Jurnal Pendidikan dan Konseling
Volume 4 Nomor 5 Tahun 2022
E-ISSN: 2685-936X dan P-ISSN: 2685-9351
Universitas Pahlawan Tuanku Tambusai

Pengembangan Aplikasi Berbasis Web dengan Python Flask untuk Klasifikasi Data Menggunakan Metode Decision Tree C4.5

Alan Chandra Darmawan[1], Lizda Iswari[2]
[1,2]Teknik Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia [1,2]
Email: 18523262@students.uii.ac.id[1], lizda.iswari@uii.ac.id[2]

Abstrak

Seiring perkembangan zaman, data merupakan salah satu alat yang dapat memberikan berbagai macam informasi berguna apabila dikelola dan diolah dengan baik dan benar. *Machine learning* adalah salah satu cabang ilmu olah data, yang dimana terdapat banyak algoritma yang dapat digunakan dalam mengolah data tergantung dari jenis dan tipe data yang digunakan. *Decision Tree C.45* adalah salah satu contoh algoritma yang mudah diimplementasikan dalam penggunaannya. Studi kasus telah dilakukan dengan mempelajari dan memahami aplikasi WEKA yang merupakan salah satu contoh dari aplikasi berkonsep *machine learning* yang terkemuka. Pada penelitian ini akan dikembangkan sebuah aplikasi yang menggunakan konsep *machine learning* berbasis *web*, dimana penggunaan algoritmanya difokuskan kepada algoritma *Decision Tree C.45*. Penggunaan algoritma ini akan menggunakan metode klasifikasi data dan metode *supervised learning* agar menghasilkan sebuah output berupa pohon keputusan dari file yang telah diunggah oleh pengguna. Sistem juga memberikan laporan klasifikasi dan juga matriks konfusi sebagai salah satu upaya penilaian dalam kinerja *machine learning* ini, agar nantinya hasil dari proses ini dapat dimanfaatkan sesuai dengan kepentingan dan kebutuhan bagi pengguna.

Kata Kunci: *Data Sains, Pembelajaran Mesin, Decision Tree C.45, Python Flask, Supervised Learning*

Abstract

Along with the times, data is one of the tools that can provide various kinds of useful information if it is managed and processed properly and correctly. Machine learning is a branch of data processing, where there are many algorithms that can be used to process data depending on the type and type of data used. Decision Tree C.45 is one example of an algorithm that is easy to implement in its use. Case studies have been carried out by studying and understanding the WEKA application, which is one example of a leading machine learning concept application. In this study, an application will be developed that uses the concept of web-based machine learning, where the use of the algorithm is focused on the Decision Tree C.45 algorithm. The use of this algorithm will use data classification methods and supervised learning methods in order to produce an output in the form of a decision tree from a file that has been uploaded by the user. The system also provides a classification report and also a confusion matrix as one of the evaluation efforts in machine learning performance, so that later the results of this process can be utilized according to the interests and needs of the user.

Keywords: *Data Science, Machine Learning, Decision Tree C.45, Python Flask, Supervised Learning*

PENDAHULUAN

Semenjak memasuki revolusi digital atau yang lebih dikenal dengan revolusi industri 4.0, segala sesuatunya kini telah beralih dan berganti menggunakan teknologi digital. Penggunaan mesin atau komputer juga meningkat seiring memasuki revolusi digital, mesin yang digunakan dilengkapi dengan kemampuan untuk menerima maupun mengumpulkan data serta memprediksi dan mengoreksi kesalahan secara mandiri. Hal ini tentu semakin memudahkan manusia untuk mendapatkan data, data sendiri dapat memberikan informasi yang berguna apabila dikelola dengan baik dan benar. Data yang baik belum menjamin terciptanya keputusan yang tepat [1], diperlukan algoritma yang tepat untuk mengolah data sesuai dengan tipe data yang dimiliki. Penelitian di *Science and Technology Studies* (STS) telah mempertanyakan klaim objektivitas dalam data [2], [3]. Data merupakan sesuatu yang sangat berharga apabila diolah menggunakan algoritma tertentu agar dapat berguna untuk memperoleh informasi. Ilmu olah data biasa disebut dengan *data science*.

