

Implementasi Algoritma Machine Learning Berbasis WEB dengan Framework Streamlit



Streamlit



Abu Tholib, M.Kom

ABU THOLIB

Buku Refrensi

**Implementasi Algoritma Machine Learning
Berbasis Web dengan Framework Streamlit**



PUSTAKA NURJA

Undang-Undang Republik Indonesia No. 19 Tahun 2002 tentang Hak Cipta

Lingkup Hak Cipta

Pasal 2

1. Hak cipta merupakan hak eksklusif bagi Pencipta atau Pemegang Hak Cipta untuk mengumumkan atau memperbanyak Ciptaannya, yang timbul secara otomatis setelah suatu ciptaan dilahirkan tanpa mengurangi pembatasan menurut peraturan perundang-undangan yang berlaku.

Ketentuan Pidana

Pasal 72

1. Barang siapa dengan sengaja dan tanpa hak melakukan perbuatan sebagaimana dimaksud dalam Pasal 2 ayat (1) atau Pasal 49 ayat (1) dan ayat (2) dipidana dengan pidana penjara masing-masing paling singkat 1 (satu) bulan dan / atau denda paling sedikit Rp.1.000.000,00 (satu juta), atau pidana penjara paling lama 7 (Tujuh) tahun dan / atau denda paling banyak Rp.5.000.000.000,00 (lima milyar rupiah).
2. Barang siapa dengan sengaja menyiarkan, memamerkan, mengedarkan, atau menjual kepada umum suatu Ciptaan atau barang hasil pelanggaran Hak Cipta atau Hak Terkait sebagaimana dimaksud pada ayat (1) dipidana dengan pidana penjara paling lama 5 (lima) tahun dan / atau denda paling banyak Rp.500.000.000,00 (lima ratus juta rupiah).

Buku Refrensi

@*abutholib*

xi + 135 halaman; 15,5 x 23 cm

1. Teknik

Penulis: Abu Tholib, M.Kom

Editor: Moh Ainol Yaqin, M.Kom

Layouter: Mochammad Faid, M.Kom

Desain Cover: Ahmad Naufal Waliyus Zain

Cetakan 1, Mei 2023

ISBN: 978-623-6757-74-1

Penerbit:

Pustaka Nurja

LP3M Universitas Nurul Jadid

Jl. KH. Zaini Mun'im Karanganyar Paiton Probolinggo
67291

Telp. (0335) 771732; CP: 082318007953

Web: <https://pustakanurja.unuja.ac.id>

email: pustakanurja@gmail.com

Hak cipta dilindungi undang-undang.

All rights reserved.

PRAKATA

Rasa syukur hanya kepada Allah SWT karena pertolonganNYA kita bisa merampungkan buku ini dan sholawat dan salam kita haturkan kepada baginda nabi Muhammad SAW sehingga Buku implementasi *machine learning* berbasis web dengan framework streamlit bisa rampung dan semoga membantu pembaca dalam memahami dan mengimplementasikan teknik *machine learning* dengan menggunakan Streamlit sebuah framework open-source yang memudahkan pembuatan aplikasi web interaktif dengan menggunakan bahasa pemrograman Python.

Dalam era digital yang semakin berkembang seperti saat ini, data menjadi salah satu hal yang sangat penting dalam pengambilan keputusan bisnis. Oleh karena itu, kemampuan untuk menganalisis data dengan menggunakan teknik *machine learning* menjadi sangat penting bagi setiap organisasi atau perusahaan. Namun, implementasi *machine learning* tidak selalu mudah karena memerlukan keahlian dan pengalaman pada bidang bahasa pemrograman, statistik dan ilmu data.

Dalam beberapa tahun terakhir, *machine learning* telah mengalami kemajuan pesat dan telah digunakan dalam berbagai aplikasi seperti analisis data, pengenalan suara dan gambar, dan bahkan pengembangan robot dan mobil tanpa awak atau *autonomous vehicles*. Oleh karena itu, pemahaman dan keterampilan dalam ilmu *machine learning* menjadi semakin penting bagi para mahasiswa, guru, dosen dan professional.

Tahun ini, kebutuhan terhadap karyawan yang memiliki kemampuan dalam penerapan *machine learning* semakin meningkat karena organisasi dan perusahaan mulai menyadari potensi besar yang dimilikinya dalam membantu

meningkatkan efisiensi bisnis mereka. Dengan menguasai *machine learning*, seseorang dapat menghasilkan model prediksi yang akurat dan dapat membantu dalam pengambilan keputusan yang lebih baik dan lebih cepat. Oleh karena itu, mempelajari *machine learning* menjadi sangat penting bagi para mahasiswa, guru, dosen dan profesional di berbagai bidang yang ingin mengembangkan keterampilan mereka dan mengambil keuntungan dari kemajuan teknologi yang ada saat ini.

Melalui buku ini, pembaca akan belajar bagaimana melakukan implementasi *machine learning* dengan mudah dan cepat menggunakan Streamlit. Buku ini disusun dengan bahasa yang mudah dipahami dan dilengkapi dengan contoh-contoh studi kasus yang akan membantu pembaca memahami konsep dan penerapan secara langsung serta dapat membantu pembaca meningkatkan kemampuan dalam pengambilan keputusan bisnis berdasarkan analisis data yang akurat dan efektif..

Terakhir, kami ingin menyampaikan terima kasih kepada seluruh pihak yang telah membantu dalam penyelesaian buku ini. Kami menyadari buku ini masih bisa di sempurnakan lagi dengan integrasi database sehingga memudahkan pengguna untuk menyimpan, mengelola, dan mengakses data lebih mudah. Semoga buku ini bermanfaat bagi pembaca dan dapat menjadi sumber referensi yang berguna. Selamat membaca !

Probolinggo, 20 Mei 2023

Penulis,

DAFTAR ISI

BAB 1 Pengenalan Framework Web Streamlit	10
1.1. Tentang Streamlit	10
1.2. Instalasi Streamlit.....	12
1.3. Aplikasi Pertama Streamlit.....	17
BAB 2 Membuat Aplikasi Dasar Streamlit	19
2.1. Dasar Streamlit.....	19
2.2. Elemen Text	22
2.3. Menampilkan Data.....	30
2.4. Menampilkan Grafik	35
2.5. Menampilkan Input Form.....	44
2.6. Menampilkan Media	55
2.7. Menampilkan Layout dan Container.....	56
2.8. Menampilkan Control Flow	63
BAB 3 Implementasi Algoritma SVM menggunakan Framework Streamlit	70
3.1. Support Vector Machine	70
3.2. Dataset.....	71
3.3. Pembersihan Data.....	72
3.4. Membuat Model	75
3.5. Evaluasi Model.....	76
3.6. Menyimpan Model	77
3.7. Aplikasi GUI SVM di Streamlit.....	78
BAB 4 Implementasi Algoritma Decision Tree menggunakan Framework Streamlit	83

4.1. Decision Tree(C4.5).....	83
4.2. Dataset.....	85
4.3. Pembersihan Data.....	86
4.4. Membuat Model.....	87
4.5. Evaluasi Model.....	88
4.6. Menyimpan Model.....	89
4.7. Aplikasi GUI Decision Tree di Streamlit.....	90
BAB 5 Implementasi Algoritma Random Forest menggunakan framework streamlit.....	93
5.1. Random Forest.....	93
5.2. Dataset.....	94
5.3. Pembersihan Data.....	96
5.4. Membuat Model.....	97
5.5. Evaluasi Model.....	98
5.6. Menyimpan Model.....	99
5.7. Aplikasi GUI Random Forest di streamlit.....	99
BAB 6 Implementasi Algoritma K-Means Clustering menggunakan framework streamlit.....	103
6.1. K-Means Clustering.....	103
6.2. Dataset.....	104
6.3. Pembersihan Data.....	106
6.4. Membuat Model.....	106
6.5. Evaluasi Model.....	109
6.6. Menyimpan Model.....	110
6.7. Implementasi Algoritma K-Means di streamlit.....	111
BAB 7 Implementasi Algoritma Naïve Bayes menggunakan streamlit.....	115

7.1 Naive Bayes	115
7.2 Dataset.....	116
7.3 Pembersihan Data.....	116
7.4 Membuat Model.....	119
7.5 Evaluasi Model.....	120
7.6 Menyimpan Model.....	121
7.7 Implementasi Algoritma Naïve Bayes di streamlit	122
DAFTAR PUSTAKA.....	125

DAFTAR GAMBAR

Gambar 1. 1 Masuk direktori d.....	14
Gambar 1. 2 Masuk ke folder datamining	14
Gambar 1. 3 masuk ke folder app.....	14
Gambar 1. 4 membuat virtual environtment python	14
Gambar 1. 5 Tampilan folder di Environtment.....	15
Gambar 1. 6 Mengaktivasi Environtment Python.....	15
Gambar 1. 7 Perintah melihat package yang terinstall.....	15
Gambar 1. 8 menginstall streamlit.....	16
Gambar 1. 9 instalasi matplotlib dan scikit-learn	16
Gambar 1. 10 Mengaktifkan aplikasi VS Code	16
Gambar 1. 11 Tampilan awal VS Code	16
Gambar 1. 12 Pembuatan file streamlit di VS Code	17
Gambar 1. 13 Tampilan awal coding streamlit	17
Gambar 1. 14 Tampilan aplikasi streamlit.....	18
Gambar 2. 1 Aplikasi pertama streamlit	20
Gambar 2. 2 Tampilan Dataframe	22
Gambar 2. 3 Tampilan Markdown.....	23
Gambar 2. 4 Tampilan title.....	24
Gambar 2. 5 Tampilan header	25
Gambar 2. 6 Tampilan Subheader	26
Gambar 2. 7 Tampilan Caption	26
Gambar 2. 8 Tampilan Code.....	27
Gambar 2. 9 Tampilan Text.....	28
Gambar 2. 10 Tampilan Latex	29
Gambar 2. 11 Tampilan Divider	30
Gambar 2. 12 Tampilan Dataframe	31
Gambar 2. 13 Tampilan Tabel.....	32

Gambar 2. 14 Tampilan Metric	33
Gambar 2. 15 Tampilan Json	34
Gambar 2. 16 Grafik Garis	36
Gambar 2. 17 Grafik Area	37
Gambar 2. 18 Grafik Batang.....	38
Gambar 2. 19 Grafik Altair.....	39
Gambar 2. 20 Menginstall Library Vega Dataset	40
Gambar 2. 21 Grafik Vega Lite	41
Gambar 2. 22 Menginstall Library Plotly	41
Gambar 2. 23 Grafik Plotly	42
Gambar 2. 24 Tampilan Peta	43
Gambar 2. 25 Tampilan Button	44
Gambar 2. 26 Tampilan Experiment Data Editor	45
Gambar 2. 27 Tampilan Button Download.....	47
Gambar 2. 28 Tampilan Checkbox	47
Gambar 2. 29 Tampilan Multi Select.....	48
Gambar 2. 30 Tampilan Slider.....	49
Gambar 2. 31 Tampilan Select Slider	50
Gambar 2. 32 Tampilan Inputan Text.....	51
Gambar 2. 33 Tampilan Date.....	52
Gambar 2. 34 Tampilan Upload	53
Gambar 2. 35 Tampilan Camera.....	54
Gambar 2. 36 Tampilan Color Picker	54
Gambar 2. 37 Tampilan Audio	55
Gambar 2. 38 Tampilan audio	56
Gambar 2. 39 Tampilan Sidebar	57
Gambar 2. 40 Tampilan Colomn	58
Gambar 2. 41 Tampilan Tab	59
Gambar 2. 42 Tampilan Expander.....	60

