

**PENGEMBANGAN APLIKASI BERBASIS WEB DENGAN
PYTHON FLASK UNTUK KLASIFIKASI DATA
MENGUNAKAN METODE
DECISION TREE C4.5**



Disusun Oleh:

N a m a : Alan Chandra Darmawan

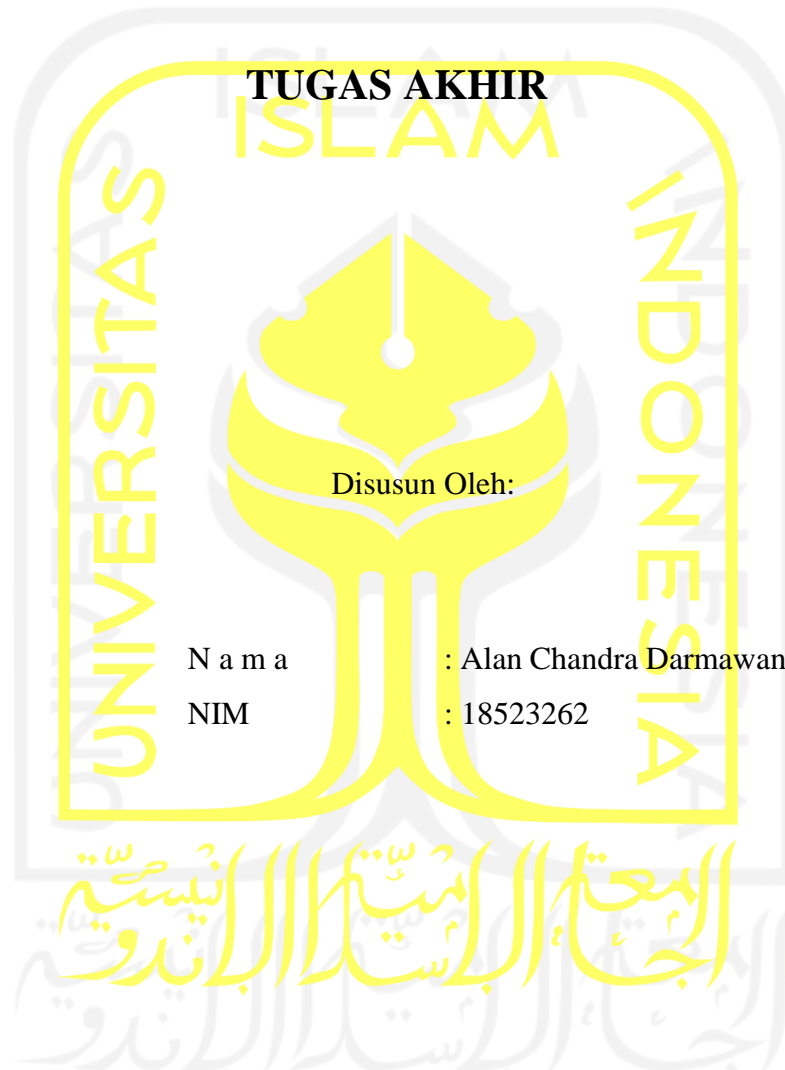
NIM : 18523262

**PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA**

2023

HALAMAN PENGESAHAN DOSEN PEMBIMBING

**PENGEMBANGAN APLIKASI BERBASIS WEB DENGAN
PYTHON FLASK UNTUK KLASIFIKASI DATA
MENGUNAKAN METODE
DECISION TREE C4.5**



Yogyakarta, 5 Januari 2023

Pembimbing,

(Lizda Iswari, S.T, M.Sc.)

HALAMAN PENGESAHAN DOSEN PENGUJI

**PENGEMBANGAN APLIKASI BERBASIS WEB DENGAN
PYTHON FLASK UNTUK KLASIFIKASI DATA
MENGUNAKAN METODE
DECISION TREE C4.5**

TUGAS AKHIR

Telah dipertahankan di depan sidang pengujian sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 5 Januari 2023

Tim Penguji

(Lizda Iswari, S.T, M.Sc.)

Anggota 1

(Chandra Kusuma Dewa, S.Kom., M.Cs.,
Ph.D.).

Anggota 2

(Elyza Gustri Wahyuni, S.T., M.Cs.).

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia



(Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Alan Chandra Darmawan

NIM : 18523262

Tugas akhir dengan judul:

PENGEMBANGAN APLIKASI BERBASIS WEB DENGAN PYTHON FLASK UNTUK KLASIFIKASI DATA MENGUNAKAN METODE DECISION TREE C4.5

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 5 Januari 2023



(Alan Chandra Darmawan)

HALAMAN PERSEMBAHAN

Dengan menyebut nama Allah yang maha pengasih lagi maha penyayang. Saya mempersembahkan segala sesuatu yang tertuang pada karya ini untuk,

Orang Tua

Kepada kedua orang tua saya yang selalu mendukung dan mengingatkan saya ketika berbuat salah sejak saat saya lahir hingga saat ini.

Kakak kandung

Kepada kakak kandung perempuan saya, sdri. Anthea Alviona Putri yang membantu saya terhadap segala permasalahan yang saya hadapi.

Dosen Pembimbing

Kepada ibu Lizda Iswari selaku dosen pembimbing penelitian saya yang sudah membimbing dan membantu saya sejak saat saya baru memulai penelitian hingga akhir.

Dosen Pembimbing Akademik

Kepada bapak Yudi Prayudi selaku Ddosen pembimbing akademik yang sudah membantu saya selama masa studi saya di UII.

Para Dosen Program Studi Teknik Informatika UII

Kepada seluruh bapak dan ibu dosen Teknik Informatika UII yang telah memberikan ilmu kepada saya baik ilmu akademik maupun non-akademik.

Teman-teman

Kepada seluruh teman-teman saya yang sudah membantu dan menghibur saya selama masa studi saya di UII.

Para Youtuber Akademik dari India

Saya berterimakasih karena sudah membantu saya menyelesaikan segala permasalahan terkait penyediaan materi yang membantu saya dan saya meminta maaf apabila terkadang saya lupa memberikan *subscribe* maupun menekan tombol *like*.

Kontributor Stackoverflow

Saya berterimakasih karena telah menyelesaikan permasalahan saya terkait *coding* dan terkait teori-teori yang telah banyak membantu saya selama masa studi saya di UII.

HALAMAN MOTO

“Dread it. Run from it. Destiny still arrives.”

(Thanos)

“Ilmu itu lebih baik daripada kekayaan, karena kekayaan itu perlu dijaga, sedangkan ilmu menjagamu.”

(Ali bin Abi Thalib)



KATA PENGANTAR

Dengan menyebut nama Allah yang maha pengasih lagi maha penyayang, dengan kalimat itu saya memulai kegiatan penelitian ini. Dan tiba saatnya penulisan akhir laporan penelitian sebagai tahap akhir dari penelitian ini. Puji dan syukur saya panjatkan kepada Allah SWT. Tuhan semesta alam berkat limpahan karunianya penelitian dengan judul “Pengembangan Aplikasi Berbasis Web dengan Python Flask Untuk Klasifikasi Data Menggunakan Metode Decision Tree C4.5” dapat berjalan dengan baik.

Memang benar adanya bahwa penelitian kali ini masih jauh dari kata sempurna. Banyak kendala maupun hambatan yang terjadi selama penelitian ini berlangsung. Namun, bukan menjadi masalah yang besar ataupun sampai mengakibatkan penelitian ini tidak dapat dilanjutkan. Penulis menganggap kendala atau hambatan tersebut menjadi sebuah tantangan yang harus dihadapi selama masa penelitian berlangsung. Dan tidak lupa saya ingin menyampaikan rasa terimakasih saya sebesar-besarnya kepada :

1. Kedua orang tua dan kakak kandung saya yang telah mendukung dan mendoakan saya hingga masa penelitian ini berakhir guna menuntaskan masa studi saya pada jenjang sarjana.
2. Ibu Lizda Iswari selaku dosen pembimbing saya yang membantu dan memandu saya dari masa awal penelitian hingga tahap yang paling akhir.
3. Kepada seluruh rekan atau teman-teman saya yang membantu dan juga menghibur saya selama masa studi saya.

Harapan saya sebagai penulis adalah bahwa laporan penulisan penelitian ini dapat membantu maupun memberikan *insight* kepada peneliti-peneliti dengan topik serupa. Semoga penelitian ini dapat berguna dan bermanfaat tidak hanya kepada saya sendiri melainkan juga kepada orang lain. Saya ucapkan terimakasih.

Yogyakarta, 5 Januari 2023

(Alan Chandra Darmawan)

SARI

Data merupakan sesuatu yang dapat memberikan informasi apabila dikelola dan diolah dengan baik dan benar, diperlukan sebuah *tools* agar dapat mengolah data. *Tools* yang dimaksud berupa sebuah alat olah data maupun cabang ilmu tentang data. *Data Science* ialah salah satu cabang ilmu yang mempelajari data dan yang mana di dalamnya terdapat banyak algoritma. Salah satu algoritma yang cukup terkenal adalah *decision tree c4.5*. Algoritma ini merupakan salah satu algoritma yang sering digunakan dalam melakukan klasifikasi data. Algoritma ini memiliki kelebihan dalam mudah diimplementasikan, mudah diinterpretasikan, dapat menangani *missing values* dan dapat menangani baik data kategorikal maupun diskrit. Selain itu diagram pohon keputusan yang merupakan salah satu hasil luaran utama pada algoritma ini adalah salah satu diagram yang dapat dengan mudah dipahami oleh manusia.

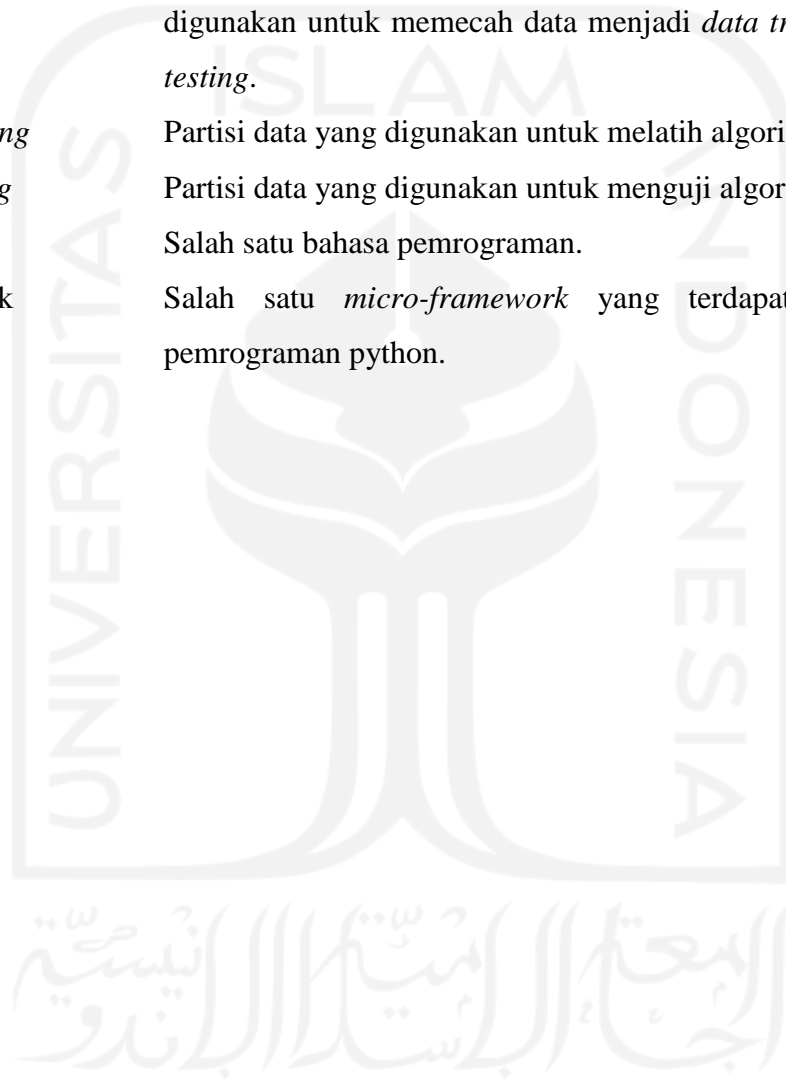
Masih terbatasnya pilihan aplikasi *machine learning* yang tersedia terutama pada aplikasi berbasis *web* menjadi salah satu alasan penelitian ini dilakukan. Hasil luaran penelitian ini adalah aplikasi *machine learning* berbasis *web* yang akan mengklasifikasi data dengan menggunakan metode *decision tree c4.5*. Aplikasi ini akan dikembangkan dengan *micro-framework* dari bahasa pemrograman python yaitu python flask. Pemilihan bahasa pemrograman python didasarkan pada bahasa pemrograman ini memiliki banyak *library* yang berhubungan dengan *machine learning*. Sedangkan python flask merupakan salah satu *micro-framework* yang terkemuka pada bahasa pemrograman python.

Aplikasi juga sudah melalui beberapa tahap pengujian seperti *black box testing*, *software testing* dan *user acceptance test*. Agar menguji aplikasi dari fungsional yang telah ditentukan, menguji aplikasi dengan aplikasi serupa yaitu WEKA dan menguji aplikasi langsung kepada pengguna. Diharapkan hasil luaran dari penelitian ini dapat dimanfaatkan oleh umum untuk melakukan klasifikasi terhadap data yang dimiliki dengan menggunakan metode *decision tree c4.5* berupa diagram pohon keputusan, visualisasi nilai klasifikasi, dapat lebih mengenali data yang dimiliki melalui tabel statistik deksriptif dan pengguna dapat mengunduh model pelatihan yang telah dilakukan. Sehingga hasil klasifikasi data tersebut dapat digunakan oleh pengguna sesuai dengan tujuan awal sebelum pengguna melakukan klasifikasi dengan aplikasi ini. Pengguna dapat mengakses aplikasi pada (alanchandrad.pythonanywhere.com).

Kata kunci: *data science*, *machine learning*, *supervised learning*, *decision tree c4.5*, klasifikasi data, python flask.

GLOSARIUM

<i>Data Science</i>	Cabang ilmu yang mempelajari tentang data.
<i>Machine Learning</i>	Salah satu cabang ilmu dari <i>data science</i> .
<i>Supervised Learning</i>	Salah satu teknik dari <i>machine learning</i> .
<i>Decision Tree</i>	Salah satu algoritma yang digunakan untuk mengklasifikasi data.
<i>Traintestsplit</i>	Salah satu library pada bahasa pemrograman python yang digunakan untuk memecah data menjadi <i>data training</i> dan <i>data testing</i> .
<i>Data Training</i>	Partisi data yang digunakan untuk melatih algoritma.
<i>Data Testing</i>	Partisi data yang digunakan untuk menguji algoritma.
Python	Salah satu bahasa pemrograman.
Python Flask	Salah satu <i>micro-framework</i> yang terdapat pada bahasa pemrograman python.



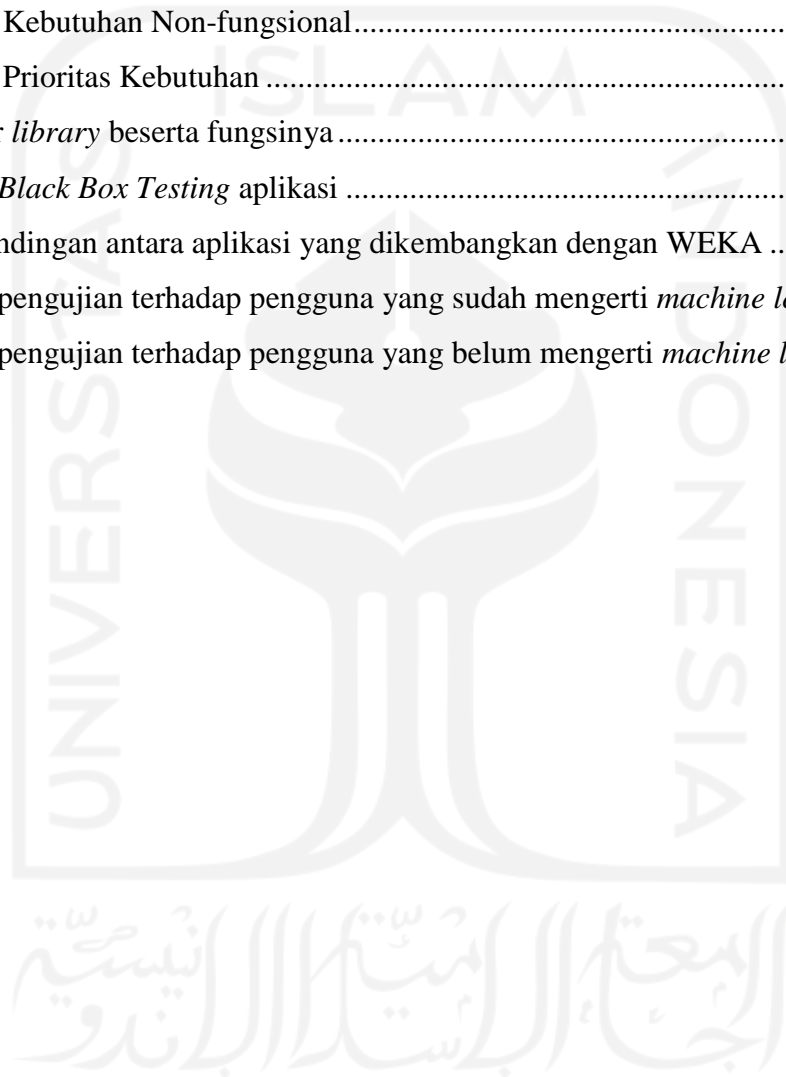
DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN DOSEN PEMBIMBING.....	ii
HALAMAN PENGESAHAN DOSEN PENGUJI.....	iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR	iv
HALAMAN PERSEMBAHAN	v
HALAMAN MOTO	vi
KATA PENGANTAR	vii
SARI.....	viii
GLOSARIUM.....	ix
DAFTAR ISI	x
DAFTAR TABEL	xii
DAFTAR GAMBAR.....	xiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	4
1.3 Batasan Masalah	4
1.4 Tujuan Penelitian	4
1.5 Manfaat Penelitian	5
1.6 Metodologi Penelitian	5
1.7 Sistematika Penelitian	6
BAB II LANDASAN TEORI	8
2.1 <i>Data Science</i>	8
2.2 <i>Machine Learning</i>	9
2.2.1 <i>Data training</i>	10
2.2.2 <i>Data testing</i>	10
2.3 <i>Supervised learning</i>	11
2.4 Klasifikasi Data.....	12
2.5 <i>Decision tree C4.5</i>	12
2.5.1 Kelebihan <i>Decision tree C4.5</i>	13
2.5.2 Kekurangan <i>Decision tree C4.5</i>	14
2.5.3 <i>Information gain</i>	14
2.5.4 Entropi	14
2.5.5 Pohon Keputusan.....	15
2.5.6 Contoh Perhitungan Manual.....	16
2.6 Bahasa Pemrograman Python	18
2.7 Python Flask.....	19
2.8 Pengujian Aplikasi	20
2.8.1 <i>Black Box Testing</i>	20
2.8.2 <i>Software testing</i>	21
2.8.3 <i>User acceptance testing</i>	21
2.9 Tinjauan Penelitian Serupa	21
BAB III METODOLOGI PENELITIAN	24
3.1 Subjek & Objek Penelitian.....	24
3.2 Indikator keberhasilan.....	24
3.3 Sumber Data.....	25
3.4 Metodologi Pengumpulan Data	25
3.4.1 Studi Literatur.....	25

3.4.2	Peninjauan <i>Dataset</i>	25
3.5	Metodologi Penelitian	26
3.5.1	Analisis Kebutuhan	27
3.5.2	Perancangan Sistem.....	38
3.5.3	Implementasi Sistem	44
3.5.4	Hasil Aplikasi	45
3.5.5	Pengujian Hasil.....	45
	BAB IV HASIL DAN PEMBAHASAN	46
4.1	Implementasi Hasil	46
4.1.1	Inisiasi Aplikasi	46
4.1.2	<i>Library Python</i>	49
4.1.3	Implementasi Kode Program.....	50
4.2	Hasil Aplikasi.....	52
4.2.1	Home	53
4.2.2	UploadData.....	54
4.2.3	Data Preprocessing	54
4.2.4	Dataframe	55
4.2.5	Data Training & Test.....	56
4.2.6	Decision Tree C4.5	57
4.3	Skenario Pengujian Hasil	58
4.3.1	Tahap Demonstrasi.....	58
4.3.2	Rangkuman Hasil	62
4.4	Hasil Pengujian	65
4.4.1	<i>Black Box Testing</i>	65
4.4.2	<i>Software Testing</i>	67
4.4.3	<i>User Acceptance Test</i>	74
4.5	Pembahasan.....	75
4.5.1	Home	75
4.5.2	Upload Data.....	75
4.5.3	Data Preprocessing	75
4.5.4	Dataframe	75
4.5.5	Data Training & Test.....	76
4.5.6	Decision Tree C4.5	76
	BAB V KESIMPULAN DAN SARAN.....	78
5.1	Kesimpulan	78
5.2	Saran.....	78
	DAFTAR PUSTAKA	79
	LAMPIRAN	84

DAFTAR TABEL

Tabel 2.1 Tabel dataset <i>Cause of Landslides</i>	16
Tabel 2.2 Tabel perhitungan entropi dan gain pada dataset.....	17
Tabel 2.3 Tabel perhitungan entropi dan gain pada atribut “Rain”	17
Tabel 2.4 Tabel perhitungan entropi dan gain pada atribut “Sloping”	18
Tabel 3.1 Tabel Kebutuhan Non-fungsional.....	36
Tabel 3.2 Tabel Prioritas Kebutuhan	37
Tabel 4.1 Daftar <i>library</i> beserta fungsinya	49
Tabel 4.2 Hasil <i>Black Box Testing</i> aplikasi	65
Tabel 4.3 Perbandingan antara aplikasi yang dikembangkan dengan WEKA	68
Tabel 4.4 Hasil pengujian terhadap pengguna yang sudah mengerti <i>machine learning</i>	74
Tabel 4.5 Hasil pengujian terhadap pengguna yang belum mengerti <i>machine learning</i>	74



DAFTAR GAMBAR

Gambar 1.1 Berbagai tipe dari teknik <i>machine learning</i>	2
Gambar 2.1 Langkah menemukan pengetahuan baru melalui <i>Data Science</i>	8
Gambar 2.2 <i>Supervised Learning</i>	11
Gambar 2.3 Struktur dari pohon keputusan	15
Gambar 2.4 Diagram pohon keputusan pada dataset "Cause of Landslides"	18
Gambar 3.1 Metode penelitian.....	27
Gambar 3.2 <i>Use case diagram</i> aplikasi	29
Gambar 3.3 Aplikasi Diagram.net	38
Gambar 3.4 Aplikasi Figma.....	39
Gambar 3.5 <i>Mockup</i> halaman Home	40
Gambar 3.6 <i>Mockup</i> halaman Upload Data	40
Gambar 3.7 <i>Mockup</i> halaman Data Preprocessing	41
Gambar 3.8 <i>Mockup</i> halaman Dataframe	41
Gambar 3.9 <i>Mockup</i> halaman Data Training & Test.....	42
Gambar 3.10 <i>Mockup</i> halaman Decision Tree C4.5	42
Gambar 3.11 <i>Activity diagram</i>	43
Gambar 3.12 <i>Class diagram</i>	44
Gambar 4.1 Membuka <i>folder</i> dengan Visual Studio Code.....	47
Gambar 4.2 <i>Install</i> virtualenv pada Visual Studio Code	47
Gambar 4.3 Membuat virtualenv dengan nama env	47
Gambar 4.4 Aktivasi virtualenv	47
Gambar 4.5 Menginstall Flask pada <i>terminal code editor</i>	47
Gambar 4.6 Membuat fail dengan nama "app.py"	48
Gambar 4.7 Contoh kode program.....	48
Gambar 4.8 Menjalankan aplikasi percobaan.....	48
Gambar 4.9 <i>Library</i> yang digunakan	49
Gambar 4.10 Kode pada halaman induk.....	51
Gambar 4.11 Penggunaan Javascript pada aplikasi	51
Gambar 4.12 Kode menampilkan halaman Data Preprocessing.....	52
Gambar 4.13 Hasil halaman Home	53
Gambar 4.14 Hasil halaman Upload Data	54
Gambar 4.15 Hasil halaman Data Preprocessing.....	54

Gambar 4.16 Hasil halaman Dataframe	55
Gambar 4.17 Hasil halaman Data Training & Test	56
Gambar 4.18 Hasil halaman Decision Tree C4.5	57
Gambar 4.19 <i>Dataset</i> "bukancsv.txt"	59
Gambar 4.20 <i>Dataset</i> "1Kolom.csv"	59
Gambar 4.21 <i>Dataset</i> "weather-numeric.csv"	60
Gambar 4.22 <i>Dataset</i> "diabetes.csv"	61
Gambar 4.23 <i>Dataset</i> "car_evaluation.csv"	62
Gambar 4.24 Notifikasi penolakan <i>dataset</i> "bukancsv.txt"	62
Gambar 4.25 Notifikasi penolakan terhadap <i>dataset</i> "1kolom.csv"	62
Gambar 4.26 Pilihan target(y) pada <i>dataset</i> "weather-numeric.csv"	63
Gambar 4.27 Diagram pohon keputusan pada <i>dataset</i> "weather-numeric.csv"	63
Gambar 4.28 Pilihan target(y) pada <i>dataset</i> "diabetes.csv"	64
Gambar 4.29 Diagram pohon keputusan pada <i>dataset</i> "diabetes.csv"	64
Gambar 4.30 Pilihan target(y) pada <i>dataset</i> "car_evaluation.csv"	65
Gambar 4.31 Diagram pohon keputusan pada <i>dataset</i> "car_evaluation.csv"	65
Gambar 4.32 Pohon keputusan "weather-numeric" pada DTC45	68
Gambar 4.33 Pohon keputusan "weather-numeric" pada WEKA	69
Gambar 4.34 Hasil Klasifikasi "weather-numeric" pada DTC45	69
Gambar 4.35 Hasil Klasifikasi "weather-numeric" pada WEKA	69
Gambar 4.36 Pohon keputusan "heart" pada DTC45	70
Gambar 4.37 Pohon keputusan "heart" pada WEKA	70
Gambar 4.38 Hasil Klasifikasi "heart" pada DTC45	71
Gambar 4.39 Hasil Klasifikasi "heart" pada WEKA	71
Gambar 4.40 Pohon Keputusan "drug200" pada DTC45	72
Gambar 4.41 Pohon Keputusan "drug200" pada WEKA	72
Gambar 4.42 Hasil Klasifikasi "drug200" pada DTC45	73
Gambar 4.43 Hasil Klasifikasi "drug200" pada WEKA	73

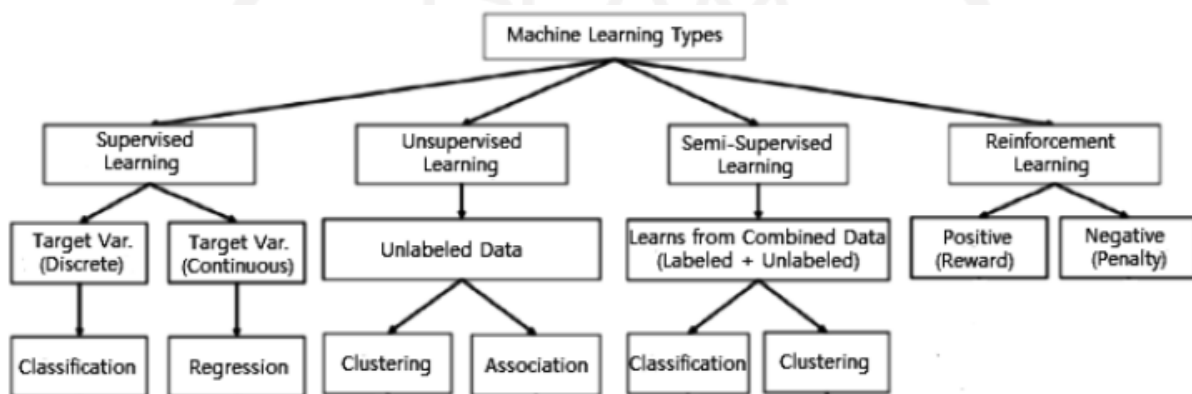
BAB I PENDAHULUAN

1.1 Latar Belakang

Sejak zaman dahulu hingga pada masa kini, data merupakan sesuatu yang dapat memberikan informasi yang berguna apabila pemiliknya dapat mengelola dan mengolah data tersebut dengan benar dan baik sesuai dengan jenis tipe data yang dimiliki oleh data tersebut. Namun, pada kenyataannya masih banyak orang yang tidak mampu memanfaatkan data yang dimiliki. Apabila data tidak dimanfaatkan, maka data hanya akan menjadi sesuatu yang memenuhi penyimpanan saja. Data yang baik saja tidak menjamin terciptanya keputusan yang tepat (Muller et al., 2019), diperlukan sebuah pemahaman tentang bagaimana cara untuk mengolah dan mengelola data yang dimiliki. Objektivitas dalam data dapat dipertanyakan (Gray et al., 2018), (Virkus & Garoufallou, 2019). Ditambah lagi dengan perkembangan zaman yang semakin lama semakin memudahkan kita untuk mendapatkan data. Oleh karena itu kebutuhan untuk pengolahan dan pengelolaan data juga semakin meningkat, diperlukan sebuah *tools* agar dapat mengolah data tersebut dengan baik dan benar. *Tools* yang dimaksud dapat berupa sebuah algoritma maupun cabang-cabang ilmu yang bersangkutan dengan pengolahan data. Salah satu cabang ilmu yang mempelajari data adalah *Data Science*. *Data Science* merupakan cabang ilmu yang mempelajari data yang terdiri dari beberapa kombinasi ilmu lain, seperti ilmu komputer, sistem informasi, matematika, manajemen, ilmu informasi, dan statistika (Virkus & Garoufallou, 2019) untuk menciptakan prediksi dan wawasan yang memiliki dampak untuk berbagai bidang ilmiah tradisional (Brodie, 2019) yang mengekstrak korelasi dengan perkiraan kemungkinan dan kesalahan (Brodie, 2015). Adapun penggiat dari cabang ilmu *data science* biasa disebut dengan *data scientist* atau statistikawan (Adhisyanda Aditya et al., 2020). Dalam pelaksanaannya para *data scientist* atau statistikawan tentu tidak lepas dalam penggunaan *machine learning* sebagai sebuah disiplin ilmu dalam praktiknya. *Machine learning* sendiri merupakan salah satu contoh dari cabang keilmuan AI (*Artificial Intelligence*).