Data science merupakan cabang ilmu olah data yang terdiri dari beberapa kombinasi ilmu lain, seperti statistika, matematika, ilmu komputer, ilmu informasi, manajemen, dan sistem informasi [3]. *Data science* sering digambarkan sebagai proses berbasis data yang rasional dari penemuan yang mengungkapkan sifat dasar dari suatu domain data [4]. *Data science* terdiri beberapa tahap seperti *data wrangling*, *modeling* dan penggunaan data. *Data wrangling* merupakan tahap mencari atau menghimpun data dari beberapa sumber, menyaring data untuk menyimpan data atau membuang data yang tidak penting, mengeksplor data termasuk memeriksa tipe data yang tersedia, karena setiap jenis tipe data properti membutuhkan perlakuan yang berbeda tergantung pada jenis tipe data, dan mengisi data kosong atau *missing values* proses ini berguna untuk meningkatkan akurasi data, pada tahap ini juga dilakukan standarisasi format data atau mengonversi data menjadi satu format yang sama. *Modeling* merupakan tahap dimana ketika membuat model data agar tercapainya tujuan yang sudah dirancang sebelum proses pengumpulan data. Tahap terakhir merupakan menggunakan data agar dapat membantu proses pembuatan keputusan. Salah satu cabang ilmu yang paling terkenal dari *data science* merupakan *machine learning*.

Machine learning merupakan sebuah cabang ilmu dari data science yang berfokus pada penggunaan data secara tepat dengan menggunakan algoritma dalam meningkatkan tingkat akurasi data. *Machine learning* memiliki peran yang sangat penting dalam pelaksanaan revolusi industri 4.0, peran *machine learning* ialah menyediakan sistem dengan kemampuan untuk belajar dan meningkatkan performa dari pengalaman secara otomatis tanpa diprogram secara khusus [5]. Menurut Alzubi, J., Nayyar, A. et al "Tergantung pada bagaimana suatu algoritma dilatih dan berdasarkan ketersediaan output saat dilatih, paradigma pembelajaran mesin dapat diklasifikasikan ke dalam sepuluh kategori, diantaranya *semi-supervised learning*, *supervised learning*, *unsupervised learning*, *hybrid learning*, *artificial neural network*, *ensemble learning*, *evolutionary learning*, *Instance-based learning*, *dimensionality reduction algorithms* dan *reinforcement learning*" [6]. Masing-masing kategori mempunyai kekurangan dan kelebihannya masing-masing.

Diantaranya *supervised learning*. *Supervised learning* adalah salah satu cabang dari *machine learning* yang membangun model prediktif dengan belajar dari sejumlah besar sampel pelatihan, dimana setiap sampel pelatihan memiliki label yang menunjukan *output*-nya [7]. Terdapat banyak algoritma yang termasuk di dalam *supervised learning*, seperti *Decision Tree*, *Linear Regression*, *Naïve Bayes*, *K-Nearest Neighbor* (KNN) dan *Support-Vector Machines* (SVM). Algoritma *Decision Tree* sendiri memiliki banyak jenis dan variasi, seperti *Iterative Dichotomiser* (ID3), *C.45*, *Classification and Regression Trees* (CART), dan *Random Forest*. Diantara beberapa pilihan algoritma *Decision Tree* di atas, Dalam jurnal yang ditulis oleh Arora, A., Gupta, B. et al telah menjelaskan kekurangan dan

kelebihan dari masing-masing algoritma *decision tree* [8]. Berikut beberapa keunggulan algoritma *Decision Tree C.45* antara lain mudah untuk diimplementasikan, dapat membuat model dengan tipe data kategorikal maupun tipe data kontinyu, mudah untuk di intrepretasikan dan dapat berhadapan dengan *missing values*.

Waikato Environment for Knowledge Analysist (WEKA) merupakan sebuah *software* atau aplikasi yang dikembangkan di Selandia Baru tepatnya di Universitas Waikato dan telah memiliki Lisensi Publik Umum GNU. Aplikasi ini sendiri berisi kumpulan algoritma *machine learning* untuk melakukan persiapan data, klasifikasi, regresi, pengelompokan, penambangan aturan asosiasi, dan visualisasi. Aplikasi ini juga dilengkapi dengan *user interface* yang memudahkan pengguna dalam melakukan akses ke segala fitur di dalam aplikasinya. Adapun penilitian ini, ditujukan untuk mengembangkan aplikasi berbasis *web* yang memiliki fitur utama berupa memberikan *output* yang berbentuk pohon keputusan dan mengklasifikasikan data dari dataset yang di unggah oleh pengguna. Aplikasi berbasis *web* yang bernama DTC45 ini akan mengklasifikasikan data menggunakan metode *Decision Tree C.45*.

