utilize a mathematical approach using data mining. Data mining has classified models that can be used to detect early onset alzheimer's disease. Some algorithms that can be used for classifications include C4.5 and Adaptive Boosting (AdaBoost) as applied to this study to classify alzheimer's disease. The second comparison of the algorithm aims to acquire which algorithm is most appropriate in the classification of alzheimer's disease. In testing both the algorithms used two testing techniques: split and the k-fold validation. At the market split selected 80% in size for training and 20% in check and validation's k-fold selected a k by 10. Applications from both algorithms obtained that k-fold validation works better than split. This is because the k-fold validation increases the percentage of value value of precision, recall, and accuracy from each algorithm. For performance on each algorithm, AdaBoost in its utility works better than C4.5 with precision value, recall and accuracy in succession, which is 91.5%, 91% and 91.15%. From results it can be concluded that algorithm AdaBoost with validation k-fold techniques performs best at classification alzheimer's disease than other algorithms and testing techniques.

Keywords: *Data Mining*; Classification; Alzheimer; C4.5; *AdaBoost*; *Percentage Split*; *K-fold Cross Validation*.

Naskah diterima 24 Juli 2023; direvisi 8 September 2023; dipublikasi 9 September 2023.
JATI is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.



1. Pendahuluan

Penyakit alzheimer adalah kondisi neurologis yang menyerang sistem saraf pada otak. Penyakit alzheimer menyebabkan hilangnya neuron dan membuat penderita sulit untuk melakukan aktivitas sehari-hari dikarenakan kebingungan, ingatan yang tidak terorganisir, dan pada akhirnya akan menghilangnya kemampuan untuk mengingat [1]. Faktor utama yang memengaruhi perkembangan penyakit alzheimer adalah usia. Penyakit ini biasanya didiagnosis pada orang yang berusia 65 tahun ke atas, tetapi tidak menutup kemungkinan terjadi pada orang yang lebih muda. Selain itu, keturunan garis genetik dan cedera otak traumatis yang cukup berat juga terkait dengan peningkatan risiko [2]. Organisasi Kesehatan Dunia (WHO) telah mengungkapkan bahwa 55 juta orang di seluruh dunia mengalami penurunan keterampilan dalam memori dan berpikir, yang sekarang

dapat mempengaruhi individu berusia 30 sampai 40 tahun. Penyakit ini menjadi salah satu tantangan terbesar dalam dunia kesehatan di abad ke-21 ini [3].

Diagnosis tahap awal untuk penyakit alzheimer dapat meningkatkan efisiensi terapi untuk penyakit ini. Diagnosa penyakit alzheimer dapat dibantu dengan pendeteksian dini. Deteksi dini penyakit alzheimer dapat memanfaatkan pendekatan matematis pada *data mining* untuk menganalisis data. Data mining memiliki berbagai metode, salah satunya klasifikasi yang dapat diterapkan untuk pendeteksian dini atau prediksi penyakit alzheimer. Klasifikasi adalah proses untuk mengidentifikasi dan mengelompokkan objek yang memiliki karakteristik yang sama ke dalam beberapa kelas yang telah ditentukan [4]. Beberapa algoritma yang bisa digunakan pada klasifikasi penyakit alzheimer adalah algoritma C4.5 dan algoritma *AdaBoost*.

Pohon keputusan adalah salah satu teknik klasifikasi yang paling banyak digunakan [5]. Salah satu algoritma pohon keputusan adalah C4.5 yang mampu menghasilkan pohon keputusan yang akurat dan mudah dipahami. Selain itu dalam penerapannya, dapat mengidentifikasi atribut yang paling berpengaruh dan atribut yang tidak berpengaruh. Beberapa penelitian sebelumnya telah memanfaatkan algoritma C4.5, seperti Muhammad Fikri Hasani dan Sutikno [6] dalam klasifikasi penyakit infeksi saluran kemih. Penelitian tersebut menghasilkan akurasi, presisi, dan *recall* dengan nilai kinerja di atas 90%. Arni Sepharni, Iwansyah Edo Hendrawan, dan Chaerur Rozikin [7] menerapkan C4.5 dalam klasifikasi penyakit jantung. Namun sayangnya, hasil penelitian tersebut menghasilkan nilai akurasi, presisi, dan *recall* di bawah 80%. Meskipun memiliki berbagai kelebihan, algoritma C4.5 mempunyai kelemahan dalam kategori skalabilitas, yang mana semakin banyak atribut maka semakin banyak pula memori yang dibutuhkan [8]. Selain itu, algoritma C4.5 cenderung menghasilkan pohon keputusan yang kompleks dan dalam beberapa kasus dapat *overfitting*. Metode lain yang bisa digunakan untuk klasifikasi penyakit alzheimer ini adalah algoritma *AdaBoost* [9].

Berbeda dengan algoritma C4.5, algoritma *AdaBoost* memiliki kelebihan yang tidak terlalu rentan terhadap *overfitting*. Dengan memperhatikan sampel-sampel yang sulit diklasifikasikan, *AdaBoost* membantu mengurangi risiko *overfitting* [10]. *AdaBoost* adalah algoritma *ensemble learning* yang menggabungkan beberapa model (*multi classifier*) untuk mencapai hasil akhir [11]. Sehingga kelemahan pada algoritma *single classifier* seperti C4.5 dapat diatasi dengan algoritma *multi classifier* salah satunya adalah *AdaBoost*. Beberapa penelitian sebelumnya yang menggunakan algoritma *AdaBoost* dalam penelitiannya adalah Agus Byna dan Muhammad Basit [12] yang menerapkan algoritma *AdaBoost* guna mengoptimasi prediksi penyakit stroke dengan hasil akurasi 98%. Eka Listiana dan Much Aziz Muslim [13] menerapkan algoritma *AdaBoost* dalam mendiagnosa penyakit ginjal kronis dengan akurasi yang diperoleh sangat baik yaitu sebesar 99.5%. Namun *AdaBoost* masih rentan terhadap *outlier* atau nilai data yang berbeda dari set data lainnya [14].

Penelitian yang berkenaan dengan klasifikasi penyakit alzheimer belum banyak dilakukan. *Dataset* yang digunakan pada penelitian ini memiliki atribut-atribut yang saling relevan dan informatif sehingga dapat memberikan hasil deteksi dan klasifikasi yang lebih baik. Pada penelitian ini akan dibandingkan performa algoritma C4.5 dan algoritma *AdaBoost* dalam klasifikasi penyakit alzheimer. Dalam pengujian kedua algoritma digunakan teknik pengujian *percentage split* dan *k-fold cross validation*. Pada *percentage split* dipilih ukuran split sebesar 80% untuk data latih dan 20% sebagai data uji. Pada teknik pengujian *k-fold cross validation* dipilih nilai k sebesar 10 dimana data akan dibagi menjadi sepuluh kelompok dan dapat bergantian sebagai data latih serta data uji sebanyak sepuluh kali. Penelitian ini akan menentukan algoritma dan teknik pengujian mana yang paling optimal dalam mendeteksi penyakit alzheimer berdasarkan kriteria akurasi, presisi, dan *recall*.

2. Metode Penelitian

Tahapan-tahapan yang akan dilakukan dalam penelitian ini ditunjukkan pada Gambar 1.