Sejak pertama kali diperkenalkan pada tahun 1950 oleh ilmuwan matematika seperti Alan Turing yang menciptakan “Training Test” untuk memeriksa kecerdasan mesin yang dibuatnya (Alzubi et al., 2018), *machine learning* sudah berkembang dengan sangat cepat hingga saat ini. Peran dalam menyediakan sebuah sistem yang memiliki kemampuan secara

husus untuk belajar dan meningkatkan kemampuan dari pengalaman secara sendirinya tanpa diprogram secara khusus (Sarker et al., 2020) adalah kekuatan utama *machine learning* pada era sekarang. Maka dari itu *machine learning* diperkirakan dapat mengubah ilmu data untuk menjadi pemimpin dalam paradigma ilmiah ilmu data baru (Sarker et al., 2020). Bahkan tidak sedikit perusahaan besar seperti Google, IBM dan Microsoft yang berlomba mematenkan ilmu pada bidang ini. Sehingga saat ini dan diperkirakan akan terus berkembang seiring berjalannya waktu sejalan dengan kebutuhan mengenai algoritma-algoritma pembelejaran cerdas. Atas dasar inilah peneliti mengambil tema *machine learning* pada penelitian kali ini.



Gambar 1.1 Berbagai tipe dari teknik *machine learning*

Sumber: Sarker(2021)

Pada *machine learning* sendiri terdapat beberapa teknik yang dapat digunakan. Masing-masing teknik *machine learning* memiliki fungsi dan kegunaannya masing-masing. Berdasarkan Gambar 1.1, maka *machine learning* terbagi menjadi 4 tipe diantaranya *Reinforcement learning*, *Semi-Supervised learning*, *Unsupervised learning* dan *Supervised learning*. Diantara 4 teknik tersebut, supervised learning dan unsupervised learning merupakan yang paling terkenal diantaranya. *Unsupervised learning* memiliki sifat yang bertolak belakang dari *supervised learning*, dimana data tidak memiliki label dan sistem tidak mengetahui *input* dan *output* yang benar. Pada umumnya *unsupervised learning* dapat digunakan untuk *association*, *dimensionality reduction* dan *clustering*. Contoh algoritma *unsupervised learning* berupa K-Means, KNN, HCA, PCA dan LDA (Nath & Nema, 2020). Sedangkan *supervised learning*. Teknik *supervised learning* membangun model prediktif dengan cara belajar dari sejumlah besar sampel pelatihan, di mana setiap contoh pelatihan memiliki label yang menunjukkan *ground-truth output*-nya. Teknik pelatihan ini juga menggunakan data yang sudah berlabel yang dimana *input* dan *output*-nya sudah diketahui (Josse et al., 2019). Tipe pelatihan *supervised learning* memiliki beberapa tipe model seperti klasifikasi dan regresi.

Klasifikasi bertujuan untuk memilih *input* yang diterima menjadi beberapa kategori atau kelas berdasarkan label data yang sudah dilatih. Sedangkan model regresi diharapkan untuk menghasilkan hubungan numerik antara *input* dan *output*-nya. Serta terdapat beberapa algoritma yang biasa digunakan dalam *supervised learning* contohnya: *linear regression*, *logistic regression*, *neural networks*, *bayesian logic*, SVM, DT dan lainnya. Terutama pada algoritma *decision tree* (DT).

Decision tree merupakan algoritma sistem pendukung keputusan yang menggunakan grafik pohon keputusan yang berbentuk seperti pohon dan efek kemungkinan setelahnya termasuk peluang terjadinya kemungkinan, biaya sumber daya dan utilitas. Grafik pohon keputusan merupakan suatu hal yang sangat membuat algoritma *decision tree* sangat spesial, karena penggunaan grafik pohon keputusan ini sangat memudahkan dalam memahami hasil perhitungan dari algoritma *decision tree* itu sendiri. Algoritma *decision tree* memiliki berbagai macam jenis, seperti: ID3, C4.5, CART dan Random Forest. Masing-masing dari jenis algoritma *decision tree* memiliki kekurangan dan kelebihan masing-masing (Arora et al., 2017). Diantara berbagai macam jenis algoritma *decision tree* yang tersedia, algoritma *decision tree c4.5* atau yang bisa disebut j48 (Mathuria et al., 2013) merupakan salah satu dari algoritma *decision tree* yang terbaik. Hal ini disebabkan algoritma ini sangat mudah untuk diimplementasikan, dapat menangani *missing values* dan dapat menyederhanakan pohon keputusan untuk mengatasi *overfitting*.

Dari penjelasan yang telah dijabarkan di atas, maka pada penelitian ini akan mengembangkan sebuah aplikasi berbasis *web* sebagai sebuah alat atau *tools* yang dapat digunakan oleh pengguna sebagai pengklasifikasi data. Aplikasi berbasis *web* ini akan mengklasifikasi data menggunakan algoritma *decision tree c4.5* atau j48. Pengembangan aplikasi berbasis *web* ini akan menggunakan bahasa pemrograman python bersama dengan sebuah *micro-framework* bernama python flask. Penggunaan bahasa pemrograman python dalam pengembangan aplikasi berbasis *web* ini, dikarenakan pada bahasa pemrograman python terdapat banyak sekali *library* yang dapat membantu dalam pembuatan *machine learning* (Lee, 2019) dibandingkan dengan bahasa pemrograman yang lainnya. Sedangkan python flask merupakan salah satu dari beberapa *micro-framework* tersedia yang menggunakan bahasa pemrograman python. Flask juga merupakan *micro-framework* yang ramah digunakan untuk membangun sebuah *machine learning* (Ghimire, 2020).

Diharapkan hasil dari pengembangan aplikasi berbasis *web* ini dapat berguna untuk pengguna yang sudah memiliki pengetahuan mengenai *machine learning* maupun dapat

membantu pengguna yang belum memiliki pengetahuan tentang *machine learning* untuk mengklasifikasi data yang dimiliki menggunakan algoritma *decision tree c4.5* atau *j48*. Karena akan disertakan dengan bantuan atau petunjuk penggunaan aplikasi dan pengetahuan tentang kata-kata maupun istilah yang sering muncul di dalam aplikasi untuk lebih mudah dimengerti oleh pengguna.

1.2 Rumusan Masalah

Berdasarkan penjabaran yang telah dijelaskan pada latar belakang maka kita dapat menarik rumusan masalah pada penelitian ini berupa Bagaimana mengembangkan aplikasi berbasis *web* yang akan mengklasifikasi data dengan metode *decision tree c4.5* menggunakan *micro-framework* python flask ?

1.3 Batasan Masalah

Adapun ruang batasan masalah yang akan membatasi ruang lingkup dalam penelitian ini sebagai berikut :

- a. Aplikasi dirancang sebagai aplikasi berbasis *web* menggunakan *micro-framework* python flask dan menggunakan *library* yang tersedia pada bahasa pemrograman python.
- b. Aplikasi hanya dapat menerima file bertipe ekstensi “.csv” pada *input* datasetnya.
- c. Pada aplikasi yang dikembangkan tidak terdapat layanan *security* maupun keamanan pada data yang digunakan.
- d. Pada aplikasi yang dikembangkan tidak terdapat *user management*.
- e. Hasil akhir dari aplikasi berbasis *web* ini menggambarkan fungsionalitas aplikasi sebagai aplikasi *machine learning*.
- f. Bentuk hasil klasifikasi pada aplikasi berbasis *web* ini berupa diagram pohon keputusan, visualisasi nilai klasifikasi, matriks konfusi, laporan klasifikasi dan tersedia model pelatihan dalam *format* “.pkl”.

1.4 Tujuan Penelitian

Berdasarkan Rumusan Masalah yang telah tercantum, maka timbulah tujuan dari penelitian ini adalah mengembangkan suatu aplikasi berbasis *web* yang mampu menerima *input* berupa fail bertipe “.csv” sehingga dapat mengklasifikasi data, menampilkan visualisasi klasifikasi, memberikan prediksi yang berkaitan dengan data yang telah diunggah oleh

pengguna menggunakan algoritma *decision tree c4.5* dan membuat model pelatihan yang dapat diunduh oleh pengguna dalam format “.pkl”.

1.5 Manfaat Penelitian

- a. Dapat mengenali karakteristik data yang diunggah dengan lebih baik dengan melihat tabel statistik deskriptif data.
- b. Mendapatkan hasil klasifikasi menggunakan algoritma *decision tree c4.5* dari data yang diunggah.
- c. Dapat mempelajari mengenai algoritma *decision tree c4.5* melalui aplikasi berbasis *web* ini.
- d. Dapat mengunduh model pelatihan yang dilakukan melalui aplikasi yang dikembangkan dalam format “.pkl”.
- e. Memahami bagaimana cara mengembangkan aplikasi berbasis *web* dengan menggunakan *micro-framework* python flask.

1.6 Metodologi Penelitian

Perihal metodologi yang akan dilakukan pada penelitian kali ini agar tercapainya tujuan yang telah ditetapkan adalah sebagai berikut :

a. Analisis Kebutuhan

Tahap analisis kebutuhan akan mengawali kegiatan pada metode penelitian ini. Pada tahap analisis kebutuhan akan dilakukan dengan mencari informasi dan pengetahuan mengenai segala sesuatu yang dibutuhkan dalam mengembangkan sebuah aplikasi *machine learning* berbasis *web*, sehingga nantinya aplikasi dapat berjalan dengan baik sesuai dengan yang diharapkan. Segala kebutuhan materi dan pengetahuan yang didapatkan pada tahap ini akan menjadi landasan teori yang mendasari penelitian ini.

b. Perancangan Sistem

Perancangan sistem merupakan tahap kedua pada metode penelitian ini. Tahap ini dilakukan dengan membuat rancangan atau konsep yang akan dikembangkan selama penelitian ini. Rancangan yang akan dibuat berupa rancangan antarmuka sebagai *frontend* dan juga “Server” sebagai *backend*, rancangan dibuat mengikuti aturan-aturan yang ada pada *micro-framework* python flask. Pembuatan *mockup* aplikasi maupun diagram-diagram seperti *activity* diagram juga akan dilakukan pada tahap ini.

c. Implementasi Sistem

Tahap berikutnya adalah implementasi sistem yang juga menjadi tahap ketiga pada metode penelitian ini. Tahap ini merupakan tahap dimana segala rencana dan juga rancangan yang telah dibuat pada tahap sebelumnya dieksekusi dan diimplementasikan. Mulai dari pembuatan antarmuka aplikasi menggunakan bahasa pemrograman HTML maupun pembuatan aplikasi menggunakan *library* yang tersedia pada bahasa pemrograman python dan dibutuhkan oleh sistem.

d. Hasil Aplikasi

Tahap keempat adalah hasil aplikasi. Pada tahap ini dilakukan dokumentasi terhadap aplikasi yang sudah dikembangkan mulai dari antarmuka aplikasi sampai ke fitur-fitur aplikasi yang telah di buat pada tahap sebelumnya.

e. Pengujian Aplikasi

Pengujian aplikasi menjadi tahap terakhir pada metode penelitian ini. Tahap ini juga merupakan tahap yang paling penting pada seluruh tahapan yang telah dilakukan. Tahap ini merupakan tahapan penentu apakah aplikasi dapat berjalan seperti yang sudah dirancang atau tidak.

1.7 Sistematika Penelitian

Penyusunan penelitian ini dituliskan dengan terstruktur dan sistematis, agar memudahkan pembaca dalam memahami pembahasan yang dibahas pada penelitian ini dan memudahkan penulis dalam melaporkan kegiatan yang telah dilakukan. Adapun rancangan sistematika penulisan ini adalah sebagai berikut :

BAB I PENDAHULUAN

Bab satu akan menjelaskan tentang masalah yang diangkat dan melatar belakangi penelitian ini. Penjabaran ditulis menjadi beberapa subbab berupa rumusan masalah yang akan dihadapi, menetapkan tujuan masalah yang diambil, manfaat yang dapat diambil dari pengangkatan masalah, menentukan batasan masalah yang akan menjadi pembatas pada penelitian ini, penggambaran umum mengenai metodologi penelitian yang diambil dan sistematika penulisan.

BAB II LANDASAN TEORI

Pada bagian landasan teori berisi seputar tinjauan pustaka yang berasal dari penelitian-penelitian dalam bentuk jurnal, buku dan lain sebagainya yang telah ada dan relevan dengan permasalahan yang diangkat sehingga dapat menjadi sumber informasi dan referensi pada penelitian ini seperti *data science*, *machine learning*, *supervised learning*, klasifikasi data, *decision tree c4.5*, bahasa pemrograman python dan python flask.

BAB III METODOLOGI PENELITIAN

Pada metode penelitian ini akan dibahas mengenai segala sesuatu yang berkaitan dengan kebutuhan yang diperlukan untuk merancang sistem dimulai dari segi analisis kebutuhan, perancangan desain yang akan digunakan, pengumpulan data dan pengetahuan yang dibutuhkan, implementasi sistem yang berisi dengan cara kerja dari metode yang akan digunakan dan selanjutnya berisi tentang hasil dari pengujian sistem dan analisa dari hasil pengujian.

BAB IV HASIL DAN PEMBAHASAN

Pada bagian ini akan membahas terkait semua proses yang telah ditetapkan pada metodologi penelitian dari tahap analisa kebutuhan hingga tahap pengujian sistem serta tahap dokumentasi aplikasi yang dikembangkan.

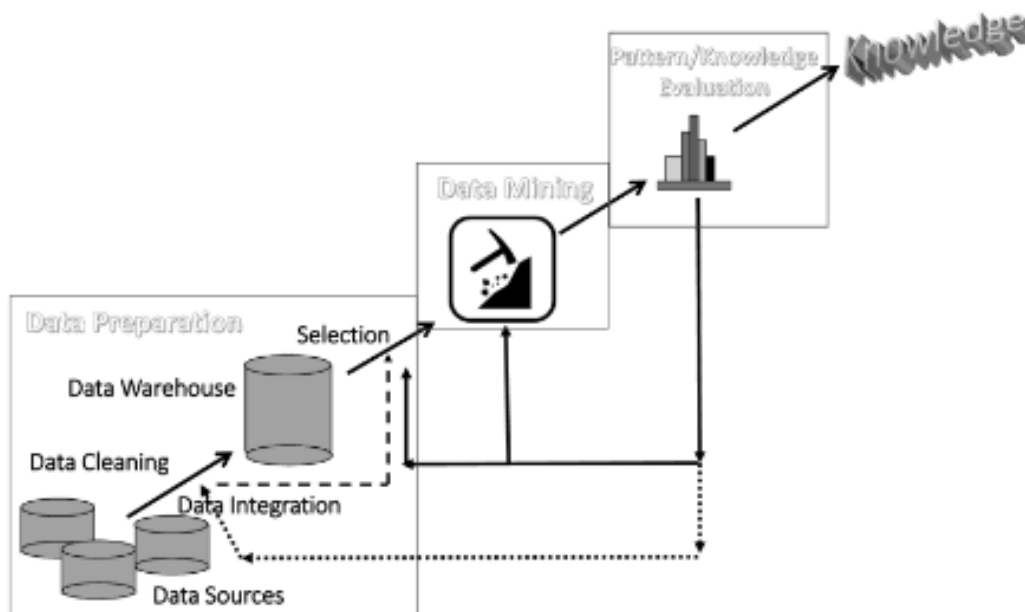
BAB V KESIMPULAN DAN SARAN

Bagian ini akan diisi dengan kesimpulan dari segala tahapan yang telah dilakukan pada penelitian ini dan juga berisi tentang saran yang dapat menjadi sumber referensi pada kegiatan penelitian berikutnya yang relevan dengan permasalahan yang diangkat pada penelitian kali ini.

BAB II LANDASAN TEORI

2.1 *Data Science*

Setelah membahas mengenai pengertian dari *data science* pada bab sebelumnya, kita dapat mengerti bahwasannya *data science* merupakan disiplin ilmu yang melakukan *preprocessing* dan analisis data dengan tujuan mengekstraksi pengetahuan yang berharga dalam data yang dimiliki. Membuat prediksi tentang fenomena tertentu dari data yang dimiliki juga merupakan salah satu tujuan dari *data science* (Rokach & Maimon, 2014). Diperlukan beberapa langkah maupun tahapan yang harus dilakukan agar hal tersebut bisa terjadi. Berdasarkan Gambar 2.1, langkah yang dapat dilakukan agar mendapatkan pengetahuan dari data yang dimiliki berupa : *Data preparation, data mining, evaluasi model & knowledge presentation*.



Gambar 2.1 Langkah menemukan pengetahuan baru melalui *Data Science*

Sumber: Han et al.(2022)

Langkah pertama merupakan *data preparation*, dalam tahap data preparation sendiri terdiri dari 4 langkah yang harus dilakukan. Pembersihan data, integrasi data, transformasi data dan seleksi data. Pembersihan data dilakukan untuk menghilangkan *noise* dan *incosistent* data, hal ini dilakukan dengan tujuan memperkecil ukuran data yang akan diproses dengan menghilangkan sebagian data yang dianggap tidak penting. Berikutnya integrasi data, langkah

ini dilakukan dengan menggabungkan data apabila sumber data yang dimiliki lebih dari satu. Selanjutnya transformasi data, dalam tahap ini data yang dimiliki akan dijadikan atau diubah menjadi format yang sesuai atau yang diinginkan untuk proses *data mining* dengan cara melakukan operasi *summary* maupun *aggregations*. Transformasi dan konsolidasi data biasanya dilakukan sebelum memasuki proses seleksi data, khususnya dalam *data warehousing*. *Data reduction* juga dapat dilakukan untuk mendapatkan data yang lebih kecil tanpa mengorbankan integritas atau yang dianggap *essential* yang dimiliki oleh data. Yang terakhir seleksi data, proses ini dilakukan dengan melakukan seleksi pada data dan menyisakan data yang relevan dengan tugas analisis yang akan dilakukan.

Langkah kedua ialah *data mining*, proses ini merupakan proses terpenting dimana teknik-teknik cerdas dilakukan untuk mengekstrak pola atau membangun model. Hampir semua teknik *data mining* bekerja dengan membangun model secara eksplisit maupun implisit dengan menggeneralisasi beberapa sampel pelatihan yang dirasa cukup. Hal ini merujuk pada pembelajaran induktif.

Langkah ketiga pada proses ini adalah evaluasi model, pada tahap ini mengidentifikasi pola-pola tertentu atau model yang mewakili *knowledge* yang terdapat dalam data. Pada proses ini biasanya dilakukan penilaian atau pengujian terhadap model yang sudah dibuat. Penilaian maupun pengujian yang dilakukan berbeda-beda tergantung dari pembelajaran yang dilakukan terhadap model.

Langkah terakhir merupakan *knowledge presentation*, setelah semua proses telah dilalui maka data yang sekarang sudah menghasilkan pengetahuan atau *knowledge* yang dapat digunakan sesuai dengan keinginan. Pengetahuan yang didapatkan dapat disajikan dalam bentuk visualisasi data maupun laporan-laporan.

2.2 *Machine Learning*

Machine Learning banyak digunakan untuk mengumpulkan pengetahuan dari sejumlah besar data. Aplikasinya terus meningkat dan berkisar dari persepsi mesin dan pemahaman teks hingga perawatan kesehatan, genomik, mobil cerdas, dan lain sebagainya. Mengingat sifat kritis dari beberapa aplikasi ini dan peran yang dimainkan *machine learning* dalam operasinya. (Sarker, 2021) menyatakan aplikasi berfungsi secara baik dikarenakan konteks analisis data dan juga komputasi. *Machine learning* pada umumnya mempunyai sistem dengan kemampuan untuk belajar dan meningkatkan kemampuan yang dimiliki secara otomatis berdasarkan pengalaman tanpa diprogram secara khusus. Terlepas dari algoritme *machine learning* yang

digunakan, kesalahan pada data dapat berdampak buruk pada kualitas model yang dihasilkan (Breck et al., 2019). Selanjutnya, sering terjadi bahwa prediksi dari model yang dihasilkan dicatat dan digunakan untuk menghasilkan lebih banyak data untuk pelatihan. Siklus umpan balik (*feedback*) semacam itu memiliki potensi untuk memperbesar kesalahan data "kecil" dan dapat menyebabkan regresi bertahap kinerja model yang dibangun selama periode waktu tertentu.

Ada beberapa jenis tugas yang dapat diselesaikan oleh *machine learning*, berikut klasifikasi tugas yang dapat diselesaikan oleh *machine learning* seperti klasifikasi, regresi, klusterisasi, *dimensional reduction*, asosiasi dan lainnya. Tentu tugas-tugas yang dapat dilakukan oleh *machine learning* ini memiliki ciri khasnya masing-masing. Hal ini tentu semakin menunjukkan kapabilitas *machine learning* dalam memecahkan permasalahan-pemmasalahan yang ada. Pada umumnya model *machine learning* dibagi menjadi dua yaitu *data training* dan *data testing*, memang tidak semua jenis algoritma pada *machine learning* memecah model menjadi *training* dan *testing*.

2.2.1 Data training

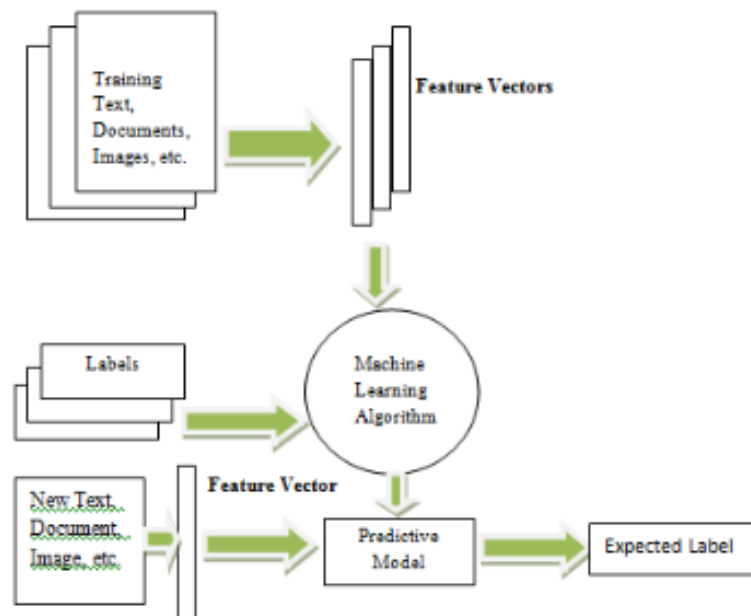
Data training merupakan bagian dari *dataset* yang digunakan untuk membuat prediksi atau untuk menjalankan segala fungsi dari algoritma dari *machine learning*. Dalam proses pelatihan, sampel dalam *data training* diambil sebagai *input* di mana fitur dipelajari oleh algoritma pembelajaran dan membangun model pembelajaran. Dengan memberikan petunjuk melalui algoritma agar mesin yang kita latih bisa mencari korelasinya sendiri melalui proses pelatihan yang dilakukan.

2.2.2 Data testing

Data testing merupakan bagian dari *dataset* yang digunakan untuk melakukan pengujian atau *testing* terhadap prediksi yang telah dilakukan. Untuk mengukur keakuratan maupun menimbang performa yang telah dilakukan. Dalam proses pengujian, model pembelajaran menggunakan mesin eksekusi untuk membuat prediksi untuk pengujian atau data produksi. Data yang diberi label adalah *output* dari model pembelajaran yang memberikan prediksi akhir atau data yang diklasifikasikan. Kumpulan data ini menguji model setelah proses pelatihan selesai dilakukan. Perlu diingat bahwa *data testing* adalah sampel-sampel yang menjadi representatif dengan masalah yang sedang dihadapi.

2.3 Supervised learning

Supervised learning memiliki ciri khusus yaitu data memiliki label yang menunjukkan hasil luarannya. Teknik *supervised learning* membangun model prediksi dengan melakukan kegiatan pembelajaran dari sejumlah besar *data training*, dimana setiap sampel pada *data training* tersebut sudah memiliki label yang menunjukkan hasil luarannya (Zhou, 2018). Di bawah *supervised learning*, satu set contoh atau modul *training* atau pelatihan disediakan dengan hasil luaran yang benar dan berdasarkan *data training* ini, algoritma belajar untuk merespons dengan lebih akurat dengan membandingkan hasil luarannya dengan yang diberikan sebagai *input*. *Supervised learning* juga dikenal sebagai belajar melalui contoh. Gambar 2.2 berikut menjelaskan konsepnya.



Gambar 2.2 *Supervised Learning*

Sumber: Alzubi et al.(2018)

Supervised learning menemukan aplikasi dalam prediksi berdasarkan data historis. Misalnya: sistem pengenalan yang menentukan apakah suatu objek dalam foto adalah mobil, manusia atau truk yang diberi gambar berwarna pada suatu objek melalui kamera, atau diberikan *e-commerce* riwayat penelusuran seseorang, rekomendasi produk oleh situs *web e-commerce*.

Tugas *supervised learning* dapat dikategorikan lebih lanjut sebagai tugas klasifikasi dan tugas regresi. Dalam kasus klasifikasi, label keluarannya adalah diskrit sedangkan pada kasus regresi label keluarannya adalah kontinu.

2.4 Klasifikasi Data

Tujuan utama dari klasifikasi data adalah untuk mengidentifikasi dan menetapkan kelas yang telah ditentukan sebelumnya menuju *instance* yang dipilih, ketika set pelatihan atau *instance* dengan label kelas diberikan. Metode klasifikasi adalah fitur pemrosesan data yang unik dari sebuah *machine learning* (Alpaydin, 2010) dan memungkinkan untuk menjalankan klasifikasi multi-kelas (Pranckevičius & Marcinkevičius, 2017). Tergantung pada jumlah kelas yang dimiliki oleh *output*, masalahnya bisa menjadi masalah klasifikasi biner atau multi-kelas. Klasifikasi berusaha untuk memprediksi kelas yang dituju dengan ketelitian atau akurasi yang paling akurat (Charbuty & Abdulazeez, 2021). Algoritma klasifikasi juga mencoba menemukan hubungan antara atribut luaran dan masukan untuk membuat sebuah model selama proses *training* (Patil & Kulkarni, 2019), (Ahmed & Hikmat Sadiq, 2018). Perbedaan mendasar antara melakukan klasifikasi dengan klasterisasi adalah pada masalah klasifikasi biasanya memiliki dasar pengkategorian sedangkan klasterisasi akan melakukan pengkategorian berdasarkan kemiripan-kemiripan yang dimiliki oleh datanya.

Masalah dalam klasifikasi biasanya di mana masalah output hanya dapat menjadi salah satu dari sejumlah kelas output yang diketahui apriori seperti Ya atau Tidak, benar dan salah, disebut masalah klasifikasi.

2.5 Decision tree C4.5

Decision tree C4.5 merupakan algoritma yang digunakan untuk menghasilkan pohon keputusan yang pertama kali dikembangkan oleh Ross Quinlan. Algoritma ini adalah perpanjangan dari algoritma ID3 (*Iterative Dichotomizer 3*) Ross Quinlan. *Decision tree C4.5* menghasilkan pohon keputusan yang dapat digunakan untuk klasifikasi dan oleh karena itu C4.5 sering disebut sebagai pengklasifikasi statistik. Algoritma ini lebih baik daripada algoritma ID3 karena mampu berurusan dengan atribut kontinu dan diskrit juga dengan nilai-nilai yang hilang atau *missing values* dan pemangkasan pohon setelah konstruksi. Ini mengarah pada pembentukan pohon yang lebih kecil, aturan yang lebih sederhana dan menghasilkan interpretasi yang lebih intuitif. Beberapa algoritma digunakan untuk mengimplementasikan pohon keputusan. Yang paling populer adalah: *Decision tree C4.5*, *Classification and Regression Tree* (CART), *Decision Tree C.50*, *Iterative Dichotomiser 3* (ID3), *Decision Tree M5* dan *Chi-Square Automatic Interaction Detection* (CHAID).

Decision tree membuat eksplisit dari semua alternatif yang mungkin dan menelusuri setiap alternatif ke kesimpulannya dalam satu tampilan, untuk memudahkan perbandingan di antara berbagai alternatif. Transparansi pada data adalah salah satu keuntungan terbaik dari *decision tree*. Keuntungan utama lainnya adalah kemampuan untuk memilih fitur yang paling bias dan sifatnya yang lebih mudah dipahami. Hal ini juga mudah untuk mengklasifikasikan dan dapat diinterpretasikan dengan mudah. Juga digunakan untuk kumpulan data kontinu dan diskrit. Penyaringan variabel dan bagian fitur cukup baik dalam *decision tree* (Rokach & Maimon, 2014). Dengan berbicara tentang kinerjanya, non-linier tidak mempengaruhi salah satu parameter dari *decision tree*. Berbicara tentang karakteristik *decision tree*. Pada algoritma CART (*Classification and Regression Tree*) dan *decision tree c4.5* memiliki karakteristik yang sama seperti yang dimiliki ID3. Perbedaannya hanyalah bahwa C4.5 dan CART keduanya dapat mengambil kumpulan data kontinu sebagai input untuk tujuan simulasi (Priyam et al., 2013). Algoritma *decision tree* digunakan untuk membagi atribut untuk diuji pada setiap node untuk menentukan apakah pemisahan adalah yang "Terbaik" di kelas individu. Partisi yang dihasilkan pada setiap cabang adalah kemungkinan yang "Murni", untuk itu kriteria pada pemisahan harus identik.

Decision Tree adalah teknik untuk mendekati fungsi target bernilai diskrit yang mewakili fungsi yang dipelajari dalam bentuk pohon keputusan (Shokri et al., 2017). Ukuran statistik seperti informasi gain, indeks gini dan entropi dihitung untuk setiap node untuk menghitung nilai node tersebut (Alzubi et al., 2018).

Sistem yang dapat menciptakan algoritma klasifikasi merupakan salah satu teknik yang banyak digunakan dalam *data mining* (bhatiya et al., 2012). Ketika berbicara *data mining*, algoritma klasifikasi memiliki kemampuan untuk meng-*handle* informasi dalam jumlah yang banyak. Hal ini juga dapat dipergunakan untuk membuat sebuah asumsi terkait dengan nama kelas kategorikal, untuk melakukan klasifikasi terhadap pengetahuan berdasarkan data training dan label kelas dan untuk melakukan klasifikasi data terbaru yang baru saja didapatkan (Nikam, 2015). Algoritma klasifikasi dalam *machine learning* terdiri dari beberapa algoritma di dalamnya dan pada makalah ini penggunaan algoritma difokuskan dengan pengguna *decision tree c4.5*.

2.5.1 Kelebihan *Decision tree C4.5*

1. Algoritma *decision tree c4.5* merupakan algoritma yang mudah untuk diimplementasikan.

2. Model yang dibuat pada algoritma *decision tree c4.5* dapat dengan mudah untuk diinterpretasikan.
3. Dapat dengan mudah menangani baik tipe data kategorikal dan kontinu.
4. Dapat menangani *noise* dan *missing values* pada data.