Decision Tree C.45

Decision Tree C.45 merupakan sebuah algoritma yang dikembangkan oleh Ross Quinlan, algoritma ini juga sering dikatakan sebagai penerus dari algoritma Decision Tree ID3 (Iterative Dichotomiser) yang juga dikembangkan oleh Ross Quinlan. C.45 menghilangkan fitur harus kategorikal dengan mendefinisikan atribut diskrit secara dinamis (berdasarkan variabel numerik) yang mempartisi nilai atribut kontinu ke dalam set interval diskrit. Algoritma ini juga mengubah hasil luaran dari ID3 menjadi aturan if-then. Keakuratan setiap aturan kemudian akan dievaluasi untuk menentukan urutan penerapannya. Apabila jumlah kelas dilambangkan dengan C dan Pi adalah probabilitas yang terkait dengan kelas i. Maka nilai entropi dapat didefinisikan sebagai :

$$\text{Entropi} = \sum_{i=1}^{c} - \text{Pi} \times \log 2(\text{Pi})$$

Sedangkan nilai indeks gini dapat diperoleh dengan rumus :

$$\text{Gini} = 1 - \sum_{i=1}^{c} (\text{Pi})^2$$

Adapun algoritma C.45 terdiri dari beberapa langkah utama, yaitu :

1. Menghitung nilai entropi pada setiap atribut.
2. Menghitung nilai indeks gini masing-masing kelas.
3. Nilai indeks gini yang tertinggi akan menjadi nilai akar pertama dan seterusnya.
4. Ulangi langkah dari awal hingga tiap cabang menemukan ujungnya.

Setelah melakukan langkah-langkah di atas maka terbentuklah pohon keputusan yang akan menjadi hasil akhir dari algoritma ini. Pengujian akan dilakukan menggunakan data testing yang telah ditentukan sebelumnya.

True Positive (TP) = Terjadi ketika hasil prediksi sebuah observasi milik kelas tertentu dan observasi sebenarnya milik kelas itu.

True Negative (TN) = Terjadi ketika hasil prediksi sebuah observasi milik kelas tertentu dan observasi sebenarnya bukan milik kelas itu.

False Positive (FP) = Terjadi ketika hasil prediksi sebuah observasi bukan milik kelas tertentu tetapi observasi sebenarnya ternyata milik kelas itu. Kesalahan ini biasa disebut dengan eror tipe 1

False Negative (FN) = Terjadi ketika hasil prediksi sebuah observasi bukan milik kelas tertentu tetapi observasi sebenarnya milik kelas itu. Kesalahan ini biasa disebut dengan eror tipe 2

Pengujian berupa laporan klasifikasi dan matriks konfusi yang akan menampilkan beberapa poin sebagai berikut :

1. Precision

   Precision juga bisa dilihat sebagai tolak ukur ketepatan dalam melakukan klasifikasi. Untuk setiap atributnya, didefinisikan sebagai rasio true positive(TP) dengan jumlah dari true positive(TP) dan false negative(FN).

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

2. Recall

   Recall merupakan tolak ukur kelengkapan dalam melakukan klasifikasi. Kemampuan algoritma dalam menemukan semua contoh true positive. Untuk setiap atributnya, didefinisikan sebagai rasio true positive(TP) dengan jumlah true positive(TP) dan false negative(FN).

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

3. F1-Score

   F1-Score rata-rata tertimbang dari precision dan recall sehingga nilai tertingginya adalah 1.0 dan yang terburuk adalah 0. Secara umum, F1-Score memiliki nilai lebih rendah dari nilai accuracy karena menimbang recall dan precision pada perhitungannya.

$$\text{F1Score} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}$$

4. Support

   Support adalah jumlah kemunculan aktual kelas dalam dataset yang telah ditentukan. Support yang tidak seimbang dalam data training dapat menunjukan kelemahan struktural dalam skor algoritma pengklasifikasi yang dilaporkan dan dapat menunjukan perlunya pengembalian sampel bertingkat atau rebalancing.

Data Classification

Salah satu ciri utama dari algoritma supervised learning adalah data classification dan regression. Data classification yang dimaksud adalah dimana ketika hasil analisis telah diprediksi akan menghasilkan output yang telah diberikan label. Tujuan dari proses data classification sendiri untuk memprediksi kelas tujuan dengan presisi tertinggi. Data classification mencari hubungan antara atribut input dan output untuk membangun sebuah model melalui proses training [9].

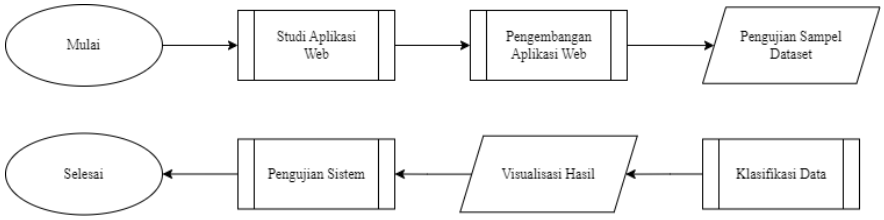
Bahasa Pemrograman Python

Python adalah bahasa pemrograman yang tercipta pada Desember 1989 oleh Guido Van memiliki tingkat bahasa bahasa yang tinggi. Python merupakan bahasa pemrograman yang cocok untuk tujuan machine learning. Python juga disebut sebagai bahasa yang mudah untuk dipelajari karena memiliki tingkat bahasa yang tinggi, ini juga membantu programmer untuk menyingkat waktu untuk mempelajari bahasa pemrogramannya [10]. Python sangat cocok untuk mengembangkan machine learning, ini terbukti dengan banyaknya library yang tersedia dalam bahasa pemrograman ini.