Gambar 2. 43 Tampilan Container	61
Gambar 2. 44 Tampilan Empty	62
Gambar 2. 45 Tampilan perintah stop	63
Gambar 2. 46 Tampilan Form	65
Gambar 2. 47 Tampilan form submit.....	67
Gambar 2. 48 Tampilan experimental rerun	68
Gambar 3. 1 Menampilkan Data Kanker	72
Gambar 3. 2 Menampilkan Data yang kosong.....	73
Gambar 3. 3 Menghapus kolom.....	74
Gambar 3. 4 Menampilkan Confusion Metric	77
Gambar 3. 5 Tampilan susunan folder	78
Gambar 3. 6 Tampilan Aplikasi GUI SVM Streamlit	82
Gambar 4.1 Menampilkan Data Diabetes	86
Gambar 4. 2 Menampilkan dataset yang kosong	86
Gambar 4. 3 Hasil Confusion Metric	89
Gambar 4.4 Tampilan Aplikasi Decision Tree	92
Gambar 5. 1 Menampilkan dataset iris	95
Gambar 5. 2 Tampilan Data yang bersih	96
Gambar 5. 3 Data Reduction	97
Gambar 5. 4 Hasil Evaluasi Model.....	98
Gambar 5. 5 Tampilan Aplikasi Random Forest	101
Gambar 6. 1 Menampilkan Data Rumah	105
Gambar 6. 2 Grafik Scater 3 cluster	108
Gambar 6. 3 Tampilan Grafik Box 3 Cluster.....	108
Gambar 6. 4 Grafik uji coba range 2 sampai 8	110
Gambar 6. 5 susunan folder k-means.....	111
Gambar 6. 6 Halaman utama klaster rumah.....	113
Gambar 7. 1 Menampilkan Data penyakit jantung	116
Gambar 7. 2 Menampilkan Info kolom dan type datanya.....	117

Gambar 7. 3 memeriksa nilai kosong118
Gambar 7. 4 Hasil output variabel target.....119
Gambar 7. 5 Hasil Akurasi dan Presisi121
Gambar 7. 6 Hasil ujicoba algoritma naïve bayes di streamlit124

BAB 1

Pengenalan Framework Web Streamlit

Deskripsi Materi:

Terdapat 3 Materi pada Bab 1 yaitu tentang framework streamlit, Instalasi streamlit dan contoh aplikasi pertama dengan streamlit. materi Awal sangat penting dipelajari secara benar dan teliti sebelum belajar implementasi web streamlit lebih jauh lagi.

1.1. Tentang Streamlit

Perkembangan teknologi kecerdasan buatan dalam beberapa dekade terakhir sungguh luar biasa pesat. Salah satu cabangnya adalah *Machine Learning* yang berfokus pada sebuah mesin/komputer yang mampu belajar dari data yang tersedia dan mengubah data tersebut menjadi informasi dan keputusan[1] perluasan implementasi Industri 4.0 dan teknologi baru memungkinkan terobosan baru dalam pengembangan berbagai metode *machine learning*, Teknologi ini telah mengalami kemajuan signifikan [2], pendekatan *machine learning* dapat digunakan untuk menyelesaikan permasalahan yang kompleks di berbagai bidang yang tidak dapat di selesaikan oleh program konvensional [3], dengan semakin meningkatnya daya komputasi dan jumlah data yang tersedia, *Machine Learning* menjadi lebih efektif dalam menangani tugas-tugas yang lebih kompleks, seperti analisis citra dan teks, pengenalan wajah, dan pemrosesan bahasa alami. Dalam beberapa dekade terakhir, *Machine Learning* juga telah menjadi pusat

perhatian bagi banyak organisasi dan perusahaan di seluruh dunia. Perusahaan dapat menggunakan *Machine Learning* untuk mengidentifikasi faktor risiko[4] dan memprediksi permintaan pasar serta situasi risiko yang mungkin terjadi [5]. Banyak aplikasi bisnis menggunakan teknologi ini untuk mempercepat proses bisnis dan meningkatkan keuntungan mereka. Para eksekutif, manajer, dan investor meningkatkan minat mereka terhadap proyek-proyek *data science*, mereka melihat semakin banyak peluang yang sangat menjanjikan kedepannya sehingga perusahaan seperti Google dan Amazon melakukan riset secara mendalam terkait dengan analisis data pada pencarian web, serta menganalisa rekomendasi produk pada perusahaan Amazon untuk meningkatkan penjualan mereka [6]. Bahkan Dalam beberapa tahun terakhir, dengan adopsi teknologi *Deep Learning* mencapai level yang jauh lebih tinggi lagi. *Deep Learning* telah meraih kesuksesan besar dibanyak aplikasi terapan di *computer vision* [7] Neural Network yang kuat dengan ratusan atau bahkan ribuan lapisan dapat digunakan untuk mempelajari pola yang lebih kompleks, seperti gambar dan suara. membawa kemampuan yang semakin dekat dengan kecerdasan manusia.

Salah satu titik kesulitan bagi para *Data Scientist* adalah dalam proses setelah mereka menemukan wawasan baru atau membuat model baru. Apa cara terbaik untuk menunjukkan hasil secara visual kepada para khalayak umum adalah dengan membuat aplikasi berbasis website, ada beberapa framework yang biasa digunakan oleh khalayak umum seperti Flask dan Django namun kekurangannya adalah kita harus membangun seluruh aplikasi web dari awal, sehingga framework streamlit menjadi pilihan dalam hal kecepatan dan kemudahan. streamlit adalah kerangka kerja utama pertama yang semuanya murni Python, sehingga

mengurangi waktu pengembangan dari berminggu-minggu menjadi beberapa jam[8].

Streamlit adalah framework Python yang digunakan untuk membangun aplikasi web dengan antarmuka pengguna interaktif untuk proyek-proyek *data science* dan *machine learning*[9]. Ini memungkinkan pengembang untuk membuat aplikasi web khusus dan tampilan data dengan sedikit usaha dan keahlian dalam pengembangan, web Streamlit dibangun di atas pustaka ilmu data yang populer seperti NumPy, Pandas, dan Matplotlib, sehingga memudahkan untuk membuat visualisasi dan komponen interaktif dalam aplikasi. Streamlit juga menyediakan API sederhana untuk input dan output data, sehingga mudah untuk terhubung ke berbagai sumber data dan API. Dengan Streamlit, pengembang dapat membuat aplikasi yang dapat digunakan di berbagai platform, seperti Heroku atau AWS. Streamlit juga memiliki fitur berbagi bawaan, di mana pengembang dapat membagikan aplikasi mereka dengan tim mereka atau komunitas yang lebih luas hanya dengan membagikan tautan.

Streamlit adalah pustaka Python yang *open source* untuk memudahkan membuat dan berbagi aplikasi web yang indah dan sesuai kebutuhan untuk *machine learning* dan *data science*[10]. Hanya dalam beberapa menit, sehingga memudahkan pengembang untuk fokus pada analisis dan visualisasi data mereka.

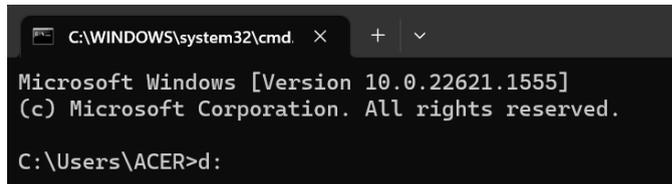
1.2. Instalasi Streamlit

Ada beberapa aplikasi yang dibutuhkan ketika kita belajar membuat aplikasi web dengan streamlit seperti :

1. Silahkan mendownload python di <https://www.python.org/downloads/> lalu install sampai selesai, perlu diketahui ketika pembuatan buku ini menggunakan python versi 3.11.0 Python adalah

bahasa pemrograman yang menyediakan toolkit untuk pembelajaran dan analisis seperti scikit-learn, numpy, scipy, pandas, dan visualisasi data menggunakan matplotlib [11].

2. Selanjutnya adalah mendownload dan install visual studio code di <https://code.visualstudio.com/download> dimana Visual Studio Code adalah editor kode yang didefinisikan ulang dan dioptimalkan untuk membangun dan men-debug aplikasi[12], kami menggunakan versi 1.73.0 waktu menggunakan pada latihan di buku ini.
3. Untuk menampilkan hasil dari coding yang sudah dibuat, kita membutuhkan web browser seperti google chrome atau web browser bawaan windows Microsoft edge. Pada saat buku ini dibuat menggunakan google chrome Version 112.0.5615.140 (Official Build) (64-bit).
4. Pemahaman dasar tentang Python juga sangat berguna untuk buku ini. Jika Anda belum mempunyai dasar pemrograman python, silakan luangkan waktu untuk mengenal dan belajar bahasa pemrograman Python dengan lebih baik menggunakan tutorial ini (<https://docs.python.org/3/tutorial/>) atau beberapa referensi yang melimpah di internet.
5. Langkah – langkah instalasi streamlit
 - a. Masuk ke cmd dengan cara tekan Windows + R selanjutnya ketik cmd lalu OK
 - b. Selanjutnya ketik D: lalu enter untuk masuk ke direktori D karena saya agendanya semua file experiment pembelajaran streamlit nantinya akan ditaruh di direktori D, kalian bebas akan ditaruh dimana nantinya file experiment kalian, jika masih bingung lebih baik ikut langkah-langkah yang sudah ada di buku ini.