2.5.2 Kekurangan *Decision tree C4.5*

1. Sedikit perubahan pada data dapat berpengaruh besar ketika menggunakan algoritma *decision tree c4.5*.
2. Apabila *data training* terlalu kecil maka algoritma *decision tree c4.5* tidak bekerja terlalu baik.

2.5.3 *Information gain*

Information gain adalah salah satu metrik yang digunakan untuk segmentasi dan sering juga disebut sebagai *mutual information*. Ini secara intuitif memberi informasi seberapa banyak pengetahuan yang diketahui terhadap nilai variabel acak (Raileanu & Stoffel, 2004). *Information gain* berbanding terbalik dengan entropi, semakin tinggi nilainya maka akan semakin baik. Apabila S melambangkan himpunan, A melambangkan atribut, maka nilai *information gain* dapat diperoleh dengan :

$$\text{Information Gain (S, A)} = \text{Entropi(S)} - \sum_{i=1}^n \times \text{Entropi(S}_i) \quad (2.1)$$

Information gain memiliki bias terhadap atribut dengan multi-variasi yang merupakan kelemahan utama dari *information gain* (Zhang & Jiang, 2012). Pemisahan data yang tidak seimbang, dimana salah satu node anak memiliki lebih banyak entri dibandingkan dengan yang lain. Dimana ratio gain cenderung lebih baik terhadap situasi tersebut.

2.5.4 Entropi

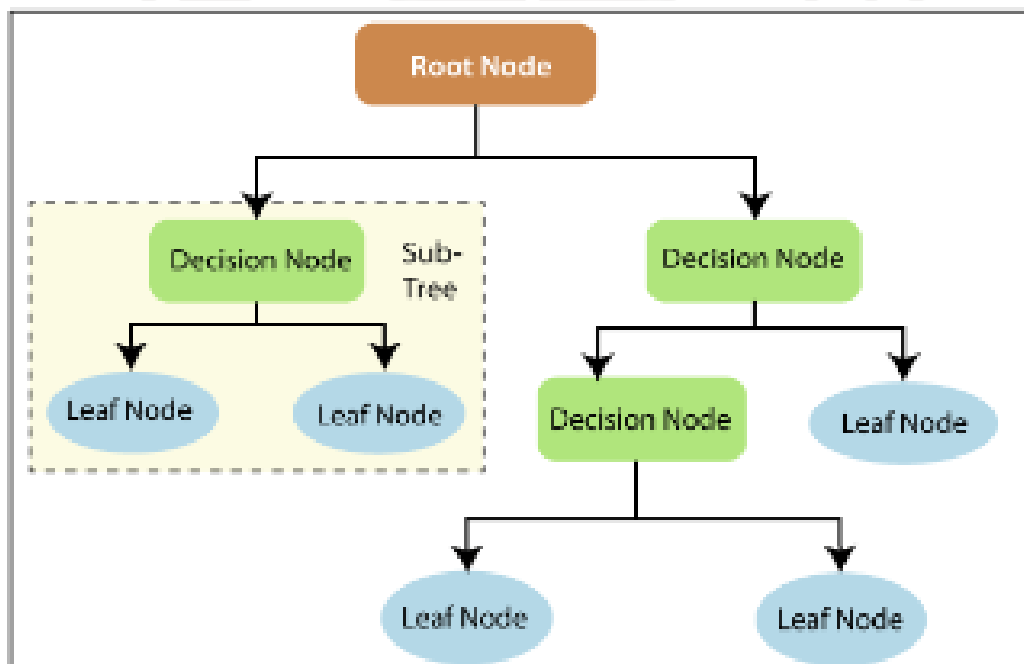
Entropi merupakan ukuran terhadap ketidakpastian yang memiliki keterkaitan dengan variabel acak. Nilai entropi meningkat selaras dengan meningkatnya ketidakpastian atau keacakan dan menurun dengan penurunan ketidakpastian atau keacakan. Nilai entropi berkisar antara 0 sampai 1.

$$\text{Entropi} = \sum_{i=1}^c - P_i \times \log_2(P_i) \quad (2.2)$$

Dimana P_i adalah rasio dari jumlah sampel pada *subset* dan nilai atribut ke-n sedangkan C melambangkan kelas. Fungsi log basis 2 digunakan karena seperti yang sudah dinyatakan di atas entropi dikodekan dalam bit 0 dan 1.

2.5.5 Pohon Keputusan

Pohon normal terdiri dari akar, cabang, dan daun. Struktur yang sama diikuti pada *decision tree*. Ini berisi simpul akar, cabang, dan simpul daun. Pengujian atribut pada setiap node internal, hasil pengujian pada label cabang dan kelas sebagai hasilnya adalah pada *node* daun (Gershman et al., 2010), (Jadhav & Channe, 2016). *Root node* adalah induk dari semua *node* dan seperti namanya itu adalah node paling atas di Pohon. Pohon keputusan adalah pohon dimana setiap simpul menunjukkan fitur (atribut), setiap mata rantai (cabang) menunjukkan keputusan (aturan) dan setiap daun menunjukkan hasil (nilai kategorikal atau berkelanjutan) (Jadhav & Channe, 2016). Karena pohon keputusan meniru pemikiran tingkat manusia sehingga sangat mudah untuk mengambil data dan membuat beberapa interpretasi yang baik. Seluruh idenya adalah membuat pohon seperti ini untuk seluruh data dan memproses satu hasil di setiap daun (Patel & Prajapati, 2018).



Gambar 2.3 Struktur dari pohon keputusan

Sumber: Charbuty & Abdulazeez(2021)

Pohon keputusan melakukan klasifikasi pada *instance* dengan mengurutkannya dari akar hingga ke beberapa simpul daun berdasarkan nilai fitur. Setiap *node* mewakili beberapa keputusan (kondisi pengujian) pada atribut *instance* sedangkan setiap cabang mewakili nilai yang mungkin untuk fitur itu. Klasifikasi sebuah *instance* dimulai pada simpul akar yang disebut simpul keputusan. Berdasarkan nilai *node*, pohon melintasi ke bawah sepanjang tepi yang sesuai dengan nilai output dari uji fitur. Proses ini berlanjut di sub-pohon yang dipimpin

oleh simpul baru di ujung tepi sebelumnya. Akhirnya, simpul daun menandakan kategori klasifikasi atau keputusan akhir. Saat menggunakan *decision tree*, fokusnya adalah pada bagaimana memutuskan atribut mana yang merupakan pengklasifikasi terbaik di setiap tingkat simpul.

2.5.6 Contoh Perhitungan Manual

1. Dataset

Pada percobaan pengklasifikasian data menggunakan algoritma *decision tree c4.5* ini akan menggunakan dataset yang didapatkan dari contoh penelitian serupa. Tabel 2.1 merupakan isi dari dataset *Cause of Landslides* (Handrianto & Farhan, 2019) sebagai berikut.

Tabel 2.1 Tabel dataset *Cause of Landslides*

Landslide Hazard Zone	Slope	Weather	Type of Soil	Cause of Landslides
High	Steep	Rain	Latosol	Yes
Middle	Sloping	Rain	Clay	Yes
High	Rather Steep	Rain	Latosol	Yes
Low	Sloping	Bright	Latosol	No
Middle	Steep	Rain	Clay	Yes
Middle	Steep	Bright	Clay	No
High	Rather Steep	Bright	Latosol	No
Middle	Steep	Rain	Clay	Yes
Middle	Sloping	Rain	Clay	No
High	Steep	Rain	Latosol	Yes
Middle	Sloping	Rain	Clay	Yes
Middle	Sloping	Rain	Latosol	No
Middle	Rather Steep	Rain	Clay	No
High	Steep	Rain	Latosol	Yes
Low	Rather Steep	Rain	Clay	No
Middle	Sloping	Rain	Latosol	No
High	Steep	Rain	Clay	Yes

2. Perhitungan Manual

Pada perhitungan manual kali ini akan menggunakan kolom “Cause of Landslides” sebagai label keluarannya. Dimana pada kolom ini hanya memiliki 2 unik nilai yaitu

“Yes” dan “No” yang menjadi acuan sebagai perhitungan entropi dan juga gain. Tabel 2.2 menunjukkan perhitungan entropi dan gain pada dataset ini.

Tabel 2.2 Tabel perhitungan entropi dan gain pada dataset

		Sum	Yes	No	Entropi	Gain
Total		17	9	8	0.998	
Landslide Hazard Zone	High	6	5	1	0.650	0.243
	Middle	9	4	5	0.991	
	Low	2	0	2	0	
Slope	Steep	7	6	1	0.591	0.238
	Rather Steep	4	1	3	0.811	
	Sloping	6	2	4	0.918	
Weather	Bright	3	0	3	0	0.610
	Rain	14	9	5	0.940	
Type of soil	Latosol	8	4	4	1	0.002
	Clay	9	5	4	0.991	

Berdasarkan perhitungan di atas dapat kita simpulkan bahwa kolom “Weather” menjadi akar dari pohon keputusannya, karena memiliki nilai gain yang lebih besar dibandingkan dengan kolom lainnya. Dan pada atribut “Bright” sudah menjadi salah satu akhir cabang (Daun) dikarenakan memiliki nilai entropi 0. Tabel 2.3 menunjukkan perhitungan selanjutnya.

Tabel 2.3 Tabel perhitungan entropi dan gain pada atribut “Rain”

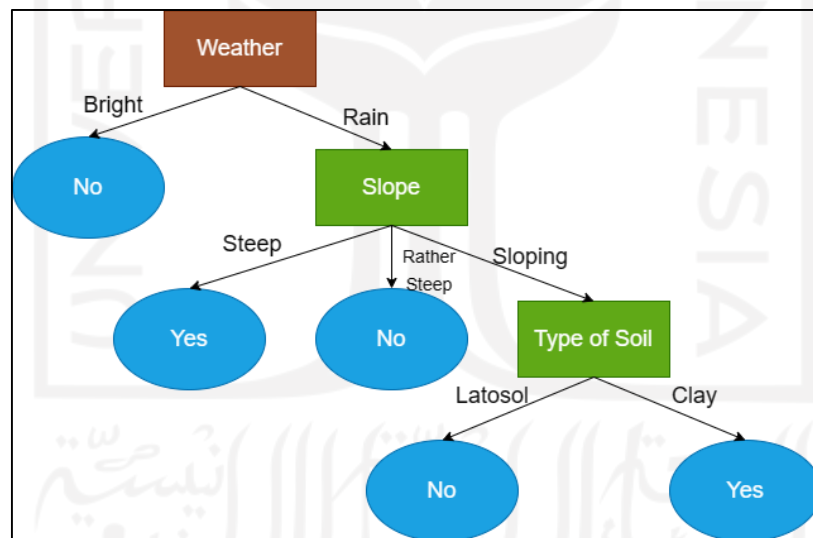
		Sum	Yes	No	Entropi	Gain
Total		14	9	5	0.940	
Landslide Hazard Zone	High	5	5	0	0	0.369
	Middle	8	4	4	1	
	Low	1	0	1	0	
Slope	Steep	6	6	0	0	0.396
	Rather Steep	3	1	2	0.918	
	Sloping	5	2	3	0.971	
Type of soil	Latosol	6	4	2	0.918	0.001
	Clay	8	5	3	0.954	

Berdasarkan perhitungan di atas dapat kita simpulkan bahwa kolom “Slope” menjadi cabang selanjutnya, karena memiliki nilai gain yang lebih besar dibandingkan dengan kolom lainnya. Dan pada atribut “Steep” sudah menjadi salah satu akhir cabang dikarenakan memiliki nilai entropi 0. Tabel 2.4 menunjukkan perhitungan selanjutnya.

Tabel 2.4 Tabel perhitungan entropi dan gain pada atribut “Sloping”

		Sum	Yes	No	Entropi	Gain
Total		5	2	3	0.971	
Landslide Hazard Zone	High	0	0	0	0	0
	Middle	5	2	3	0.970	
	Low	0	0	0	0	
Type of soil	Latosol	2	0	2	0	0.419
	Clay	3	2	1	0.918	

Maka berdasarkan perhitungan yang telah dilakukan, Gambar 2.4 menunjukkan diagram pohon keputusan dari dataset “Cause of Landslides” dengan menggunakan kolom “Cause of Landslides” sebagai label keluarannya.



Gambar 2.4 Diagram pohon keputusan pada dataset ”Cause of Landslides”

2.6 Bahasa Pemrograman Python

Didirikan pada Desember 1989 oleh Guido Van, python adalah salah satu bahasa pemrograman umum tingkat tinggi yang paling populer. Python adalah bahasa pemrograman yang diketik secara dinamis, sering disebut pengetikan bebek dalam rekayasa perangkat lunak (Grinberg, 2018). Ini adalah bahasa yang ditafsirkan. Karena, bahasa ini tidak memerlukan waktu kompilasi terpisah, melainkan dikompilasi di *byte* kode dan dieksekusi secara instan.

Salah satu alasan popularitas Python adalah karena keterbacaannya. Karena kode yang dapat dibaca adalah impian bagi seorang *software engineer*, itu penting karena pengembang harus dapat memahami pentingnya kode terlepas dari pembuatnya. Keterbacaan kode memang tergantung pada penulisnya. Tapi Python selangkah lebih maju dalam hal ini dengan sangat linguistik. Jenis bahasa pemrograman. Ini juga disebut-sebut sebagai cara mudah untuk mempelajari bahasa pemrograman. Ini sangat membantu *programmer* untuk mengurangi waktu yang dihabiskan untuk mempelajari bahasa pemrograman lain.

2.7 Python Flask

Python flask adalah *framework* yang terbilang kecil menurut sebagian besar standar bagi *framework*. Terbilang cukup untuk disebut "*micro-framework*" dan cukup kecil apabila kita sudah terbiasa dengan python flask. Kita dapat membaca dan memahami semua sumber kodenya.

Meski dapat terlihat kecil bukan berarti python flask bisa dianggap kurang dari *framework-framework* yang lain. Python flask dibangun sebagai *micro-framework* yang dapat diperluas dari bawah hingga ke atas. Itu berarti flask memiliki inti yang solid pada layanan dasarnya, sementara ekstensi yang dimilikinya menyediakan sisanya. Kita juga dapat memilih paket ekstensi yang kita inginkan, karena dapat memilih paket ekstensi yang kita inginkan. Maka kita tidak akan memiliki kelebihan ukuran fail yang akan menambah berat ukuran fail yang kita miliki.

Python flask mempunyai beberapa dependensi utama untuk mendukungnya yaitu *Routing*, *debugging* dan *Web Server Gateway Interface (WSGI)*. *Subsystems* python flask berasal dari "Werkzeug". *Support* untuk *template* yang dimiliki python flask didukung oleh "Jinja2" dan untuk *command-line integration* berasal dari "Click" (Grinberg, 2018). Inilah semua dependensi yang dimiliki oleh python flask yang semua dependensi utamanya juga dimiliki oleh Armin Ronacher selaku dari pencipta python flask.

Flask tidak memiliki dependensi utama atau bawaan pada sistem *database*-nya, pada validasi *web-forms*, *user authenticating*, dan *high-level tasks* lainnya. Layanan utama ini dan lainnya merupakan pelayanan utama yang biasanya terintegrasi dengan dependensi utama pada *framework* umum lainnya. Sebagai pengembang, pastinya kita memiliki rasa bimbang dalam memilih ekstensi yang cocok atau tepat dalam proyek yang kita bangun atau bahkan mungkin membuat sendiri apabila diperlukan. Ini tentu berbanding terbalik dengan *framework-*

framework lainnya yang dapat dikatakan besar, dimana sebagai besar pilihan yang telah kita buat terkadang sulit atau tidak mungkin untuk diubah lagi.

2.8 Pengujian Aplikasi

Pengujian aplikasi merupakan sebuah langkah pengujian atau pembuktian aplikasi yang telah dikembangkan terhadap hal-hal yang berkenaan dengan sesuatu yang diharapkan atau diminta pada aplikasi yang dibangun. Hal ini ditujukan sebagai upaya pembuktian untuk menunjukkan kualitas dari aplikasi yang dikembangkan. Langkah ini akan mempertanyakan apakah aplikasi yang sudah dikembangkan dapat berjalan sesuai dengan yang diharapkan atau tidak. Pengujian akan berlangsung sebagai berikut.

2.8.1 *Black Box Testing*

Black box testing juga biasa disebut pengujian fungsional, teknik pengujian fungsional yang merancang kasus uji berdasarkan informasi dari spesifikasi. Dengan *black box testing*, penguji perangkat lunak tidak boleh memiliki akses ke sumber internal kode itu sendiri. *Black box testing* juga tidak berkaitan dengan mekanisme internal suatu sistem. Pengujian ini hanya berfokus pada *output* yang dihasilkan sebagai respons terhadap *input* dan kondisi eksekusi yang dipilih (Liu & Kuan Tan, 2009).

Kode tersebut murni dianggap sebagai “*big black box*” bagi penguji yang tidak dapat melihat ke dalam “*box*”. Penguji perangkat lunak hanya mengetahui bahwa informasi dapat dimasukkan ke dalam *black box* dan *black box* akan mengirimkan sesuatu kembali (*feedback*). Hal ini dapat dilakukan murni berdasarkan pengetahuan spesifikasi kebutuhan. Penguji mengetahui apa yang diharapkan dari *black box* untuk dikirim dan menguji untuk memastikan *black box* mengirimkan apa yang seharusnya dikirim (Mitra et al., 2011).

Mengapa Pengujian *Black Box*? *Black box testing* memainkan peran penting dalam pengujian perangkat lunak, ini membantu validasi fungsionalitas sistem secara keseluruhan. *Black box testing* dilakukan berdasarkan kebutuhan pelanggan-sehingga persyaratan yang tidak lengkap atau tidak terduga dapat dengan mudah diidentifikasi dan dapat diatasi nanti. *Black box testing* dilakukan berdasarkan perspektif pengguna akhir. Kepentingan utama dari *black box testing* adalah menangani *input* yang valid dan tidak valid dari perspektif pengguna (Murnane & Reed, 2001).

Kapan menggunakan *Black Box Testing*? *Black box testing* dilakukan dari awal siklus pengembangan proyek perangkat lunak. Semua anggota tim pengujian harus dilibatkan sejak awal proyek dikerjakan. Selama tahap pengujian, penguji perlu dilibatkan dari fase

pengumpulan dan analisis kebutuhan pengguna. Pada tahap desain data uji dan skenario uji perlu disiapkan.

2.8.2 Software testing

Software testing adalah teknik yang paling sering digunakan untuk memverifikasi dan memvalidasi kualitas perangkat lunak (Shao et al., 2008). *Software testing* merupakan prosedur yang mengeksekusi program atau sistem dengan maksud dan tujuan untuk menemukan kesalahan didalamnya. Hal ini dapat diukur menjadi padat akan karya dan mahal, yang menyumbang lebih dari 50% dari total biaya pengembangan perangkat lunak (Sharma & Subhash Chandra, 2010). Pengujian perangkat lunak adalah aktivitas signifikan dari *Software Development Life Cycle* (SDLC). Ini membantu dalam mengembangkan kepercayaan pengembang bahwa suatu program melakukan apa yang dimaksudkan untuk dilakukan. Dengan kata lain, kita dapat mengatakan itu adalah proses mengeksekusi program dengan maksud untuk menemukan kesalahan.

Software testing merupakan area luas yang terutama terdiri dari bidang teknis dan non-teknis yang berbeda (Nidhra, 2012), seperti spesifikasi persyaratan, pemeliharaan, proses, desain dan implementasi dalam masalah manajemen dalam rekayasa perangkat lunak.

2.8.3 User acceptance testing

User acceptance testing (UAT) dapat didefinisikan sebagai pengujian “dari perspektif pengguna dan pemangku kepentingan lain untuk siapa (*software*) dibuat atau diperoleh”. Tujuan mengelola risiko, membangun kepercayaan, menilai proses bisnis sudah benar, dan memastikan bahwa perangkat lunak siap untuk dirilis. UAT mencoba mendemonstrasikan kemampuan fungsional awal hingga fungsional akhir dari sistem perangkat lunak. Tujuan utamanya adalah untuk mengevaluasi kesiapan sistem untuk penggunaan operasional. Kriteria penerimaan adalah ciri ataupun tanda bahwa suatu sistem atau komponen harus memenuhi syarat agar dapat diterima oleh pengguna, pelanggan, atau entitas berwenang lainnya (Suárez & Gómez, 2008).

Di akhir atau setelah UAT, tangkap pelajaran yang dipetik dan secara aktif bekerja untuk memasukkan peningkatan dalam sesi UAT di masa mendatang. Ini adalah salah satu perilaku kepercayaan inti.

2.9 Tinjauan Penelitian Serupa

Pada bagian ini akan membahas penelitian-penelitian yang serupa dan relevan dengan masalah yang diangkat pada penelitian kali ini. Masalah yang difokuskan pada tinjauan pustaka

ini adalah mengenai penelitian algoritma *decision tree c4.5* sebagai permasalahan utamanya, terlebih jika terdapat topik mengenai implementasi algoritma *decision tree c4.5* ke dalam aplikasi.

Kesimpulan akan disajikan untuk merincikan tinjauan pustaka ini dengan menggunakan tabel analisis yang dapat dilihat pada **Error! Reference source not found.** untuk mempermudah pembaca memahami hasil laporan penelitian kali ini.

Tabel analisis tinjauan pustaka

No	Judul	Studi Kasus	Sitasi
1	Metode Decision Tree Algoritma C.45 Dalam Mengklasifikasi Data Penjualan Bisnis Gerai Makanan Cepat Saji	Implementasi Algoritma C4.5 Dalam Kasus Data Penjualan Bisnis Gerai Makanan Cepat Saji.	(Cynthia & Ismanto, 2018)
2	C.45 Algorithm for Classification of Causes of Landslides	Implementasi Algoritma C4.5 Dalam Kasus Klasifikasi Penyebab Tanah Longsor.	(Handrianto & Farhan, 2019)
3	Utilization of Prediction Data for Prospective Decision Customers Insurance Using the Classification Method of C.45 and Naive Bayes Algorithms	Prediksi Klasifikasi Terhadap Keputusan Calon Asuransi Nasabah.	(Pasaribu et al., 2019)
4	Analisa Kinerja Algoritma C.45 Dalam Memprediksi Hasil Belajar	Implementasi Algoritma C4.5 Terhadap Prediksi Hasil Belajar.	(Ardiansyah Sembiring et al., 2018)
5	Classification Based on Decision Tree Algorithm for Machine Learning	Algoritma Klasifikasi berbasis <i>Decision Tree</i> .	(Charbuty & Abdulazeez, 2021)
6	Penerapan Data Mining Untuk Memprediksi Potensi Pendonor Darah Menjadi Pendonor Tetap Menggunakan Metode Decision Tree C.45	Implementasi Algoritma <i>Decision Tree C4.5</i> Pada Kasus Potensi Pendonor Darah.	(Atmaja et al., 2017)
7	ANALISA PREDIKSI MAHASISWA DROP OUT MENGGUNAKAN	Implementasi Algoritma C.45 & ID3 Pada Kasus	(Rajendra Haidar et al., 2020)

	METODE DECISION TREE DENGAN ALGORITMA ID3 dan C4.5.	Prediksi Mahasiswa <i>Drop Out</i> .	
8	An online software for decision tree classification and visualization using C4.5 algorithm (ODTC).	Aplikasi Visualisasi Menggunakan <i>Decision Tree C4.5</i>	(Das et al., 2014)

Tujuan yang bisa didapatkan sesuai dengan beberapa judul penelitian lain yang terdapat pada tabel analisis tinjauan diatas adalah :

- a. Algoritma *Decision Tree C4.5* dapat menyelesaikan masalah klasifikasi pada berbagai kasus keseharian kita.
- b. Salah satu metode pengujian kinerja algoritma merupakan membagi sejumlah besar data menjadi *data training* dan *data testing*.
- c. Pada algoritma *decision tree c4.5* memerlukan satu kolom sebagai atribut luaran (*label output*) sedangkan kolom lainnya sebagai atribut masukan (*label input*) dalam pengimplementasiannya.
- d. Hasil luaran akhir pada klasifikasi data menggunakan metode decision tree c4.5 dapat berupa diagram keputusan maupun nilai-nilai hasil klasifikasi.
- e. Sudah terdapat penelitian mengenai aplikasi klasifikasi menggunakan algoritma *decision tree c4.5* sebelumnya, namun pada aplikasi tersebut belum terdapat fitur yang dapat mengunduh model pelatihan dan tidak terdapat detail yang menampilkan partisi data untuk melatih maupun menguji kinerja algoritma.

BAB III

METODOLOGI PENELITIAN

3.1 Subjek & Objek Penelitian

Subjek penelitian merupakan sesuatu yang menjadi pusat pembahasan atau komponen utama yang akan menjadi topik perbincangan pada penelitian. Subjek pada penelitian umumnya dapat berupa sebuah kelompok, individu, benda, topik maupun rumpun ilmu. Sedangkan objek penelitian merupakan suatu perilaku atau fenomena yang terjadi atau diberikan pada subjek penelitian yang akan diamati selama penelitian berlangsung. Subjek dan objek penelitian akan sangat menentukan bagaimana penelitian berlangsung, karena akan memengaruhi penentuan rumusan, batasan, tujuan, manfaat dan metodologi penelitian yang menjadi dasar pada sebuah penelitian. Subjek pada penelitian ini akan difokuskan kepada algoritma *decision tree c4.5*, sedangkan objek pada penelitian ini adalah implementasi algoritma *decision tree c4.5*.

3.2 Indikator keberhasilan

Indikator keberhasilan adalah suatu hal yang dapat menentukan keberhasilan dalam suatu penelitian. Keberhasilan dari sebuah penelitian merupakan hal penentu apakah tujuan dari penelitian dapat berjalan sesuai yang sudah diharapkan atau tidak. Adapun indikator keberhasilan pada penelitian kali ini sebagai berikut.

1. Aplikasi dapat menerima atau menolak fail yang diunggah oleh pengguna dengan beberapa peraturan seperti.
 - a) Fail yang diunggah oleh pengguna merupakan fail yang memiliki tipe ekstensi “.csv”.
 - b) Apabila fail yang diunggah oleh pengguna memiliki kolom kurang dari 3, maka aplikasi dapat menolak fail yang diunggah.
2. Aplikasi dapat membaca dan membedakan tipe data dari fail yang diunggah oleh pengguna.
3. Aplikasi dapat menerima *input* “traintestsplit” dan “targety” yang telah dipilih oleh pengguna.

4. Aplikasi dapat melakukan maupun menampilkan hasil perhitungan algoritma *decision tree c4.5* dari fail yang diunggah oleh pengguna dan membuat model pelatihan yang dapat diunduh oleh pengguna dalam format “.pkl”.

3.3 Sumber Data

Ketersediaan sumber data akan memiliki pengaruh yang besar pada jalannya penelitian, apabila terjadi kekurangan sumber data juga akan menghambat jalannya penelitian. Pada penelitian ini akan menggunakan jenis data sekunder sebagai sumber datanya. Data sekunder yang didapatkan berasal dari beberapa publikasi ilmiah, jurnal, buku, maupun sumber tertulis lainnya yang memiliki kredibilitas dan memiliki pembahasan yang relevan dengan penelitian ini. Data sekunder yang telah diterbitkan oleh pihak lain bisa saja memiliki tujuan komersial maupun non komersial (Suliyanto, 2017). Selain itu penelitian ini juga akan menggunakan beberapa *dataset* yang bersifat umum sebagai *input* untuk aplikasi yang dikembangkan.

3.4 Metodologi Pengumpulan Data

Menentukan teknik atau metode pengumpulan data merupakan salah satu hal pokok dalam melakukan sebuah penelitian. Dengan menggunakan metode pengumpulan data yang tepat memudahkan peneliti dalam hal mencapai tujuan dari penelitian yang dilakukan. Penelitian ini akan menggunakan metode dokumentasi sebagai metode pengumpulan datanya. Dokumentasi dalam mengumpulkan data sekunder guna memperoleh informasi yang relevan dengan permasalahan yang diangkat.

3.4.1 Studi Literatur

Kegiatan studi literatur merupakan serangkaian kegiatan yang berhubungan dengan mencari, mengumpulkan, menyaring dan mengolah segala data relevan dari beberapa sumber yang didapatkan sebagai bahan dasar untuk penelitian yang dilakukan (Kartiningrum, 2015). Studi literatur pada penelitian ini akan difokuskan kepada subjek dan objek dari penelitian ini yaitu algoritma *decision tree c4.5* dan implementasi algoritma *decision tree c4.5* menggunakan python flask.

3.4.2 Peninjauan Dataset

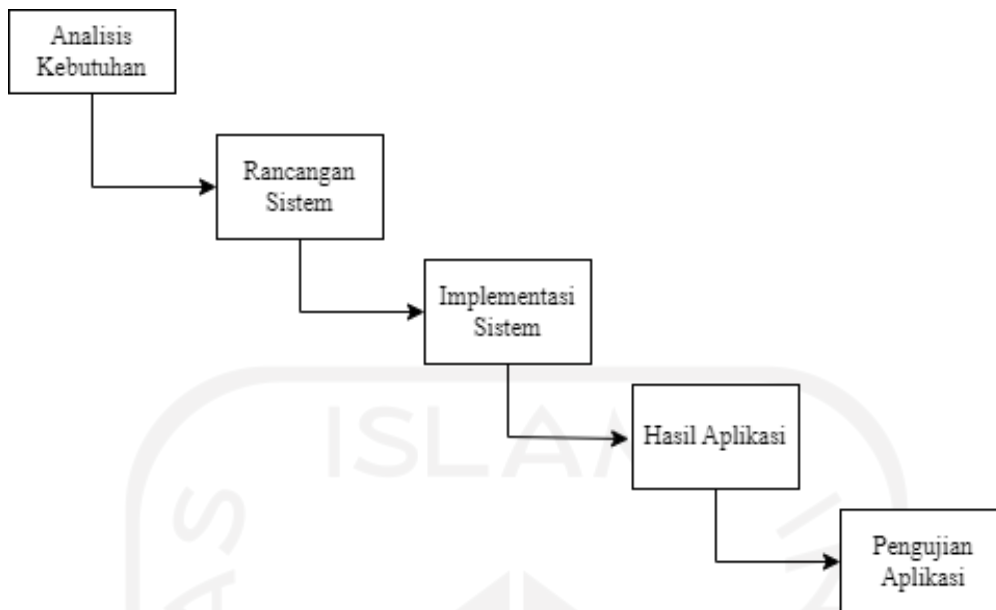
Selain dari studi literatur terhadap algoritma *decision tree c4.5*, pada penelitian ini juga dilakukan peninjauan *dataset* sebagai salah satu metode pengumpulan datanya. Hal ini dilakukan mengingat pada penelitian ini membutuhkan *dataset* sebagai masukan kepada

aplikasi *machine learning* dengan algoritma *decision tree c4.5* berbasis *web* yang akan dikembangkan pada penelitian kali ini. Kriteria dataset yang akan digunakan dalam penelitian kali ini dijabarkan sebagai berikut.