Python Flask

Flask menggunakan bahasa pemrograman python dan termasuk dalam jenis microframework yang dimiliki oleh bahasa pemrograman ini. Flask sendiri memiliki 3 dependensi, subsystems disediakan oleh Werkzeug, template-nya didukung oleh Jinja2, dan command-line integration dari Click [11]. Python Flask dirilis untuk pertama kali pada 1 April 2010 oleh Armin Ronacher.

METODE



Gambar 1. Diagram Alir Perancangan Sistem.

Studi Kasus Aplikasi

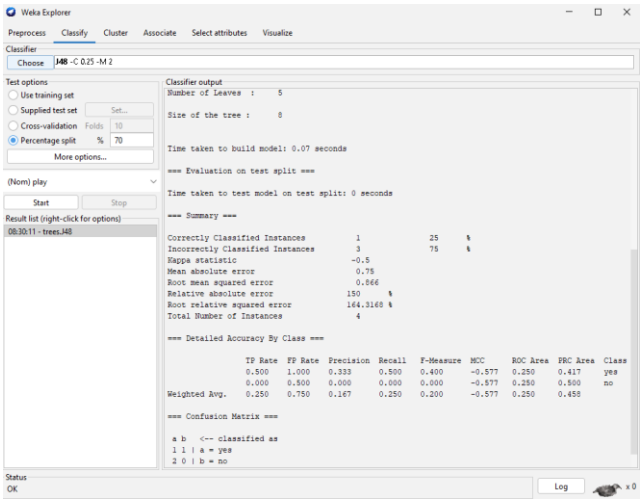
Langkah awal dalam memulai perancangan sistem, dimulai dengan tahap studi kasus dengan cara memahami dan mempelajari aplikasi serupa. Aplikasi desktop berbasis machine learning yang sebelumnya telah dibahas terlebih dahulu pada pendahuluan, yaitu Waikato Environment for Knowledge Analysist (WEKA). Weka merupakan sebuah tools yang berupa aplikasi desktop dimana pengguna dapat melakukan baik klasifikasi atau klasterisasi terhadap dataset yang dipilih oleh pengguna. Pada aplikasi ini tampilan awal berupa GUI.

Gambar 2. GUI WEKA

Setelah memilih fitur “Explorer” aplikasi akan membuka jendela baru secara sendirinya dan anda akan memasuki tahap preprocess pada aplikasi ini, pada tahap ini anda dapat mengunggah dataset yang anda inginkan. Lalu pada tahap classify anda dapat memilih algoritma classifier, terdapat banyak algoritma pilihan yang dapat anda pilih. Pada tahap classify terdapat “Test options” yang dimana anda dapat memilih bagaimana cara data yang anda miliki akan dilatih, terdapat empat cara yaitu menggunakan training-set, supplied test-set, cross-validation dan percentage split. Sedangkan di bawah kolom “Test options” terdapat pilihan kolom target(y) yang bisa anda pilih. Hasil dari proses klasifikasi dapat dilihat pada bagian kanan aplikasi, hasil yang akan ditampilkan berupa detil informasi proses klasifikasi, lama pembuatan model, summary (TP Rate, FP Rate, Precision, Recall, F-Measure, ROC Area dan PRC Area), detail akurasi berdasarkan kelas dan matriks konfusi.

Agar dapat lebih memahami aplikasi ini, maka dilakukan percobaan klasifikasi menggunakan algoritma decision tree c.45. Percobaan dilakukan terhadap dataset “weather-numeric.csv” yang terdiri dari 5 atribut dan 14 baris. Tahap classify akan menggunakan algoritma “J48” yang merupakan

nama algoritma decision tree c.45 yang terdapat di WEKA, pada pilihan “Test options” akan menggunakan percentage split sebesar 70%. Yang berarti dataset akan terbagi menjadi 70% data training dan 30 % data testing. Pada pilihan target(y) atau label output akan dipilih atribut “play” yang merupakan salah satu atribut kategorikal yang terdapat pada dataset ini. Berikut hasil klasifikasi seperti yang tertera pada gambar 3 di bawah ini.