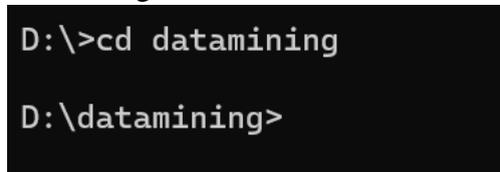


```
C:\WINDOWS\system32\cmd. x + v
Microsoft Windows [Version 10.0.22621.1555]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ACER>d:
```

Gambar 1. 1 Masuk direktori d

- c. Langkah selanjutnya adalah masuk ke folder datamining (folder datamining saya sudah buat sebelumnya di direktori D tinggal klik kanan new->folder, artinya folder datamining ini sudah saya buat sebelumnya), dengan cara ketik cd datamining

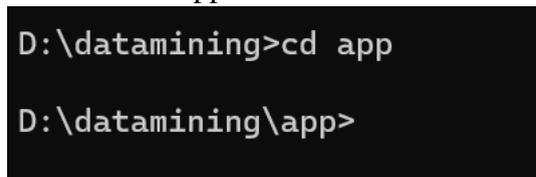


```
D:\>cd datamining

D:\datamining>
```

Gambar 1. 2 Masuk ke folder datamining

- d. Langkah selanjutnya adalah masuk ke folder app (folder app saya sudah buat sebelumnya), dengan cara ketik cd app

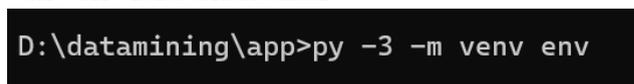


```
D:\datamining>cd app

D:\datamining\app>
```

Gambar 1. 3 masuk ke folder app

- e. Langkah selanjutnya adalah membuat virtual environment python dengan cara ketik py -3 -m venv env lalu tekan enter

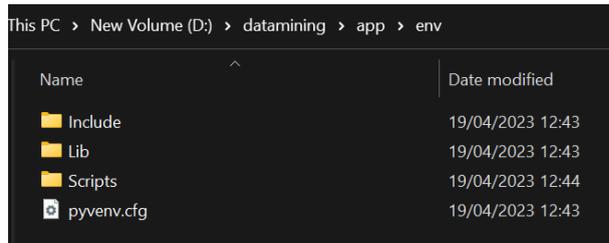


```
D:\datamining\app>py -3 -m venv env
```

Gambar 1. 4 membuat virtual environtment python

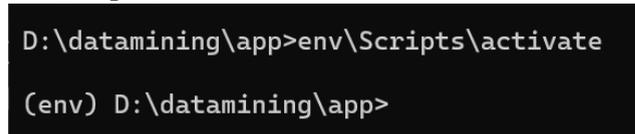
venv adalah sebuah environment manager yang digunakan untuk membuat lingkungan python virtual yang terisolasi.

Sehingga ada tampilan seperti dibawah ini di folder env



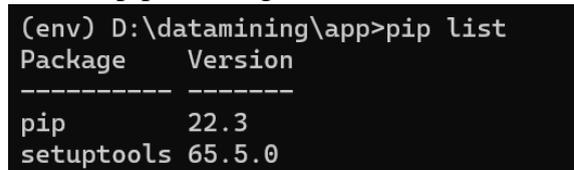
Gambar 1. 5 Tampilan folder di Environment

- f. Menjalankan aktivasi dengan mengetik `env\scripts\activate` lalu enter



Gambar 1. 6 Mengaktivasi Environment Python

- g. Melihat pip list dengan cara



Gambar 1. 7 Perintah melihat package yang terinstall

- h. Langkah selanjutnya adalah install streamlit dengan cara ketik `pip install streamlit` lalu enter pada saat buku ini dibuat kami menggunakan streamlit versi 1.12.0 sebagai catatan (butuh koneksi internet ketika install streamlit) seperti pada gambar

```
(env) D:\datamining\app>pip install streamlit
Collecting streamlit
  Using cached streamlit-1.21.0-py2.py3-none-any.whl (9.7 MB)
Collecting altair<5,>=3.2.0 (from streamlit)
  Using cached altair-4.2.2-py3-none-any.whl (813 kB)
```

Gambar 1. 8 menginstall streamlit

- i. Selanjutnya kita melakukan instalasi library yang dibutuhkan seperti matplotlib dan scikit-learn dengan cara ketik pip install matplotlib scikit-learn

```
(env) D:\datamining\app>pip install matplotlib scikit-learn
Collecting matplotlib
  Using cached matplotlib-3.7.1-cp311-cp311-win_amd64.whl (7.6 MB)
Collecting scikit-learn
  Using cached scikit_learn-1.2.2-cp311-cp311-win_amd64.whl (8.3 MB)
Collecting contourpy>=1.0.1 (from matplotlib)
  Using cached contourpy-1.0.7-cp311-cp311-win_amd64.whl (162 kB)
```

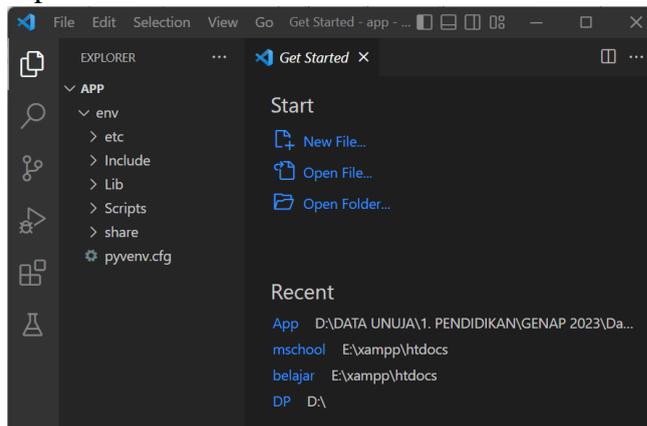
Gambar 1. 9 instalasi matplotlib dan scikit-learn

- j. Langkah selanjutnya kita mencoba menjalankan vs code di cmd dengan cara ketik code . lalu enter seperti pada gambar dibawah ini

```
(env) D:\datamining\app>code .
```

Gambar 1. 10 Mengaktifkan aplikasi VS Code

- k. Selanjutnya akan tampil vs code yang sudah aktif seperti dibawah ini

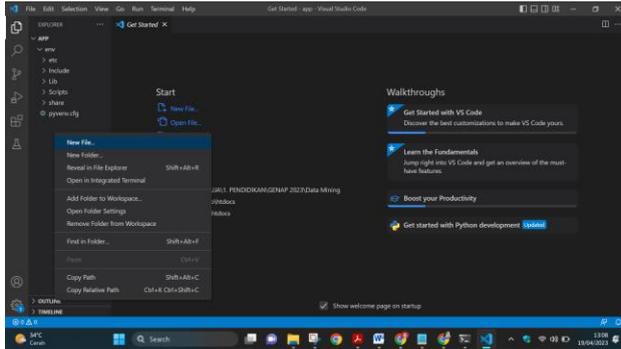


Gambar 1. 11 Tampilan awal VS Code

1.3. Aplikasi Pertama Streamlit

a. Langkah Pertama

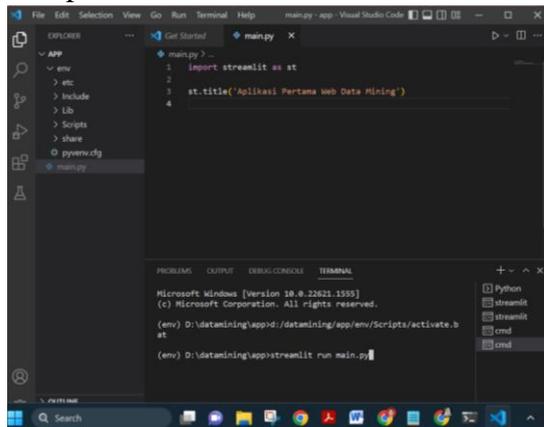
Langkah yang perlu dilakukan adalah membuat new file dengan nama **main.py** seperti pada gambar dibawah ini :



Gambar 1. 12 Pembuatan file streamlit di VS Code

b. Langkah kedua

Selanjutnya file **main.py** yang sudah dibuat sebelumnya di ketikkan code seperti dibawah ini dengan **import streamlit as st** dan memberikan title pada tampilan awal website,



Gambar 1. 13 Tampilan awal coding streamlit

c. Langkah ketiga

Perintah untuk menjalankan aplikasi streamlit yang sudah dibuat dengan cara ketik :

```
streamlit run main.py
```

lalu tekan enter, main.py adalah nama file yang sudah dibuat sebelumnya, pada studi kasus lainnya nanti kalian harus menyesuaikan dengan nama file yang sudah dibuat. jika berhasil maka akan tampil di web browser seperti pada contoh di bawah ini. Pada saat pembuatan buku ini kami menggunakan web browser google chrome Version 112.0.5615.140 (Official Build) (64-bit).



Gambar 1. 14 Tampilan aplikasi streamlit

BAB 2

Membuat Aplikasi Dasar Streamlit

Deskripsi Materi:

Pada bab 2 ini membahas tentang dasar-dasar pembuatan aplikasi di streamlit. Dengan membagi menjadi 6 sub bab materi (elemen text, menampilkan data, menampilkan grafik, menampilkan input form, menampilkan media dan layout & container). Untuk memudahkan dalam memahami setiap materi di buku ini, kami memberikan contoh coding yang bisa kalian ketik ulang disertai dengan penjelasan fungsi dari perintah yang diketik serta hasil program ketika dijalankan di web browser.

2.1. Dasar Streamlit

Pada contoh dibawah ini kita mencoba untuk membuat program sederhana dengan menulis argument yang akan ditampilkan di web browser dengan perintah `st.write()`. Pada langkah awal dipersilahkan untuk membuat file dengan nama `main.py` seperti pada contoh dibawah ini :



File main.py

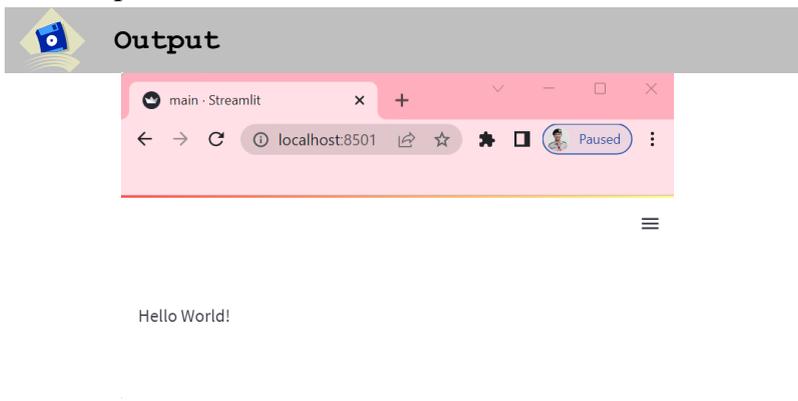
```
import streamlit as st
st.write('Hello World!')
```



Penjelasan

Pada baris pertama kita import library streamlit sebagai `st`, Fungsi `import streamlit as st` adalah untuk mengimpor modul atau pustaka Streamlit ke

dalam skrip Python Anda. Streamlit adalah kerangka kerja yang digunakan untuk membangun aplikasi web interaktif dengan menggunakan Python. Dengan mengimpor modul Streamlit menggunakan sintaks `import streamlit as st`, Anda dapat menggunakan berbagai fitur dan fungsi yang disediakan oleh Streamlit dalam skrip Anda. Setelah mengimpor modul Streamlit, Anda dapat menggunakan berbagai fungsi yang disediakan oleh Streamlit untuk membuat antarmuka pengguna yang interaktif dan dinamis. Beberapa contoh fungsi yang sering digunakan dalam Streamlit adalah `st.write()`, `st.title()`, `st.header()`, dan banyak lagi., selanjutnya pada baris kedua kita ingin menulis argumen dengan perintah `st.write(“)` dengan isinya adalah string. selanjutnya untuk menjalankan aplikasinya silahkan melihat gambar 1.14 Tampilan aplikasi streamlit yang dijelaskan secara rinci sebelum pembahasan di bab ini.