1. Fail yang memiliki ekstensi selain dari “.csv”. Hal ini ditujukan untuk melihat apakah sistem mampu menolak fail selain “.csv” yang diunggah.
2. Fail “.csv” yang hanya memiliki satu kolom. Hal ini ditujukan untuk melihat apakah sistem mampu menolak fail yang memiliki kolom kurang dari 2.
3. Fail “.csv” dengan kolom full diskrit dan memiliki lebih dari sama dengan 2 kolom.
4. Fail “.csv” dengan kolom full kategorikal dan memiliki lebih dari sama dengan 2 kolom.
5. Fail “.csv” dengan kolom baik diskrit maupun kategorikal dan memiliki lebih dari sama dengan 2 kolom.

3.5 Metodologi Penelitian

Demi tercapainya tujuan pada penelitian, maka dibutuhkan sebuah proses-proses dalam menjalankan penelitian tersebut. Proses-proses dalam menjalankan sebuah penelitian itu biasa disebut dengan metode penelitian. Pada penelitian ini akan menerapkan metode *waterfall* sebagai metode penelitiannya. Metode *waterfall* atau yang sering disebut dengan “Classic Life Cycle” (Pressman, 2005). Model *waterfall* ini juga merupakan metode yang paling sering digunakan dalam Rekayasa Perangkat Lunak hingga saat ini (Suhirman et al., 2021). Inti dari metode *waterfall* adalah mengerjakan setiap tahapannya secara berurutan selangkah demi selangkah hingga tahapan paling akhir. Gambar 3.1 akan menggambarkan metode penelitian yang diterapkan pada penelitian ini.



Gambar 3.1 Metode penelitian

Analisis kebutuhan akan mengawali kegiatan pada pengembangan aplikasi berbasis web ini. Tahapan ini dilakukan dengan mencari informasi atau pengetahuan mengenai kebutuhan-kebutuhan yang diperlukan pada penelitian ini. Setelah itu penelitian dilanjutkan pada tahap perancangan sistem. Tahapan perancangan sistem ini akan dilakukan dengan merancang desain antarmuka aplikasi dan membuat berbagai macam diagram yang akan menggambarkan aplikasi yang dikembangkan. Selanjutnya penelitian akan dilanjutkan dengan tahap implementasi sistem. Pada tahap implementasi sistem ini akan mengeksekusi segala rancangan-rancangan yang telah dibuat pada tahap sebelumnya. Tahap keempat pada metode penelitian ini adalah tahap hasil aplikasi. Tahap ini akan menguji aplikasi yang berhasil dikembangkan sebagai bentuk verifikasi dan validasi apakah aplikasi yang dikembangkan dapat berjalan sesuai yang diharapkan atau tidak.

3.5.1 Analisis Kebutuhan

Dalam tahapan ini seluruh kebutuhan yang dibutuhkan akan melalui tahap analisis *system requirements specification*. *System requirements specification* (SRS) adalah tahapan yang dibutuhkan dalam menganalisis keperluan yang dibutuhkan oleh sistem. SRS juga merupakan tahap pokok pada setiap proyek Rekayasa Perangkat Lunak. Pada penelitian ini aplikasi akan dikembangkan dengan *micro-framework* dari python yaitu python flask dengan bantuan *software* penyunting sumber kode seperti Visual Studio Code dan lainnya. Kebutuhan pada penelitian ini diperoleh berdasarkan riset-riset maupun penelitian-penelitian yang sudah ada. Pada tahap ini terdapat empat jenis kebutuhan yang terbagi menjadi kebutuhan fungsional,

kebutuhan non-fungsional dan prioritas kebutuhan serta terdapat tambahan mengenai gambaran umum aplikasi yang akan dijelaskan lebih lanjut sebagai berikut.

a. Gambaran Umum

Agar memudahkan pembaca dalam memahami aplikasi dengan lebih baik, berikut penjelasan mengenai gambaran umum aplikasi.

1. Tentang Aplikasi

Aplikasi ini merupakan sebuah aplikasi *machine learning* berbasis *web* dibuat menggunakan *micro-framework* pada bahasa pemrograman python yaitu python flask yang memiliki fitur utama mengklasifikasi data yang diunggah oleh pengguna menggunakan algoritma *decision tree c4.5*.

2. Fungsi Aplikasi

Terdapat beberapa fungsi yang dimiliki oleh aplikasi berbasis *web* ini. Diantaranya adalah melakukan klasifikasi pada data yang diunggah oleh pengguna, menampilkan hasil klasifikasi data, memberikan keterangan pada data dengan menampilkan statistik deskriptif dari data, pengguna juga dapat melakukan *data preprocessing* dengan menghapus satu atau lebih kolom yang diinginkan, menampilkan partisi data (x training, x test, y training dan y test), melakukan *screenshot* pada antarmuka aplikasi yang diinginkan dan mengunduh model pelatihan yang telah dilakukan oleh aplikasi dalam format “.pkl”.

3. Pengguna Aplikasi

Target utama pada pengguna aplikasi adalah seseorang yang ingin melakukan klasifikasi terhadap data yang dimiliki dengan menggunakan algoritma *decision tree c4.5*. Tidak terdapat spesifikasi khusus pada pengguna aplikasi untuk memiliki pengetahuan mengenai algoritma *decision tree c4.5* lebih lanjut atau tidak. Karena pada aplikasi sudah terdapat informasi-informasi seputar algoritma *decision tree c4.5* yang dapat membuat pengguna lebih mengenal lebih jauh terhadap algoritma ini.

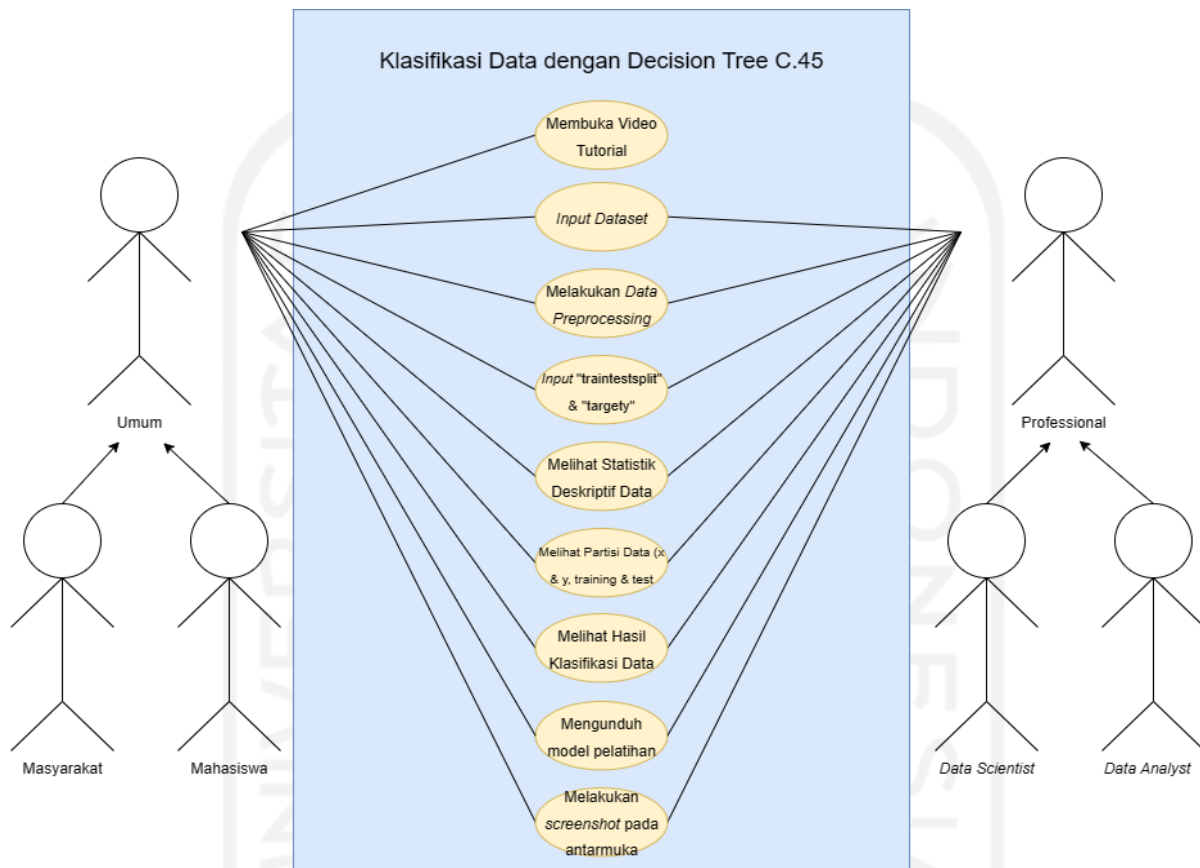
4. Kendali Aplikasi

Aplikasi dapat diakses dengan menggunakan *web browser* yang dimiliki pengguna baik dari *mobile device* maupun *personal computer (PC)*.

b. *Functional Specifications Requirements*

Kebutuhan *functional specifications requirements* merupakan fungsional yang membahas tentang fitur atau fungsi utama pada aplikasi. Kebutuhan fungsional pada

aplikasi ini dijelaskan pada Gambar 3.2 mengenai *use case diagram* aplikasi. Dimana *actor* pada *use case diagram* dikategorikan menjadi dua yaitu umum dan professional. Perbedaan dalam kedua actor hanya terdapat pada actor “umum” memiliki *use case* “Membuka Video Tutorial”, sedangkan pada “professional” tidak memerlukan *use case* tersebut.



Gambar 3.2 *Use case diagram* aplikasi

Semua fitur yang terdapat dalam use case pada **Error! Reference source not found.** akan dirincikan lebih lanjut dengan penjelasan mengenai deskripsi, langkah dan kebutuhan yang dibutuhkan oleh sistem sebagai berikut :

1. Membuka video tutorial
 - a) Deskripsi

Fitur ini merupakan salah satu fitur opsional yang terdapat dalam aplikasi yang artinya apabila pengguna tidak menggunakan fitur ini, maka tidak akan mengganggu jalannya aplikasi. Fitur ini juga merupakan langkah awal agar pengguna dapat lebih memahami tentang bagaimana cara kerja dari aplikasi. Sehingga diharapkan selama pengguna menggunakan aplikasi dapat dengan

lebih terarah dan terhindar dari kebingungan ketika menggunakan aplikasi berbasis *web* ini.

b) Langkah

- 1) Pengguna menekan *button* “Disini” yang terdapat pada halaman “Home”.
- 2) Pengguna akan diarahkan menuju Youtube yang merupakan situs dimana video tutorial berada.
- 3) Pengguna menonton video tutorial hingga pengguna dapat memahami bagaimana cara kerja aplikasi.
- 4) Selesai,

c) Kebutuhan

- 1) Sistem dapat membuka halaman Home dan menampilkan *button* “Disini”.
- 2) Sistem dapat mengarahkan pengguna ke video tutorial yang terdapat pada situs Youtube.

2. *Input dataset*

a) Deskripsi

Fitur *input dataset* merupakan salah satu fitur utama yang dimiliki oleh aplikasi. Apabila fitur ini tidak dilakukan oleh pengguna maka akan banyak fitur-fitur utama lainnya yang tidak dapat diakses oleh pengguna seperti : Melakukan *data preprocessing*, *input* “*traintestsplitted*” & “*targety*”, melihat statistik deskriptif data, melihat partisi data dan melihat hasil klasifikasi data. Pada fitur ini terdapat spesifikasi terhadap data yang diunggah oleh pengguna yang berupa fail yang diunggah harus memiliki ekstensi bertipe “.csv” dan pada fail yang diunggah terdapat dua atau lebih kolom di dalamnya.

b) Langkah

- 1) Pengguna menekan *button* “Choose File” pada halaman “Upload Data”.
- 2) Pengguna memilih fail “.csv” yang diinginkan pada penyimpanan *device*-nya untuk diunggah dengan *button* “Open”.
- 3) Pengguna menekan *button* “Submit” pada halaman “Upload Data”.
- 4) Selesai.

c) Kebutuhan

- 1) Sistem dapat membuka halaman “Upload Data” dan menampilkan *button* “Choose file” dan “Submit”.
- 2) Sistem dapat memunculkan *pop-up* penyimpan pada *device* pengguna.

- 3) Sistem dapat menerima apabila fail yang diunggah memiliki ekstensi “.csv” dan memiliki lebih dari sama dengan dua kolom di dalamnya. Namun sistem juga dapat menolak fail pengguna apabila syarat tersebut tidak terpenuhi oleh fail yang diunggah.
- 4) Sistem dapat menampilkan data yang diunggah oleh pengguna apabila syarat fail sudah terpenuhi.

3. Melakukan *data preprocessing*

a) Deskripsi

Fitur *data preprocessing* merupakan fitur yang dapat diakses hanya apabila pengguna sudah melakukan fitur *input dataset*. Pada fitur ini pengguna dapat memilih satu atau lebih kolom pada *dataset* yang diunggah untuk dihapus. Kolom yang sudah dihapus akan secara permanen hilang pada fail yang dimiliki oleh aplikasi. Apabila pengguna ingin mengakses kolom yang sudah terhapus pada fitur ini sebelumnya, maka pengguna harus mengunggah ulang fail yang dimilikinya. Pengguna juga perlu mengingat bahwa aplikasi hanya akan menerima fail yang memiliki kolom lebih dari sama dengan dua pada failnya. Apabila setelah melakukan fitur ini dan kolom yang dimiliki oleh fail menjadi kurang dari dua, maka secara otomatis sistem akan menolak fail yang telah diunggah dan pengguna perlu melakukan fitur *input dataset* lagi.

b) Langkah

- 1) Pengguna sudah melakukan fitur “Input Dataset”.
- 2) Pengguna menekan *button* “klik disini” pada halaman “Upload data”.
- 3) Pengguna memilih kolom yang diinginkan untuk dihapus pada pilihan yang disediakan.
- 4) Pengguna menekan *button* “Submit”.
- 5) Selesai.

c) Kebutuhan

- 1) Sistem dapat menampilkan *button* “Klik disini” hanya ketika pengguna sudah melakukan “*input dataset*” pada halaman “Upload Data”.
- 2) Sistem dapat menampilkan pilihan berupa nama kolom yang terdapat pada fail yang diunggah oleh pengguna.

- 3) Sistem dapat menghapus kolom yang telah dipilih oleh pengguna dan menyimpan perubahan yang telah ditetapkan.
 - 4) Sistem dapat menghapus fail yang diunggah oleh pengguna apabila setelah fitur ini telah selesai dilakukan dan pada fail yang diunggah terdapat kurang dari dua kolom tersisa.
 - 5) Sistem dapat menampilkan hanya button “Kembali” pada halaman Data Preprocessing (Hal ini terjadi ketika pengguna mengakses “url” *data preprocessing* namun belum melakukan “input dataset”).
4. *Input “traintestsplit” & “targety”*
- a) Deskripsi

Fitur ini merupakan salah satu fitur utama yang harus dilakukan agar pengguna dapat melihat hasil klasifikasi data yang diunggah. Fitur ini akan berguna apabila pengguna sudah melakukan fitur “*input dataset*”. Apabila pengguna belum melakukannya namun sudah menekan *button* “Submit” pada halaman “Dataframe”, maka fitur ini tidak akan berguna. Pada fitur ini pengguna menentukan pilihan “*traintestsplit*” antara 0.1 hingga 0.9 yang ditampilkan dalam *button* “range”. Sedangkan pilihan “*targety*” akan diprioritaskan kepada kolom kategorikal, namun apabila *dataset* yang diunggah tidak memiliki kolom kategorikal. Maka pilihan “*targety*” akan berisi kolom diskrit.
 - b) Langkah
 - 1) Pengguna telah melakukan fitur “*input dataset*”.
 - 2) Pengguna memilih pilihan “*traintestsplit*” dan “*targety*” yang diinginkan dan menekan *button* “Submit”.
 - 3) Selesai.
 - c) Kebutuhan
 - 1) Sistem dapat menyimpan fail yang telah diunggah oleh pengguna pada fitur “*input dataset*”.
 - 2) Sistem dapat membedakan antara kolom kategorikal dan kolom diskrit yang terdapat pada fail yang diunggah.
 - 3) Sistem dapat menampilkan pilihan pada “*traintestsplit*” berupa *button* “range” antara 0.1 hingga 0.9.

- 4) Sistem dapat menampilkan pilihan pada “targety” dan memberi prioritas kepada kolom kategorikal sebagai pilihan.
- 5) Sistem dapat menyimpan pilihan “traintestsplit” dan “targety” yang telah dipilih oleh pengguna.

5. Melihat statistik deskriptif data

a) Deskripsi

Fitur melihat statistik deskriptif data merupakan salah satu fitur utama pada aplikasi ini. Pengguna dapat mengakses fitur ini pada halaman “Dataframe”. Sistem akan menampilkan sebuah tabel yang berisikan *count*, *unique*, *top*, *freq*, *mean*, standar deviasi, nilai minimal, kuartil 1, kuartil 2, kuartil 3 dan nilai maksimal. Fitur ini dapat diakses apabila pengguna telah melakukan “*input dataset*”.

b) Langkah

- 1) Pengguna telah melakukan “*input dataset*”.
- 2) Pengguna dapat melihat tabel Statistik Deskriptif pada halaman “Dataframe”.
- 3) Selesai.

c) Kebutuhan

- 1) Sistem dapat menyimpan fail yang telah diunggah oleh pengguna pada fitur “*input dataset*”.
- 2) Sistem dapat membedakan antara kolom kategorikal dan kolom diskrit pada fail yang diunggah.
- 3) Sistem dapat menampilkan tabel statistik deskriptif pada halaman “Dataframe”.

6. Melihat partisi data

a) Deskripsi

Fitur melihat partisi data merupakan fitur yang dapat diakses oleh pengguna apabila pengguna telah melakukan fitur “*input dataset*” dan telah melakukan “*input traintestsplit & targety*”. Pada fitur ini pengguna dapat melihat *X.training*, *X_test*, *y_training* & *y_test* yang sebelumnya telah dipecah dari *dataset*. Pengguna dapat mengakses fitur ini pada halaman “Data Training & Tes”.

b) Langkah

- 1) Pengguna telah melakukan “*input dataset*”.
 - 2) Pengguna telah melakukan “*input traintestsplrit & targety*”.
 - 3) Pengguna dapat melihat partisi data yang terdapat pada halaman “Data Training & Tes”.
 - 4) Selesai.
- c) Kebutuhan
- 1) Sistem dapat menyimpan fail yang telah diunggah oleh pengguna pada fitur “*input dataset*”.
 - 2) Sistem dapat membedakan antara kolom kategorikal dan kolom diskrit yang terdapat pada fail yang diunggah.
 - 3) Sistem dapat menyimpan pilihan “*traintestsplrit*” & “*targety*” yang telah dipilih oleh pengguna.
 - 4) Sistem dapat memecah data sesuai dengan pilihan-pilihan yang telah ditentukan oleh pengguna.
 - 5) Sistem dapat melakukan *encoding* terhadap kolom kategorikal yang terdapat pada fail yang diunggah.
 - 6) Sistem dapat menampilkan partisi data (X_{training} , X_{test} , y_{training} & y_{test}) pada halaman “Data Training & Tes”.
7. Melihat hasil klasifikasi data
- a) Deskripsi
- Fitur ini merupakan fitur utama dan tujuan akhir dari aplikasi berbasis *web* ini. Pengguna dapat mengakses fitur ini apabila pengguna telah melakukan fitur “*input dataset*” dan “*input trainingtestsplrit & targety*”. Fitur ini dapat diakses pada halaman “Decision Tree C4.5”. Pada fitur ini pengguna akan disajikan visualisasi hasil klasifikasi menggunakan algoritma *decision tree c4.5* berupa grafik pohon keputusan, matriks konfusi, laporan klasifikasi, nilai training-set akurasi, training-set skor, test-set skor dan akurasi model.
- b) Langkah
- 1) Pengguna telah melakukan “*input dataset*”.
 - 2) Pengguna telah melakukan “*input traintestsplrit & targety*”.
 - 3) Pengguna dapat melihat hasil klasifikasi yang terdapat pada halaman “Decision Tree C4.5”
 - 4) Selesai.

c) Kebutuhan

- 1) Sistem dapat menyimpan fail yang telah diunggah oleh pengguna pada fitur “*input dataset*”.
- 2) Sistem dapat membedakan antara kolom kategorikal dan kolom diskrit yang terdapat pada fail yang diunggah.
- 3) Sistem dapat menyimpan pilihan “*traintestsplit*” & “*targety*” yang telah dipilih oleh pengguna.
- 4) Sistem dapat memecah data sesuai dengan pilihan-pilihan yang telah ditentukan oleh pengguna.
- 5) Sistem dapat melakukan *encoding* terhadap kolom kategorikal yang terdapat pada fail yang diunggah.
- 6) Sistem dapat melakukan komputasi dengan menggunakan algoritma *decision tree c.45* terhadap data yang diunggah.
- 7) Sistem dapat menampilkan hasil komputasi yang terdapat pada halaman “Decision Tree C4.5”.

8. Mengunduh Model Pelatihan

a) Deskripsi

Fitur mengunduh model pelatihan merupakan fitur dimana pengguna mendapatkan file bertipe “.pkl” dari pelatihan model yang sudah dilakukan oleh aplikasi.

b) Langkah

- 1) Pengguna telah melakukan “*input dataset*”.
- 2) Pengguna telah melakukan “*input traintestsplit & targety*”.
- 3) Pengguna mengunjungi halaman “Decision Tree C45” dan menekan *button* “Download Model”.
- 4) Selesai.

c) Kebutuhan

- 1) Sistem dapat menyimpan fail yang telah diunggah oleh pengguna pada fitur “*input dataset*”.
- 2) Sistem dapat membedakan antara kolom kategorikal dan kolom diskrit yang terdapat pada fail yang diunggah.
- 3) Sistem dapat menyimpan pilihan “*traintestsplit*” & “*targety*” yang telah dipilih oleh pengguna.

- 4) Sistem dapat memecah data sesuai dengan pilihan-pilihan yang telah ditentukan oleh pengguna.
- 5) Sistem dapat melakukan *encoding* terhadap kolom kategorikal yang terdapat pada fail yang diunggah.
- 6) Sistem dapat melakukan komputasi dengan menggunakan algoritma *decision tree c.45* terhadap data yang diunggah.
- 7) Sistem dapat mengirimkan model pelatihan bertipe “.pkl” ke *device* yang digunakan pengguna.

d) Deskripsi

Fitur mengunduh model pelatihan merupakan salah satu fitur yang dapat digunakan pengguna ketika pengguna menginginkan model pelatihan yang telah dilakukan oleh aplikasi. Dimana aplikasi akan mengirimkan file bertipe “.pkl” ke *device* yang digunakan oleh pengguna.

e) Langkah

- 5) Pengguna menekan *button* “Screenshot” pada halaman yang diinginkan.
- 6) Setelah melihat gambar pratinjau pengguna dapat menekan *button* “Download” untuk menyimpan gambar tersebut atau menekan *button* “Kembali” apabila ingin membatalkannya.
- 7) Selesai.

f) Kebutuhan

- 8) Sistem dapat menampilkan *button* “Screenshot”.
- 9) Sistem dapat menampilkan halaman pratinjau terhadap antarmuka yang diinginkan.
- 10) Sistem dapat mengirimkan fail “.png” ke penyimpanan pengguna apabila pengguna menekan *button* “Download”.

c. *Non-Functional Requirements*

Kebutuhan *non-functional specifications requirements* merupakan kebutuhan yang akan membahas selain dari kebutuhan utama atau *functional specifications requirements*. Tabel 3.1 akan menjelaskan faktor-faktor yang dibutuhkan pada kebutuhan non-fungsional pada aplikasi ini. Berikut beberapa faktor kebutuhan non-fungsional yang diperlukan.

Tabel 3.1 Tabel Kebutuhan Non-fungsional

No	Kebutuhan	Deskripsi	Tool
----	-----------	-----------	------

1	<i>Hardware</i>	Perangkat keras yang dibutuhkan untuk mengakses program.	<i>Personal Computer (PC) atau Mobile Device</i>
2	<i>Software</i>	Perangkat lunak yang dibutuhkan untuk mengakses program.	OS, Browser
3	Antarmuka / <i>User Interface (UI)</i>	Sebagai penghubung antara pengguna dan sistem melalui tampilan pada layar perangkat keras.	Bootstrap (Visual Studio Code)
4	Server	Penyedia layanan hosting maupun file yang berisi kode-kode pemrograman	Python Flask (Visual Studio Code) & Pythonanywhere

d. *Prioritization Requirements*

Prioritization requirements merupakan sub bagian paling akhir pada tahap analisis kebutuhan. Tahap ini tidak kalah penting, terutama apabila menggunakan pengembangan aplikasi menggunakan metode *waterfall*. Fungsional ini akan mengurutkan fitur-fitur yang terdapat pada aplikasi mulai dari yang paling penting untuk dikembangkan hingga ke fitur terakhir. Kebutuhan fungsional dan kebutuhan non-fungsional yang akan diurutkan berdasarkan kepentingannya dengan skala terkecil adalah skala paling penting. Tabel 3.2 akan menjelaskan prioritas kebutuhan yang terdapat pada aplikasi berbasis *web* ini.

Tabel 3.2 Tabel Prioritas Kebutuhan

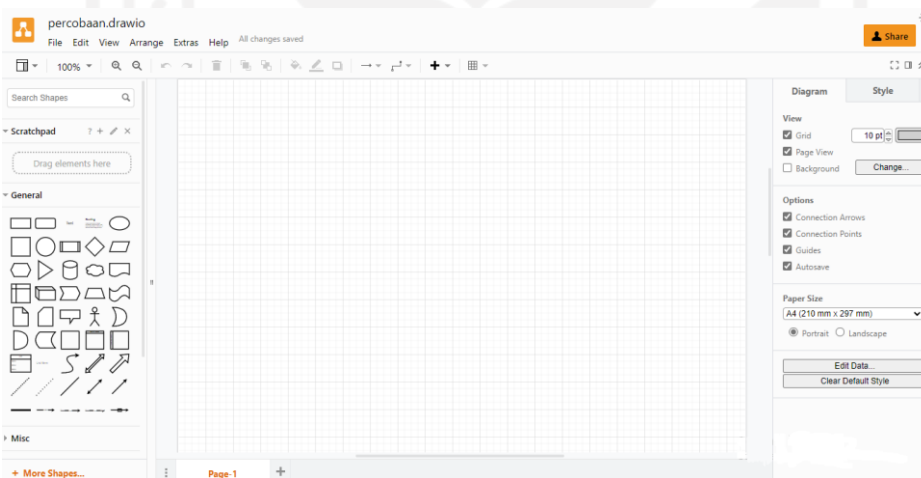
No	Fitur	Halaman pada Aplikasi	Prioritas
1	Input dataset	Upload Data	Prioritas 1
2	Tampilan dataset	Upload Data	Prioritas 9
3	Data preprocessing	Upload Data	Prioritas 10
4	Input <i>traintestsplitted & targety</i>	Dataframe	Prioritas 2
5	Statistik deskriptif data	Dataframe	Prioritas 11
6	Tampilan partisi data	Data Training & Tes	Prioritas 8
7	Tampilan diagram pohon keputusan	Decision Tree C4.5	Prioritas 3
8	Konfusi matriks	Decision Tree C4.5	Prioritas 4

9	Nilai hasil klasifikasi	Decision Tree C4.5	Prioritas 5
10	Laporan Klasifikasi	Decision Tree C4.5	Prioritas 6
11	Unduh model pelatihan	Decision Tree C4.5	Prioritas 7
12	Screenshot	Semua halaman	Prioritas 12

3.5.2 Perancangan Sistem

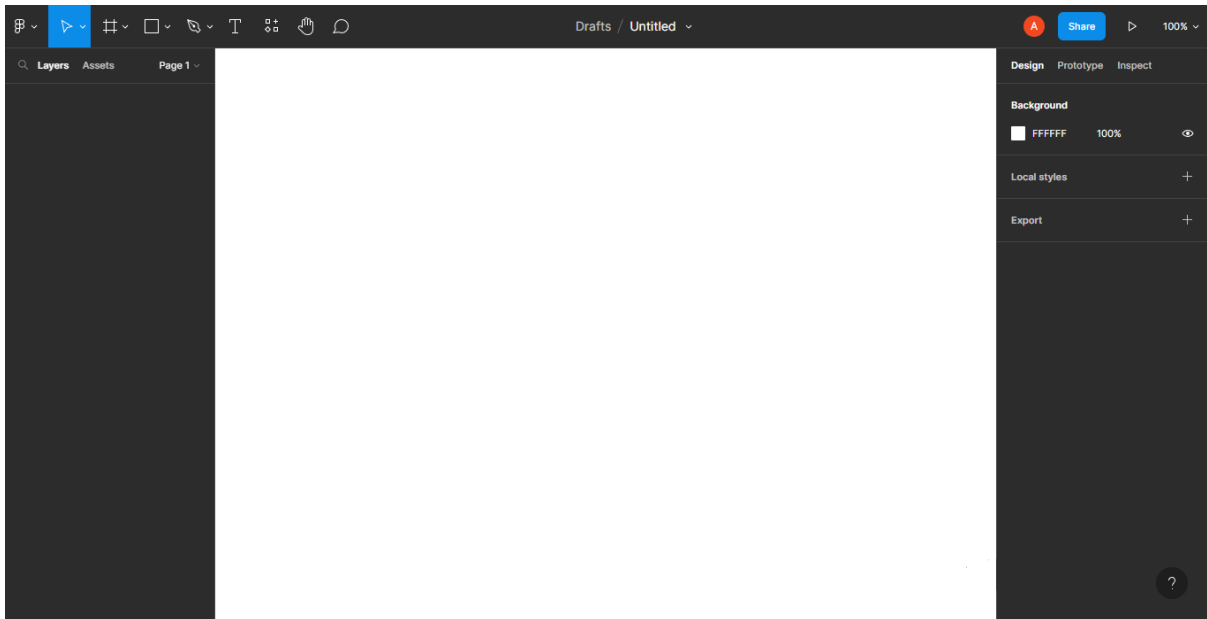
Tahap perancangan merupakan tahap dimana pengembang membuat garis besar aplikasi baik untuk antarmuka, sistemnya dan interaksi antara pengguna dengan sistem. Namun rancangan yang tercipta pada tahap ini belum merupakan hasil akhir dari aplikasi. Aplikasi bisa saja mendapat tambahan atau pengurangan selama masa implementasi. Rancangan yang dibuat pada tahap ini memiliki tiga jenis rancangan yaitu rancangan antarmuka (*Mockup*), *activity diagram* dan *class diagram*.