Gambar 3. Klasifikasi data menggunakan WEK

Pengembangan Aplikasi Web

Tahapan berikutnya ialah membangun sebuah aplikasi berbasis web menggunakan Python Flask. Pada tahapan pengembangan Aplikasi berbasis web ini, akan dibagi menjadi 2 jenis script, yaitu :

1. UI

   Pada bagian User Interface(UI) merupakan bagian yang difokuskan kepada tampilan dari aplikasi berbasis web ini. Ada dua bahasa pemrograman yang digunakan pada tahap ini, HTML dan Javascript. Bahasa pemrograman HTML merupakan inti dari bagian ini, sedangkan penggunaan javascript hanya terdapat pada beberapa bagian. Pada script ini juga disesuaikan dengan aturan atau ketetapan dari Jinja2 yang merupakan salah satu dari 3 dependensi yang dimiliki oleh python flask.

2. Server

   Pada bagian server akan berisikan code dari seluruh proses yang akan dilakukan oleh aplikasi. Mulai dari proses input dataset oleh user, pemecahan data menjadi data training dan data test, pemilihan target(y) oleh user, pengklasifikasian data menggunakan algoritma decision tree c.45, dan visualisasi hasil menggunakan pohon keputusan.

3. Pengujian Sampel Dataset

   Sampel dataset yang akan digunakan dalam menguji sistem merupakann dataset yang sama pada saat melakukan studi kasus aplikasi WEKA. “weather-numeric.csv” memiliki 5 kolom yang terdiri dari 2 kolom diskrit dan 3 kolom kategori dan terdiri dari 14 baris di dalamnya. Dataset ini adalah dataset yang diberikan ketika anda mengunduh aplikasi WEKA.

4. Klasifikasi Data

   Tahap klasifikasi data adalah tahapan inti dari penilitian ini, karena pada tahapan ini yang menjadi penentu apakah aplikasi ini dapat bekerja dengan baik atau tidak. Pada tahap ini akan

dibuat menggunakan bahasa pemrograman python agar sistem dapat mengklasifikan data menggunakan algoritma decision tree c.45 yang nantinya hasil dari klasifikasi dapat digunakan sesuai dengan kepentingan pengguna.

5. Visualisasi Hasil

Tahap visualisasi data merupakan tahap dimana aplikasi akan menampilkan hasil dari proses klasifikasi yang telah dilakukan. Hasil klasifikasi akan ditampilakn menjadi 3 bagian :

a. Pohon Keputusan

Pohon keputusan merupakan sebagai salah satu cara untuk visualisasi terhadap proses klasifikasi yang dilakukan oleh algoritma decision tree. Pohon keputusan juga merupakan salah satu keunggulan bagi algoritma decision tree karena membuat hasil dari klasifikasi dapat lebih mudah dimengerti oleh pengguna.

b. Laporan Klasifikasi

Laporan klasifikasi yang akan ditampilkan dalam proses visualisasi meliputi precision, recall, F1-Score dan support. Detail dari laporan klasifikasi sendiri sebelumnya sudah dibahas pada tinjauan pustaka.

c. Matriks Konfusi

Konfusi matriks merupakan sebuah tool untuk meringkas kinerja yang dilakukan oleh sistem klasifikasi. Matriks konfusi akan memberikan gambaran tentang kinerja model klasifikasi dan jenis kesalahan yang dihasilkan.

d. Pengujian Sistem

Tahapan pengujian sistem menjadi tahapan terakhir yang dilakukan dalam merancang sistem. Tahapan pengujian ini akan dilakukan berulang kali guna mendapatkan hasil yang memuaskan. Apabila keseluruhan tahapan perancangan dapat berjalan dengan baik maka penelitian dapat dilanjutkan menuju tahap implementasi sistem.

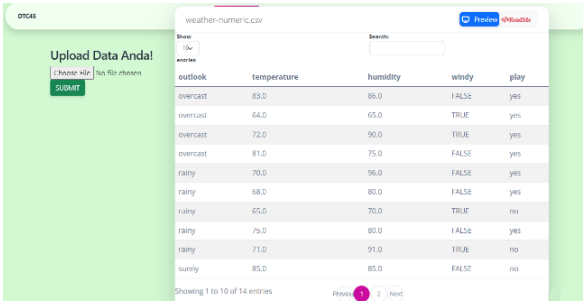
Implementasi Sistem

1. Tampilan Aplikasi

Tampilan pada aplikasi didesain sederhana dan informatif, ini bertujuan untuk memudahkan pengguna dalam menggunakan aplikasi berbasis web ini. Aplikasi ini memiliki navbar yang berisi home, upload data, dataframe, data training&set dan decision tree c.45. Mengenai penjelasan tentang aplikasi dan tujuan dibuatnya aplikasi ini terdapat pada halaman “Home” dan istilah-istilah matematis yang dibutuhkan oleh sistem juga diberikan penjelasan singkat pada setiap halamannya agar memudahkan pengguna.