Gambar 2. 1 Aplikasi pertama streamlit

Pada praktek Selanjutnya kita akan menampilkan dataframe, silahkan membuat file dengan nama **main1.py** seperti dibawah ini.

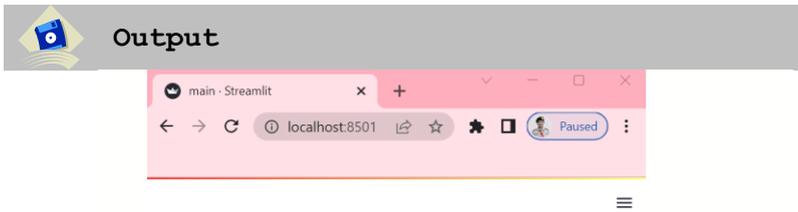


```
File main1.py
1 import streamlit as st
2 import pandas as pd
3
4 st.write(1234)
5 st.write(pd.DataFrame({
6     'first column': [1, 2, 3, 4],
7     'second column': [10, 20, 30, 40],
8 })))
```



Penjelasan

Pada baris kedua kita import library pandas sebagai pd. Pandas adalah library open source pada Python yang sering digunakan untuk memproses data, pada baris keempat kita ingin menulis argumen dengan perintah `st.write(1234)` dengan isinya adalah bilangan, `st.write()` juga menerima format data lain, DataFrame adalah struktur data berlabel dua dimensi dalam pandas, yang merupakan pustaka manipulasi dan analisis data yang populer di Python. DataFrame merepresentasikan data dalam bentuk tabel, mirip dengan tabel dalam spreadsheet atau database. DataFrame terdiri dari baris dan kolom, di mana setiap kolom dapat memiliki tipe data yang berbeda (misalnya, numerik, string, boolean) dan diberi label dengan nama kolom. Contoh penerapan dataframe seperti pada baris kelima sampai ke delapan



1234

	first column	second column
0	1	10
1	2	20
2	3	30
3	4	40

Gambar 2. 2 Tampilan Dataframe

2.2. Elemen Text

Pada pembahasan elemen teks yang kita gunakan seperti **st.markdown()**, **st.title()**, **st.header()**, **st.subheader()**, **st.caption()**, **st.code()**, **st.teks()**, **st.lateks()**, **st.divider()**, yang akan kita jelaskan secara rinci pada pembahasan dibawah ini.

a. Perintah **st.markdown()**



File markdown.py

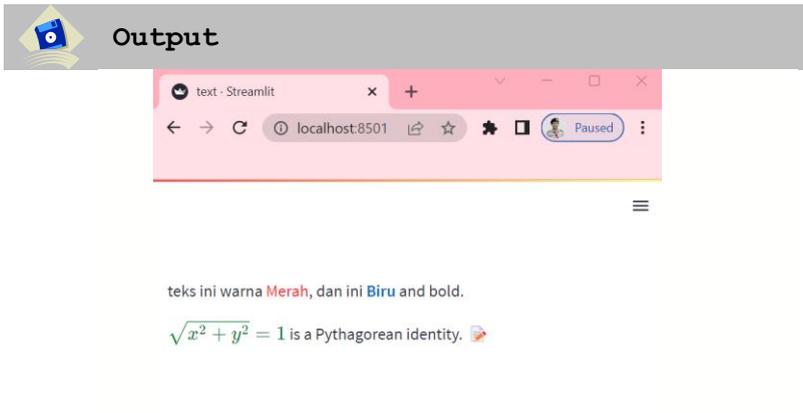
```
1 import streamlit as st
2
3 st.markdown("teks ini warna :red[Merah], dan ini **:blue[Biru]** and bold.")
4 st.markdown(":green[ $\sqrt{x^2+y^2}=1$ ] is a Pythagorean identity. :pencil:")
```



Penjelasan

Fungsi dari **st.markdown()** adalah menampilkan string yang diformat sebagai Markdown. Teks berwarna, menggunakan sintaks `:color[teks` yang akan diwarnai, di mana warna perlu diganti dengan salah satu warna yang didukung berikut: biru, hijau,

oranye, merah, ungu. Sedangkan pada baris ke empat kita mencoba menampilkan contoh sebuah rumus akar kuadrat dengan perintah `sqrt`, sedangkan untuk menambah emoji `:pencil:` kalau kalian tertarik memberikan variasi emoji bisa kunjungi link berikut <https://share.streamlit.io/streamlit/emoji-shortcodes>



Gambar 2. 3 Tampilan Markdown

b. Perintah `st.title()`

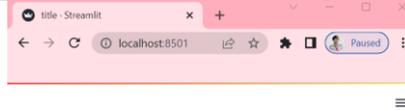
File `title.py`

```
title.py > ...
1 import streamlit as st
2
3 st.title('SELAMAT BELAJAR STREAMLIT')
4 st.title('WEB DEVELOPMENT FOR MACHINE LEARNING')
```

Penjelasan

Fungsi dari `st.title()` adalah menampilkan teks dalam format judul, biasanya ini digunakan sebagai informasi awal di aplikasi yang akan kita buat

Output



SELAMAT BELAJAR STREAMLIT
WEB DEVELOPMENT FOR MACHINE
LEARNING

Gambar 2. 4 Tampilan title

c. Perintah `st.header()`

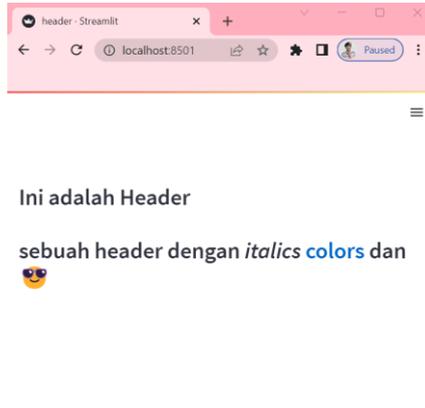
File `header.py`

```
header.py > ...  
1 import streamlit as st  
2  
3 st.header('Ini adalah Header')  
4 st.header('sebuah header dengan _italics_ :blue[colors] dan :sunglasses:')
```

Penjelasan

Fungsi dari `st.header()` adalah elemen-elemen yang mewakili konten pengantar, pada dasarnya sekelompok alat bantu pengantar atau navigasi

Output



Gambar 2. 5 Tampilan header

d. Perintah `st.subheader()`

File `subheader.py`

```
subheader.py > ...
1 import streamlit as st
2
3 st.subheader('Ini adalah sub Header')
4 st.subheader('sebuah header dengan italics :blue[colors] dan :sunglasses:')
```

Penjelasan

Fungsi dari `st.subheader()` adalah Subjudul atau judul tambahan yang muncul tepat setelah judul utama pada halaman website kita

Output



Ini adalah sub Header
sebuah header dengan *italics colors* dan 😊

Made with Streamlit

Gambar 2. 6 Tampilan Subheader

e. Perintah `st.caption()`

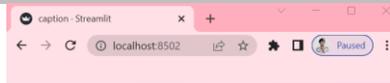
File `subheader.py`

```
caption.py > ...  
1 import streamlit as st  
2 st.caption('Ini adalah tampilan caption')
```

Penjelasan

Fungsi dari `st.caption()` adalah menampilkan teks dengan font paling kecil, ini bisa digunakan untuk penjelasan atau keterangan dari sebuah halaman.

Output



Ini adalah tampilan caption

Gambar 2. 7 Tampilan Caption

f. Perintah `st.code()`

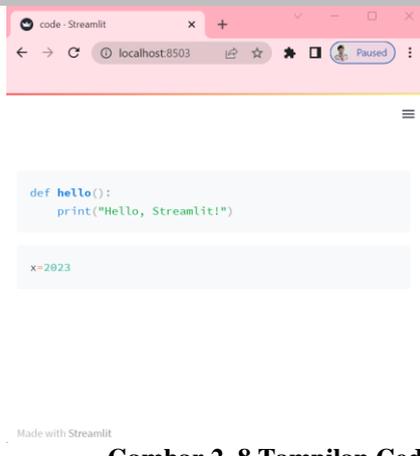
 **File code.py**

```
code.py > ...
1 import streamlit as st
2 code = '''def hello():
3     print("Hello, Streamlit!")'''
4 st.code(code, language='python')
5 st.code('x=2023')
```

Penjelasan

Fungsi dari `st.code()` adalah menampilkan blok kode dengan penyorotan sintaks opsional

Output



Gambar 2. 8 Tampilan Code

g. Perintah `st.text()`

File `code.py`

```
text.py > ...  
1 import streamlit as st  
2  
3 st.text("ini adalah contoh text di streamlit")
```

Penjelasan

Fungsi dari `st.text()` adalah menampilkan text di website, fungsi ini jarang digunakan karena bisa menggunakan fungsi `st.title()` atau `st.header()`

Output



Gambar 2.9 Tampilan Text

h. Perintah `st.latex()`

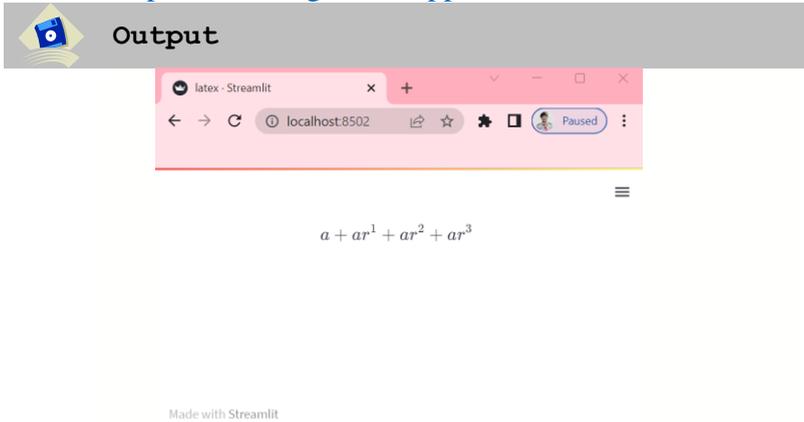
File `latex.py`

```
latex.py > ...  
1 import streamlit as st  
2 st.latex(r''' a+a r^1+a r^2+a r^3 ''')
```

Penjelasan

Fungsi dari `st.latex()` adalah Menampilkan ekspresi matematika yang diformat sebagai LaTeX. Fungsi LaTeX secara lengkap yang didukung tercantum di

<https://katex.org/docs/supported.html>



Gambar 2. 10 Tampilan Latex

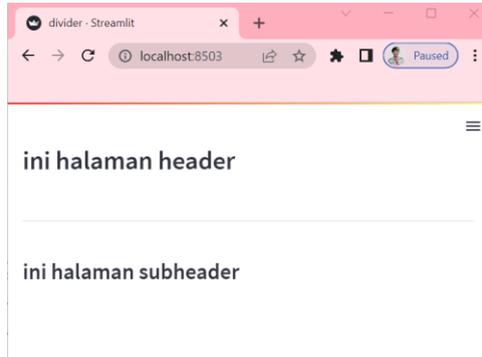
i. Perintah `st.divider()`

```
File latex.py
divider.py > ...
1 import streamlit as st
2 st.header('ini halaman header')
3 st.divider()
4 st.subheader('ini halaman subheader')
```

Penjelasan

Fungsi dari `st.divider()` adalah menampilkan baris secara horizontal, biasanya digunakan sebagai pembagi antara dua keterangan seperti contoh dibawah ini.