Baik *activity diagram* dan *class diagram* pada penelitian ini akan dibuat pada salah satu *software* bertipe *open source* yaitu **Diagram.net**. *Software* ini dapat diakses secara *offline* dengan mengunduh aplikasinya maupun secara *online* dengan mengakses situs “**Diagram.net**”. *Software* ini juga merupakan salah satu perangkat lunak pembuat diagram alir gratis yang terkemuka.



Gambar 3.3 Aplikasi Diagram.net

Rancangan antarmuka (*Mockup* aplikasi) berfokus pada bagaimana rancangan antarmuka aplikasi terlihat ketika nanti aplikasi sudah dibuat. *Mockup* aplikasi pada penelitian ini akan dibuat pada salah satu aplikasi *open source* yaitu Figma. Aplikasi ini bisa diakses baik secara *online* maupun *offline*. Figma juga merupakan salah satu aplikasi untuk mendesain antarmuka yang terkemuka.

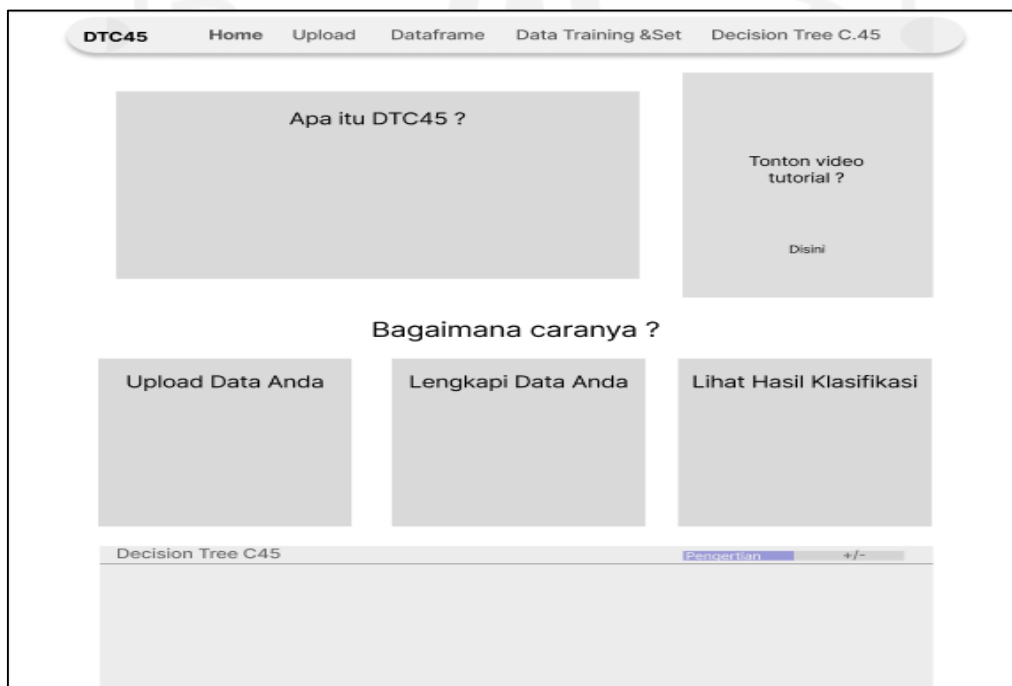


Gambar 3.4 Aplikasi Figma

a. Rancangan Antarmuka (*Mockup*)

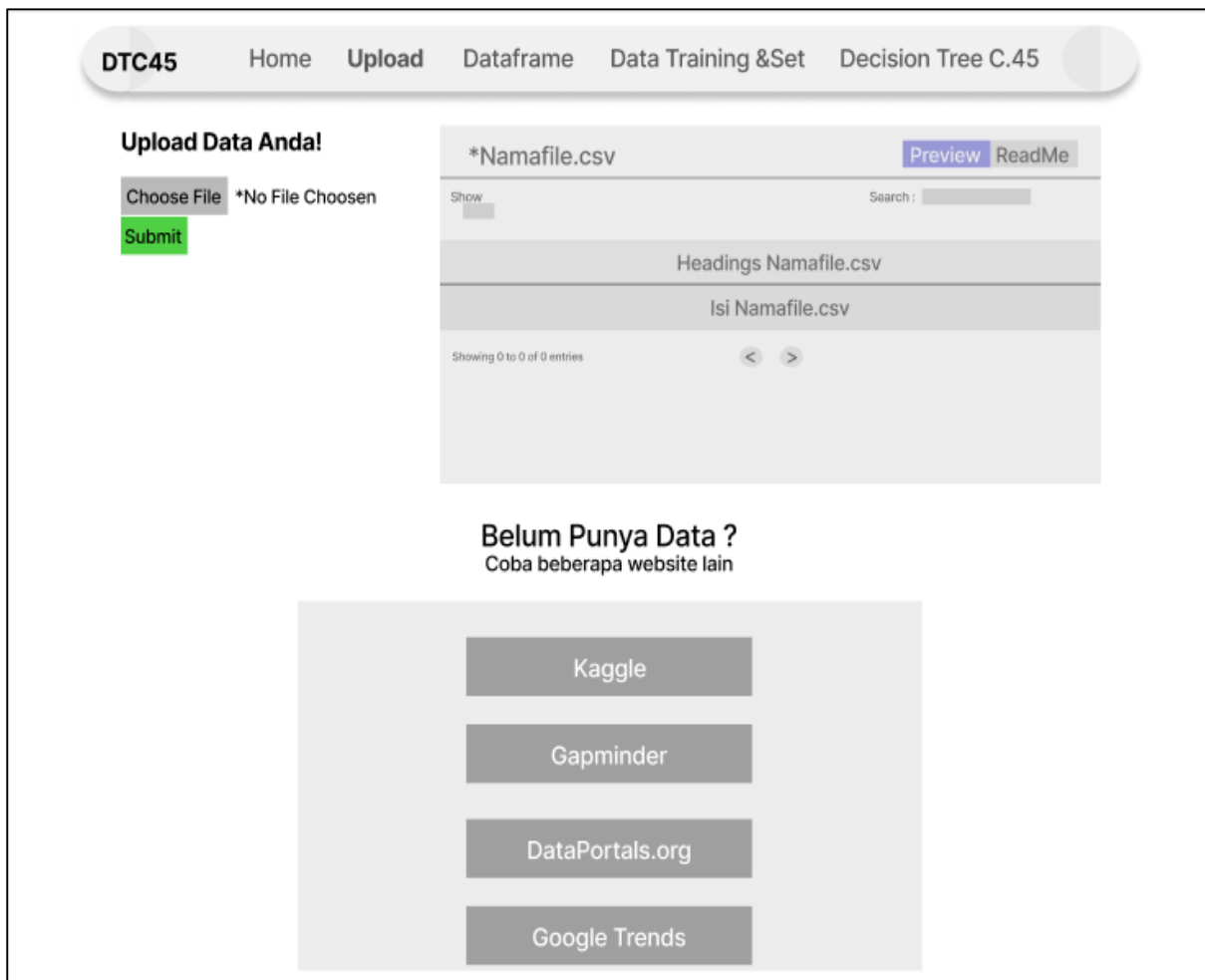
Perancangan antarmuka pada aplikasi ini akan berdasarkan pada tiap-tiap halaman yang dimiliki oleh aplikasi. *Mockup* aplikasi ini juga mengikuti aturan dari Jinja2 dan Template selaku dependensi dari python flask. Selain peraturan-peraturan pada dependensi dari python flask, perancangan ini dibuat berdasarkan Bootstrap selaku kerangka kerja antarmuka gratis yang digunakan pada pengembangan aplikasi berbasis *web* ini.

1. Home

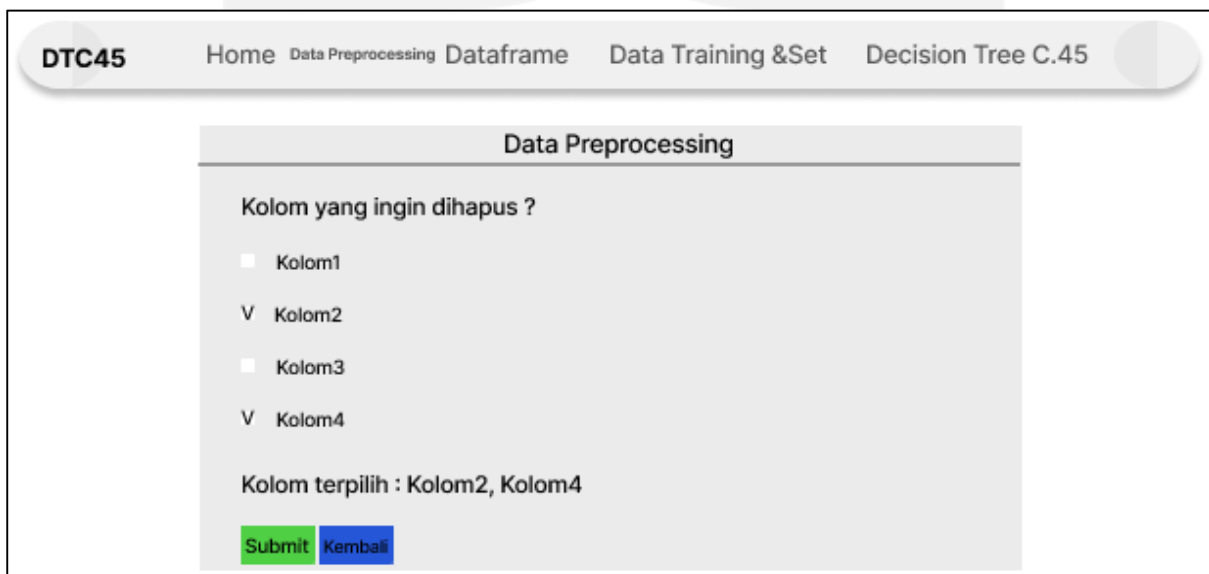


Gambar 3.5 *Mockup* halaman Home

2. Upload Data

Gambar 3.6 *Mockup* halaman Upload Data

3. Data Preprocessing



Gambar 3.7 *Mockup* halaman Data Preprocessing

4. Dataframe

DTC45 Home Upload **Dataframe** Data Training &Set Decision Tree C.45

***Namafile.csv**

2 Kolom Diskrit (*lisit nama kolom) (5,14) Datashape 3 Kolom Katergorikal (*lisit nama kolom)

Lengkapi Data Anda!
 Traintestsplit: 0,1 — 0,9 0,5
 Target(y):
 Nama Kolom: v
 Submit

Dataframe Preview ReadMe

Show Search:

Headings Namafile.csv

Isi Namafile.csv

Showing 0 to 0 of 0 entries < >

Null Values			
Kolom 1	Kolom 2	Kolom 3	Kolom 4
0	1	0	0
Total Null Values = 1			

Statistik Deskriptif

Nama Kolom

Isi

Gambar 3.8 *Mockup* halaman Dataframe

5. Data Training & Test

DTC45 Home Upload **Dataframe** **Data Training &Set** Decision Tree C.45

***Namafile.csv**

X Train Test Split 0.5 Y (*lisit nama kolom) (*lisit nama kolom)

Preview X X.Training X.Test

Show Search:

Heading

Isi

Showing 0 to 0 of 0 entries < >

Train Test Data X Data Y

Penjelasan Penjelasan

Preview Y y.training y.test

Show Search:

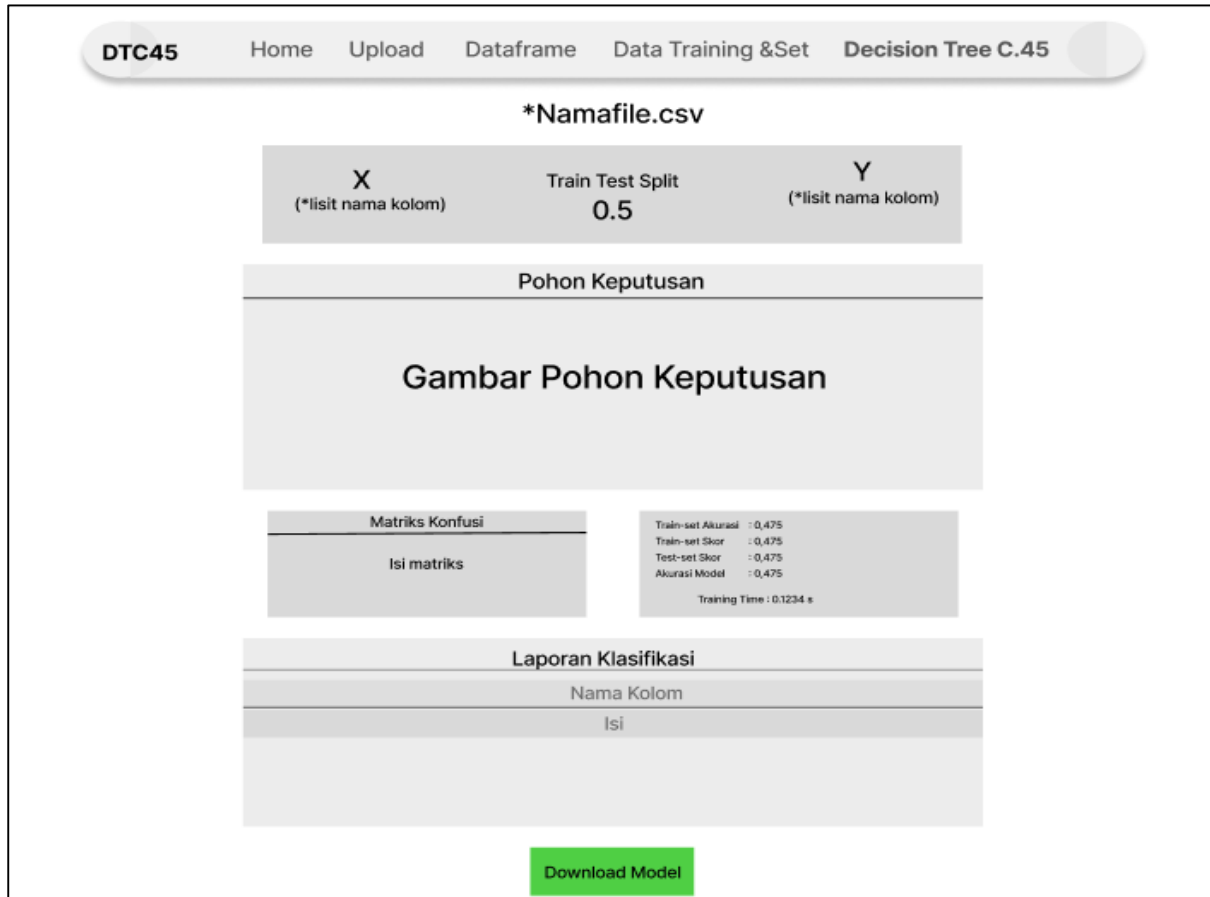
Heading

Isi

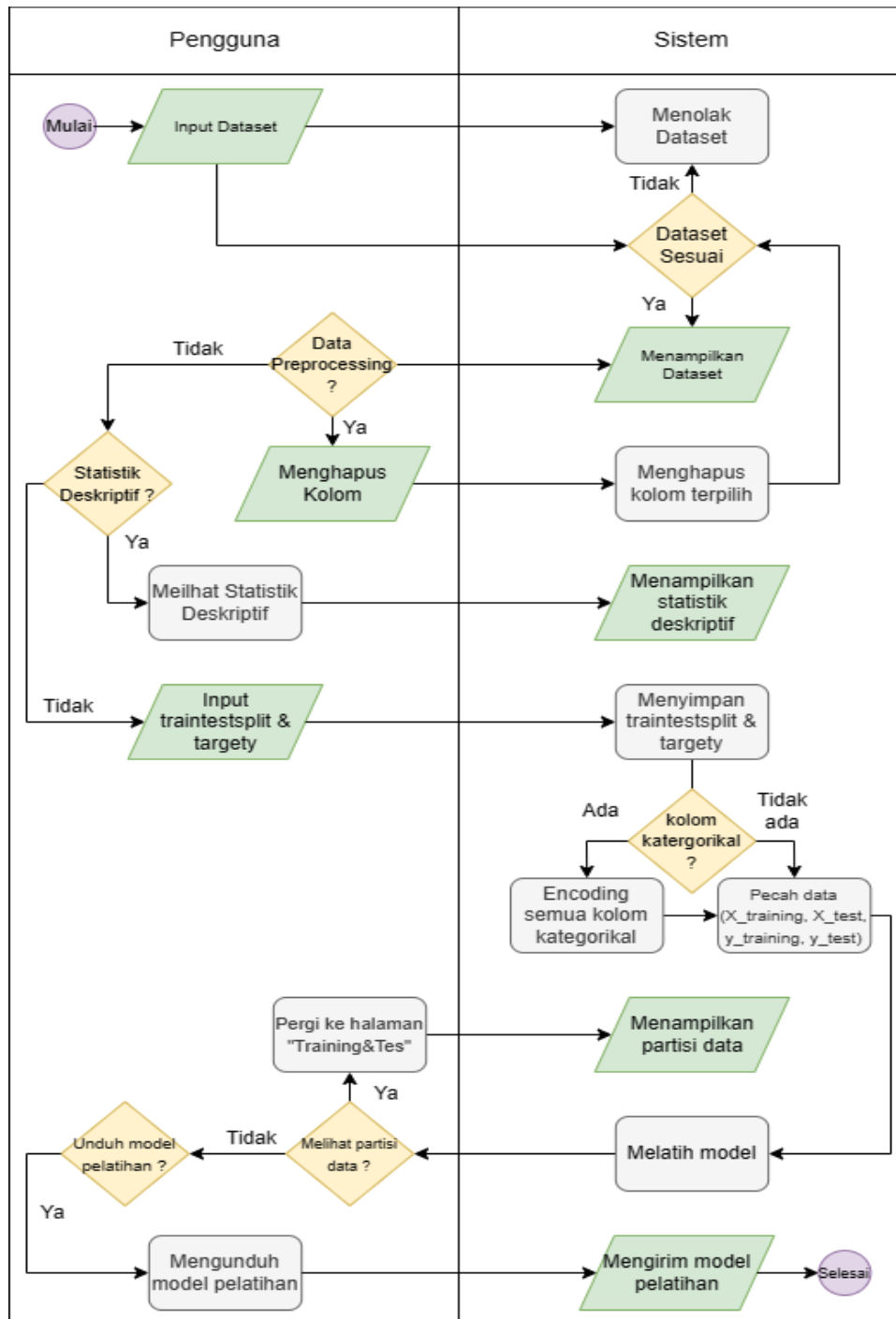
Showing 0 to 0 of 0 entries < >

Gambar 3.9 *Mockup* halaman Data Training & Test

6. Decision Tree C4.5

Gambar 3.10 *Mockup* halaman Decision Tree C4.5b. *Activity Diagram*

Gambar 3.11 merupakan gambar *activity diagram* pada aplikasi ini. Dimana pelaku pada *activity diagram* ini hanya terdapat dua yaitu pengguna dan sistem. Dimana kegiatan yang dilambangkan oleh jajar genjang berwarna hijau merupakan *input* bagi pengguna dan melambangkan *output* pada sistem. Belah ketupat berwarna oranye melambangkan percabangan. Persegi panjang dengan warna abu-abu cerah melambangkan kegiatan atau aktivitas yang dilakukan oleh pengguna maupun sistem. Dan lambing terakhir adalah lingkaran dengan warna ungu melambangkan awal dan akhir dari *activity diagram* ini. *Activity diagram* diawali dengan pengguna yang melakukan *input dataset* ini merupakan kegiatan pertama dan terpenting dalam *activity diagram* ini sampai dengan tahap visualisasi klasifikasi data dengan algoritma *decision tree c4.5*.

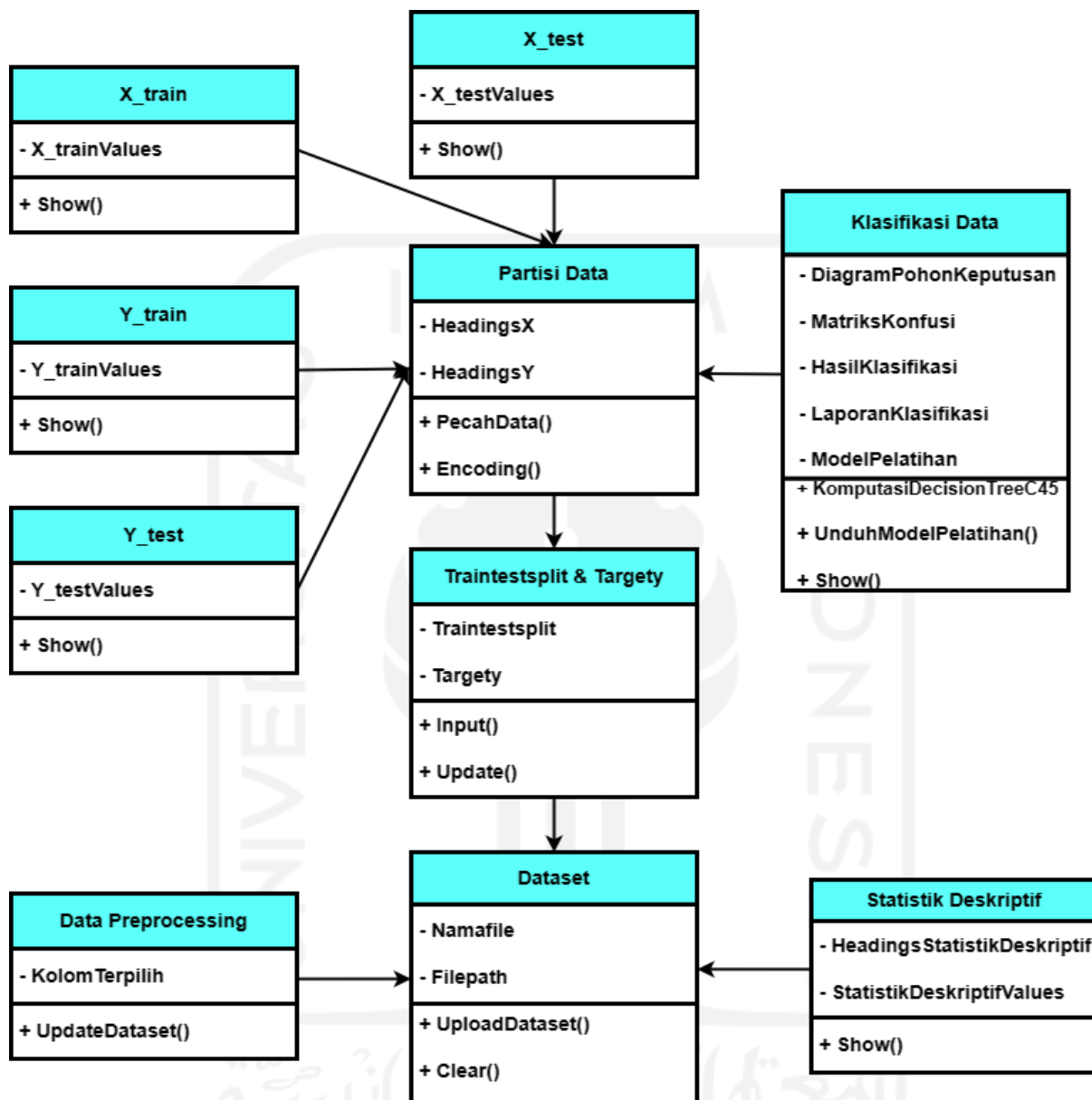


Gambar 3.11 Activity diagram

c. Class Diagram

Class diagram pada aplikasi ini terdiri dari 1 kelas dan 3 subkelas. Dimana komponen atas pada bagian kelas dan subkelas berisikan nama, komponen tengah berisikan atribut yang dimiliki oleh kelas maupun subkelas dan komponen bawah terdapat operasi atau fungsi yang dapat dilakukan oleh kelas maupun subkelasnya. Pada diagram ini juga digambarkan bagaimana hubungan antara satu sama lain yang dimana

semua subkelas merupakan pewarisan dari kelas “Dataset”. Gambar 3.12 menggambarkan bagaimana *class diagram* yang dimiliki oleh aplikasi berbasis *web* ini.



Gambar 3.12 *Class diagram*

3.5.3 Implementasi Sistem

Tahap implementasi merupakan tahapan penerapan segala sesuatu yang telah direncanakan pada tahap perancangan sistem, tahap ini adalah tahap dimana pengekseskusan rencana yang telah dirancang. Pada tahap ini akan difokuskan kepada pengembangan aplikasi *machine learning* menggunakan *decision tree c4.5* berbasis *web* dengan *micro-framework* python flask. Pengembangan aplikasi berbasis *web* ini akan membuat kode-kode dengan bahasa pemrograman python dengan bantuan *library* yang dimiliki bahasa pemrograman tersebut yang relevan dengan algoritma *decision tree c4.5*.

3.5.4 Hasil Aplikasi

Tahap hasil aplikasi sebagai bentuk hasil luaran dari tahap implementasi sistem pada penelitian ini. Pada tahap ini akan ditampilkan hasil implementasi berupa dokumentasi antarmuka aplikasi yang telah dikembangkan, tahap ini juga akan menunjukkan apakah fungsionalitas aplikasi yang telah disebutkan sebelumnya dapat bekerja atau tidak sebelum lanjut ke tahap pengujian.

3.5.5 Pengujian Hasil

Pengujian hasil merupakan tahapan terakhir pada rangkaian metode penelitian ini. Tahap ini akan menjadi tolak ukur apakah aplikasi yang dikembangkan dapat dikatakan berhasil dikembangkan dan dapat diterima oleh pengguna atau tidak. Pengujian hasil pada penelitian ini akan dilakukan sebagai berikut :

1. *Black Box Testing* : Pengujian ini akan menguji semua fungsionalitas sistem baik *functional specifications requirements, non-functional requirements* dan *prioritizations requirements*.
2. *Software Testing* : Pengujian *software testing* dilakukan dengan melakukan membandingkan aplikasi yang dikembangkan dengan aplikasi yang memiliki kegunaan serupa. Dalam hal ini aplikasi WEKA digunakan sebagai aplikasi pembanding.
3. *User Acceptance Test* : Pengujian ini akan mengujikan aplikasi yang telah dikembangkan kepada calon pengguna, yang dilakukan untuk mendapatkan feedback kekurangan ataupun kelebihan dari aplikasi yang telah dikembangkan.

BAB IV

HASIL DAN PEMBAHASAN

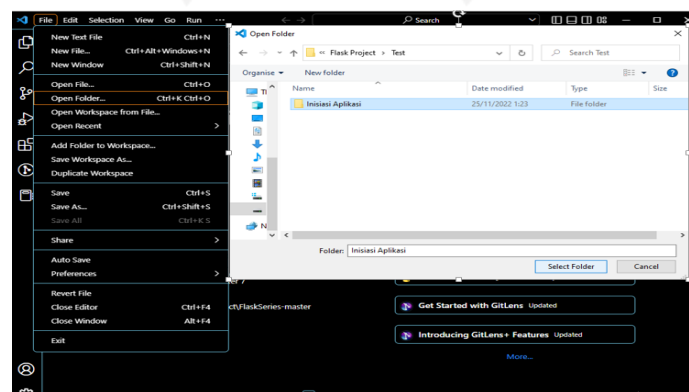
4.1 Implementasi Hasil

Berakhirnya tahap rancangan sistem menandakan dimulainya tahap implementasi hasil. Pada tahap implementasi akan mencoba merealisasikan setiap rancangan yang sudah dibuat pada tahap sebelumnya. Mulai dari *activity diagram*, *class diagram* dan *mockup aplikasi* akan dibuat ada tahap ini. Pengerjaan tahap implementasi akan dibagi menjadi beberapa tahap. Berikut pembagian tahapan-tahapan pada tahap implementasi hasil.

4.1.1 Inisiasi Aplikasi

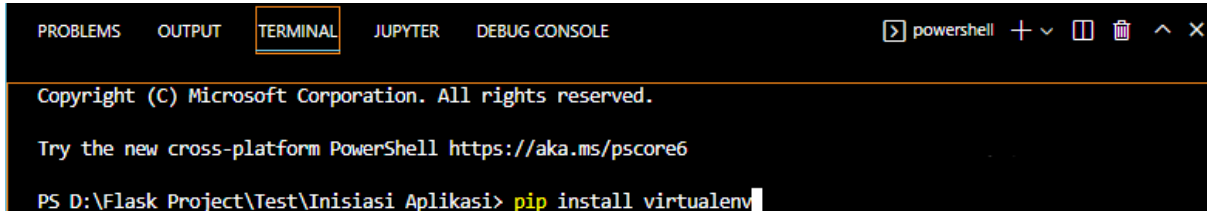
Sesuai namanya tahap inisiasi merupakan tahap yang mengawali kegiatan implementasi. Sebelum memasuki langkah awal pada tahap inisiasi hal pertama yang harus dilakukan yaitu memilih aplikasi *code editor*. Dimana pada pengembangan aplikasi kali ini pengembang menggunakan aplikasi Visual Studio Code sebagai aplikasi *code editor*. Alasan pemilihan aplikasi ini dikarenakan aplikasi ini merupakan salah satu *code editor* bersifat *open source* yang memiliki banyak ekstension yang dapat membantu pengembang selama masa pengembangan aplikasi. Setelah pemilihan aplikasi *code editor* tahap inisiasi dapat dilakukan dengan tahapan sebagai berikut.

1. Memastikan pada *device* yang digunakan sudah memiliki python *set-up*. Jika belum maka dapat mengunduhnya pada situs resmi python lalu melakukan instalasi pada *device* anda.
2. Membuat *folder* kosong pada *device* anda.
3. Membuka aplikasi Visual Studio Code (*Code editor*).
4. Membuka *folder* yang telah dibuat dengan Visual Studio Code seperti yang terlihat pada Gambar 4.1.



Gambar 4.1 Membuka *folder* dengan Visual Studio Code

5. Install “virtualenv” pada *terminal code editor* dengan *command* “pip install virtualenv” seperti pada Gambar 4.2 (Langkah ini tidak perlu dilakukan juga sudah menginstall virtualenv).