Gambar 1. Rangkaian Proses yang Dilakukan dalam Penelitian

2.1. Deskripsi Data

Data yang digunakan pada penelitian ini adalah dataset yang diambil melalui situs Kaggle (<https://bit.ly/3MmyQ3L>) dalam format csv. Dataset Alzheimer ini terdiri dari 373 data dengan 15 atribut yang meliputi *Subject ID*, *MRI ID*, *Group*, *Gender*, *Hand*, *Age*, *EDUC*, *SES*, *MMSE*, *CDR*, *eTIV*, *nWBV*, *ASF*. Atribut target pada dataset ini adalah *Group* yang meliputi 3 kelas yaitu *demented*, *nondemented* dan *converted*. Kelas *converted* akan dikategorikan ke dalam kelas *demented*. Dengan demikian, jumlah data untuk kelas

demented adalah 183 dan data kelas *nondemented* berjumlah sebanyak 190 data. Atribut-atribut yang digunakan pada data alzheimer beserta tipe atribut disajikan pada Tabel 1.

Tabel 1. Atribut beserta Deskripsinya

| Nama Atribut | Tipe Atribut | Nilai | Missing Data |
|-------------------|--------------|---|--------------|
| <i>Subject ID</i> | Kategorik | ID pemeriksaan subjek 1-142 | <i>Null</i> |
| <i>MRI ID</i> | Kategorik | ID Tes 1-354 | <i>Null</i> |
| <i>Group</i> | Kategorik | 0: <i>Demented</i> 1: <i>Non-Demented</i> | <i>Null</i> |
| <i>Visit</i> | Numerik | Jumlah pemeriksaan subjek (1-5) | <i>Null</i> |
| <i>MR Delay</i> | Numerik | Nilai keterlambatan subjek sejak melakukan pemeriksaan (0-2639) | <i>Null</i> |
| <i>M/F</i> | Kategorik | F: <i>Female</i> M: <i>Male</i> | <i>Null</i> |
| <i>Hand</i> | Kategorik | R: <i>Right</i> L: <i>Left</i> | <i>Null</i> |
| <i>Age</i> | Numerik | Usia subjek pada saat dilakukan pemeriksaan (60-98 tahun) | <i>Null</i> |
| <i>EDUC</i> | Kategorik | Tingkat pendidikan (6-23) | <i>Null</i> |
| <i>SES</i> | Kategorik | Status Sosial Ekonomi 1 (<i>highest statue</i>) – 5 (<i>lowest statue</i>) | 19 |
| <i>MMSE</i> | Numerik | Nilai pemeriksaan keadaan mental 0 (<i>worst value</i>) – 30 (<i>best value</i>) | 2 |
| <i>CDR</i> | Kategorik | Tingkat demensia klinis 0= <i>No Dementia</i> , 0.5= <i>Very Mild AD</i> , 1= <i>Mild AD</i> , 2= <i>Moderate AD</i> | <i>Null</i> |
| <i>eTIV</i> | Numerik | Perkiraan total volume intrakranial (1106-2004) | <i>Null</i> |
| <i>nWBV</i> | Numerik | Volume keseluruhan otak yang dinormalisasikan di ekspersika dengan persentase dari semua voxel (0.644-0.837) | <i>Null</i> |
| <i>ASF</i> | Numerik | Faktor skala atlas (0.876 -1.587) | <i>Null</i> |

2.2. Preprocessing Data (Pemrosesan Data)

Data penyakit alzheimer dari Tabel 1 terdiri dari 15 atribut. Atribut yang tidak memiliki pengaruh akan dihapus sehingga yang awalnya terdapat 15 atribut, menjadi sebanyak 12 atribut, yang terdiri dari *Group*, *MR Delay*, *M/F*, *Hand*, *Age*, *EDUC*, *SES*, *MMSE*, *CDR*, *eTIV*, *nWBV*, *ASF*. Dimana atribut atau kolom *Group* merupakan keputusan atau label dari kriteria yang dialami oleh penderita, sedangkan 11 kolom atau atribut lainnya adalah kriteria seseorang yang dikatakan menderita atau tidak menderita penyakit alzheimer.

a. Missing Data (Data Kosong/Hilang)

Dalam dataset penyakit alzheimer terdapat dua atribut yang mempunyai data yang hilang di dalamnya yaitu 19 record pada kolom *SES* dan 2 record pada kolom *MMSE*. Data yang hilang akan dilakukan imputasi. Pada kolom *SES* dan *MMSE* dilakukan imputasi dengan menggunakan *mean*, yaitu rata-rata nilai dari seluruh data pada kolom yang akan diimputasi.

b. Normalisasi Data

Pada dataset penyakit alzheimer dari Tabel 1 dapat dilihat bahwa untuk atribut *MMSE* dan *eTIV* memiliki range nilai yang cukup jauh, sehingga dilakukan normalisasi data. Tujuan dilakukannya normalisasi adalah menyamakan range nilai atribut dengan skala tertentu agar dapat menghasilkan klasifikasi yang lebih baik. Normalisasi data menggunakan penskalaan min-max dengan rumus seperti pada persamaan (1) [15]:

$$Normalized(x) = \frac{\min Range\ x + (x\ \min\ Value)(\max\ Range - \min\ Range)}{\max\ Value - \min\ Value} \quad (1)$$

c. Pengujian Data

Pada pengolahan data awal juga digunakan dua teknik pengujian yaitu *percentage split* dan *k-fold cross validation*. *Percentage split* dalam *data mining* adalah teknik untuk membagi dataset menjadi dua subset, yaitu subset pelatihan (*training set*) dan subset pengujian (*testing set*) [16]. Pembagian tersebut bertujuan untuk melatih model agar dapat mengenali pola berdasarkan data yang ada dan dapat menguji performa hasil prediksi dari model pada subset pengujian [17]. Pembagian data pada penelitian ini sebesar 80% untuk data latih dan 20% untuk data uji. Kemudian pada teknik pengujian *k-fold cross validation* langkah awalnya adalah data yang tersedia dipartisi menjadi k segmen dengan ukuran yang sama atau mendekati. Setelah itu data latih bagi proses iterasi atau pengulangan serta data validasi dilakukan. Sebelum melakukan proses *k-fold* data dibagi menjadi

tingkatan, kemudian mengelola ulang data untuk setiap lipatan data dan merupakan perwakilan yang sesuai dari seluruh data yang ada. Pada penelitian ini dipilih nilai k sebesar 10.