2. Dataset

Terkait dataset yang diunggah oleh pengguna terdapat beberapa persyaratan yang diinginkan oleh sistem. Pertama dataset yang diunggah harus memiliki ekstensi file bertipe “.csv”, apabila pengguna mengunggah file yang memiliki ekstensi berbeda akan ditolak oleh sistem. Kedua file csv yang diunggah menggunakan comma separator, apabila pengguna mengunggah file csv menggunakan separator selain comma. Maka file akan tetap terunggah tetapi sistem akan membaca file tersebut hanya memiliki satu kolom saja. Untuk pengalaman yang lebih baik, sangat dianjurkan untuk mengunggah file csv yang menggunakan comma separator. Untuk meninjau terlebih dahulu file yang diunggah oleh pengguna yang terdapat pada halaman “Upload Data”.



Gambar 4. Tampilan Halaman "Upload Data".

3. Klasifikasi Data dengan Decision Tree C.45

Setelah selesai mengunggah dataset dan file diterima oleh sistem, sistem akan mengenali setiap tipe data yang ada terdapat di dataset. Hal ini ditujukan untuk memberikan prioritas kepada tipe data kategorikal sebagai pilihan dalam pemilihan target(y), apabila dalam dataset tidak terdapat kolom dengan tipe data kategorikal maka pilihan dialihkan kepada kolom dengan tipe data diskrit. Setelah pengguna melengkapi isian "traintestsplit" dan "target(y)". Maka sistem memecah dataset menjadi dataset x dan dataset y.

Dataset X merupakan seluruh kolom selain kolom yang terpilih pada pilihan target(y). Pemilihan target y ditujukan sebagai label(output) bagi dataset x(input). Traintestsplit akan memecah baik dataset x maupun dataset y sebesar "traintestsplit" terpilih untuk menjadi test size, sedangkan (1 – test size) akan menjadi train size. Langkah selanjutnya adalah melakukan encoding terhadap semua kolom kategorikal, apabila dalam dataset yang diunggah oleh pengguna tidak terdapat kolom kategorikal. Maka langkah ini akan diabaikan oleh sistem dan lanjut ke langkah berikutnya.

Langkah selanjutnya sistem membuat model pelatihan algoritma decision tree menggunakan kriteria indeks gini. Model tersebut selanjutnya dilatih. Proses pembagian data training dan data testing serta pembuatan model dan pelatihan model dilakukan dengan menggunakan library sckit-learn.

```
C45_model = tree.DecisionTreeClassifier(criterion = "gini")
C45_model.fit(X_train, y_train)
```

Gambar 5. Pseudocode pemodelan dan pelatihan model Decision Tree C.45.

Untuk membandingkan hasil klasifikasi, aplikasi DTC45 akan diuji menggunakan dataset yang sama seperti yang sudah dilakukan sebelumnya pada aplikasi WEKA. Dataset "weather-numeric.csv" berisi seperti pada gambar di bawah.

No.	1: outlook Nominal	2: temperature Numeric	3: humidity Numeric	4: windy Nominal	5: play Nominal
1	sunny	85.0	85.0	FALSE	no
2	sunny	80.0	90.0	TRUE	no
3	overcast	83.0	86.0	FALSE	yes
4	rainy	70.0	96.0	FALSE	yes
5	rainy	68.0	80.0	FALSE	yes
6	rainy	65.0	70.0	TRUE	no
7	overcast	64.0	65.0	TRUE	yes
8	sunny	72.0	95.0	FALSE	no
9	sunny	69.0	70.0	FALSE	yes
10	rainy	75.0	80.0	FALSE	yes
11	sunny	75.0	70.0	TRUE	yes
12	overcast	72.0	90.0	TRUE	yes
13	overcast	81.0	75.0	FALSE	yes
14	rainy	71.0	91.0	TRUE	no

Gambar 5. "weather-numeric.csv".

Pengujian dilakukan dengan variabel "traintestsplit" dan "target(y)" yang sama seperti ketika studi aplikasi. Variabel "traintestsplit" sebesar 0.3 dan "target(y)" berisikan atribut "play". Maka dataset akan terbagi menjadi 4 dengan rincian sebesar 70% data training dan 30% data testing, dataset X terdiri dari (outlook, temperature, humidity dan windy) dan dataset y tediri kolom "play". Serta kolom bertipe data kategorikal (kolom "windy" dan "play") sudah melewati tahap encoding. Pada tahap encoding kolom "windy" merubah nilai "false" menjadi 1.0 dan nilai "true" menjadi 2.0. Sedangkan kolom "play" merubah nilai "yes" menjadi 1 dan "no" menjadi 2.