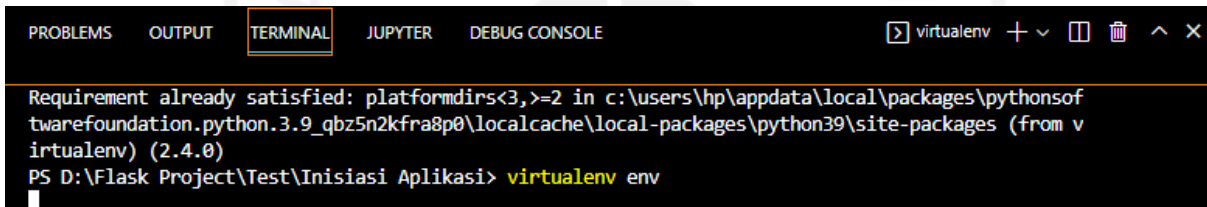
```

PROBLEMS OUTPUT TERMINAL JUPYTER DEBUG CONSOLE
Copyright (C) Microsoft Corporation. All rights reserved.
Try the new cross-platform PowerShell https://aka.ms/pscore6
PS D:\Flask Project\Test\Inisiasi Aplikasi> pip install virtualenv

```

Gambar 4.2 Install virtualenv pada Visual Studio Code

6. Membuat *environment* pada *terminal code editor* dengan *command* “virtualenv (nama environment anda)” seperti yang terlihat pada Gambar 4.3.



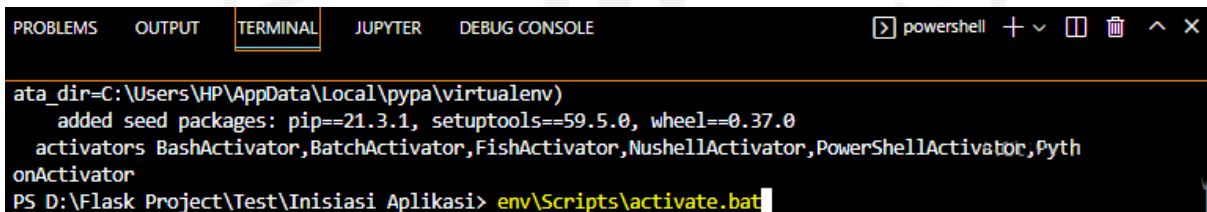
```

PROBLEMS OUTPUT TERMINAL JUPYTER DEBUG CONSOLE
Requirement already satisfied: platformdirs<3,>=2 in c:\users\hp\appdata\local\packages\pythonsoftw
arefoundation.python.3.9_qbz5n2kfra8p0\localcache\local-packages\python39\site-packages (from v
irtualenv) (2.4.0)
PS D:\Flask Project\Test\Inisiasi Aplikasi> virtualenv env

```

Gambar 4.3 Membuat virtualenv dengan nama env

7. Mengaktifkan virtualenv yang baru saja dibuat dengan *command* “env\Scripts\activate.bat” seperti yang terlihat pada Gambar 4.4.



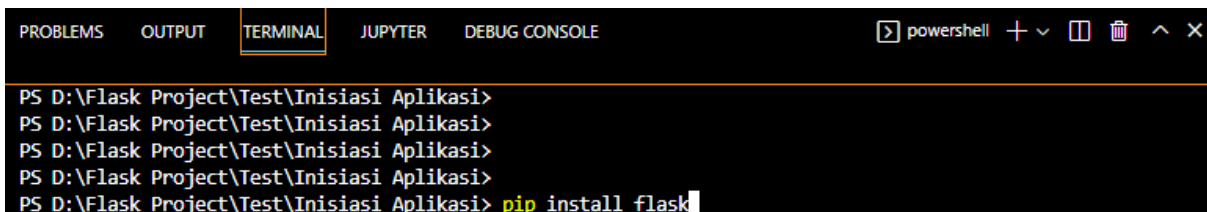
```

PROBLEMS OUTPUT TERMINAL JUPYTER DEBUG CONSOLE
ata_dir=C:\Users\HP\AppData\Local\pypa\virtualenv)
added seed packages: pip==21.3.1, setuptools==59.5.0, wheel==0.37.0
activators BashActivator,BatchActivator,FishActivator,NushellActivator,PowerShellActivator,Pyth
onActivator
PS D:\Flask Project\Test\Inisiasi Aplikasi> env\Scripts\activate.bat

```

Gambar 4.4 Aktivasi virtualenv

8. Menginstall flask pada *terminal code editor* dengan *command* “pip install flask” seperti pada Gambar 4.5 (Langkah ini bisa anda lewati apabila anda sudah pernah menginstall flask).



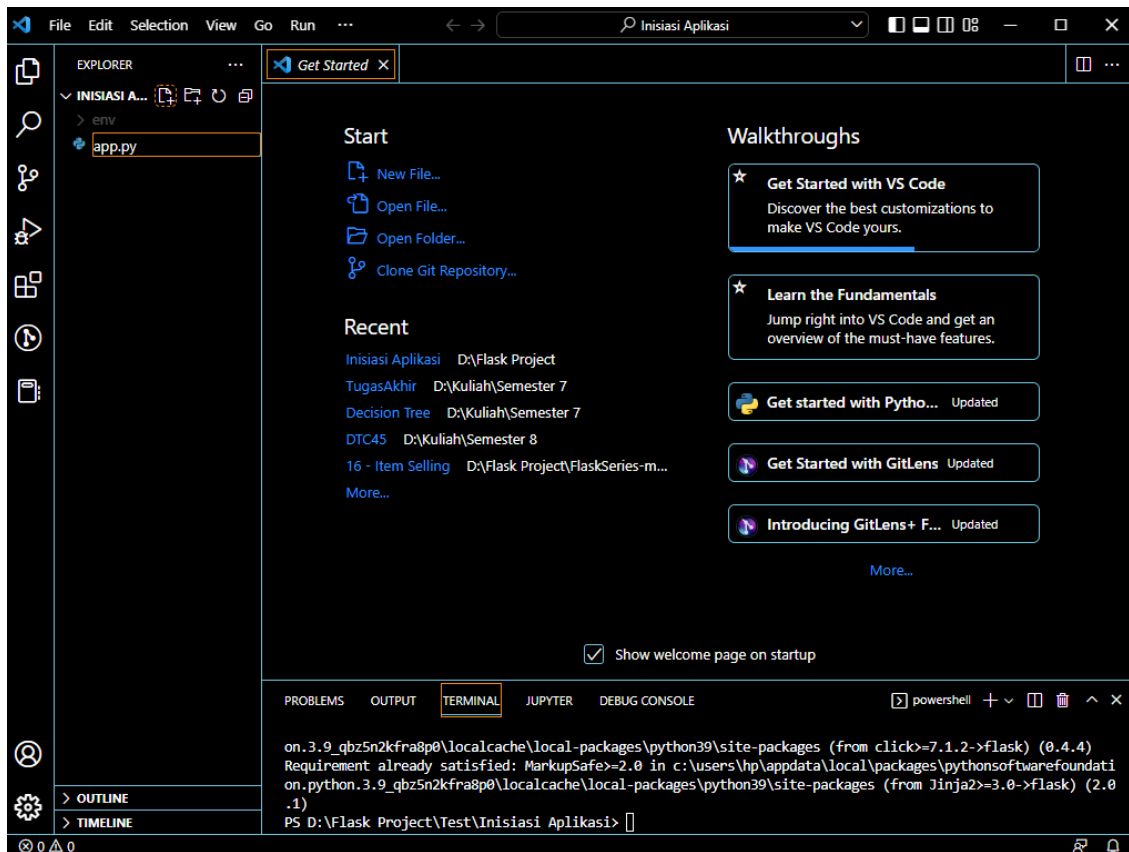
```

PROBLEMS OUTPUT TERMINAL JUPYTER DEBUG CONSOLE
PS D:\Flask Project\Test\Inisiasi Aplikasi>
PS D:\Flask Project\Test\Inisiasi Aplikasi>
PS D:\Flask Project\Test\Inisiasi Aplikasi>
PS D:\Flask Project\Test\Inisiasi Aplikasi>
PS D:\Flask Project\Test\Inisiasi Aplikasi> pip install flask

```

Gambar 4.5 Menginstall Flask pada *terminal code editor*

9. Membuat fail dengan ekstensi “.py” seperti yang terlihat pada Gambar 4.6.



Gambar 4.6 Membuat fail dengan nama "app.py"

10. Menulis kode program sebagai percobaan seperti yang terlihat pada Gambar 4.7.

```

1. from flask import Flask
2.
3. app = Flask(__name__)
4.
5. @app.route('/')
6. def index():
7.     return "Hello World"
8.
9. if __name__ == "__main__":
10.    app.run(debug=True)

```

Gambar 4.7 Contoh kode program

11. Menjalankan aplikasi percobaan dengan *command* "python app.py" pada *terminal* seperti pada Gambar 4.8.

Gambar 4.8 Menjalankan aplikasi percobaan

12. Maka aplikasi bisa diakses pada "localhost:5000".

4.1.2 Library Python

```
import graphviz
import base64
import os
import pyscreenshot
import random
import string
import pandas as pd
import numpy as np
import category_encoders as ce
from flask import Flask , render_template, request, redirect, session,
flash, url_for, send_file
from sklearn import tree
from werkzeug.utils import secure_filename
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report,
confusion_matrix
```

Gambar 4.9 Library yang digunakan

Library merupakan sebuah hal yang terpenting sebelum menentukan bahasa pemrograman yang ingin digunakan. Ketersediaan *library* dalam sebuah bahasa pemrograman bahkan dapat menjadi sebuah alasan dalam memilih suatu bahasa pemrograman karena *library* dapat membantu meringankan pekerjaan pengembang dalam masa pengembangan sebuah aplikasi. Tabel 4.1 merupakan tabel yang menjelaskan daftar library yang digunakan pada pengembangan aplikasi ini beserta fungsi yang dimiliki.

Tabel 4.1 Daftar *library* beserta fungsinya

No	Library	Fungsi
1	graphviz	Visualisasi pada diagram pohon keputusan.
2	base64	Melakukan <i>encode</i> pada variabel “pohon keputusan” dan <i>decode</i> untuk bisa ditampilkan pada antarmuka.
3	os	Membuat <i>folder</i> secara otomatis dalam aplikasi.
4	pyscreenshot	Melakukan <i>screenshot</i> .
5	random	Membuat nama <i>random</i> pada fail <i>screenshot</i> .
6	string	Membuat nama <i>random</i> pada fail <i>screenshot</i> .
7	pandas	Mendapatkan akses terhadap fail yang diunggah oleh pengguna.
8	numpy	Transformasi pada <i>array</i> .

9	flask	Membuka fitur yang terdapat pada flask.
10	sklearn	Komputasi algoritma <i>decision tree c4.5</i> dan visualisasi hasil komputasi.
11	werkzeug	Mengamankan nama fail yang telah diunggah oleh pengguna.

4.1.3 Implementasi Kode Program

a. Antarmuka

Bagian antarmuka atau *User Interface* (UI) merupakan bagian dimana pengguna dapat berinteraksi dengan sistem, dengan kata lain bagian ini merupakan penghubung antara pengguna dan sistem. Bagian ini juga biasa disebut dengan *front-end*. Pada aplikasi ini bagian antarmuka akan ditulis menggunakan bahasa pemrograman HTML dan Javascript. Penggunaan bahasa pemrograman Javascript hanya terdapat pada beberapa bagian kecil aplikasi agar terlihat lebih responsif ketika digunakan oleh pengguna.

Aplikasi ini juga menggunakan beberapa fitur yang terdapat pada python flask yaitu Jinja2 dengan penerapan halaman induk utama yang diturunkan kepada anak-anaknya. Penggunaannya terlihat dengan kode “{% %}”. Pada Jinja2 juga dapat menerapkan percabangan atau perulangan agar halaman dapat lebih diatur sesuai dengan keinginan pengembang. Gambar 4.10 merupakan contoh kode untuk halaman utama.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>
      {% block title %}
      {% endblock %}
    </title>
  </head>

  <body class="background">
    <h1 class="text-center">
      Analisa Data Menggunakan Metode Decision Tree with Python
      </img>
    </h1>
    <div class="text-center">
      </img>
    </div>
```



```

<h1 class="text-center"> Alan Chandra Darmawan </h1>
<h2 class="text-center"> 18523262 </h2>
{% block navbar %} {% endblock %}
<div class="container">
  {% with messages = get_flashed_messages(with_categories=true) %}
    {% if messages %}
      {% for category, message in messages %}
        <div class="alert alert-{{ category }} alert-
dismissible fade show" role="alert">
          <span class="text-white"> {{ message }} </span>
          <button type="button" class="btn-close" data-bs-
dismiss="alert" aria-label="Close">
            <span aria-hidden="true"></span>
          </button>
        </div>
      {% endfor %}
    {% endif %}
  {% endwith %}
</div>
{% block body %}    {% endblock %}
{% block script %}  {% endblock %}
</body>
</html>

```

Gambar 4.10 Kode pada halaman induk

Penggunaan bahasa pemrograman Javascript hanya terdapat pada beberapa bagian saja. Seperti pada *slider* atau pada tampilan tabel yang dimana hal itu ditujukan untuk membuat tampilan lebih responsif. Gambar 4.11 merupakan contoh penggunaan bahasa pemrograman Javascript pada aplikasi.

```

{% block script %}
<script>
  var slider = document.getElementById("slider");
  var selector = document.getElementById("selector");
  var SelectValue = document.getElementById("SelectValue");
  SelectValue.innerHTML = slider.value;

  slider.oninput = function()
  {
    SelectValue.innerHTML = this.value/10;
    selector.style.left = this.value + "&";
  }
</script>
{% endblock %}

```

Gambar 4.11 Penggunaan Javascript pada aplikasi

b. Server

Pada bagian server merupakan tempat dimana kode keseluruhan fungsi berada pada aplikasi. Bagian ini biasanya juga disebut sebagai bagian *back-end*. Pada aplikasi ini tidak menggunakan *database*. Fungsi *database* diganti dengan penggunaan “session” yang dimiliki oleh python flask. Namun “session” hanya dapat menyimpan nilai satu variabel saja, persis seperti yang dibutuhkan pada aplikasi ini. Variabel yang menempati isi “session” merupakan variabel dimana nama fail berada. Gambar 4.12 merupakan contoh salah satu fungsi yang terdapat pada aplikasi.

```
@app.route('/datapreprocessing') #(Route untuk Blank Load Upload Page)
def LoadDP():

    if "dataframe" in session: #Pengecekan session sudah terisi apa tidak
        namafile = session["dataframe"]
        df = pd.read_csv("uploads/{}".format(namafile))
        headings = df.columns #Menyimpan nama kolom
        return render_template("datapreprocessing.html", tnamafile = "{}"
        .format(namafile), headings = headings)

    else: #Apabila belum ada file yang di upload
        namafile = "Belum upload data"
        headings = "Null"
        flash("Anda belum mengupload data anda", "danger")
        return render_template("datapreprocessing.html", tnamafile = "{}"
        .format(namafile), headings = headings)
```

Gambar 4.12 Kode menampilkan halaman Data Preprocessing


4.2 Hasil Aplikasi

Setelah tahap implementasi telah usai, maka saatnya memasuki tahap hasil aplikasi. Pada tahap ini akan berisikan tentang dokumentasi dari hasil aplikasi utuh yang telah dirancang pada tahap perancangan sistem dan telah diimplementasikan pada tahap implementasi kode program.

Pada tahap ini akan ditampilkan total 6 halaman yang merupakan hasil jadi aplikasi. Keenam halaman ini berupa halaman *home* (halaman *landing*), *upload data*, *dataframe*, *data training & tes* dan *decision tree c4.5*. Tampilan aplikasi pada tahap ini akan menggunakan *dataset* “weather-numeric.csv” sebagai sampel *dataset* yang didapatkan dari aplikasi *machine learning* berbasis *desktop* WEKA dan akan menggunakan *input* “traintestsplitted” sebesar 0.2 serta pada “targety” berupa kolom “play”. Berikut dokumentasi hasil jadi aplikasi yang akan dirincikan sebagai berikut.

4.2.1 Home

Analisa Data Menggunakan Metode Decision Tree with Python




Alan Chandra Darmawan
18523262

DTC45 **HOME** Upload Data Dataframe Data Training & Set Decision Tree C.45 Screenshot

Apa itu DTC45 ?

DTC45 adalah Sebuah aplikasi berbasis web yang dikembangkan sebagai bentuk akhir atau luaran dari proyek Tugas Akhir yang akan mengklasifikasi data yang anda upload menggunakan algoritma Decision Tree C.45


Teknik Informatika, Fakultas Teknologi Industri
Universitas Islam Indonesia



Tonton video tutorial ?

DISINI


Bagaimana caranya ?



Upload Data Anda !

Upload data anda pada halaman upload page, pastikan data yang anda upload sudah sesuai dengan apa yang diinginkan oleh sistem.


[Upload Page.](#) →



Lengkapi Data Anda

Pastikan anda juga tidak lupa melengkapi data dengan mengisi trainintestsplrit dan sudah memilih target Y dari salah satu kolom pada data yang anda upload.

[Dataframe Page.](#) →



Lihat Hasil Klasifikasi

Ketika semua kebutuhan yang dibutuhkan oleh sistem telah anda lengkapi, silahkan lihat hasil klasifikasi data anda pada halaman Decision Tree C.45

[Decision Tree C.45 Page.](#) →

Algorithm Decision Tree C45 Pengertian +/-

Decision tree C4.5 adalah algoritma yang digunakan untuk mengklasifikasi data dan menghasilkan pohon keputusan sebagai salah satu hasilnya. Algoritma merupakan perpanjangan dari algoritma ID3. Algoritma ini dianggap lebih baik karena mampu berurusan dengan atribut kontinu dan diskrit.

Gambar 4.13 Hasil halaman Home

4.2.2 UploadData

Analisa Data Menggunakan Metode Decision Tree with Python

UNIVERSITAS ISLAM INDONESIA

Alan Chandra Darmawan
18523262

DTC45 Home **UPLOAD DATA** Dataframe Data Training & Set Decision Tree C.45 Screenshot

Data yang anda upload adalah : weather-numeric.csv

Apakah ada kolom yang ingin anda hapus ?
[KLIK DISINI](#)

Upload Data Anda!

Upload Data Anda!
Choose File | No file chosen
[SUBMIT](#)

outlook	temperature	humidity	windy	play
overcast	83.0	86.0	FALSE	yes
overcast	64.0	65.0	TRUE	yes
overcast	72.0	90.0	TRUE	yes
overcast	81.0	75.0	FALSE	yes
rainy	70.0	96.0	FALSE	yes
rainy	68.0	80.0	FALSE	yes
rainy	65.0	70.0	TRUE	no
rainy	75.0	80.0	FALSE	yes
rainy	71.0	91.0	TRUE	no
sunny	85.0	85.0	FALSE	no

Showing 1 to 10 of 14 entries

Belum Punya Data ?
Coba beberapa website open source berikut !

- [KAGGLE.COM](#)
- [KAGGLE.COM](#)
- [GOOGLE TRENDS](#)
- [DATAPORTALS.ORG](#)
- [GOOGLE PUBLIC DATA](#)

Gambar 4.14 Hasil halaman Upload Data

4.2.3 Data Preprocessing

Analisa Data Menggunakan Metode Decision Tree with Python

UNIVERSITAS ISLAM INDONESIA

Alan Chandra Darmawan
18523262

DTC45 Home **DATA PREPROCESSING** Dataframe Data Training & Set Decision Tree C.45 Screenshot

Data Preprocessing

Apakah ada kolom yang ingin anda hapus ?

- outlook
- temperature
- humidity
- windy
- play

Kolom terpilih :


[SUBMIT](#) [KEMBALI](#)


ReadMe

Kolom yang anda hapus akan terhapus permanen oleh file yang diterima oleh sistem
Apabila anda ingin menggunakan kolom yang sudah anda hapus, maka anda harus mengupload ulang data anda.


Gambar 4.15 Hasil halaman Data Preprocessing

4.2.4 Dataframe

Analisa Data Menggunakan Metode Decision Tree with Python 



Alan Chandra Darmawan
18523262

DTC45 Home Upload Data **DATAFRAME** Data Training & Set Decision Tree C45 

Belum upload data

?

Kolom Diskrit

Tidak diketahui

?

Datashape

Tidak diketahui

?

Kolom Kategorikal

Tidak diketahui

Dataframe [Preview](#) [ReadMe](#)

Show 10 entries Search:

N	u	I	I
No data available in table			

Showing 0 to 0 of 0 entries Previous Next

Null Values


N	u	I	I
Total null values : ?			


Statistik Deskriptif

	N	u	I	I
count				
unique				
top				
freq				
mean				
std				
min				
25%				
50%				
75%				
max				

Gambar 4.16 Hasil halaman Dataframe

4.2.5 Data Training & Test

Analisa Data Menggunakan Metode Decision Tree with Python 



Alan Chandra Darmawan
18523262

DTC45 Home Upload Data Dataframe **DATA TRAINING & SET** Decision Tree C45 Screenshot

Belum upload data

Kolom X

Tidak diketahui

Train Tes Split

?

Kolom Y

Tidak diketahui

Preview X X.Train X.Test

Show 10 entries

Search:

T i d a k d i k e t a h u i

No data available in table

Showing 0 to 0 of 0 entries Previous Next

Kolom X Kolom Y

Kolom-kolom yang digunakan sebagai label masukan atau input dalam pelatihan algoritma

Train Test

Bagian dataset yang akan dilatih untuk membuat prediksi maupun menjalankan algoritma

Preview Y y.train y.test

Show 10 entries

Search:

Tidak diketahui


No data available in table

Showing 0 to 0 of 0 entries Previous Next

Gambar 4.17 Hasil halaman Data Training & Test

4.2.6 Decision Tree C4.5

Analisa Data Menggunakan Metode Decision Tree with Python



Alan Chandra Darmawan
18523262

DTCC45 Home Upload Data Dataframe Data Training & Set **DECISION TREE C4.5** Screenshoot

Belum upload data

Kolom X Tidak diketahui
Train Tes Split ?
Kolom Y Tidak diketahui

Pohon Keputusan

Pohon Keputusan

Matriks Konfusi

Training-set Akurasi : ?
Training-set Skor : ?
Test-set Skor : ?
Akurasi Model : ?
Training time :

Precision Recall
Precision : Nilai yang mewakili suatu akurasi dalam hasil klasifikasi yang dilakukan.

F1 Score Support
F1-Score : Nilai rata-rata tertimbang dari precision dan recall dalam hasil klasifikasi yang dilakukan.

Laporan Klasifikasi

Tidak diketahui

accuracy
macro_avg
weighted_avg

LULAMAN LAPORAN DOWNLOAD MODEL

Gambar 4.18 Hasil halaman Decision Tree C4.5

4.3 Skenario Pengujian Hasil

4.3.1 Tahap Demonstrasi

Pada tahap demonstrasi aplikasi akan diberikan beberapa sampel dataset baik yang memenuhi sudah syarat maupun yang tidak. Hal ini dianggap perlu dilakukan dengan tujuan untuk melihat apakah sistem mampu mengenali syarat yang telah ditentukan dan menolak sampel yang tidak memenuhi syarat tersebut. Berikut beberapa syarat agar fail yang diunggah diterima oleh sistem.

1. Fail yang diunggah harus memiliki tipe ekstensi “.csv”.
2. Fail yang diunggah harus memiliki lebih dari sama dengan 2 kolom (1 kolom sebagai variabel y dan yang lainnya sebagai x).

Maka pada tahap demonstrasi total akan diberikan lima sampel fail yang dimana terdiri dua sampel yang tidak memenuhi syarat dan tiga sampel lainnya memenuhi syarat. Pada dua fail yang tidak memenuhi syarat tersebut salah satu fail tidak memiliki ekstensi “.csv” dan yang satunya sudah memiliki ekstensi “.csv” namun hanya memiliki satu kolom di dalamnya. Sedangkan pada tiga sampel yang sudah memenuhi syarat akan berisikan satu fail yang memiliki baik kolom kategorikal maupun kolom diskrit didalamnya, satu fail hanya akan memiliki kolom kategorikal dan fail terakhir hanya akan memiliki kolom diskrit saja. Hal ini dilakukan dengan tujuan memberikan gambaran terhadap bagaimana aplikasi yang dikembangkan dapat tetap bekerja walaupun diberikan *input* yang berbeda-beda.

Pada tahap ini tidak akan menampilkan seluruh hasil dari aplikasi. Melainkan hanya menunjukkan tampilan dari beberapa fitur saja yang terdapat pada aplikasi seperti bagian upload data, input “*traintestsplit & targety*” dan hasil klasifikasi berupa diagram pohon keputusan saja. Berikut beberapa sampel fail yang telah ditentukan.

- a. bukancsv.txt

Sampel *dataset* ini tidak memenuhi kriteria pada syarat fail yang diinginkan oleh sistem. Penggunaan *dataset* ini sebagai sampel ialah untuk menunjukkan bahwa aplikasi mampu menolak *dataset* ini dikarenakan tidak memiliki tipe ekstensi “.csv”. Gambar 4.19 menunjukkan sampel pada *dataset* bukancsv.txt.