2.3. Penerapan C4.5

Algoritma C4.5 adalah salah satu algoritma klasifikasi yang menggunakan representasi struktur pohon. Dalam algoritma ini, setiap node menunjukkan nilai atribut, setiap cabang menunjukkan nilai atribut, dan setiap daun menunjukkan kelas. Berbagai langkah yang diambil untuk membuat pohon keputusan algoritma C4.5 adalah sebagai berikut [18]:

1. Tentukan atribut sebagai akar

Setiap atribut yang dipilih untuk menjadi akar berdasarkan nilai *gain ratio* tertinggi dari semua atribut [19]. Persamaan (2) berisi rumus untuk menemukan nilai *gain ratio*.

$$GainRatio(S, A) = \frac{Gain(S, A)}{SplitInfo(S, A)} \quad (2)$$

Dengan S sebagai himpunan kasus, A sebagai atribut, $Gain(S, A)$ adalah *information gain* pada atribut A dan $SplitInfo(S, A)$ adalah *split information* pada atribut A . *Split information* menyatakan informasi potensial atau dikenal juga sebagai entropi pada atribut A terhadap banyaknya himpunan kasus. Perhitungan *split information* dapat dilihat pada persamaan (3).

$$SplitInfo(S, A) = - \sum_{i=1}^n \frac{S_i}{S} \log_2 \frac{S_i}{S} \quad (3)$$

Dimana S merupakan himpunan kasus, A sebagai atribut, S_i adalah jumlah sampel atribut A . *Gain* merupakan tingkat pengaruh suatu atribut terhadap pengambilan keputusan atau ukuran efektivitas suatu variabel dalam mengklasifikasikan data [20]. Persamaan (4) menunjukkan rumus yang digunakan untuk menghitung *gain*.

$$Gain(S, A) = Entropi(S) - \sum_{i=1}^n \frac{|S_i|}{|S|} * Entropi(S_i) \quad (4)$$

Dengan N sebagai jumlah partisi atribut A , $|S_i|$ adalah jumlah partisi ke i , dan $|S|$ adalah jumlah partisi dalam S . Perhitungan entropi digunakan untuk menentukan seberapa informatif sebuah atribut yang dimasukkan untuk menghasilkan atribut [21]. Sementara itu, cara menghitung nilai entropi ditunjukkan pada persamaan (5).

$$Entropi(S) = \sum_{i=1}^n -p_i * \log_2 p_i \quad (5)$$

Dalam hal ini, S merupakan himpunan kasus, n merupakan jumlah partisi atribut A , A adalah atribut, dan p_i adalah proporsi S_i terhadap S .

2. Pada setiap nilai dibuat cabang.
3. Bagi kasus dalam cabang.
4. Pada setiap cabang, ulangi prosedur di atas untuk sampai semua kasus pada cabang memiliki kelas yang sama.

2.4. Penerapan Adaptive Boosting

Adaboost dalam penerapannya diawali dengan membangun model dan membuat prediksi. Kemudian memberikan bobot lebih pada observasi yang salah diklasifikasikan. Selanjutnya membangun model berikutnya, dan dapat diulangi langkah awal, setelah itu dibuat model akhir dengan menggunakan rata-rata dari semua model. Beberapa tahapan algoritma *AdaBoost* bekerja adalah sebagai berikut [22]:

1. Menentukan bobot pada tiap data, digunakan persamaan (6):

$$w(x_i, y_i) = \frac{1}{N} \quad (6)$$

Dimana w merupakan nilai *weight* (bobot), (x_i, y_i) adalah data pelatihan ke- i dimana $i = \{1, 2, \dots, n\}$, dan N adalah banyaknya data.

2. Setelah mendapatkan nilai bobot untuk setiap data, maka dilanjutkan dengan mencari nilai Entropi dan *Information Gain*. Untuk mencari nilai Entropi dan *Information Gain* dapat digunakan persamaan (7) dan persamaan (8):

$$Entropi(S) = \sum_{i=1}^n -p_i \log_2(p_i) \quad (7)$$

$$Information\ Gain = Entropi(S) - \sum_{i=1}^n \frac{|S_i|}{|S|} Entropi(S_i) \quad (8)$$

Dimana pada persamaan (7), S merupakan himpunan kasus, p_i adalah probabilitas dari S_i terhadap S . Kemudian pada persamaan (8), $|S_i|$ merupakan jumlah kasus pada partisi ke- i , dan $|S|$ merupakan jumlah kasus pada S .

3. Setelah didapatkan nilai Entropi dan *Information Gain*, langkah selanjutnya adalah mengklasifikasi data berdasarkan nilai *Information Gain* tertinggi. Setelah itu perbarui distribusi sehingga nilai bobot menjadi baru. Selanjutnya menentukan *Error Rate* dan bobot. Untuk mencari *Error Rate* dapat menggunakan persamaan (9):

$$\varepsilon_t = \frac{1}{N}(x) \quad (9)$$

Untuk ε_t adalah nilai *Error Rate* yaitu jumlah ketidaktepatan klasifikasi pada data latih dibagi dengan ukuran data latih, t ialah nilai jumlah percobaan atau iterasi, N adalah banyaknya data, dan x ialah banyaknya data yang berbeda antara hasil dengan target (*predict*). Untuk mencari bobot dapat menggunakan persamaan (10):

$$\alpha_t = \log\left(\frac{1 - \varepsilon_t}{\varepsilon_t}\right) \quad (10)$$

Dimana α_t merupakan nilai bobot untuk pengklasifikasi, kemudian t ialah nilai jumlah percobaan atau iterasi, dan ε_t merupakan nilai *Error Rate*.

4. Selanjutnya dilakukan pembaruan untuk bobot dengan pengamatan yang salah, dapat digunakan persamaan (11):

$$D_{t+1} = \frac{1}{N} e^{\alpha_t} \quad (11)$$

Dimana D_{t+1} merupakan pembaruan nilai bobot untuk pengamatan yang salah, N banyaknya data, dan α_t merupakan nilai bobot. Setelah melakukan pembaruan bobot tersebut, seluruh nilai bobot terbaru akan dijumlahkan untuk menormalisasi nilai agar nilai tersebut menjadi 1. Normalisasi dilakukan dengan membagi bobot baru. Selanjutnya prediksi final dari stump yang terbentuk dapat dilihat pada bobot-bobot baru yang telah dibuat.

2.5. Confusion Matrix

Confusion matrix merupakan sebuah tabel yang menyatakan klasifikasi jumlah data uji yang tepat diklasifikasikan dan jumlah data uji yang salah diklasifikasikan [20]. Dalam penaksiran kinerja suatu model menggunakan *confusion matrix*, ada empat istilah sebagai representasi dari hasil proses klasifikasi, yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN) [23]. Berikut *confusion matrix* untuk klasifikasi dua kelas menurut pada Tabel 2.

Tabel 2. *Confusion Matrix*

| Kelas | | Prediksi | |
|--------|---------|----------|---------|
| | | Positif | Negatif |
| Aktual | Positif | TP | FN |
| | Negatif | FP | TN |

Ada ukuran evaluasi kinerja yang digunakan dalam klasifikasi nilai-nilai dalam *confusion matrix*, yaitu akurasi, presisi, dan *recall*. Berdasarkan rumus perhitungan, performansi *confusion matrix* ditunjukkan pada persamaan (12), persamaan (13), dan persamaan (14) [24].

$$Akurasi = \frac{TP + TN}{TP + FP + TN + FN} \quad (12)$$

$$Presisi = \frac{TP}{FP + TP} \quad (13)$$

$$Recall = \frac{TP}{FN + TP} \quad (14)$$

3. Hasil dan Pembahasan

3.1. C4.5

Hasil perhitungan *gain* pada algoritma C4.5 dengan menggunakan *percentage split* dan *k-fold cross validation* menghasilkan *gain* tertinggi yang sama, yaitu atribut CDR. Nilai *gain* tertinggi di peroleh menunjukkan bahwa jika nilai CDR dari pasien tersebut belum melebihi dari nilai yang ditentukan maka pasien tersebut akan dilabeli 0. Untuk nilai CDR yang tidak memenuhi syarat maka harus melihat atribut lainnya untuk mendiagnosa penyakit alzheimer. Dari perhitungan C4.5 menggunakan *percentage split* diperoleh aturan linguistik untuk klasifikasi penyakit alzheimer sebagai berikut :