Output



Gambar 2. 11 Tampilan Divider

2.3. Menampilkan Data

Pada materi ini anda akan mempelajari bagaimana memvisualisasi data tersebut dengan cepat dan interaktif Untuk itulah web Streamlit sebenarnya dibuat. beberapa perintah yang digunakan adalah `st.dataframe()`. `st.tabel()`. `st.metric()`. `st.json()` untuk lebih jelasnya kita langsung praktek dan dengan penjelasan yang sangat rinci seperti dibawah ini :

a. Perintah `dataframe()`

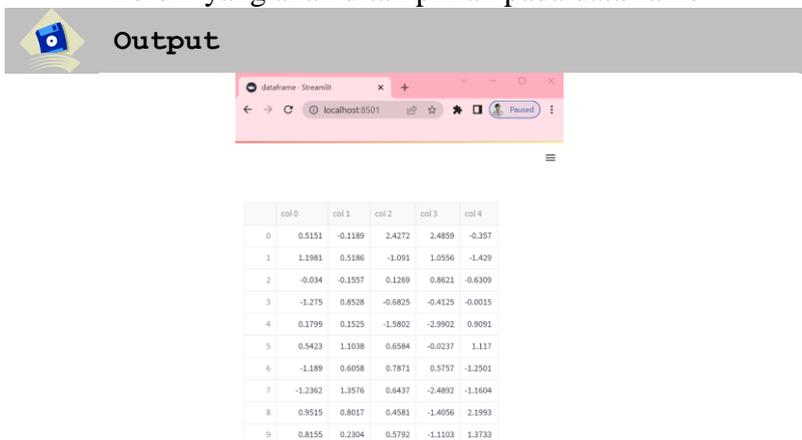
 **File dataframe.py**

```
dataframe.py > ...
1 import streamlit as st
2 import pandas as pd
3 import numpy as np
4 st.title('Belajar DataFrame')
5 df = pd.DataFrame(
6     np.random.randn(10, 5),
7     columns=('col %d' % i for i in range(5)))
8 st.dataframe(df)
```

 **Penjelasan**

pada baris kedua kita import pandas sebagai pd

dengan fungsi **pandas** adalah sebuah library pada bahasa pemrograman Python untuk memanipulasi dan analisis data. Secara khusus, pandas menawarkan struktur data dan operasi untuk memanipulasi tabel numerik dan deret waktu, selanjutnya pada baris ketiga kita import numpy sebagai np. fungsi **NumPy** adalah sebuah library pada bahasa pemrograman Python yang digunakan untuk bekerja dengan Array. NumPy juga memiliki fungsi untuk bekerja dalam domain aljabar linier, transformasi fourier, dan matriks. Pada baris keenam kumpulan sampel secara acak dari 10 baris dan 5 kolom yang akan ditampilkan pada dataframe



Gambar 2. 12 Tampilan Dataframe

belajar DataFrame Python adalah langkah penting dalam mempelajari analisis data, pengolahan data, dan pemrosesan data yang efisien. DataFrame memberikan kerangka kerja yang kuat untuk mengelola, memanipulasi, dan menganalisis data dengan mudah menggunakan bahasa pemrograman Python

b. Perintah st.table()

File table.py

```
table.py > ...  
1 import streamlit as st  
2 import pandas as pd  
3 import numpy as np  
4 st.title('Belajar DataFrame')  
5 df = pd.DataFrame(  
6     np.random.randn(10, 5),  
7     columns=('Kolom %d' % i for i in range(5)))  
8 st.table(df)
```

Penjelasan

Menampilkan tabel secara statis. Ini berbeda dari st.dataframe karena tabel dalam hal ini statis: seluruh isinya diletakkan langsung di halaman.

Output



Belajar Membuat Tabel

	Kolom 0	Kolom 1	Kolom 2	Kolom 3	Kolom 4
0	-0.6043	0.2331	-0.2340	0.8202	0.7862
1	1.9588	-0.2090	1.2166	-0.6956	0.2502
2	1.2293	0.7220	0.0640	1.2227	-0.8655
3	-0.7636	0.8768	-1.2756	0.1944	0.6463
4	0.6366	-1.6980	0.8636	-0.2315	1.8041
5	0.9114	0.6995	-0.0839	0.6768	-0.0901
6	-0.6909	0.6325	0.5482	-0.0033	0.3381
7	0.2443	-0.8020	-0.3807	1.7937	0.4308
8	0.7817	0.2251	0.3380	0.7596	0.1339
9	-1.7560	-0.3291	-0.9595	1.9128	-0.1388

Gambar 2. 13 Tampilan Tabel

c. File metric.py



File metric.py

```
metric.py > ...
1 import streamlit as st
2 st.metric(label="Temperature", value="70 °F", delta="1.2 °F")
3 st.caption('dibawah ini kombinasi coding st.metric() dengan st.column()')
4 st.divider()
5 kolom1, kolom2, kolom3 = st.columns(3)
6 kolom1.metric("Temperature", "70 °F", "1.2 °F")
7 kolom2.metric("Wind", "9 mph", "-8%")
8 kolom3.metric("Humidity", "86%", "4%")
```

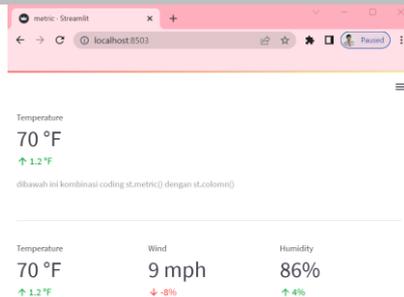


Penjelasan

Fungsi **st.metric()** adalah menampilkan metrik dalam font tebal besar, dengan indikator opsional tentang bagaimana metrik berubah seperti pada gambar dibawah ini.



Output



Gambar 2. 14 Tampilan Metric

Dengan menggunakan **st.metric()**, Anda dapat membuat panel metrik yang menggambarkan kinerja sistem atau aplikasi dalam waktu nyata. Anda dapat memperbarui nilai metrik secara otomatis atau berdasarkan interaksi pengguna.

d. Perintah st.json()

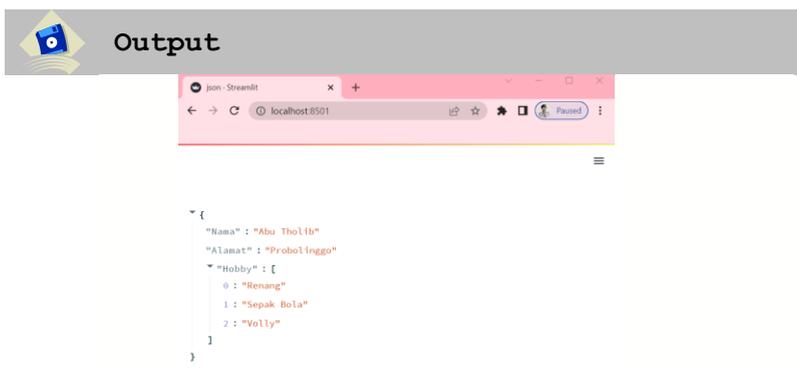


```
File metric.py
json.py > ...
1 import streamlit as st
2
3 st.json({
4     'Nama': 'Abu Tholib',
5     'Alamat': 'Probolinggo',
6     'Hobby': [
7         'Renang',
8         'Sepak Bola',
9         'Volly',
10    ],
11 })
```



Penjelasan

Fungsi st.json() adalah menampilkkan objek atau string sebagai string JSON yang ditampilkan dengan sangat menarik.



Gambar 2. 15 Tampilan Json

2.4. Menampilkan Grafik

Pada materi ini kita akan membahas kemampuan untuk memvisualisasikan data. Grafik dan visualisasi membantu kita memahami data secara intuitif dan membuat pola atau tren yang mungkin tidak terlihat dari data mentah. Streamlit adalah platform yang memungkinkan pengembang dan analis data untuk membuat aplikasi web interaktif yang memungkinkan pengguna untuk mengeksplorasi dan memvisualisasikan data dengan mudah dan cepat. Kita akan menjelaskan pembuatan grafik dengan beberapa perintah seperti `st.line_chart()`, `st.area_chart()`, `st.bar_chart()`, `st.pyplot()`, `st.altair_chart()`, `st.vega_lite_chart()`, `st.plotly_chart()`, `st.bokeh_chart()`, `st.pydeck_chart()`, `st.graphviz_chart()`, `st.map()`. Kami memberikan banyak contoh kode dan demo aplikasi untuk membantu Anda memahami konsep-konsep dasar dan mempraktekkannya dengan cepat. materi ini akan membantu Anda meningkatkan kemampuan visualisasi data Anda dengan menggunakan Streamlit dan membantu Anda membuat aplikasi web interaktif yang menarik dan informatif. Selanjutnya kita langsung praktek saja ya. Kalian langsung saja membuat file dengan nama `line.py` dengan isi coding seperti dibawah ini :

a. Perintah `st.line()`



File line.py

```
line.py > ...
1 import streamlit as st
2 import pandas as pd
3 import numpy as np
4 chart_data = pd.DataFrame(
5     np.random.randn(5, 3),
6     columns=['Kolom1', 'Kolom2', 'Kolom3'])
7 st.table(chart_data)
8 st.line_chart(chart_data)
```

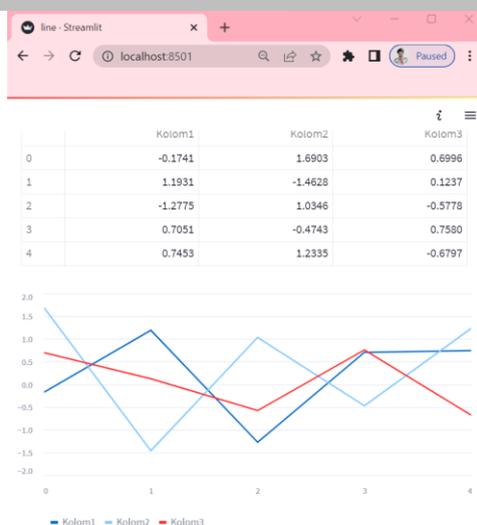


Penjelasan

Fungsi `st.line_chart()` adalah menampilkan grafik dalam bentuk garis. Pada baris kesatu sampai ketiga kita tetap import library streamlit, pandas dan numpy agar ketika dibutuhkan untuk manipulasi data. Selanjutnya pada baris keempat memanggil fungsi `pd.dataframe` untuk membuat table baris dan kolom. Pada baris ketujuh kita menampilkan data dalam bentuk table dengan perintah `st_table()`. Pada baris kedelapan kita menampilkan grafik baris dengan perintah `st.line_chart()` dengan mengambil data dari variable `chart_data`