```
Suhu,Kelembaban,Berangin,Bermain
29.4,85, Tidak Berangin,Tidak
26.6,90,Berangin,Tidak
28.3,78,Tidak Berangin,Ya
21.1, 96,Tidak Berangin,Ya
20,80,Tidak Berangin,Ya
18.3,70,Berangin,Tidak
17.7,65,Berangin,Ya
22.2,95,Tidak Berangin,Tidak
20.5,70,Tidak Berangin,Ya
23.8,80,Tidak Berangin,Ya
```

Gambar 4.19 *Dataset* "bukancsv.txt"

b. 1Kolom.csv

Pada sampel *dataset* ini tidak memenuhi kriteria pada syarat fail yang diinginkan oleh sistem. Walaupun sudah memiliki tipe ekstensi ".csv", sampel *dataset* ini hanya memiliki 1 kolom saja. Gambar 4.20 merupakan sampel *dataset* 1Kolom.csv.

outlook
sunny
sunny
overcast
rainy
rainy
overcast
sunny
sunny
rainy
sunny
overcast
overcast
rainy

Gambar 4.20 *Dataset* "1Kolom.csv"

c. weather-numeric.csv

Sampel *dataset* ini merupakan salah satu sampel yang telah memenuhi syarat yang diinginkan oleh sistem. Sampel ini didapatkan dari salah satu aplikasi *machine learning* berbasis *desktop* WEKA (Waikato Environment for Knowledge Analysis). Pada sampel ini terdapat baik kolom kategorikal maupun kolom diskrit. Nantinya pada isian "traintestspit & targety" pada pengujian sampel ini akan menggunakan 0.2 sebagai nilai "traintestsplit" dan pada isian "targety" berisikan kolom "play". Gambar 4.21 merupakan isi dari sampel *dataset* weather-numeric.csv.

weather-numeric.csv [Preview](#) [ReadMe](#)

Show entries

Search:

outlook	temperature	humidity	windy	play
overcast	83.0	86.0	FALSE	yes
overcast	64.0	65.0	TRUE	yes
overcast	72.0	90.0	TRUE	yes
overcast	81.0	75.0	FALSE	yes
rainy	70.0	96.0	FALSE	yes
rainy	68.0	80.0	FALSE	yes
rainy	65.0	70.0	TRUE	no
rainy	75.0	80.0	FALSE	yes
rainy	71.0	91.0	TRUE	no
sunny	85.0	85.0	FALSE	no

Showing 1 to 10 of 14 entries

Previous **1** 2 Next

Gambar 4.21 *Dataset* "weather-numeric.csv"

d. diabetes.csv

Dataset ini juga merupakan salah satu sampel yang telah memenuhi syarat yang diinginkan oleh sistem. Sampel ini didapatkan dari salah satu *website data open source* yaitu Kaggle. Yang dimana pada sampel ini terdapat 768 baris dan 9 kolom. Seluruh kolom pada sampel ini merupakan kolom tipe diskrit. Nantinya pada isian "traintestsplit & targety" pada pengujian sampel ini akan menggunakan 0.4 sebagai nilai "traintestsplit" dan pada isian "targety" berisikan kolom "outcome". Gambar 4.22 menunjukkan isi dari sampel *dataset* diabetes.csv.

diabetes.csv Preview ReadMe

Show: 10 entries

Search:

Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DbtsPdgrFnctn	Age	Outcom
0	137	40	35	168	43.1	2.288	33	1
0	118	84	47	230	45.8	0.551	31	1
0	180	66	39	0	42	1.893	25	1
0	100	88	60	110	46.8	0.962	31	0
0	146	82	0	0	40.5	1.781	44	0
0	105	64	41	142	41.5	0.173	22	0
0	109	88	30	0	32.5	0.855	38	1
0	131	0	0	0	43.2	0.27	26	1
0	101	65	28	0	24.6	0.237	22	0
0	125	96	0	0	22.5	0.262	21	0

Showing 1 to 10 of 768 entries

Previous 1 2 3 4 5 ... 77 Next

Gambar 4.22 Dataset "diabetes.csv"

e. car_evaluation.csv

Dataset ini merupakan sampel terakhir yang akan diuji kepada sistem selama tahap demonstrasi. Sampel ini juga merupakan salah satu *dataset* yang telah memenuhi syarat yang diinginkan oleh sistem. Sampel ini juga didapatkan dari salah satu *website data open source* Kaggle. Yang dimana pada sampel ini terdapat 1728 baris dan 7 kolom. Seluruh kolom pada sampel ini merupakan kolom tipe kategorikal. Nantinya pada isian "traintestspit & targety" pada pengujian sampel ini akan menggunakan 0.5 sebagai nilai "traintestsplit" dan pada isian "targety" berisikan kolom "class". Gambar 4.23 merupakan sampel *dataset* car_evaluation.csv.

car_evaluation.csv Preview ReadMe

Show entries

Search:

buying	maint	doors	persons	lug_boot	safety	class
high	vhigh	2	2	small	low	unacc
high	vhigh	2	2	small	med	unacc
high	vhigh	2	2	small	high	unacc
high	vhigh	2	2	med	low	unacc
high	vhigh	2	2	med	med	unacc
high	vhigh	2	2	med	high	unacc
high	vhigh	2	2	big	low	unacc
high	vhigh	2	2	big	med	unacc
high	vhigh	2	2	big	high	unacc
high	vhigh	2	4	small	low	unacc

Showing 1 to 10 of 1,728 entries

Previous **1** 2 3 4 5 ... 173 Next

Gambar 4.23 Dataset "car_evaluation.csv"

4.3.2 Rangkuman Hasil

Bagian ini akan menunjukkan kepada bagian input "targety" yang dimana sistem mampu memberi prioritas kepada kolom kategorikal sebagai pilihan utama dan apabila tidak terdapat kolom kategorikal pada fail yang diunggah oleh pengguna maka pilihan akan berubah menjadi kolom diskrit yang terdapat pada fail. Dan pada bagian ini juga akan menunjukkan diagram pohon keputusan sebagai salah satu visualisasi hasil klasifikasi yang telah dilakukan oleh sistem.

- bukancsv.txt

Anda belum mengupload data anda

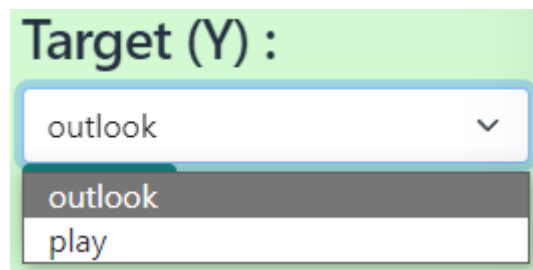
Gambar 4.24 Notifikasi penolakan dataset "bukancsv.txt"

- 1Kolom.csv

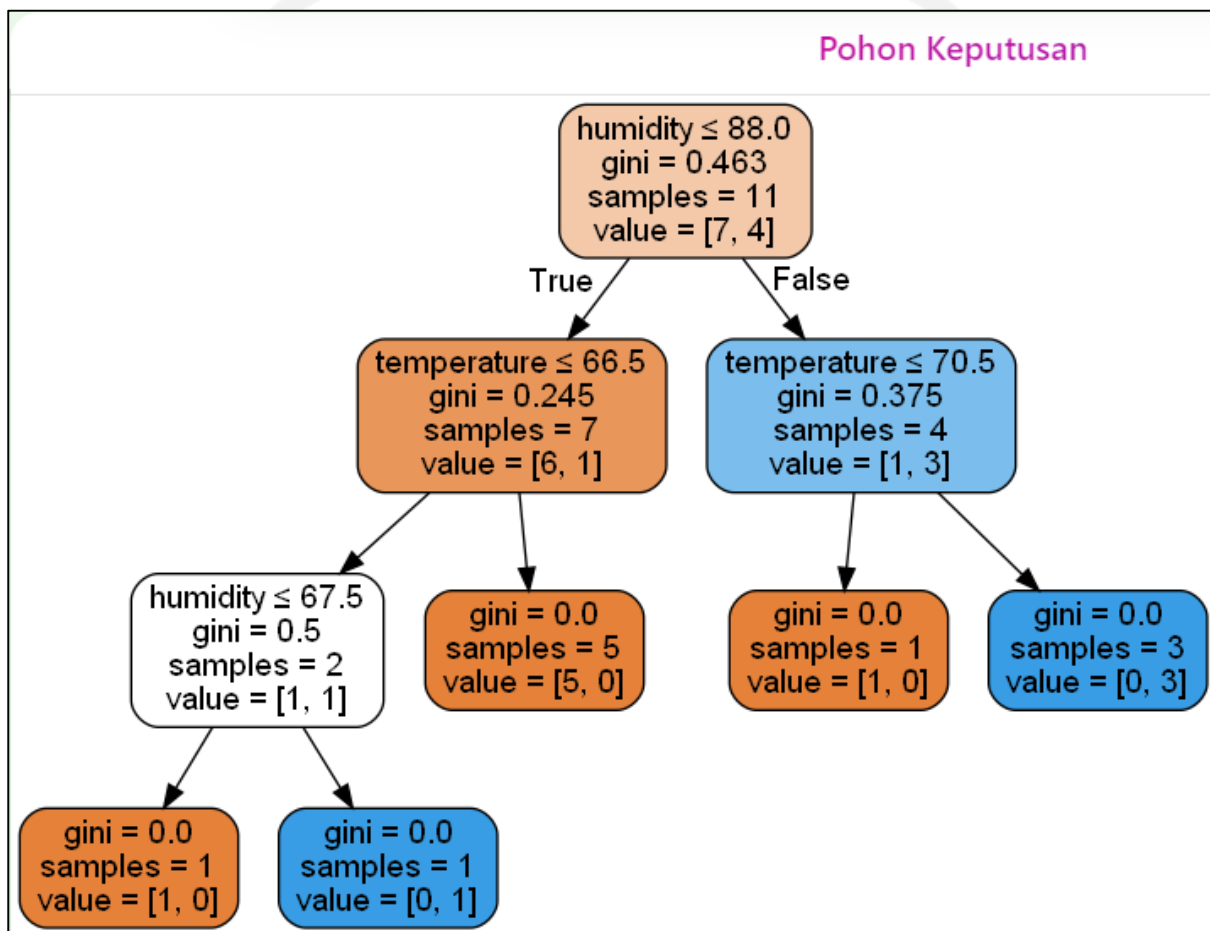
Data yang anda upload memiliki kolom kurang dari 2

Gambar 4.25 Notifikasi penolakan terhadap dataset "1kolom.csv"

c. weather-numeric.csv

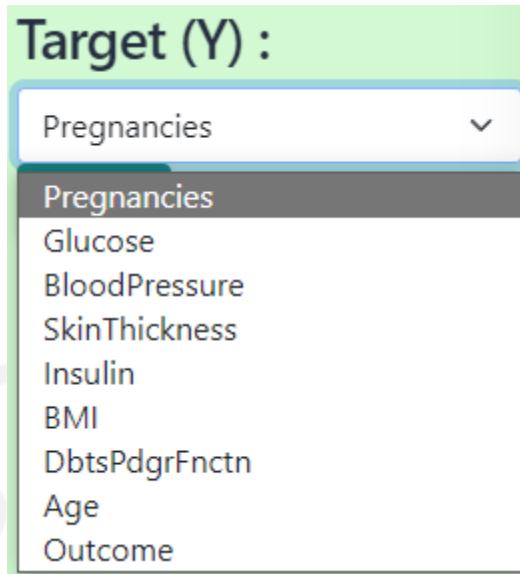


Gambar 4.26 Pilihan target(y) pada dataset "weather-numeric.csv"

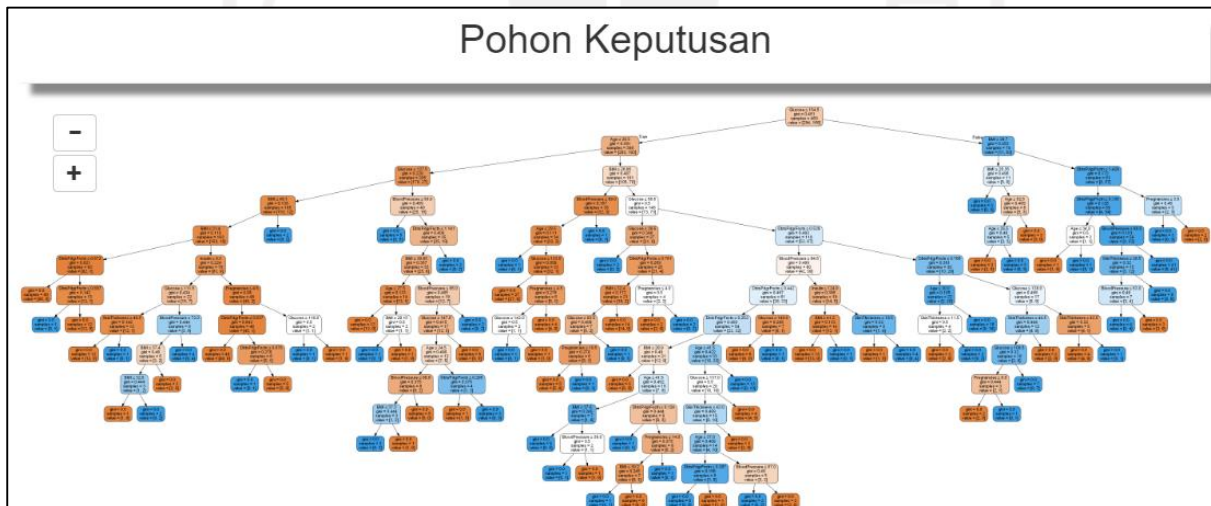


Gambar 4.27 Diagram pohon keputusan pada dataset "weather-numeric.csv"

d. diabetes.csv

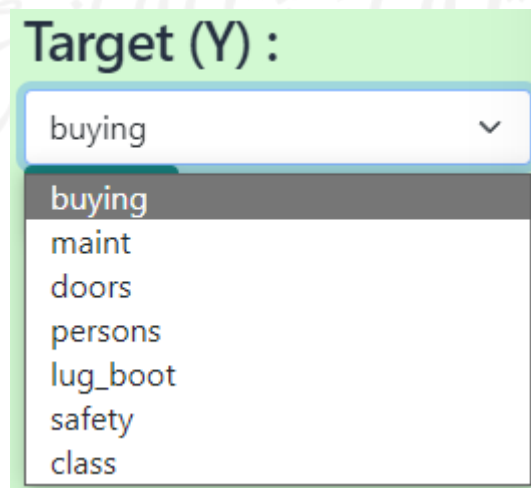


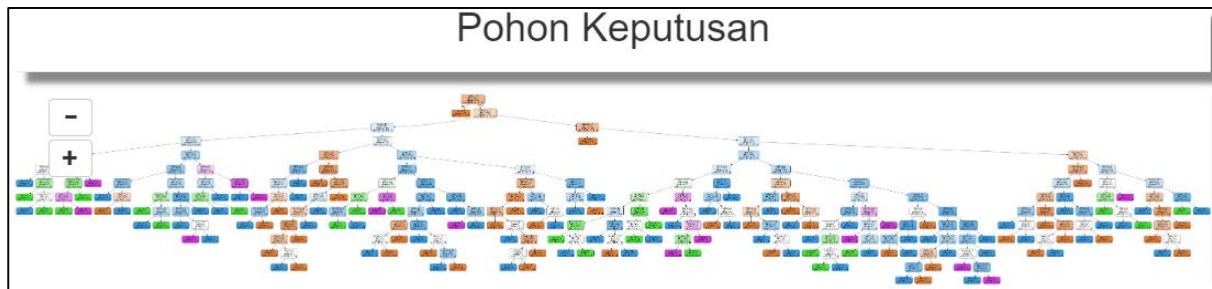
Gambar 4.28 Pilihan target(y) pada *dataset* "diabetes.csv"



Gambar 4.29 Diagram pohon keputusan pada *dataset* "diabetes.csv"

e. *car_evaluation.csv*



Gambar 4.30 Pilihan target(y) pada *dataset* "car_evaluation.csv"Gambar 4.31 Diagram pohon keputusan pada *dataset* "car_evaluation.csv"

4.4 Hasil Pengujian

4.4.1 Black Box Testing

Pada pengujian *black box testing* akan menguji segala sesuatu yang telah diharapkan kepada aplikasi yang terdapat pada kebutuhan fungsional aplikasi dengan hasil akhir aplikasi. Tabel 4.2 menjelaskan hasil uji *black box testing* pada aplikasi ini.

Tabel 4.2 Hasil *Black Box Testing* aplikasi

No.	Kebutuhan fungsional	Pengujian	Hasil yang diharapkan	Hasil Pengujian	Status
Home					
1	Membuka video tutorial	Menekan <i>button</i> "Disini"	Membuka video tutorial yang terdapat pada situs Youtube	Sesuai dengan harapan	Diterima
Upload Data					
2	<i>Input dataset</i>	Mengunggah fail yang memenuhi syarat	Fail diterima oleh aplikasi	Sesuai dengan harapan	Diterima
3		Mengunggah fail yang tidak memenuhi syarat	Fail ditolak oleh aplikasi	Sesuai dengan harapan	Diterima
Data Preprocessing					

4	Meleakukan data preprocessing	Menghapus satu atau lebih kolom (Kolom tersisa lebih dari sama dengan 2)	Kolom berhasil dihapus dan fail di-update oleh aplikasi	Sesuai dengan harapan	Diterima
5		Menghapus satu atau lebih kolom (Kolom tersisa hanya 1)	Fail menjadi tidak memenuhi syarat dan ditolak oleh aplikasi	Sesuai dengan harapan	Diterima
Dataframe					
6	<i>Input</i> “traintestsplit & targety”	Melakukan <i>input</i> pada fail yang memiliki baik kolom kategorikal maupun kolom diskrit	<i>Input</i> berhasil diterima dan aplikasi memberikan prioritas kepada kolom kategorikal sebagai pilihan “targety”	Sesuai dengan harapan	Diterima
7		Melakukan input pada fail yang hanya memiliki kolom diskrit	<i>Input</i> berhasil diterima dan aplikasi memberikan pilihan “targety” pada kolom diskrit	Sesuai dengan harapan	Diterima
8	Melihat statistik deskriptif data	Melihat tabel statistik deskriptif data	Aplikasi menampilkan statistik deskriptif <i>dataset</i>	Sesuai dengan harapan	Diterima

Data Training & Tes					
9	Melihat partisi data	Membuka halaman Data Training & Tes dan melihat partisi data	Aplikasi mampu memecah data sesuai dengan <i>input</i> yang diberikan oleh pengguna dan menampilkannya	Sesuai dengan harapan	Diterima
Decision Tree C4.5					
10	Melihat hasil klasifikasi data	Membuka halaman “Decision Tree C4.5” dan melihat hasil klasifikasi data.	Aplikasi mampu menampilkan diagram pohon keputusan, matriks konfusi, nilai hasil klasifikasi dan laporan klasifikasi	Sesuai dengan harapan	Diterima
11	Mengunduh model pelatihan	Membuka halaman “Decision Tree C4.5” dan menekan <i>button</i> “Download Model”.	Aplikasi mampu mengirimkan model pelatihan dalam format “.pkl” ke <i>device</i> pengguna	Sesuai dengan harapan	Diterima

Berdasarkan penjabaran pada pengujian *black box* yang sudah dilakukan, maka semua kebutuhan fungsional yang sudah direncanakan pada tahap perancangan sistem dapat dilakukan oleh hasil akhir aplikasi dan sudah memenuhi dari apa yang diharapkan oleh pengembang.

4.4.2 Software Testing

Pada tahap *software testing* aplikasi akan diuji terhadap salah satu aplikasi *machine learning* berbasis *desktop* yaitu WEKA (Waikato Environment for Knowledge Analysis). Pengujian ini dilakukan dengan tujuan untuk mendapatkan komparasi terhadap aplikasi serupa.

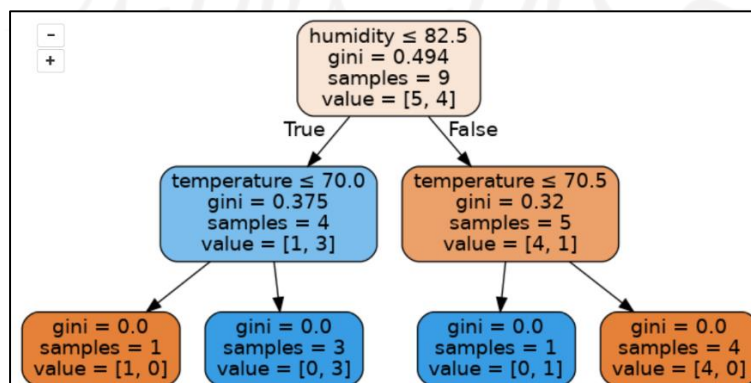
Pada pengujian ini akan membandingkan hasil dari klasifikasi dengan melakukan algoritma, metode dan dataset yang sama. Pada aplikasi WEKA algoritma *decision tree c4.5* bernama J48. Pengujian ini menggunakan *dataset* “weather-numeric.csv”, “heart.csv” dan “superstore_data.csv”. Tabel 4.3 merupakan tabel perbandingan antara aplikasi dan aplikasi WEKA.

Tabel 4.3 Perbandingan antara aplikasi yang dikembangkan dengan WEKA

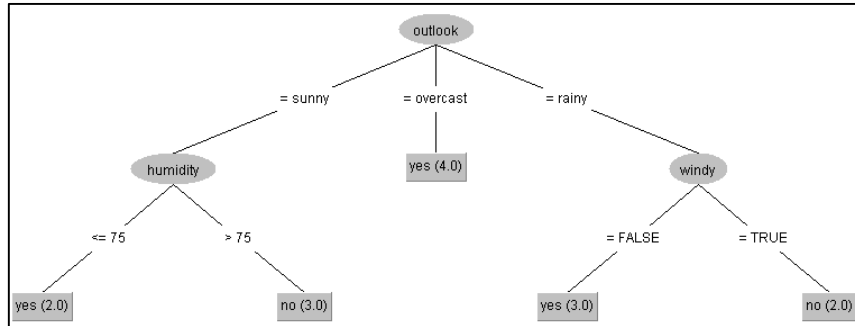
No.	Perbandingan	Aplikasi yang dikembangkan	WEKA
1	Basis aplikasi	Web	Desktop
2	Bahasa Pemrograman	Python	Java
3	Teknik pengujian hasil	Traintestsplit (<i>percentage split</i> pada aplikasi WEKA)	<i>Training test, supplied test set, cross-validation</i> dan <i>percentage split</i> (Traintestsplit pada aplikasi yang dikembangkan)
4	Format fail yang dapat digunakan	.csv	.arff, .arff.gz, .names, .data, .json, .json.gz, .libsvm, .m, .dat, .bsi, .xrff dan .xrff.gz

a. Dataset “weather-numeric.csv”

Dataset ini terdiri dari 14 baris dan 5 kolom, pengujian menggunakan dataset ini akan dilakukan dengan menggunakan pilihan *test-options* “Percentage split” sebesar 70% pada WEKA sedangkan pada aplikasi yang dikembangkan dilakukan “traintestsplit” sebesar 0.3, dimana pada masing-masing aplikasi akan membagi data menjadi *data training* sebesar 0.7 dari dataset dan 0.3 menjadi *data testing*. Untuk label output dipilih kolom “play”. Berikut hasil perhitungan dari masing-masing aplikasi.



Gambar 4.32 Pohon keputusan “weather-numeric” pada DTC45



Gambar 4.33 Pohon keputusan “weather-numeric” pada WEKA



Gambar 4.34 Hasil Klasifikasi “weather-numeric” pada DTC45

```

=== Evaluation on test split ===
Time taken to test model on test split: 0 seconds

=== Summary ===

Correctly Classified Instances      1      25  %
Incorrectly Classified Instances   3      75  %
Kappa statistic                    -0.5
Mean absolute error                 0.75
Root mean squared error             0.866
Relative absolute error             150  %
Root relative squared error        164.3168 %
Total Number of Instances          4

=== Detailed Accuracy By Class ===

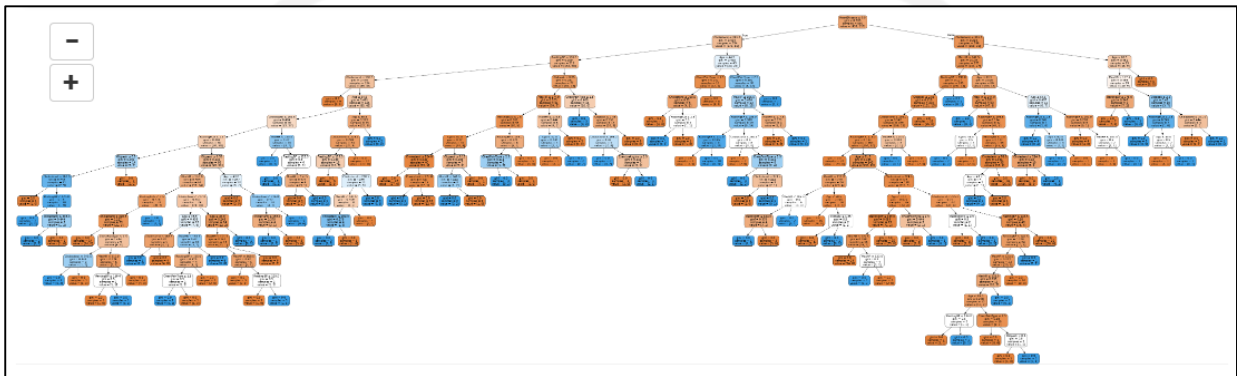
          TP Rate  FP Rate  Precision  Recall  F-Measure  MCC  ROC Area  PRC Area  Class
0.000  0.500  0.000  0.000  0.000  -0.577  0.250  0.500  no
0.500  1.000  0.333  0.500  0.400  -0.577  0.250  0.417  yes
Weighted Avg.  0.250  0.750  0.167  0.250  0.200  -0.577  0.250  0.458