Maka dataset akan terbagi menjadi seperti :

Data Training :

X				Y
Outlook	**Temperature**	**Humidity**	**Windy**	**Play**
1.0	69.0	70.0	1.0	1
1.0	80.0	90.0	2.0	1
1.0	72.0	95.0	1.0	1
1.0	75.0	70.0	2.0	1
2.0	83.0	86.0	1.0	1
2.0	64.0	65.0	2.0	1
3.0	71.0	91.0	2.0	2
3.0	68.0	80.0	1.0	2
3.0	70.0	96.0	1.0	2

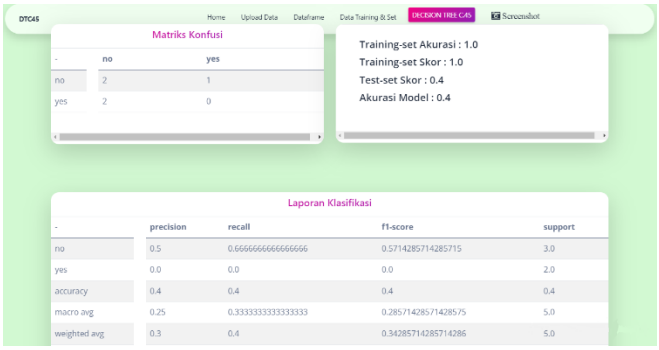
Data Testing :

X				Y
Outlook	**Temperature**	**Humidity**	**Windy**	**Play**
1.0	85.0	85.0	1.0	1
2.0	72.0	90.0	2.0	1
2.0	81.0	75.0	1.0	1
3.0	75.0	80.0	1.0	2
3.0	65.0	70.0	2.0	2

Selanjutnya sistem memulai pelatihan terhadap model data training dan hasil visualisasi klasfikasi dapat dilihat pada halaman "Decision Tree C.45".

4. Visualisasi Klasifikasi

Hasil dari klasifikasi akan divisualisasi dan dapat anda lihat pada halaman “Decision Tree C.45”. Visualisasi yang ditampilkan berupa diagram pohon keputusan menggunakan package graphviz. Pada halaman ini anda juga dapat melihat laporan klasifikasi dan juga matriks konfusi. Laporan klasifikasi akan menampilkan precision, recall, F1-Score dan support dari klasifikasi yang telah dilakukan. Terdapat juga matriks konfusi yang dapat dilihat oleh pengguna. Hal ini ditujukan agar pengguna dapat lebih memahami tentang klasifikasi yang telah dilakukan dan melihat hasil evaluasi performa dari model yang telah dilatih.



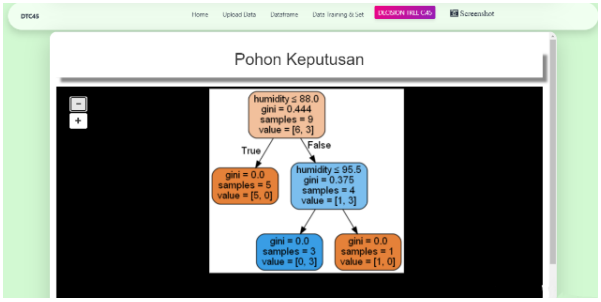
Gambar 6. Visualisasi Klasifikasi “weather-numeric.csv” menggunakan DTC45.

Komparasi Hasil

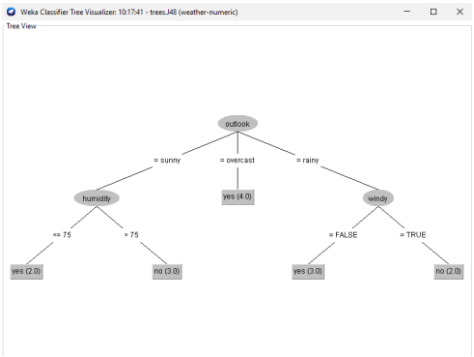
Aplikasi yang dikembangkan “DTC45” dan aplikasi yang menjadi bahan studi kasus WEKA (Waikato Environment for Knowledge Analysist), keduanya merupakan sebuah tools dimana pengguna dapat melakukan klasifikasi terhadap dataset yang dipilih oleh pengguna. Namun, kedua aplikasi ini juga memeliki perbedaan yang akan dijelaskan melalui tabel di bawah ini.