IF CDR ≤ 0.25 AND MMSE ≤ 0.827 THEN 0

IF CDR ≤ 0.25 AND MMSE > 0.827 AND MR Delay ≤ 534.5 AND Age ≤ 82.5 AND EDUC ≤ 19.0 AND nWBV ≤ 0.793 AND ASF ≤ 1.187 THEN 1

IF CDR ≤ 0.25 AND MMSE > 0.827 AND MR Delay ≤ 534.5 AND Age ≤ 82.5 AND EDUC ≤ 19.0 AND nWBV ≤ 0.793 AND ASF > 1.187 AND ASF ≤ 1.192 AND EDUC ≤ 11.5 THEN 1

IF CDR ≤ 0.25 AND MMSE > 0.827 AND MR Delay ≤ 534.5 AND Age ≤ 82.5 AND EDUC ≤ 19.0 AND nWBV ≤ 0.793 AND ASF > 1.187 AND ASF ≤ 1.192 AND EDUC > 11.5 THEN 0

IF CDR ≤ 0.25 AND MMSE > 0.827 AND MR Delay ≤ 534.5 AND Age ≤ 82.5 AND EDUC ≤ 19.0 AND nWBV ≤ 0.793 AND ASF > 1.187 AND ASF > 1.192 AND SES ≤ 1.5 AND Age ≤ 72.0 THEN 0

IF CDR ≤ 0.25 AND MMSE > 0.827 AND MR Delay ≤ 534.5 AND Age ≤ 82.5 AND EDUC ≤ 19.0 AND nWBV ≤ 0.793 AND ASF > 1.187 AND ASF > 1.192 AND SES ≤ 1.5 AND Age > 72.0 THEN 1

IF CDR ≤ 0.25 AND MMSE > 0.827 AND MR Delay ≤ 534.5 AND Age ≤ 82.5 AND EDUC ≤ 19.0 AND nWBV ≤ 0.793 AND ASF > 1.187 AND ASF > 1.192 AND SES > 1.5 THEN 1

IF CDR ≤ 0.25 AND MMSE > 0.827 AND MR Delay ≤ 534.5 AND Age ≤ 82.5 AND EDUC ≤ 19.0 AND nWBV > 0.793 AND nWBV ≤ 0.802 THEN 0

IF CDR ≤ 0.25 AND MMSE > 0.827 AND MR Delay ≤ 534.5 AND Age ≤ 82.5 AND EDUC ≤ 19.0 AND nWBV > 0.793 AND nWBV > 0.802 THEN 1

IF CDR ≤ 0.25 AND MMSE > 0.827 AND MR Delay ≤ 534.5 AND Age ≤ 82.5 AND EDUC > 19.0 AND Age ≤ 76.5 THEN 1

IF CDR ≤ 0.25 AND MMSE > 0.827 AND MR Delay ≤ 534.5 AND Age ≤ 82.5 AND EDUC > 19.0 AND Age > 76.5 THEN 1

IF CDR ≤ 0.25 AND MMSE > 0.827 AND MR Delay ≤ 534.5 AND Age > 82.5 AND ASF ≤ 1.203 THEN 1

IF CDR ≤ 0.25 AND MMSE > 0.827 AND MR Delay ≤ 534.5 AND Age > 82.5 AND ASF > 1.203 AND SES ≤ 2.5 THEN 0

IF CDR ≤ 0.25 AND MMSE > 0.827 AND MR Delay ≤ 534.5 AND Age > 82.5 AND ASF > 1.203 AND SES > 2.5 AND ASF ≤ 1.228 THEN 0

IF CDR ≤ 0.25 AND MMSE > 0.827 AND MR Delay ≤ 534.5 AND Age > 82.5 AND ASF > 1.203 AND SES > 2.5 AND ASF > 1.228 THEN 1

IF CDR \leq 0.25 AND MMSE $>$ 0.827 AND MR Delay $>$ 534.5 THEN 1
IF CDR $>$ 0.25 AND eTIV \leq 0.659 THEN 0
IF CDR $>$ 0.25 AND eTIV $>$ 0.659 AND Age \leq 80.5 THEN 0
IF CDR $>$ 0.25 AND eTIV $>$ 0.659 AND Age $>$ 80.5 AND CDR \leq 0.75 THEN 1
IF CDR $>$ 0.25 AND eTIV $>$ 0.659 AND Age $>$ 80.5 AND CDR \leq 0.75 THEN 0

Berikut aturan linguistik yang diperoleh dari algoritma C4.5 menggunakan *k-fold cross validation* untuk klasifikasi penyakit alzheimer:

IF CDR \leq 0.25 AND MMSE \leq 0.827 THEN 0
IF CDR \leq 0.25 AND MMSE $>$ 0.827 AND SES \leq 1.5 AND EDUC \leq 15.0 THEN 0
IF CDR \leq 0.25 AND MMSE $>$ 0.827 AND SES \leq 1.5 AND EDUC $>$ 15.0 AND MR Delay \leq 220.5 AND nWBV \leq 0.696 THEN 0
IF CDR \leq 0.25 AND MMSE $>$ 0.827 AND SES \leq 1.5 AND EDUC $>$ 15.0 AND MR Delay \leq 220.5 AND nWBV $>$ 0.696 AND eTIV \leq 0.219 THEN 0
IF CDR \leq 0.25 AND MMSE $>$ 0.827 AND SES \leq 1.5 AND EDUC $>$ 15.0 AND MR Delay \leq 220.5 AND nWBV $>$ 0.696 AND eTIV $>$ 0.219 AND ASF \leq 1.113 AND eTIV \leq 0.605 THEN 0
IF CDR \leq 0.25 AND MMSE $>$ 0.827 AND SES \leq 1.5 AND EDUC $>$ 15.0 AND MR Delay \leq 220.5 AND nWBV $>$ 0.696 AND eTIV $>$ 0.219 AND ASF \leq 1.113 AND eTIV $>$ 0.605 AND nWBV \leq 0.722 AND eTIV \leq 0.702 THEN 0
IF CDR \leq 0.25 AND MMSE $>$ 0.827 AND SES \leq 1.5 AND EDUC $>$ 15.0 AND MR Delay \leq 220.5 AND nWBV $>$ 0.696 AND eTIV $>$ 0.219 AND ASF \leq 1.113 AND eTIV $>$ 0.605 AND nWBV \leq 0.722 AND eTIV $>$ 0.702 THEN 1
IF CDR \leq 0.25 AND MMSE $>$ 0.827 AND SES \leq 1.5 AND EDUC $>$ 15.0 AND MR Delay \leq 220.5 AND nWBV $>$ 0.696 AND eTIV $>$ 0.219 AND ASF \leq 1.113 AND eTIV $>$ 0.605 AND nWBV $>$ 0.722 THEN 1
IF CDR \leq 0.25 AND MMSE $>$ 0.827 AND SES \leq 1.5 AND EDUC $>$ 15.0 AND MR Delay \leq 220.5 AND nWBV $>$ 0.696 AND eTIV $>$ 0.219 AND ASF $>$ 1.113 AND Age \leq 70 AND Age \leq 65 THEN 1
IF CDR \leq 0.25 AND MMSE $>$ 0.827 AND SES \leq 1.5 AND EDUC $>$ 15.0 AND MR Delay \leq 220.5 AND nWBV $>$ 0.696 AND eTIV $>$ 0.219 AND ASF $>$ 1.113 AND Age \leq 70 AND Age $>$ 65 THEN 0
IF CDR \leq 0.25 AND MMSE $>$ 0.827 AND SES \leq 1.5 AND EDUC $>$ 15.0 AND MR Delay \leq 220.5 AND nWBV $>$ 0.696 AND eTIV $>$ 0.219 AND ASF $>$ 1.113 AND Age $>$ 70 THEN 1
IF CDR \leq 0.25 AND MMSE $>$ 0.827 AND SES \leq 1.5 AND EDUC $>$ 15 AND MR Delay $>$ 220.5 THEN 1
IF CDR \leq 0.25 AND MMSE $>$ 0.827 AND SES $>$ 1.5 AND nWBV \leq 0.793 AND nWBV \leq 0.731 AND nWBV \leq 0.73 AND Age \leq 85.5 THEN 1
IF CDR \leq 0.25 AND MMSE $>$ 0.827 AND SES $>$ 1.5 AND nWBV \leq 0.793 AND nWBV \leq 0.731 AND nWBV \leq 0.73 AND Age $>$ 85.5 AND Age \leq 86.5 THEN 0
IF CDR \leq 0.25 AND MMSE $>$ 0.827 AND SES $>$ 1.5 AND nWBV \leq 0.793 AND nWBV \leq 0.731 AND nWBV \leq 0.73 AND Age $>$ 85.5 AND Age $>$ 86.5 AND MR Delay \leq 758 THEN 1
IF CDR \leq 0.25 AND MMSE $>$ 0.827 AND SES $>$ 1.5 AND nWBV \leq 0.793 AND nWBV \leq 0.731 AND nWBV \leq 0.73 AND Age $>$ 85.5 AND Age $>$ 86.5 AND MR Delay $>$ 758 THEN 1
IF CDR \leq 0.25 AND MMSE $>$ 0.827 AND SES $>$ 1.5 AND nWBV \leq 0.793 AND nWBV $>$ 0.731 THEN 1
IF CDR \leq 0.25 AND MMSE $>$ 0.827 AND SES $>$ 1.5 AND nWBV $>$ 0.793 AND nWBV \leq 0.8 AND MR Delay \leq 347.5 AND Age \leq 68.5 THEN 0
IF CDR \leq 0.25 AND MMSE $>$ 0.827 AND SES $>$ 1.5 AND nWBV $>$ 0.793 AND nWBV \leq 0.8 AND MR Delay \leq 347.5 AND Age $>$ 68.5 THEN 1
IF CDR \leq 0.25 AND MMSE $>$ 0.827 AND SES $>$ 1.5 AND nWBV $>$ 0.793 AND nWBV \leq 0.8 AND MR Delay $>$ 347.5 THEN 1
IF CDR \leq 0.25 AND MMSE $>$ 0.827 AND SES $>$ 1.5 AND nWBV $>$ 0.793 AND nWBV $>$ 0.8 THEN 1
IF CDR $>$ 0.25 AND eTIV \leq 0.661 THEN 0
IF CDR $>$ 0.25 AND eTIV $>$ 0.661 AND Age \leq 80.5 THEN 0
IF CDR $>$ 0.25 AND eTIV $>$ 0.659 AND Age $>$ 80.5 AND MMSE \leq 0.75 THEN 0
IF CDR $>$ 0.25 AND eTIV $>$ 0.659 AND Age $>$ 80.5 AND MMSE \leq 0.75 THEN 1

Setelah diperoleh aturan linguistik di atas, dibuatlah proses uji mengukur keberhasilan algoritma C4.5 menggunakan teknik pengujian *percentage split* dan *k-fold cross validation* dengan *confusion matrix*. Berikut *confusion matrix* yang dihasilkan oleh algoritma C4.5 dengan seperti terlihat pada Tabel 3.

Tabel 3. *Confusion Matrix* pada C4.5

| <i>Percentage Split</i> | | | | <i>K-fold Cross Validation</i> | | | |
|-------------------------|---|----------|----|--------------------------------|---|----------|-----|
| Kelas | | Prediksi | | Kelas | | Prediksi | |
| | | 0 | 1 | | | 0 | 1 |
| Aktual | 0 | 28 | 4 | Aktual | 0 | 168 | 15 |
| | 1 | 5 | 38 | | 1 | 25 | 165 |

Berdasarkan Tabel 3 terlihat bahwa ada 75 data yang diprediksi untuk *percentage split* dan 373 data yang diprediksi untuk *k-fold cross validation*. Diketahui terdapat 28 data diprediksi secara benar sebagai kondisi 0 dan 4 data yang seharusnya masuk kondisi 0 tetapi diprediksi sebagai kondisi 1. Pada kondisi 1 ada 38 data yang diprediksi secara benar dan 5 data yang seharusnya masuk kondisi 1 tetapi diprediksi sebagai kondisi 0. Pada *confusion matrix* dengan teknik pengujian *k-fold cross validation* terdapat 168 data diprediksi secara benar sebagai kondisi 0 dan 15 data yang seharusnya masuk kondisi 0 tetapi diprediksi sebagai kondisi 1. Pada kondisi 1 ada 165 data yang diprediksi secara benar dan 25 data yang seharusnya masuk kondisi 1 tetapi diprediksi sebagai kondisi 0. Dimana dari hasil *confusion matrix* dengan menggunakan *percentage split* dan *k-fold cross validation* di atas dapat diketahui akurasi, presisi dan *recall* dari tiap-tiap kondisi yang dapat dilihat pada Tabel 4.

Tabel 4. Perbandingan *Percentage Split* dan *K-fold Cross Validation* pada C4.5

| Teknik Pengujian | Presisi (%) | <i>Recall</i> (%) | Akurasi (%) |
|--------------------------------|-------------|-------------------|-------------|
| <i>Percentage Split</i> | 87.5 | 88 | 88 |
| <i>K-fold Cross Validation</i> | 89.5 | 89.5 | 89.27 |

Berdasarkan Tabel 4 dapat diketahui pengujian dengan *k-fold cross validation* lebih baik dibandingkan dengan *percentage split*. Hal tersebut terlihat dari nilai presisi, *recall*, dan akurasi dari *k-fold cross validation* lebih tinggi daripada *percentage split*. Nilai presisi dengan *k-fold cross validation* 2% lebih baik daripada *percentage split*. Nilai *recall* dengan *k-fold cross validation* 1.5% lebih baik daripada hasil *recall* dengan *percentage split*. Pada nilai akurasi dengan menggunakan *k-fold cross validation* juga lebih baik dibandingkan *percentage split*.

3.2. Adaptive Boosting (AdaBoost)

Penerapan *AdaBoost* dan penggunaan dua teknik pengujian, yakni *percentage split* dan *k-fold cross validation* pada penelitian ini mendapatkan hasil perhitungan *confusion matrix* seperti yang terlihat pada Tabel 5.