=== Confusion Matrix ===
a b  <-- classified as
0 2 | a = no
1 1 | b = yes
  
```

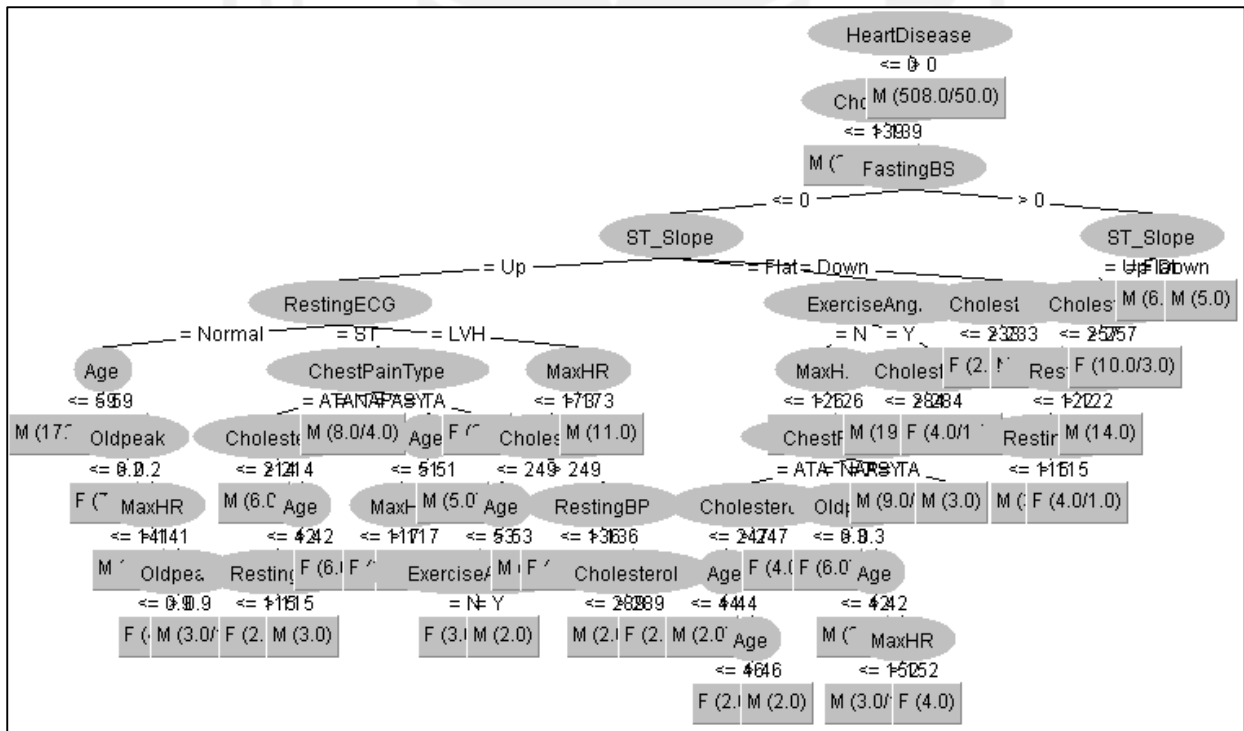
Gambar 4.35 Hasil Klasifikasi “weather-numeric” pada WEKA

b. Dataset “heart.csv”

Dataset ini terdiri dari 918 baris dan 12 kolom, pengujian menggunakan dataset ini akan dilakukan dengan menggunakan “Percentage split” sebesar 60% pada WEKA sedangkan pada aplikasi yang dikembangkan dilakukan “traintestsplrit” sebesar 0.4, dimana pada masing-masing aplikasi akan membagi data menjadi *data training* sebesar 0.6 dari dataset dan 0.4 menjadi *data testing*. Untuk label output dipilih kolom “Sex”. Berikut hasil perhitungan dari masing-masing aplikasi.



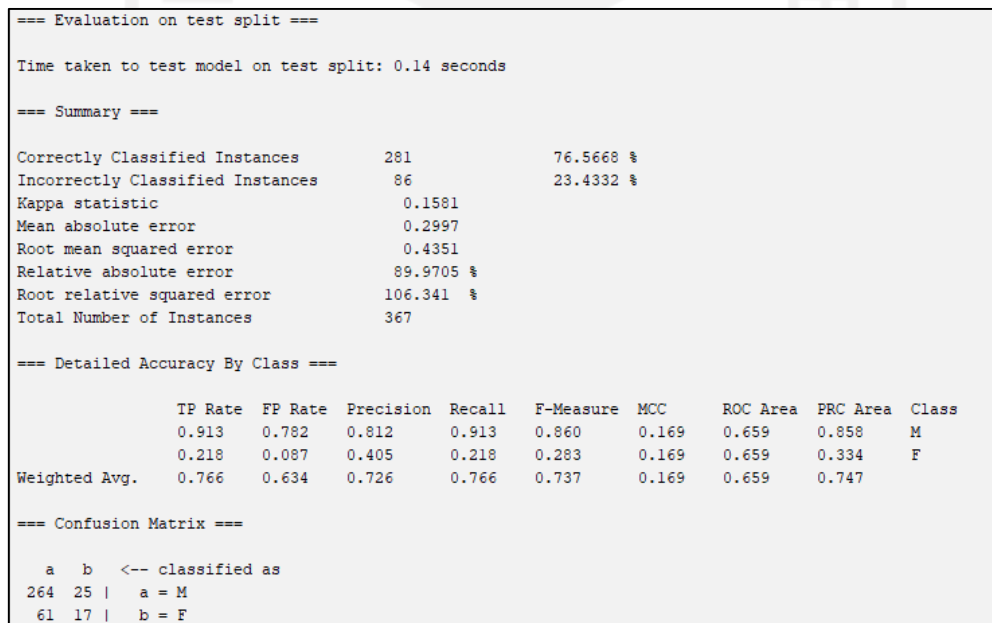
Gambar 4.36 Pohon keputusan “heart” pada DTC45



Gambar 4.37 Pohon keputusan “heart” pada WEKA



Gambar 4.38 Hasil Klasifikasi “heart” pada DTC45

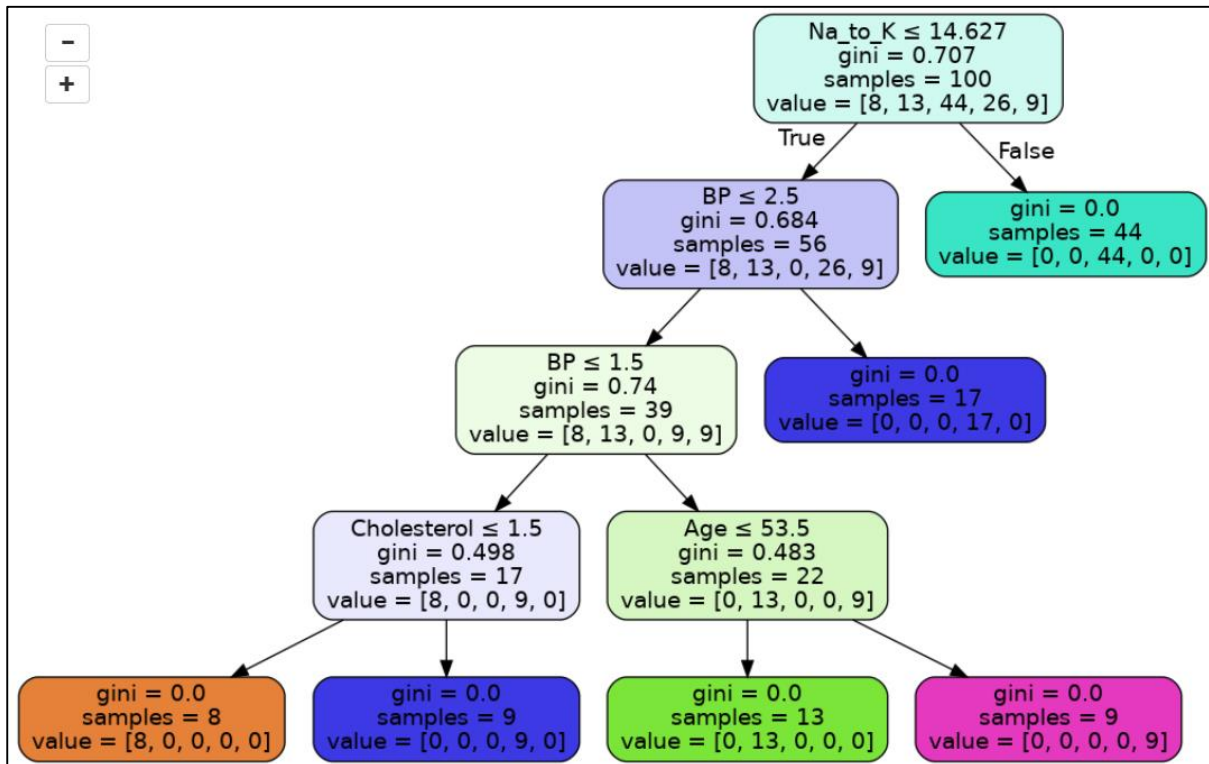


Gambar 4.39 Hasil Klasifikasi “heart” pada WEKA

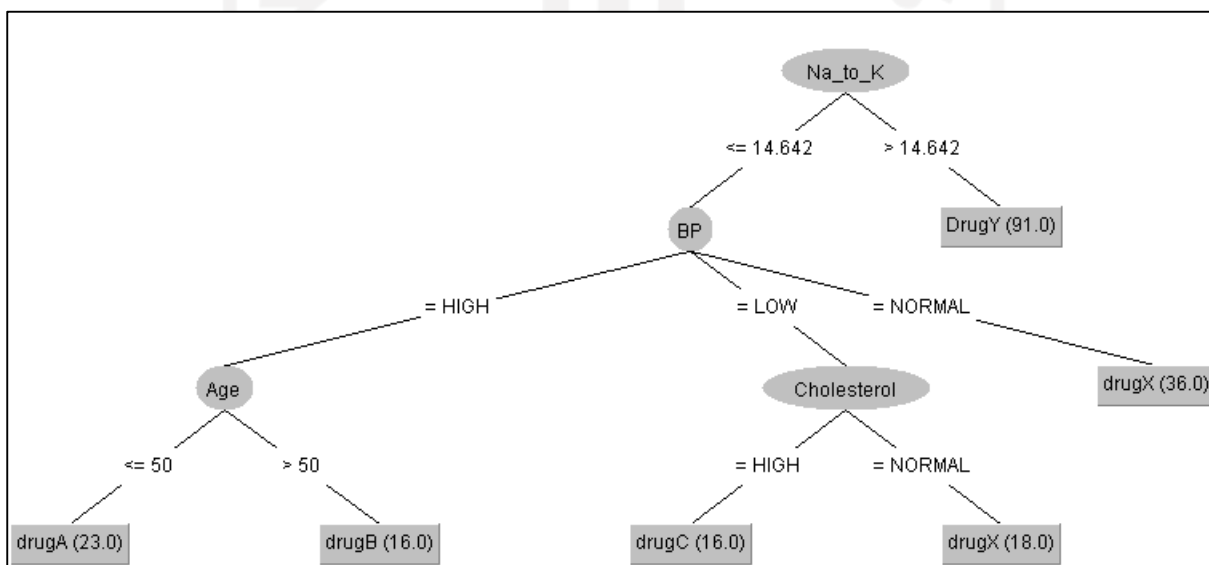
c. Dataset “drug200.csv”

Dataset ini terdiri dari 200 baris dan 6 kolom, pengujian menggunakan dataset ini akan dilakukan dengan menggunakan “Percentage split” sebesar 50% pada WEKA

sedangkan pada aplikasi yang dikembangkan dilakukan “traintestsplit” sebesar 0.5, dimana pada masing-masing aplikasi akan membagi data menjadi *data training* sebesar 0.5 dari dataset dan 0.5 menjadi *data testing*. Untuk label output dipilih kolom “Drug”. Berikut hasil perhitungan dari masing-masing aplikasi.



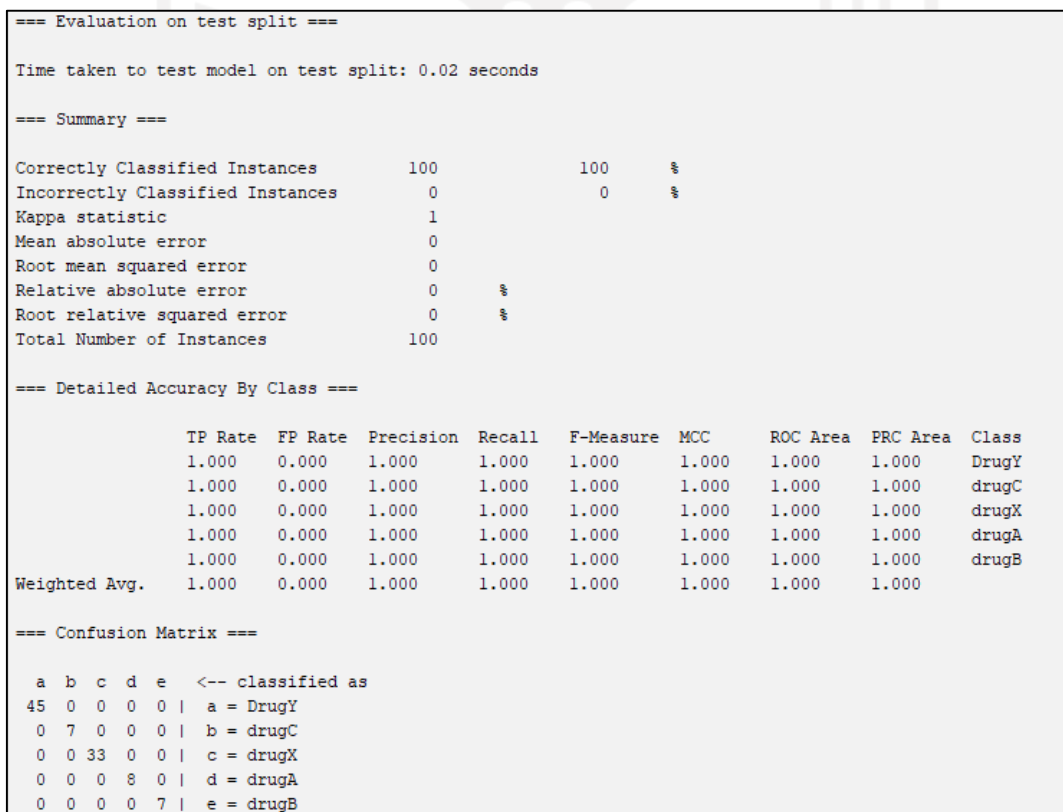
Gambar 4.40 Pohon Keputusan “drug200” pada DTC45



Gambar 4.41 Pohon Keputusan “drug200” pada WEKA



Gambar 4.42 Hasil Klasifikasi “drug200” pada DTC45



Gambar 4.43 Hasil Klasifikasi “drug200” pada WEKA

4.4.3 User Acceptance Test

Pengujian ini dilakukan dengan menguji langsung aplikasi kepada penggunanya, terdapat 6 pengguna yang melakukan pengujian ini. Pada awal pengujian pengguna ditanya apakah sudah tahu mengenai *machine learning* atau tidak, 3 diantaranya mengaku sudah mengetahui mengenai *machine learning* sementara 3 lainnya belum mengetahui tentang *machine learning*. Setelah itu pengguna melakukan kegiatan seperti menonton *video tutorial*, menggunakan aplikasi dan terakhir menjawab beberapa pertanyaan. Tabel 4.4 dan Tabel 4.5 merupakan jawaban dari pengguna setelah melakukan kegiatan pengujian.

Tabel 4.6 Hasil pengujian terhadap pengguna yang sudah mengerti *machine learning*.

No	Pertanyaan	SS (Sangat Setuju)	S (Setuju)	C (Cukup)	KS (Kurang Setuju)	TS (Tidak Setuju)
1	Jika anda memiliki data untuk diklasifikasikan menggunakan algoritma decision tree c4.5, apakah anda mau menggunakan aplikasi ini ?	-	2	1	-	-
2	Apakah video tutorial yang tersedia membantu anda dalam menggunakan aplikasi ?	1	2	-	-	-
3	Apakah aplikasi tergolong mudah untuk digunakan ?	1	2	-	-	-
4	Apakah materi pembelajaran mengenai algoritma decision tree c4.5 dalam aplikasi membantu anda ?	-	1	2	-	-
5	Apakah kinerja dari aplikasi memuaskan menurut anda ?	1	1	1	-	-

Tabel 4.7 Hasil pengujian terhadap pengguna yang belum mengerti *machine learning*.

No	Pertanyaan	SS (Sangat Setuju)	S (Setuju)	C (Cukup)	KS (Kurang Setuju)	TS (Tidak Setuju)
1	Jika anda memiliki data untuk diklasifikasikan menggunakan algoritma decision tree c4.5, apakah anda mau menggunakan aplikasi ini ?	2	1	-	-	-
2	Apakah video tutorial yang tersedia membantu anda dalam menggunakan aplikasi ?	3	-	-	-	-

3	Apakah aplikasi tergolong mudah untuk digunakan ?	-	2	1	-	-
4	Apakah materi pembelajaran mengenai algoritma decision tree c4.5 dalam aplikasi membantu anda ?	1	2	-	-	-
5	Apakah kinerja dari aplikasi memuaskan menurut anda ?	-	3	-	-	-

4.5 Pembahasan

4.5.1 Home

Pada halaman ini sistem sama sekali tidak melakukan perhitungan maupun menerima sebuah *input* dari pengguna. Melainkan hanya bagian dimana pengguna dapat lebih mengenali aplikasi baik dari tujuan dan cara penggunaan aplikasi atau *video tutorial* terdapat pada bagian ini. Bagian ini juga merupakan halaman pertama yang akan dilihat oleh pengguna.

4.5.2 Upload Data

Pada halaman ini sistem tidak melakukan perhitungan namun pada bagian ini sistem dapat menerima maupun menolak input yang diberikan oleh pengguna. *Output* pada bagian ini hanya terdapat pada tampilan pratinjau data yang telah diunggah.

4.5.3 Data Preprocessing

Bagian *data preprocessing* merupakan bagian yang hanya dapat dilakukan ketika sistem sudah menerima data sebagai *input*-nya. Pada bagian ini pengguna dapat melakukan *data manipulating* dengan menghapus kolom-kolom yang diinginkan dan membarui data yang diunggah.

4.5.4 Dataframe

Pada bagian ini terdapat suatu *input* yang harus diberikan pengguna guna melihat hasil klasifikasi data. Pada halaman ini pengguna juga wajib mengisi “*traintestsplrit*” dan “*targety*” agar klasifikasi dapat dilakukan. Bagian ini juga merupakan bagian dimana pengguna dapat lebih mengenal lebih jauh terhadap data yang telah diunggah. Pada bagian ini terdapat statistik deksriptif dari data dan *null values* yang dimiliki oleh pengguna yang menampilkan beberapa nilai sebagai berikut.

1. *Count* : Total jumlah baris selain *null variables* dalam suatu kolom.
2. *Unique* : Jumlah nilai data berbeda yang terdapat dalam suatu kolom.
3. *Top* : Nilai data terbanyak (paling sering muncul) yang terdapat dalam suatu kolom.

4. *Frequensi* : Total jumlah nilai data terbanyak dalam suatu kolom.
5. *Mean* : Nilai rata-rata yang dimiliki oleh suatu kolom.
6. *Standard Deviation* : Nilai rata-rata sebaran data pada suatu kolom.
7. *Min* : Nilai data paling rendah yang dimiliki suatu kolom.
8. 25% (*Quartile 1*) : Nilai paling rendah ketika data pada suatu kolom dibagi menjadi 4 bagian yang sama.
9. 50% (*Quartile 2*) : Nilai tengah ketika data pada suatu kolom dibagi menjadi 4 bagian yang sama.
10. 75% (*Quartile 3*) : Nilai paling tinggi ketika data pada suatu kolom dibagi menjadi 4 bagian yang sama.
11. *Max* : Nilai data paling tinggi yang dimiliki suatu kolom.

4.5.5 Data Training & Test

Pada bagian ini pengguna dapat melihat setiap partisi data yang telah dipecah oleh sistem menggunakan *input* yang telah diberikan oleh pengguna. Berikut faktor pemecahan data pada komputasi aplikasi.

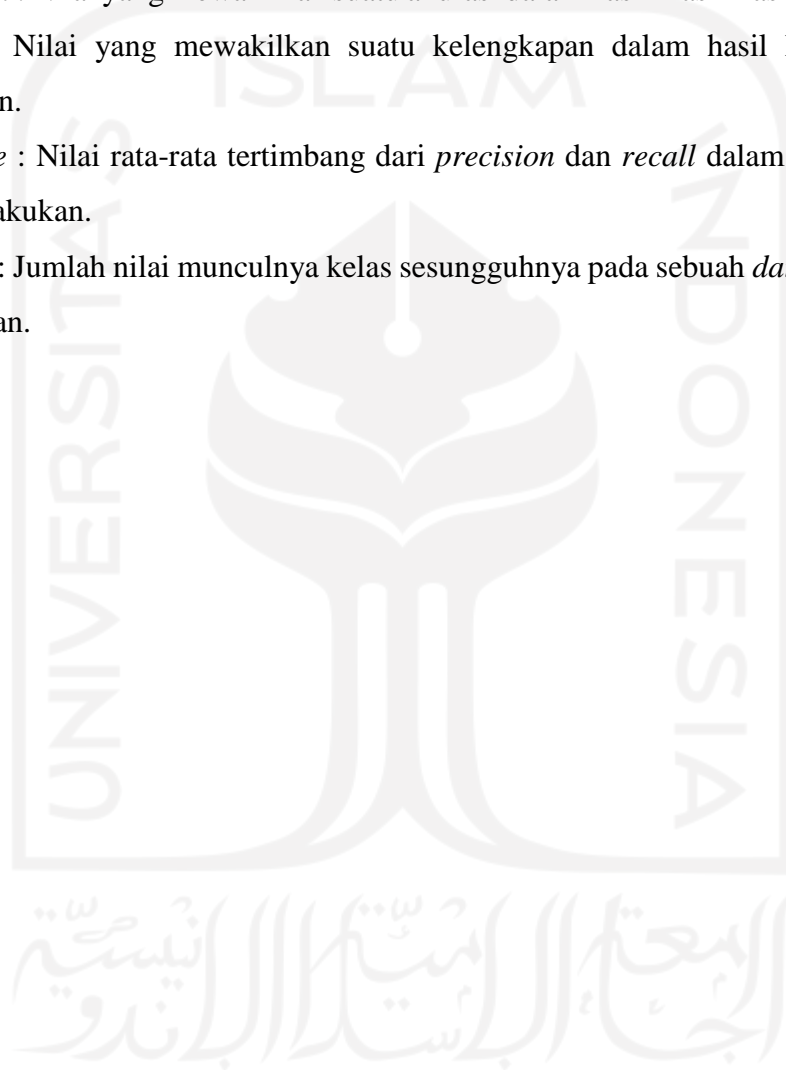
1. Variabel x : Sebuah variabel yang digunakan sebagai variabel masukan (*input*) pada suatu algoritma *supervised learning*.
2. Variabel y : Sebuah variabel yang digunakan sebagai variabel luaran (*output*) pada suatu algoritma *supervised learning*.
3. *Data training* : Bagian dari *dataset* yang digunakan untuk melatih sebuah algoritma *machine learning*.
4. *Data testing* : Bagian dari *dataset* yang digunakan untuk menguji sebuah performa dari suatu pembelajaran.

4.5.6 Decision Tree C4.5

Bagian dimana semua hasil klasifikasi maupun hasil pengujian terhadap hasil klasifikasi divisualisasikan. Hasil klasifikasi dan hasil pengujian klasifikasi yang ditampilkan berupa sebagai berikut.

1. Pohon keputusan : Visualisasi hasil keputusan yang pada tiap percabangannya mewakili aturan-aturan berdasarkan hasil klasifikasi.
2. Matriks konfusi : Sebuah matriks yang mewakilkan tingkat akurasi terhadap hasil klasifikasi.
3. Training-set akurasi : Nilai yang menyatakan seberapa baik model dapat belajar dari *training* yang sudah dilakukan.

4. Training-set skor : Nilai yang merepresentasikan bagaimana tergeneralisasikannya model *training* yang dibuat.
5. Test-set skor : Nilai yang merepresentasikan bagaimana tergeneralisasikannya model *testing* yang dibuat.
6. Akurasi model : Nilai akurasi atau ketepatan terhadap suatu model *machine learning* yang dibangun.
7. *Precision* : Nilai yang mewakili suatu akurasi dalam hasil klasifikasi yang dilakukan.
8. *Recall* : Nilai yang mewakili suatu kelengkapan dalam hasil klasifikasi yang dilakukan.
9. *F1-Score* : Nilai rata-rata tertimbang dari *precision* dan *recall* dalam hasil klasifikasi yang dilakukan.
10. *Support* : Jumlah nilai munculnya kelas sesungguhnya pada sebuah *dataset* yang sudah ditentukan.



BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Pada penelitian kali ini aplikasi *machine learning* berbasis *web* menggunakan algoritma *decision tree c4.5* berhasil melalui tahap pengujian dan dapat memenuhi apa yang sudah diharapkan pengembang. Berdasarkan hasil pengujian ini aplikasi sudah dapat dipublikasikan ke umum. Pada aplikasi ini pengguna dapat melakukan hasil klasifikasi data yang diunggah, melakukan manipulasi data dengan menghapus kolom yang diinginkan, lebih mengenali data yang dimiliki dengan melihat statistik deskriptif data dan melihat partisi data dengan *input* diberikan oleh pengguna. Setelah menggunakan aplikasi ini pengguna dapat menggunakan hasil klasifikasi data yang telah dilakukan sesuai dengan keinginan atau sesuai dengan kehendak pengguna.

5.2 Saran

Tentu saja penelitian yang dilakukan masih memiliki kekurangan dan tidak sempurna. Berikut beberapa saran atau masukan yang dapat menjadi acuan pada penelitian-penelitian serupa berikutnya.

1. Aplikasi yang dikembangkan masih tidak terdapat visualisasi berupa diagram-diagram yang dapat membuat pengguna lebih mengenali data yang dimiliki. Diagram-diagram yang menggambarkan karakteristik terhadap kolom pada dataset.
2. Pada aplikasi tidak terdapat fitur untuk mengunduh hasil klasifikasi. Walaupun pada aplikasi terdapat fitur *screenshot*, namun akan lebih mudah bagi pengguna untuk menggunakan hasil klasifikasi data yang dilakukan dengan fitur untuk mengunduh laporan klasifikasi.
3. Pada aplikasi tidak menggunakan *database* dan hanya menggunakan “*session*” dan *global variable* untuk menyimpan nilai suatu *input*, hal ini menyebabkan apabila data yang diunggah oleh pengguna memiliki baris atau kolom yang banyak akan membuat aplikasi berjalan lambat pada saat membuka halaman “Decision Tree C4.5”.

DAFTAR PUSTAKA

- Adhisyanda Aditya, M., Dicky Mulyana, R., Putu Eka, I., & Rheno Widiyanto, S. (2020). *Seminar Nasional Teknologi Komputer & Sains (SAINTEKS) Penggabungan Teknologi Untuk Analisa Data Berbasis Data Science*.
- Ahmed, N. S., & Hikmat Sadiq, M. (2018). Clarify of the Random Forest Algorithm in an Educational Field. *ICOASE 2018 - International Conference on Advanced Science and Engineering*, 179–184. <https://doi.org/10.1109/ICOASE.2018.8548804>
- Alpaydin, E. (2010). *Introduction to Machine Learning* (4th ed.). Google Books. https://books.google.co.id/books?hl=id&lr=&id=tZnSDwAAQBAJ&oi=fnd&pg=PR7&dq=Alpaydin++E&ots=F3YP6_9rxl&sig=52KKh8FdsrlU1k6U2pe7hOdLMXc&redir_esc=y#v=onepage&q=Alpaydin%20%20E&f=false
- 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>
- Ardiansyah Sembiring, M., Fitri Larasati Sibuea, M., Sapta, A., Studi Sistem Informasi, P., & Royal, S. (2018). Analisa Kinerja Algoritma C.45 Dalam Memprediksi Hasil Belajar. *JOURNAL OF SCIENCE AND SOCIAL RESEARCH*, 1(1), 73–79. <https://doi.org/10.54314/JSSR.V1I1.110>
- Arora, A., Gupta, B., Uttarakhand, P., & Rawat, I. A. (2017). Analysis of Various Decision Tree Algorithms for Classification in Data Mining. *International Journal of Computer Applications*, 163(8), 975–8887.
- Atmaja, K. J., Bagus, I., Anandita, G., Kadek, N., & Dewi, C. (2017). Penerapan Data Mining Untuk Memprediksi Potensi Pendonor Darah Menjadi Pendonor Tetap Menggunakan Metode Decision Tree C.45. *S@CIES*, 7(2), 101–108. <https://doi.org/10.31598/SACIES.V7I2.284>
- bhatiya, rakesh, Ravi Kumar, G., Kumar, R., & Verma, R. (2012). Classification Algorithms for Data Mining: A Survey. *International Journal of Innovations in Engineering and Technology (IJJET)*, 1(2).
- Breck, E., Polyzotis, N., Roy, S., Whang, S. E., & Zinkevich, M. (2019). *DATA VALIDATION FOR MACHINE LEARNING*.

- Brodie, M. L. (2015). *Chapter4. Understanding Data Science: An Emerging Discipline for Data-Intensive Discovery - Getting Data Right [Book]*. <https://www.oreilly.com/library/view/getting-data-right/9781491935361/ch04.html>
- Brodie, M. L. (2019). What Is Data Science? In *Applied Data Science* (pp. 101–130). Springer International Publishing. https://doi.org/10.1007/978-3-030-11821-1_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>
- Cynthia, E. P., & Ismanto, E. (2018). Metode Decision Tree Algoritma C.45 Dalam Mengklasifikasi Data Penjualan Bisnis Gerai Makanan Cepat Saji. *Jurasik (Jurnal Riset Sistem Informasi Dan Teknik Informatika)*, 3, 1. <https://doi.org/10.30645/jurasik.v3i0.60>
- Das, S., Dahiya, S., & Bharadwaj, A. (2014). An online software for decision tree classification and visualization using C4.5 algorithm (ODTC). *2014 International Conference on Computing for Sustainable Global Development, INDIACom 2014*, 962–965. <https://doi.org/10.1109/INDIACom.2014.6828107>
- Gershman, A., Melsels, A., Karl-Heinz, L., Rokach, L., Schclar, A., & Sturm, A. (2010, June). Decision Tree Based Recommender System. *10th International Conference on Innovative Internet Community Services*. <http://www.gi-ev.de/service/publikationen/lni/>
- Ghimire, D. (2020). *Comparative study on Python web frameworks: Flask and Django*.
- Gray, J., Gerlitz, C., & Bounegru, L. (2018). Data infrastructure literacy. *Big Data and Society*, 5(2). <https://doi.org/10.1177/2053951718786316>
- Grinberg, M. (2018). *Flask Web Development: Developing Web Applications with Python*. In *Google Books*. O'Reilly media, Inc. https://books.google.co.id/books?hl=en&lr=&id=cVIPDwAAQBAJ&oi=fnd&pg=PT25&dq=flask&ots=xOFShl4lcZ&sig=te5YR_qJLr7SMkAu7nraeQYvPxY&redir_esc=y#v=onepage&q=flask&f=false
- Han, J., Pei, J., & Tong, H. (2022). *Data Mining: Concepts and Techniques*. Elsevier, Inc. https://books.google.co.id/books?hl=id&lr=&id=NR1oEAAAQBAJ&oi=fnd&pg=PP1&dq=data+mining&ots=_M7JLIvflY&sig=wWBC4ByGMQi4hBUjNAX3yOTrb4&redir_esc=y#v=onepage&q=data%20mining&f=false
- Handrianto, Y., & Farhan, M. (2019). C.45 Algorithm for Classification of Causes of Landslides. *Sinkron: Jurnal Dan Penelitian Teknik Informatika*, 4(1), 120–127. <https://doi.org/10.33395/SINKRON.V4I1.10154>

- Jadhav, S. D., & Channe, H. P. (2016). Efficient Recommendation System Using Decision Tree Classifier and Collaborative Filtering. *International Research Journal of Engineering and Technology*, 03(08). www.irjet.net
- Josse, J., Prost, N., Scornet, E., & Varoquaux, G. (2019). *On the consistency of supervised learning with missing values*. <http://arxiv.org/abs/1902.06931>
- Kartiningrum, E. D. (2015). *PANDUAN PENYUSUNAN STUDI LITERATUR*.
- Lee, W.-M. (2019). Python Machine Learning. In *Wei Meng Lee*. John Wiley & Sons, Inc. https://books.google.co.id/books?hl=en&lr=&id=9FOQDwAAQBAJ&oi=fnd&pg=PP2&dq=machine+learning+python+basic&ots=p_olArRPBD&sig=M-93EdbTTOaaTg4c7y28Kn2YV_8&redir_esc=y#v=onepage&q=machine%20learning%20python%20basic&f=false
- Liu, H., & Kuan Tan, H. B. (2009). Covering code behavior on input validation in functional testing. *Information and Software Technology*, 51(2), 546–553. <https://doi.org/10.1016/J.INFSOF.2008.07.001>
- Mathuria, M., Alam, L., an Haq, N. F., Mamun, T., Bhargava, N., Sharma, G., & Bhargava, R. (2013). Decision Tree Analysis on J48 Algorithm for Data Mining. *International Journal of Advanced Research in Computer Science and Software Engineering*, 3(6), 2277. <http://www.cs.waikato.ac.nz/ml/weka>.
- Mitra, P., Chatterjee, S., & Ali, N. (2011). Graphical analysis of MC/DC using automated software testing. *ICECT 2011 - 2011 3rd International Conference on Electronics Computer Technology*, 3, 145–149. <https://doi.org/10.1109/ICECTECH.2011.5941819>
- 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>
- Murnane, T., & Reed, K. (2001). On the effectiveness of mutation analysis as a black box testing technique. *Proceedings of the Australian Software Engineering Conference, ASWEC, 2001-January*, 12–20. <https://doi.org/10.1109/ASWEC.2001.948492>
- Nath, A., & Nema, S. (2020). Clustering Visualization and Class Prediction using Flask of Benchmark Dataset for Unsupervised Techniques in ML. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, 9, 2278–3075. <https://doi.org/10.35940/ijitee>

- Nidhra, S. (2012). Black Box and White Box Testing Techniques - A Literature Review. *International Journal of Embedded Systems and Applications*, 2(2), 29–50. <https://doi.org/10.5121/ijesa.2012.2204>
- Nikam, S. S. (2015). A comparative study of classification techniques in data mining algorithms. *Oriental Journal of Computer Science & Technology*, 8(1), 13–19. www.computerscijournal.org
- Pasaribu, U. S., Husniah, H., Sari, R. K. N., Dwiasnati, S., & Devianto, Y. (2019). Utilization of Prediction Data for Prospective Decision Customers Insurance Using the Classification Method of C.45 and Naive Bayes Algorithms. *Journal of Physics: Conference Series*, 1179(1), 012023. <https://doi.org/10.1088/1742-6596/1179/1/012023>
- Patel, H. H., & Prajapati, P. (2018). Study and Analysis of Decision Tree Based Classification Algorithms. *International Journal of Computer Sciences and Engineering*, 6(10), 74–78. <https://doi.org/10.26438/ijcse/v6i10.7478>
- Patil, S., & Kulkarni, U. (2019). Accuracy prediction for distributed decision tree using machine learning approach. *Proceedings of the International Conference on Trends in Electronics and Informatics, ICOEI 2019, 2019-April*, 1365–1371. <https://doi.org/10.1109/ICOEI.2019.8862580>
- Pranckevičius, T., & Marcinkevičius, V. (2017). Comparison of Naive Bayes, Random Forest, Decision Tree, Support Vector Machines, and Logistic Regression Classifiers for Text Reviews Classification. *Baltic Journal of Modern Computing*, 5(2). <https://doi.org/10.22364/bjmc.2017.5.2.05>
- Pressman, R. S. (2005). *Software Engineering: A Practitioner's Approach* (6th ed.). The McGraw-Hill Companies, Inc. https://books.google.co.id/books?hl=id&lr=&id=bL7QZHtWvaUC&oi=fnd&pg=PA1&dq=pressman&ots=O8zfcUwIbl&sig=njLvfuNVec394ZJoLNyglICUp0I&redir_esc=y#v=onepage&q=pressman&f=false
- Priyam, A., Gupta, R., Rathee, A., & Srivastava, S. (2013). *Comparative Analysis of Decision Tree Classification Algorithms* (Vol. 3, Issue 2). <http://inpressco.com/category/ijcet>
- Raileanu, L. E., & Stoffel, K. (2004). Theoretical Comparison between the Gini Index and Information Gain Criteria. *Annals of Mathematics and Artificial Intelligence* 2004 41:1, 41(1), 77–93. <https://doi.org/10.1023/B:AMAI.0000018580.96245.C6>
- Rajendra Haidar, L., Sedyono, E., Iriani, A., & Notohamidjojo Blotongan Sidorejo, J. O. (2020). ANALISA PREDIKSI MAHASISWA DROP OUT MENGGUNAKAN

- METODE DECISION TREE DENGAN ALGORITMA ID3 dan C4.5. *Jurnal Transformatika*, 17(2), 97–106.
<https://doi.org/10.26623/TRANSFORMATIKA.V17I2.1609>
- Rokach, L., & Maimon, O. (2014). Data Mining with Decision Trees: Theory and Applications, 2nd Edition. *Data Mining with Decision Trees: Theory and Applications, 2nd Edition*, 81, 1–305. <https://doi.org/10.1142/9097>
- Sarker, I. H. (2021). Machine Learning: Algorithms, Real-World Applications and Research Directions. *SN Computer Science*, 2(3). <https://doi.org/10.1007/s42979-021-00592-x>
- 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(1). <https://doi.org/10.1186/s40537-020-00318-5>
- Shao, D., Khurshid, S., & Perry, D. E. (2008). A Case for White-box Testing Using Declarative Specifications Poster Abstract. 137–137. <https://doi.org/10.1109/TAIC.PART.2007.36>
- Sharma, M., & Subhash Chandra, B. (2010). Automatic generation of test suites from decision table - Theory and implementation. *Proceedings - 5th International Conference on Software Engineering Advances, ICSEA 2010*, 459–464. <https://doi.org/10.1109/ICSEA.2010.78>
- Shokri, R., Stronati, M., Song, C., & Shmatikov, V. (2017). Membership Inference Attacks Against Machine Learning Models. *Proceedings - IEEE Symposium on Security and Privacy*, 3–18. <https://doi.org/10.1109/SP.2017.41>
- Suhirman, S., Hidayat, A. T., Saputra, W. A., & Saifullah, S. (2021). Website-Based E-Pharmacy Application Development to Improve Sales Services Using Waterfall Method. *International Journal of Advances in Data and Information Systems*, 2(2), 114–129. <https://doi.org/10.25008/ijadis.v2i2.1226>
- Suárez-Figueroa, M. C., & Gómez-Pérez, A. (2008). First Attempt towards a Standard Glossary of Ontology Engineering Terminology. *Managing Ontologies and Lexical Resources. TKE 2008 8th International Conference on Terminology and KE. | The 8th International Conference on Terminology and Knowledge Engineering | August 18-August 21, 2008 | Copenhagen, DENMARK*.
- Suliyanto. (2017). *METODE PENELITIAN KUANTITATIF*.
- Virkus, S., & Garoufallou, E. (2019). Data science from a library and information science perspective. *Data Technologies and Applications*, 53(4), 422–441. <https://doi.org/10.1108/DTA-05-2019-0076/FULL/XML>

- Zhang, X., & Jiang, S. (2012). A splitting criteria based on similarity in decision tree learning. *Journal of Software*, 7(8), 1775–1782. <https://doi.org/10.4304/jsw.7.8.1775-1782>
- 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>

LAMPIRAN

Dataset Cause of Landslides

Landslide Hazard Zone	Slope	Weather	Type of Soil	Cause of Landslides
High	Steep	Rain	Latosol	Yes
Middle	Sloping	Rain	Clay	Yes
High	Rather Steep	Rain	Latosol	Yes
Low	Sloping	Bright	Latosol	No
Middle	Steep	Rain	Clay	Yes
Middle	Steep	Bright	Clay	No
High	Rather Steep	Bright	Latosol	No
Middle	Steep	Rain	Clay	Yes
Middle	Sloping	Rain	Clay	No
High	Steep	Rain	Latosol	Yes
Middle	Sloping	Rain	Clay	Yes
Middle	Sloping	Rain	Latosol	No
Middle	Rather Steep	Rain	Clay	No
High	Steep	Rain	Latosol	Yes
Low	Rather Steep	Rain	Clay	No
Middle	Sloping	Rain	Latosol	No
High	Steep	Rain	Clay	Yes

Dataset weather-numeric.csv

outlook	temperature	humidity	windy	play
----------------	--------------------	-----------------	--------------	-------------

sunny	85.0	85.0	FALSE	no
sunny	80.0	90.0	TRUE	no
overcast	83.0	86.0	FALSE	yes
rainy	70.0	96.0	FALSE	yes
rainy	68.0	80.0	FALSE	yes
rainy	65.0	70.0	TRUE	no
overcast	64.0	65.0	TRUE	yes
sunny	72.0	95.0	FALSE	no
sunny	69.0	70.0	FALSE	yes
rainy	75.0	80.0	FALSE	yes
sunny	75.0	70.0	TRUE	yes
overcast	72.0	90.0	TRUE	yes
overcast	81.0	75.0	FALSE	yes
rainy	71.0	91.0	TRUE	no