Output



Gambar 2. 16 Grafik Garis

Dengan menerapkan `st.line_chart()` kalian akan dapat memvisualisasikan data yang ada lebih menarik, sehingga memudahkan dalam pembacaan grafik secara garis

b. Perintah st.area()



File area.py

```
area.py > ...
1 import streamlit as st
2 import pandas as pd
3 import numpy as np
4 chart_data = pd.DataFrame(
5     np.random.randn(5, 3),
6     columns=['Kolom1', 'Kolom2', 'Kolom3'])
7 st.table(chart_data)
8 st.area_chart(chart_data)
```

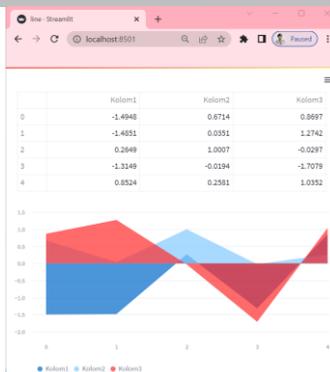


Penjelasan

Fungsi `st.area_chart()` adalah menampilkan grafik dalam bentuk area. Tidak ada perubahan coding dari baris kesatu sampai ketujuh, pada baris kedelapan kita akan menampilkan grafik area dengan perintah `st.area_chart()` dengan mengambil data dari variabel `chart_data`. Sehingga dibawah ini dapat kita lihat beberapa perbedaan dari hasil percobaan kita dengan perintah menggunakan `st.line_chart()` seperti di web browser ketika coding sudah dijalankan.



Output



Gambar 2. 17 Grafik Area

c. Perintah st.bar.py



File bar.py

```
bar.py > ...
1 import streamlit as st
2 import pandas as pd
3 import numpy as np
4 chart_data = pd.DataFrame(
5     np.random.randn(5, 3),
6     columns=['Kolom1', 'Kolom2', 'Kolom3'])
7 st.table(chart_data)
8 st.bar_chart(chart_data)
```

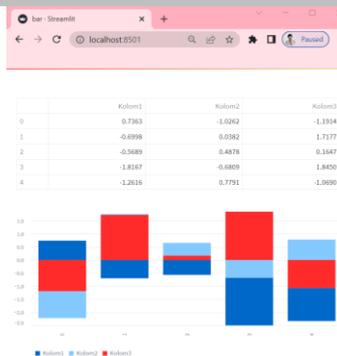


Penjelasan

Fungsi `st.bar_chart()` adalah menampilkan Grafik dalam bentuk batang. Tidak ada perubahan coding dari baris kesatu sampai ketujuh, pada baris kedelapan kita akan menampilkan Grafik batang dengan perintah `st.bar_chart()` dengan mengambil data dari variabel `chart_data`. Sehingga dibawah ini dapat kita lihat beberapa perbedaan dari hasil percobaan kita dengan perintah sebelumnya seperti di web browser ketika coding sudah dijalankan.



Output



Gambar 2. 18 Grafik Batang

d. Perintah st.altair()

File altair.py

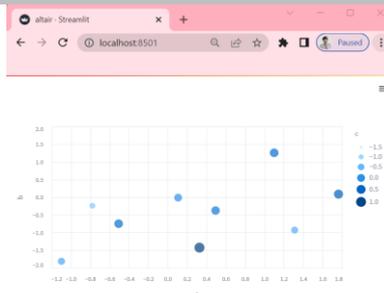
```
altair.py > ...
1 import streamlit as st
2 import pandas as pd
3 import numpy as np
4 import altair as alt
5 chart_data = pd.DataFrame(
6     np.random.randn(10, 3),
7     columns=['a', 'b', 'c'])
8 c = alt.Chart(chart_data).mark_circle().encode(
9     x='a', y='b', size='c', color='c', tooltip=['a', 'b', 'c'])
10 st.altair_chart(c, use_container_width=True)
```

Penjelasan

Fungsi `st.altair_chart()` adalah menampilkan grafik dengan library yang dimiliki oleh altair agar visualisasi data lebih menarik.

Contoh lengkap Grafik altair bisa dilihat di situs <https://altair-viz.github.io/gallery/>

Output



Gambar 2. 19 Grafik Altair

Catatan : sebelum kita praktek import data dari `vega_datasets` di streamlit maka kita perlu menginstall dulu library yang dibutuhkan dengan cara ketik `pip install`

vega_datasets seperti pada gambar dibawah ini

```
(env) D:\datamining\app>pip install vega_datasets
Collecting vega_datasets
  Downloading vega_datasets-0.9.0-py3-none-any.whl (210 kB)
    |#####| 218.0/218.0 kB 754.1 kB/s eta 0:00:00
Requirement already satisfied: pandas in d:\datamining\app\env\lib\site-packages (from vega_datasets) (1.5.3)
Requirement already satisfied: python-dateutil>=2.8.1 in d:\datamining\app\env\lib\site-packages (from pandas->vega_datasets) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in d:\datamining\app\env\lib\site-packages (from pandas->vega_datasets) (2023.3)
Requirement already satisfied: numpy>=1.21.0 in d:\datamining\app\env\lib\site-packages (from pandas->vega_datasets) (1.24.2)
Requirement already satisfied: six>=1.5 in d:\datamining\app\env\lib\site-packages (from python-dateutil>=2.8.1->pandas->vega_datasets) (1.16.0)
Installing collected packages: vega_datasets
Successfully installed vega_datasets-0.9.0
```

Gambar 2. 20 Menginstall Library Vega Dataset

e. Perintah st.vegalite()

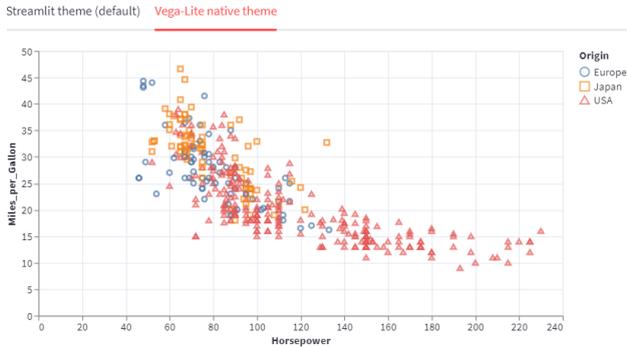
File altair.py

```
vegalite.py > ...
1 import streamlit as st
2 from vega_datasets import data
3 source = data.cars()
4 chart = {
5     "mark": "point",
6     "encoding": {
7         "x": {
8             "field": "Horsepower",
9             "type": "quantitative",
10        },
11        "y": {
12            "field": "Miles_per_Gallon",
13            "type": "quantitative",
14        },
15        "color": {"field": "Origin", "type": "nominal"},
16        "shape": {"field": "Origin", "type": "nominal"},
17    },
18 }
19 tab1, tab2 = st.tabs(["Streamlit theme (default)", "Vega-Lite native theme"])
20 with tab1:
21     # This is the default. So you can also omit the theme argument.
22     st.vega_lite_chart(source, chart, theme="streamlit", use_container_width=True)
23 with tab2:
24     st.vega_lite_chart(source, chart, theme=None, use_container_width=True)
```

Penjelasan

Fungsi `st.vega_lite_chart()` adalah menampilkan Grafik menggunakan library `vega_lite` dengan kelebihan visualisasi lebih menarik.

Output



Gambar 2. 21 Grafik Vega Lite

Catatan : sebelum kita praktek terkait dengan plotly kita butuh import plotly di streamlit dengan cara menginstall dulu dengan cara ketik **pip install plotly** lalu enter seperti pada contoh dibawah ini :

```
(env) D:\datamining\app>pip install plotly
Collecting plotly
  Downloading plotly-5.14.1-py2.py3-none-any.whl (15.3 MB)
     |-----| 15.3/15.3 MB 3.8 MB/s eta 0:00:00
Collecting tenacity>=6.2.0 (from plotly)
  Downloading tenacity-8.2.2-py3-none-any.whl (24 kB)
Requirement already satisfied: packaging in d:\datamining\app\env\lib\site-packages
Installing collected packages: tenacity, plotly
Successfully installed plotly-5.14.1 tenacity-8.2.2
```

Gambar 2. 22 Menginstall Library Plotly

f. Perintah st.plot()



```

plot.py > ...
1 import plotly.express as px
2 import streamlit as st
3 df = px.data.gapminder()
4 fig = px.scatter(
5     df.query("year==2007"),
6     x="gdpPercap",
7     y="lifeExp",
8     size="pop",
9     color="continent",
10    hover_name="country",
11    log_x=True,
12    size_max=60,
13 )
14 tab1, tab2 = st.tabs(["Streamlit theme (default)", "Plotly native theme"])
15 with tab1:
16     # This is the default. So you can also omit the theme argument.
17     st.plotly_chart(fig, theme="streamlit", use_container_width=True)
18 with tab2:
19     # Use the native Plotly theme.
20     st.plotly_chart(fig, theme=None, use_container_width=True)

```

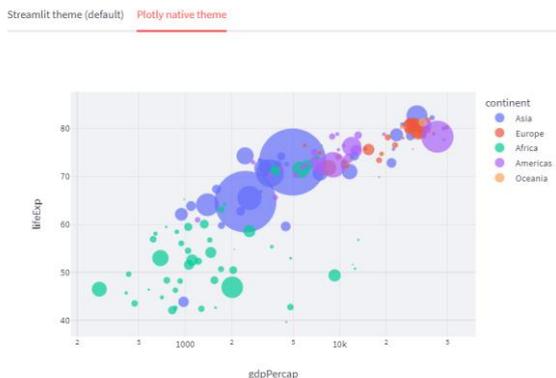


Penjelasan

Fungsi `st.plotly_chart()` adalah Menampilkan Grafik Plotly interaktif. Plotly adalah library pembuatan Grafik untuk Python. fungsi ini sangat mirip dengan argumen untuk fungsi `plot()` Plotly . Anda dapat menemukan lebih banyak tentang Plotly di <https://plot.ly/python>



Output



Gambar 2. 23 Grafik Plotly

g. Perintah st.map()



File map.py

```
map.py > ...
1 import streamlit as st
2 import pandas as pd
3 import numpy as np
4 df = pd.DataFrame(
5     np.random.randn(1000, 2) / [50, 50] + [37.76, -12
6     columns=['lat', 'lon'])
7 st.map(df)
```