Tabel 5. *Confusion Matrix* pada AdaBoost

| <i>Percentage Split</i> | | | | <i>K-fold Cross Validation</i> | | | |
|-------------------------|---|----------|----|--------------------------------|---|----------|-----|
| Kelas | | Prediksi | | Kelas | | Prediksi | |
| | | 0 | 1 | | | 0 | 1 |
| Aktual | 0 | 27 | 5 | Aktual | 0 | 165 | 18 |
| | 1 | 3 | 40 | | 1 | 15 | 175 |

Berdasarkan Tabel 5 dapat diketahui bahwa terdapat 75 data yang diprediksi pada pengujian menggunakan *percentage split*. Diketahui pula terdapat 373 data yang diprediksi menggunakan pengujian *k-fold cross validation*. Pada teknik pengujian *percentage split* bisa dilihat terdapat 27 data diprediksi secara benar sebagai kondisi 0 dan ada 5 data yang seharusnya masuk kondisi 0 tetapi diprediksi sebagai kondisi 1. Algoritma *AdaBoost* menggunakan *percentage split* sangat baik dalam memprediksi kondisi 1. Terdapat 40 data pada kondisi 1 yang diprediksi secara benar dan hanya ada 3 data yang diprediksi salah. Selanjutnya pada teknik pengujian *k-fold cross validation* dapat diketahui terdapat 165 data diprediksi secara benar sebagai kondisi 0 dan 18 data yang seharusnya masuk kondisi 0 tetapi diprediksi sebagai kondisi 1. Selain itu terdapat 175 data yang diprediksi benar sebagai kondisi 1. Kesalahan prediksi pada kondisi 1 adalah sebanyak 15 data, yang mana seharusnya 15 data tersebut masuk kondisi 1 namun diprediksi sebagai kondisi 0. Kemudian dari penerapan *percentage split* dan *k-fold cross validation* pada *AdaBoost* dapat diketahui nilai akurasi, presisi, dan *recall* dari setiap kondisi yang dapat dilihat pada Tabel 6.

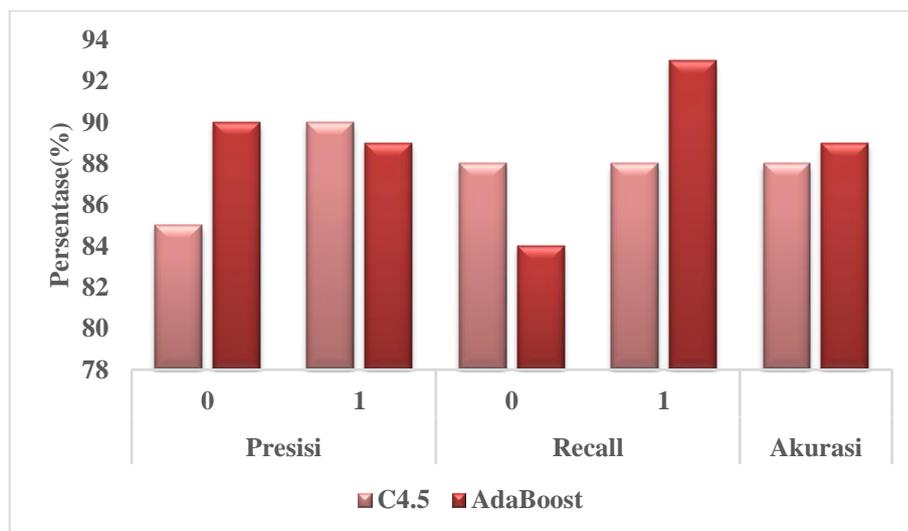
Tabel 6. Perbandingan *Percentage Split* dan *K-fold Cross Validation* pada *AdaBoost*

| Teknik Pengujian | Presisi(%) | Recall (%) | Akurasi (%) |
|--------------------------------|------------|------------|-------------|
| <i>Percentage Split</i> | 89.5 | 88.5 | 89 |
| <i>K-fold Cross Validation</i> | 91.5 | 91 | 91.15 |

Berdasarkan Tabel 6 dapat diperoleh informasi bahwa hasil kinerja pengujian dengan *k-fold cross validation* lebih baik dibandingkan dengan *percentage split*. Hal tersebut dapat dilihat dari nilai presisi, *recall* dan akurasi dari pengujian *k-fold cross validation* lebih tinggi dibandingkan dengan hasil presisi dan akurasi dari pengujian *percentage split*. Penggunaan teknik pengujian *k-fold cross validation* pada *AdaBoost* memperoleh nilai presisi yang lebih tinggi 2% dari hasil presisi dengan *percentage split*. Penggunaan teknik pengujian *k-fold cross validation* juga menghasilkan nilai *recall* yang lebih tinggi 2.5% dari hasil *recall* yang dihasilkan oleh *percentage split*. Nilai akurasi dari pengujian menggunakan *k-fold cross validation* menghasilkan nilai akurasi yakni sebesar 91.15% yang mana lebih tinggi dibandingkan nilai akurasi yang dihasilkan oleh pengujian *percentage split* yaitu sebesar 89%.

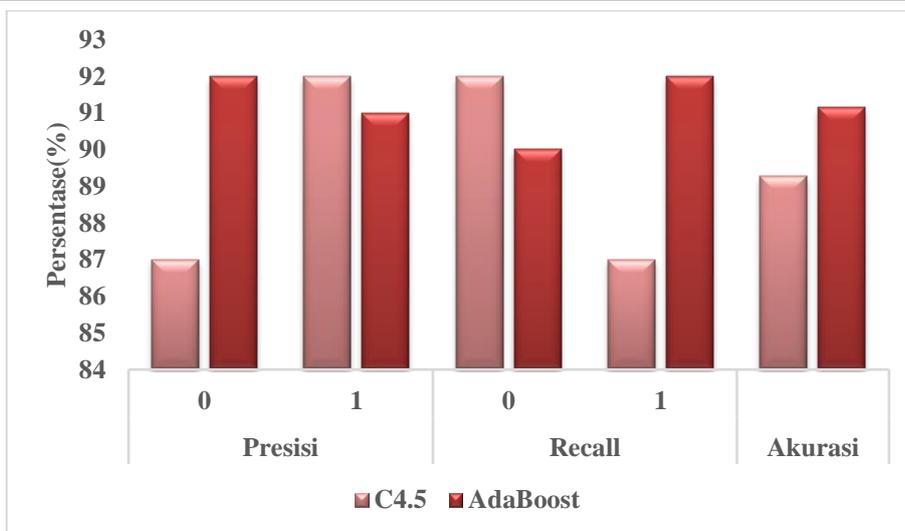
3.3. Perbandingan Hasil

Berdasarkan hasil dari klasifikasi algoritma C4.5 dan *AdaBoost* pada penyakit alzheimer menggunakan teknik pengujian *percentage split* dan *k-fold cross validation* diperoleh bahwa kedua algoritma tersebut bekerja dengan baik. Perbandingan hasil algoritma C4.5 dan *AdaBoost* dengan menggunakan teknik pengujian *percentage split* ditunjukkan pada Gambar 2.