	DTC45	**WEKA**
Jenis Aplikasi	**Website**	**Desktop**
Machine Learning	Supervised Learning (Decision Tree C.45)	Supervised Learning dan Unsupervised Learning
Pengujian Klasifikasi	Traintestsplit	training set, supplied test set, cross-validation & percentage split
Ekstensi File Import	.csv	.names, .data, .csv, .arff,
Tampilan Data Training & Data Test	Ada	Tidak Ada

Perbedaan juga dapat ditemukan pada hasil visualisasi pohon keputusan diantara dua aplikasi ini.



Gambar 7. Pohon Keputusan DTC45.



Gambar 8. Pohon Keputusan WEKA.

Perbedaan hasil klasifikasi disebabkan oleh penggunaan library sckit-learn dalam melakukan pemodelan dan pelatihan model pada aplikasi DTC45. Pada library Sckit-learn menggunakan versi algoritma CART (Classification and Regression Tree) yang dioptimalkan. Namun, implementasi scikit-learn tidak mendukung variabel kategorikal untuk saat ini.

SIMPULAN

Dari penelitian yang telah dilakukan dan aplikasi berbasis *web* yang telah dikembangkan, maka dapat ditarik kesimpulan sebagai berikut : Aplikasi berbasis *web* yang dikembangkan dapat mengklasifikasi data dibangun menggunakan bahasa pemrograman *python* dan menggunakan *microframework python flask*. Klasifikasi data pada aplikasi yang dikembangkan difokuskan menggunakan algoritma *decision tree c.45* Proses klasifikasi dapat berjalan apabila pengguna mengikuti ketentuan sistem dalam *file* yang diunggah dan telah mengisi segala keperluan yang diperlukan. Hasil visualisasi klasifikasi data yang dilakukan oleh aplikasi dapat digunakan sesuai dengan kepentingan dan kebutuhan yang diinginkan oleh pengguna.

DAFTAR PUSTAKA

Muller, M., Lange, I., Wang, D., Piorkowski, D., Tsay, J., Vera Liao, Q., Dugan, C., & Erickson, T. (2019, May 2). How data science workers work with data. *Conference on Human Factors in Computing Systems - Proceedings*. https://doi.org/10.1145/3290605.3300356.

Gray, J., Gerlitz, C., & Bounegru, L. (2018). Data infrastructure literacy. Big Data & Soc. 5(2), 1-13.

Virkus, S. & Garoufallou, E. (2019), "Data science from a library and information science perspective", Data Technologies and Applications, Vol. 53 No. 4, pp. 422-441. https://doi.org/10.1108/DTA-05-2019-0076.

Kemper, J. & Kolkman, D. (2018). Transparent to whom? No algorithmic accountability without a critical audience. Info. Comm. & Soc. DOI: 10.1080/1369118X.2018.1477967.

Sarker, I. H., Kayes, A. S. M., Badsha, S., Alqahtani, H., Watters, P., & Ng, A. (2020). Cybersecurity data

science: an overview from machine learning perspective. Journal of Big Data, 7:41. https://doi.org/10.1186/s40537-020-00318-5.

Alzubi, J., Nayyar, A., & Kumar, A. (2018). Machine Learning from Theory to Algorithms: An overview. Journal of Physics: Conference Series, 1142(1). https://doi.org/10.1088/1742-6596/1142/1/012012.

Zhou, Z. H (2018). A brief introduction to weakly supervised learning. National Science Review, 5(1), 44-53. https://doi.org/10.1093/NSR/NWX106.

Arora, A., Gupta, B., Uttarakhand, P., & Rawat, I. A. (2017). Analysis of Various Decision Tree Algorithms for Classification in Data Mining Cite this paper Related papers Analysis of Classification Techniques in Data Mining. ijesrt journal Dat a Mining Applicat ion in Enrollment Management : A Case Study Saurabh Pal Analysis of Various Decision Tree Algorithms for Classification in Data Mining. In International Journal of Computer Applications (Vol. 163, Issue 8).

Charbuty, B., & Abdulazeez, A. (2021). Classification Based on Decision Tree Algorithm for Machine Learning. Journal of Applied Science and Technology Trends, 2(01), 20–28. https://doi.org/10.38094/jastt20165.

Ghimire, D. (2020). Comparative study on Python web frameworks: Flask and Django.

Grinberg, M. (2018). Flask Web Development: Developing Web Applications with Python - Miguel Grinberg - Google Books. Google Books. https://books.google.co.id/books?hl=en&lr=&id=cVlPDwAAQBAJ&oi=fnd&pg=PT25&dq=flask&ots=xOFShl4lcZ&sig=te5YR_qJLr7SMkAu7nraeQYvPxY&redir_esc=y#v=onepage&q=flask&f=false.