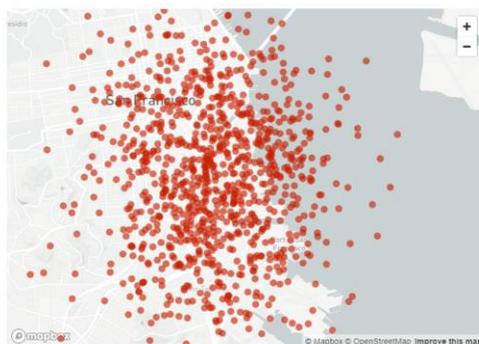


Penjelasan

Fungsi `st.map()` adalah menampilkan peta dengan titik di atasnya dengan pemusatan otomatis dan pembesaran otomatis. Visualisasi Data Geografis dengan Fungsi `st.maps()` memungkinkan Anda memvisualisasikan data geografis pada peta interaktif. Anda dapat menampilkan titik, garis, atau area pada peta berdasarkan data geografis yang Anda miliki. Ini membantu pengguna aplikasi Anda memahami dan mengeksplorasi data dalam konteks geografis



Output



Gambar 2. 24 Tampilan Peta

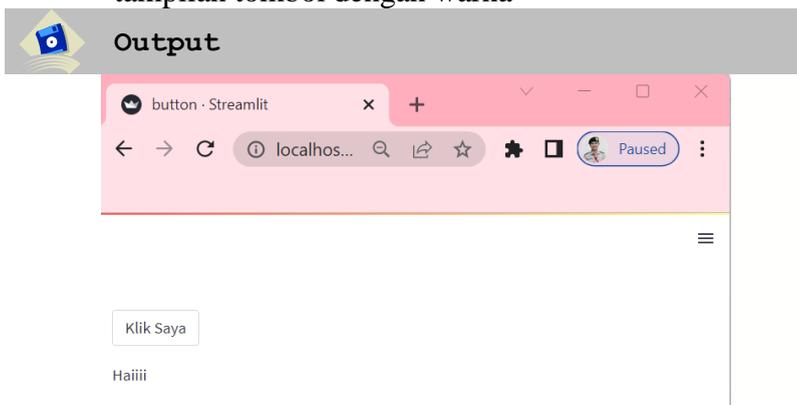
2.5. Menampilkan Input Form

a. Perintah `st.button()`

```
File button.py
button.py > ...
1 import streamlit as st
2 if st.button('Klik Saya'):
3     st.write('Selamat Belajar Button')
4 else:
5     st.write('Haiiii')
```

Penjelasan

Fungsi `st.button()` adalah merupakan sebuah coding yang digunakan untuk menciptakan sebuah tombol klik. Merupakan suatu hal yang sangat lazim digunakan untuk membuka form atau menjalankan suatu perintah. Dibawah ini contoh hasil ketika dijalankan di web browser yang tanpa modifikasi tampilan tombol dengan warna



Gambar 2. 25 Tampilan Button

b. Perintah `st.experimental_data_editor()`

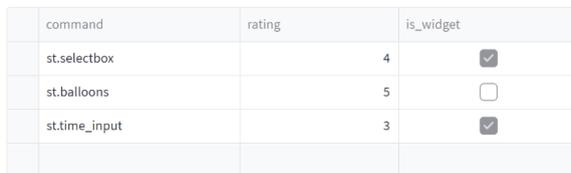
File `dataeditor.py`

```
dataeditor.py > ...
1 import streamlit as st
2 import pandas as pd
3 df = pd.DataFrame(
4     [
5         {"command": "st.selectbox", "rating": 4, "is_widget": True},
6         {"command": "st.balloons", "rating": 5, "is_widget": False },
7         {"command": "st.time_input", "rating": 3, "is_widget": True},
8     ]
9 )
10 tampil = st.experimental_data_editor(df, num_rows="dynamic", use_container_width=True)
```

Penjelasan

Fungsi `st.experimental_data_editor()` adalah menampilkan widget editor data yang memungkinkan Anda mengedit DataFrames dan banyak struktur data lainnya dalam UI seperti tabel. Pada baris ketiga kita membuat dataframe secara statis. Selanjutnya pada baris kesepuluh kita mengatur dataframe sekiranya bisa di edit dengan perintah `st.experimental_data_editor()`. Dengan parameter `num_rows="dinamic"` fungsinya Menentukan apakah pengguna dapat menambahkan dan menghapus baris di editor data. Juga parameter `use_container_width` kita buat `True` agar lebar tampilan dataframe menyesuaikan dengan web browser.

Output



command	rating	is_widget
st.selectbox	4	<input checked="" type="checkbox"/>
st.balloons	5	<input type="checkbox"/>
st.time_input	3	<input checked="" type="checkbox"/>

Gambar 2. 26 Tampilan Experiment Data Editor

c. Perintah `st.download_button()`



File `download.py`

```
download.py > ...
1 import streamlit as st
2 import pandas as pd
3 import numpy as np
4 @st.cache_data
5 def convert_df(df):
6     #convert dataframe to csv
7     return df.to_csv().encode('utf-8')
8 #membuat contoh dataframe yang akan di download
9 df = pd.DataFrame(np.random.randn(800, 2))
10 csv = convert_df(df)
11 st.download_button(
12     label="Download data",
13     data=csv,
14     file_name='data.csv',
15     mime='text/csv',
16 )
```

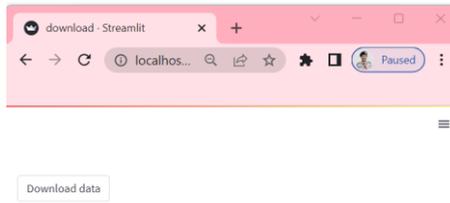


Penjelasan

Fungsi `st.download_button()` adalah Menampilkan widget tombol unduh. Ini berguna saat Anda ingin menyediakan cara bagi pengguna untuk mengunduh file langsung dari aplikasi Anda. Pada latihan ini saya membuat data random dengan menggunakan `pd.dataframe()` sebanyak 800 kolom dan 2 baris. Perhatikan bahwa data yang akan diunduh disimpan dalam memori saat pengguna terhubung. Pada baris kesebelas kita membuat tombol download dengan perintah `st.download_button()` dengan nama file ketika di download `data.csv`. ada beberapa type file download yang bisa dipelajari seperti text, binary, jpg yang masih belum kami jelaskan di buku ini.



Output



Gambar 2. 27 Tampilan Button Download

d. Perintah `st.checkbox()`

Pada file `checkbox.py` saya memasukkan 3 perintah inputan (`checkbox`, `radio`, `selectbox`) dengan tujuan kemudahan dan juga meringkas materi

File `checkbox.py`

```
checkbox.py > ...  
1 import streamlit as st  
2 st.checkbox('ini adalah checkbox')  
3 st.radio("Option",('Laki-laki', 'Perempuan'))  
4 st.selectbox("Menu",('Ayam', 'ikan bakar'))
```

Penjelasan

Fungsi dari perintah `st.checkbox()` adalah

Output

A screenshot of the Streamlit application's output. It displays a checkbox labeled 'ini adalah checkbox' which is currently unchecked. Below it is a radio button group titled 'Option' with two options: 'Laki-laki' (selected) and 'Perempuan'. At the bottom is a select box titled 'Menu' with a dropdown arrow, showing 'Ayam' as the selected option and 'ikan bakar' as another available option.

Gambar 2. 28 Tampilan Checkbox

e. Perintah `st.multiselect()`



File `multiselect.py`

```
multiselect.py > ...
1 import streamlit as st
2 options = st.multiselect('Hobby Kamu ?',['Swimming', 'Basket', 'Soccer',])
3 st.write('Kamu Memilih:', options)
```

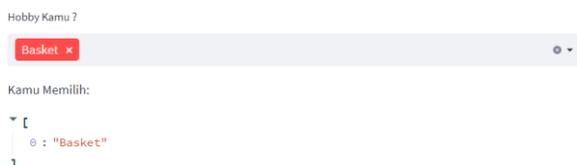


Penjelasan

Fungsi `st.multiselect()` adalah menampilkan widget dengan multi pilihan, ini bisa digunakan ketika dibutuhkan tidak hanya 1 pilihan saja sehingga memudahkan dalam pemilihan opsi.



Output



Gambar 2. 29 Tampilan Multi Select

f. Perintah `st.slider()`



File `slider.py`

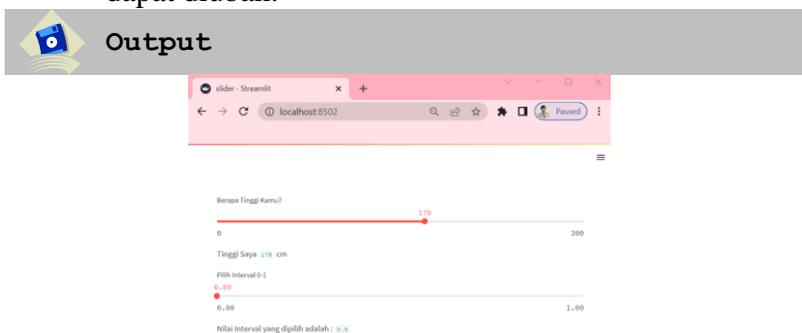
```
slider.py > ...
1 import streamlit as st
2 tinggi = st.slider('Berapa Tinggi Kamu?', 0, 300, 170)
3 st.write("Tinggi Saya ", tinggi, 'cm')
4 nilai = st.slider('Pilih Interval 0-1', min_value=0.0, max_value=1.0, st
5 st.write('Nilai Interval yang dipilih adalah :', nilai)
```



Penjelasan

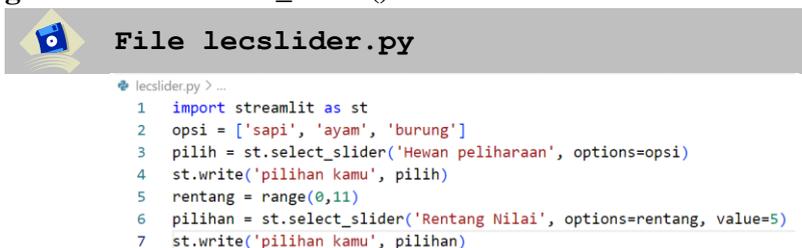
Fungsi `st.slider()` adalah Menampilkan widget penggeser. Ini mendukung tipe `int`, `float`, `date`, `time`,

dan datetime. Ini juga memungkinkan Anda merender penggeser rentang yang sering digunakan di data science maupun ML. dengan meneruskan tuple atau daftar dua elemen sebagai nilainya Perbedaan antara `st.slider()` dan `st.select_slider()` adalah bahwa slider hanya menerima data numerik atau tanggal/waktu dan menggunakan rentang sebagai input, sedangkan `select_slider` menerima tipe data apa pun dan mengambil serangkaian opsi yang dapat diubah.