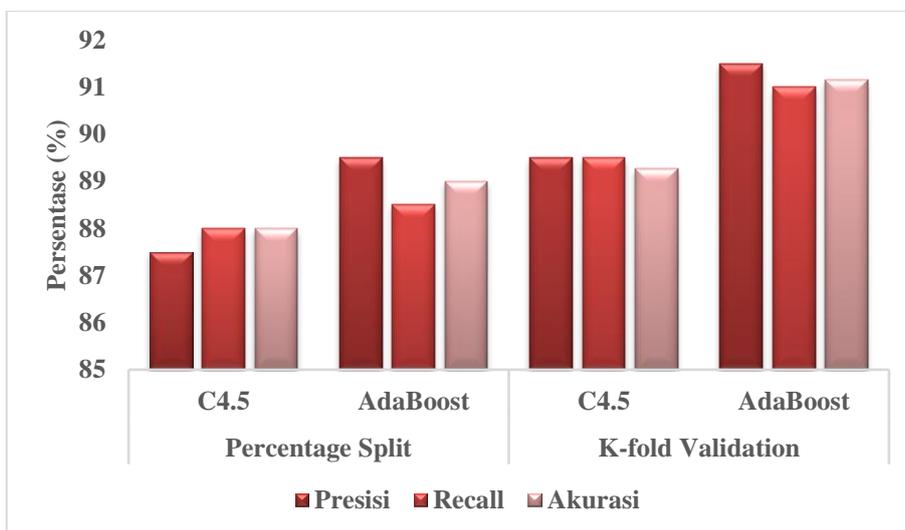
Gambar 2. Nilai Presisi, *Recall* dan Akurasi Hasil Prediksi C4.5 dan *AdaBoost* dengan Teknik *Percentage Split*

Berdasarkan Gambar 2 dapat diperoleh informasi bahwa dengan menggunakan teknik pengujian *percentage split*, algoritma *AdaBoost* pada kondisi 0 menghasilkan nilai presisi 5% lebih besar dari C4.5. Sebaliknya, pada kondisi 1 nilai presisi C4.5 lebih baik daripada *AdaBoost*. Pada kondisi 0 dengan algoritma C4.5 menghasilkan nilai *recall* 4% lebih baik dibandingkan dengan *AdaBoost*. Sementara nilai *recall* pada kondisi 1, *AdaBoost* memperoleh nilai *recall* yang lebih baik sebesar 5% dari C4.5. Pada Gambar 2, algoritma *AdaBoost* menghasilkan nilai akurasi yaitu 89% yang mana lebih besar dibandingkan dengan algoritma C4.5 yang mempunyai nilai akurasi sebesar 88%. Selanjutnya perbandingan hasil algoritma C4.5 dan *AdaBoost* dengan menggunakan teknik pengujian *k-fold cross validation* dapat dilihat pada Gambar 3.



Gambar 3. Nilai Presisi, *Recall* dan Akurasi Hasil Prediksi C4.5 dan *AdaBoost* dengan Teknik *K-fold Cross Validation*

Berdasarkan Gambar 3 dapat diketahui bahwa dengan menggunakan teknik pengujian *k-fold cross validation*, *AdaBoost* menghasilkan nilai presisi pada kondisi 0 yang lebih besar sejauh 5% dari C4.5. Pada kondisi 1, C4.5 dan *AdaBoost* menghasilkan nilai presisi yang hampir sama besarnya. Nilai *recall* yang dihasilkan pada kondisi 0 oleh algoritma C4.5 lebih baik 2% dibandingkan dengan *AdaBoost*. Sementara pada kondisi 1 *AdaBoost* menghasilkan nilai *recall* 5% lebih baik dari C4.5. Dengan menggunakan teknik *k-fold cross validation*, nilai akurasi yang dihasilkan oleh C4.5 adalah sebesar 89.27%. Algoritma *AdaBoost* menghasilkan nilai akurasi sebesar 91.15%. Sehingga pada teknik pengujian *k-fold cross validation* algoritma yang memiliki kinerja lebih baik adalah *AdaBoost*. Adapun nilai rata-rata presisi, *recall*, dan juga nilai akurasi dari seluruh kelas dengan menggunakan teknik *percentage split* dan *k-fold cross validation* ditunjukkan pada Gambar 4.



Gambar 4. Rata-rata Nilai Presisi, *Recall*, Akurasi dari C4.5 dan *AdaBoost* dengan Teknik *Percentage Split* dan *K-fold Cross Validation*

Pada Gambar 4 dengan menggunakan teknik *percentage split*, C4.5 mempunyai nilai rata-rata presisi sebesar 87.5% dan nilai rata-rata *recall* sebesar 88%. *AdaBoost* menghasilkan nilai rata-rata presisi sebesar 89.5% dan 88.5% untuk nilai rata-rata *recall*. Pada Gambar 4 dapat diketahui pula dengan menggunakan teknik *k-fold cross validation*, C4.5 memiliki nilai rata-rata presisi dan *recall* yang sama yaitu sebesar 89.5%. Kemudian *AdaBoost* menghasilkan nilai rata-rata presisi 91.5% dan *recall* sebesar 91%. Nilai akurasi yang dihasilkan kedua algoritma memiliki perbedaan yang tidak terlalu jauh. C4.5 menghasilkan nilai akurasi

sebesar 88% dengan teknik *percentage split* dan akurasi sebesar 89.27% dengan teknik *k-fold cross validation*. *AdaBoost* memiliki nilai akurasi sebesar 89% pada teknik *percentage split*. Penggunaan teknik *k-fold cross validation* pada algoritma ini menghasilkan nilai akurasi sebesar 91.15%.

Dengan demikian, berdasarkan hasil skor presisi, *recall*, dan akurasi dari masing-masing algoritma, performa yang lebih baik dihasilkan oleh *AdaBoost* dengan penggunaan *k-fold cross validation* pada penelitian perbandingan algoritma C4.5 dengan algoritma *AdaBoost* untuk mengklasifikasi penyakit alzheimer ini.

4. Kesimpulan

Berdasarkan penelitian yang telah dilakukan, dapat ditunjukkan bahwa penggunaan teknik pengujian *k-fold cross validation* mampu meningkatkan kinerja algoritma C4.5 dan *AdaBoost* dalam mengklasifikasi penyakit alzheimer. Hal ini dapat terlihat dari hasil kinerja oleh teknik *k-fold cross validation* yang lebih tinggi dibandingkan dengan hasil kinerja dari teknik *percentage split*. Untuk kinerja masing-masing algoritma dapat dilihat bahwa algoritma *AdaBoost* bekerja lebih baik dibandingkan algoritma C4.5. Hal ini dikarenakan hasil kinerja oleh *AdaBoost* selalu lebih tinggi persentasenya dibandingkan dengan C4.5, baik pada penggunaan teknik *percentage split* maupun *k-fold cross validation*. Dari hasil tersebut dapat disimpulkan bahwa penerapan algoritma *AdaBoost* dengan menggunakan *k-fold cross validation* merupakan metode yang paling tepat dalam mengklasifikasikan penyakit alzheimer. Hasil pemodelan yang telah dibangun pada penelitian ini nantinya bisa dikembangkan menjadi mesin pendeteksi dini otomatis penyakit alzheimer yang diharapkan dapat bermanfaat untuk pengguna khususnya bagi tenaga kesehatan.

Daftar Pustaka

- [1] Z. Breijyeh and R. Karaman, "Comprehensive Review on Alzheimer's Disease :," *Molecules*, vol. 25, p. 5789, 2020, doi: 10.3390/molecules25245789.
- [2] M. V. F. Silva, C. D. M. G. Loures, L. C. V. Alves, L. C. De Souza, K. B. G. Borges, and M. D. G. Carvalho, "Alzheimer's disease: Risk factors and potentially protective measures," *Journal of Biomedical Science.*, vol. 26, no. 1, pp. 1–11, 2019, doi: 10.1186/s12929-019-0524-y.
- [3] P. Scheltens *et al.*, "Alzheimer's disease," *The Lancet*, vol. 388, no. 10043, pp. 505–517, 2016, doi: 10.1016/S0140-6736(15)01124-1.
- [4] S. Hendrian, "Algoritma Klasifikasi Data Mining Untuk Memprediksi Siswa Dalam Memperoleh Bantuan Dana Pendidikan," *Faktor Exacta*, vol. 11, no. 3, pp. 266–274, 2018, doi: 10.30998/faktorexacta.v11i3.2777.
- [5] A. Cherfi, K. Noura, and A. Ferchichi, "Very Fast C4.5 Decision Tree Algorithm," *Applied Artificial Intelligence.*, vol. 32, no. 2, pp. 119–137, 2018, doi: 10.1080/08839514.2018.1447479.
- [6] M. F. Hasani and Sutikno, "Implementasi Algoritma C4.5 Untuk Klasifikasi Penyakit Infeksi Saluran Kemih Berbasis Web," *Jurnal Ilmiah Informatika Komputer.*, vol. 26, no. 3, pp. 233–246, 2021, doi: 10.35760/ik.2021.v26i3.4710.
- [7] A. Sapharni, I. E. Hendrawan, and C. Rozikin, "Klasifikasi Penyakit Jantung dengan Menggunakan Algoritma C4.5," *STRING (Satuan Tulisan Riset dan Inovasi Teknologi.*, vol. 7, no. 2, p. 117, 2022, doi: 10.30998/string.v7i2.12012.
- [8] Z. Gustiana, "Penerapan Algoritma C 4.5 Dalam Sistem Pendukung Keputusan Evaluasi Kinerja Fasilitator Pamsimas (Studi Kasus Di Kabupaten Kampar)," *Djtechno Jurnal Teknologi Informasi*, vol. 1, no. 1, pp. 20–28, 2021, doi: 10.46576/djtechno.v1i1.967.
- [9] Z. Fan, F. Xu, C. Li, and L. Yao, "Application of KPCA and AdaBoost algorithm in classification of functional magnetic resonance imaging of Alzheimer's disease," *Neural Computing and Applications.*, vol. 32, no. 10, pp. 5329–5338, 2020, doi: 10.1007/s00521-020-04707-y.
- [10] Z. Ghadiri Modarres, M. Shabankhah, and A. Kamandi, "Making AdaBoost Less Prone to Overfitting on Noisy Datasets," in *2020 6th International Conference on Web Research (ICWR)*, 2020, pp. 251–259, doi: 10.1109/ICWR49608.2020.9122292.
- [11] T. R. Mahesh *et al.*, "AdaBoost Ensemble Methods Using K-Fold Cross Validation for Survivability with the Early Detection of Heart Disease," *Computational Intelligence and Neuroscience*, 2022, doi: 10.1155/2022/9005278.
- [12] A. Byna and M. Basit, "Penerapan Metode Adaboost Untuk Mengoptimasi Prediksi Penyakit Stroke Dengan Algoritma Naïve Bayes," *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, vol. 9, no. 3, pp. 407–411, 2020, doi: 10.32736/sisfokom.v9i3.1023.
- [13] E. Listiana and M. A. Muslim, "Penerapan AdaBoost untuk Klasifikasi Support Vector Machine Guna Meningkatkan Akurasi pada Diagnosa Chronic Kidney Disease," in *Prosiding SNATIF (Buku 3)*, 2017, pp. 875–881.
- [14] O. Hornyák and L. B. Iantovics, "AdaBoost Algorithm Could Lead to Weak Results for Data with

- Certain Characteristics,” *Mathematics*, vol. 11, no. 8, p. 1801, Apr. 2023, doi: 10.3390/math11081801.
- [15] D. A. Nasution, H. H. Khotimah, and N. Chamidah, “Perbandingan Normalisasi Data untuk Klasifikasi Wine Menggunakan Algoritma K-NN,” *Computer Engineering Science and System Journal*, vol. 4, no. 1, p. 78, 2019, doi: 10.24114/cess.v4i1.11458.
- [16] V. R. Joseph and A. Vakayil, “SPlit: An Optimal Method for Data Splitting,” *Technometrics*, vol. 64, no. 2, pp. 166–176, 2022, doi: 10.1080/00401706.2021.1921037.
- [17] Y. A. Rindri and A. Fitriyani, “Analisis Perbandingan Kinerja Algoritma Multilayer Perceptron dan K-Nearest Neighbor pada Klasifikasi Tipe Migrain,” *Jurnal Teknologi dan Informasi (JATI)*, vol. 13, no. 1, 2023, doi: 10.34010/jati.v13i1.
- [18] N. P. Wong, F. N. S. Damanik, Christine, E. S. Jaya, and R. Rajaya, “Perbandingan Algoritma C4.5 dan Classification and Regression Tree (CART) Dalam Menyeleksi Calon Karyawan,” *Jurnal SIFO Mikroskil*, vol. 20, no. 1, pp. 11–18, 2019, doi: 10.55601/jsm.v20i1.622.
- [19] Normah, I. Yulianti, D. Novianti, M. N. Winnarto, A. Zumarniansyah, and S. Linawati, “Comparison of Classification C4.5 Algorithms and Naïve Bayes Classifier in Determining Merchant Acceptance on Sponsorship Program,” *Journal of Physics: Conference Series*, vol. 1641, no. 1, 2020, doi: 10.1088/1742-6596/1641/1/012006.
- [20] F. Husna, H. Rahman, and J. Juhari, “Implementasi Data Mining Menggunakan Algoritma C4.5 pada Klasifikasi Penjualan Hijab,” *Jurnal Riset Mahasiswa. Matematika*, vol. 2, no. 2, pp. 40–46, 2022, doi: 10.18860/jrmm.v2i2.14891.
- [21] V. Junita and F. A. Bachtiar, “Klasifikasi Aktivitas Manusia menggunakan Algoritme Decision Tree C4.5 dan Information Gain untuk Seleksi Fitur,” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 3, no. 10, pp. 9426–9433, 2020.
- [22] A. Bisri and R. S. Wahono, “Penerapan Adaboost untuk Penyelesaian Ketidakseimbangan Kelas pada Penentuan Kelulusan Mahasiswa dengan Metode Decision Tree,” *Journal of Intelligent System*, vol. 1, no. 1, pp. 27–32, 2015.
- [23] I. Dünsch and G. Gediga, “Confusion Matrices and Rough Set Data Analysis,” *Journal of Physics: Conference Series*, vol. 1229, no. 1, 2019, doi: 10.1088/1742-6596/1229/1/012055.
- [24] A. Desiani *et al.*, “Penerapan Metode Support Vector Machine Dalam Klasifikasi Bunga Iris,” *IJAI (Indonesian Journal of Applied Informatics)*, vol. 7, no. 1, pp. 12–18, 2022, doi: 10.20961/ijai.v7i1.61486.