Gambar 2. 30 Tampilan Slider

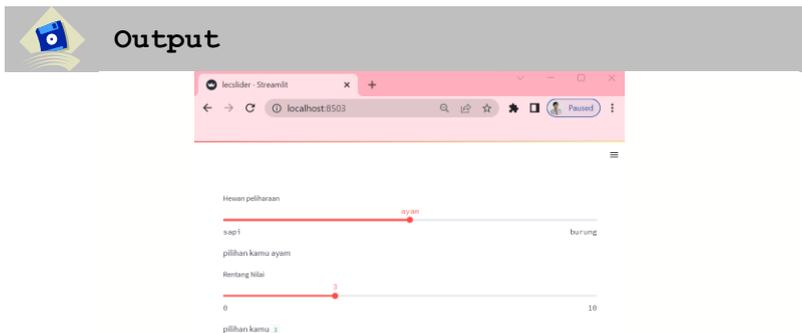
g. Perintah `st.select_slider()`



Penjelasan

Fungsi `st.select_slider()` adalah menampilkan widget penggeser untuk memilih item dari daftar. Perbedaan antara `st.select_slider` dan `st.slider` adalah

bahwa `select_slider` menerima tipe data apa pun dan mengambil serangkaian opsi yang dapat diubah, sedangkan slider hanya menerima data numerik atau tanggal/waktu dan menggunakan rentang sebagai input.



Gambar 2. 31 Tampilan Select Slider

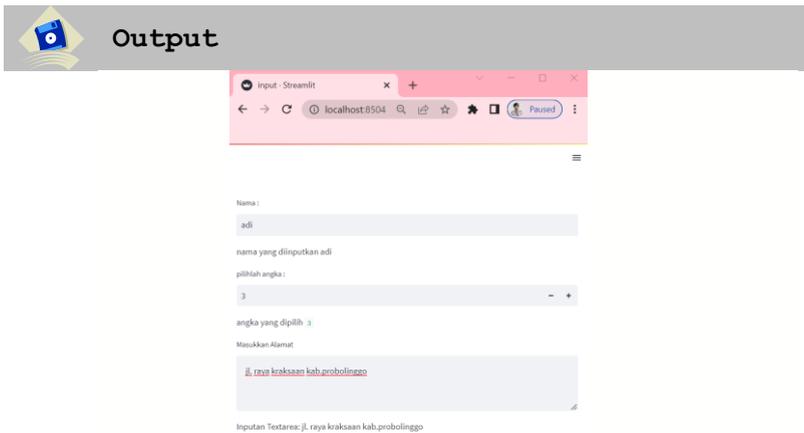
h. Perintah `st.text_input()`



Penjelasan

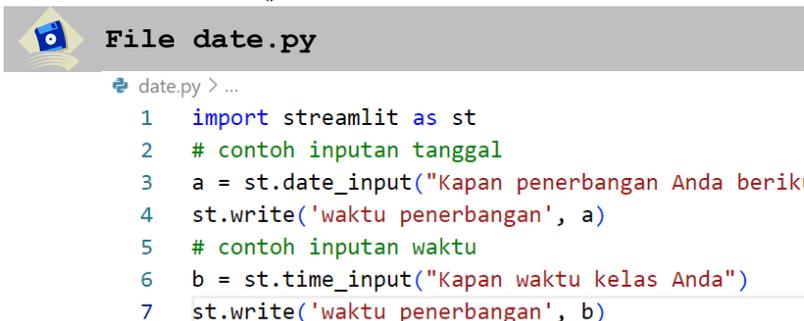
File `input.py` saya memasukkan tiga fungsi inputan untuk merangkum pembahasan yang pertama adalah `st.text_input()` fungsinya adalah menampilkan inputan berupa text. Sedangkan `st.number_input()` fungsinya adalah menampilkan inputan berupa

angka yang saya tambah minimal nilai dengan 0 dan maksimal nilai 10 yang bisa dirubah sesuai dengan kebutuhan, selanjutnya `st.text_area()` fungsinya adalah menampilkan input text multi baris ketika kebutuhan inputan text yang lebih panjang.



Gambar 2. 32 Tampilan Inputan Text

i. Perintah `st.date()`

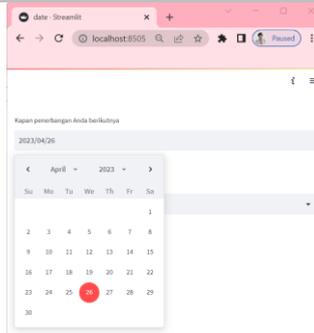


Penjelasan

Fungsi `st.date_input()` adalah menampilkan inputan tanggal yang secara default mengambil tanggal yang

ada di komputer anda. Sedangkan fungsi `st.time_input()` adalah menampilkan inputan waktu

Output



Gambar 2. 33 Tampilan Date

j. Perintah `st.file_upload()`

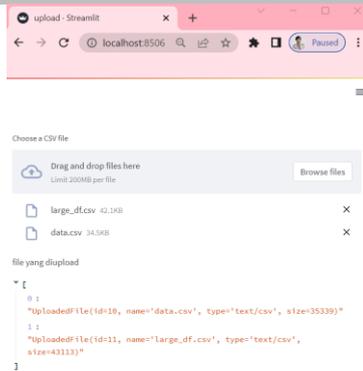
File upload.py

```
upload.py > ...
1 import streamlit as st
2 upload = st.file_uploader("Choose a CSV file", accept_multiple_files=True)
3 st.write('file yang diupload', upload)
```

Penjelasan

Fungsi `st.file_uploader()` adalah Menampilkan widget pengunggah file. Secara default, file yang diunggah dibatasi hingga 200MB. Kalian tinggal klik browse file dan memilih file yang akan di upload. Pada baris kedua fungsi dari `accept_multiple_file` jika nilai `True` maka mengizinkan untuk upload beberapa file sekaligus, jika nilai `False` maka hanya mengizinkan satu file saja.

Output



Gambar 2. 34 Tampilan Upload

k. Perintah `st.camera_input()`

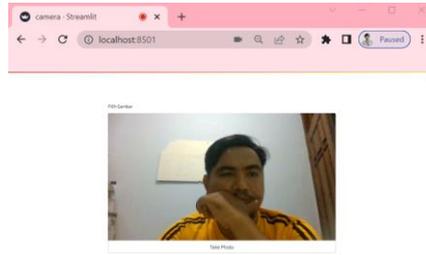
File `camera.py`

```
camera.py > ...  
1 import streamlit as st  
2 picture = st.camera_input("Pilih Gambar")  
3 if picture:  
4     st.image(picture)
```

Penjelasan

Fungsi `st.camera_input()` adalah menampilkan inputan gambar dari camera pengguna

Output



Gambar 2. 35 Tampilan Camera

1. Perintah `st.color_picker()`

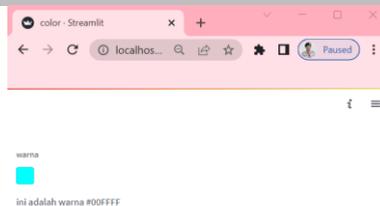
File `camera.py`

```
color.py > ...  
1 import streamlit as st  
2 warna = st.color_picker('warna', value='#00FFFF')  
3 st.write('ini adalah warna', warna)
```

Penjelasan

Fungsi `st.color_picker()` adalah Menampilkan widget pemilih warna. Parameter yang digunakan dalam value menggunakan warna hexadecimal, jika tidak ada maka default warnanya adalah hitam.

Output



Gambar 2. 36 Tampilan Color Picker

2.6. Menampilkan Media

a. Perintah `st.image()`

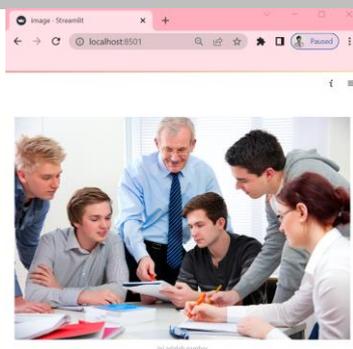
File `image.py`

```
image.py > ...  
1 import streamlit as st  
2 from PIL import Image  
3 image = Image.open('bimbingan.jpg')  
4 st.image(image, caption='ini adalah gambar')
```

Penjelasan

Fungsi `st.image()` adalah menampilkan gambar, dibaris kedua kita import library Pillow dengan nama `PIL` dengan fungsi utamanya adalah manipulasi gambar, pada baris ketiga kita membuka file gambar dengan `bimbingan.jpg` (file ini sudah satu folder dengan file `image.py`). selanjutnya pada baris keempat kita menampilkan file gambar dengan perintah `st.image()` dengan memberikan `caption` seperti diatas

Output



Gambar 2. 37 Tampilan Audio

b. File audio.py

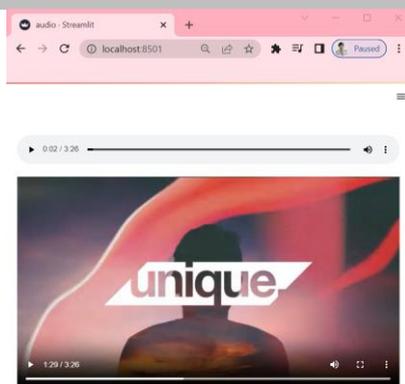
File audio.py

```
audio.py > ...  
1 import streamlit as st  
2 st.audio("audio/baby.mp3")  
3 st.video("audio/baby.mp4")
```

Penjelasan

Fungsi **st.audio()** adalah menampilkan file audio dengan format mp3, sedangkan fungsi **st.video()** adalah menampilkan file audio dengan format mp4

Output



Gambar 2. 38 Tampilan audio

2.7. Menampilkan Layout dan Container

Pada sub bab ini menjelaskan bagaimana konsep pengaturan tata letak tampilan layar di web browser agar lebih interaktif dan menarik dengan beberapa opsi yang bisa

kita lakukan dengan menggunakan sidebar, column, tab, expander, container dan empty.

a. Perintah st.sidebar()

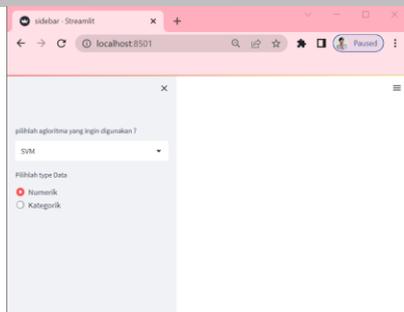
File sidebar.py

```
sidebar.py > pilih_
1 import streamlit as st
2 # Menggunakan notasi objek
3 pilih = st.sidebar.selectbox(
4     "pilihlah algoritma yang ingin digunakan ?",
5     ("SVM", "KNN", "K-Means")
6 )
7 # Menggunakan notasi "with"
8 with st.sidebar:
9     pilih_radio = st.radio(
10        "Pilihlah type Data",
11        ("Numerik", "Kategorik")
12    )
```

Penjelasan

Pada pembuatan sidebar kita tetap membutuhkan library streamlit, sehingga pada baris kesatu menggunakan import streamlit, ada dua cara untuk menampilkan sidebar, pertama kita menggunakan notasi objek seperti pada baris ketiga, cara kedua kita menggunakan notasi with seperti pada baris kedelapan.

Output



Gambar 2. 39 Tampilan Sidebar

b. Perintah st.column()



File columns.py

```
column.py > ...
1 import streamlit as st
2 import pandas as pd
3 import numpy as np
4 col1, col2 = st.columns(2)
5 with col1:
6     chart_data = pd.DataFrame(
7         np.random.randn(5, 3),
8         columns=['Kolom1', 'Kolom2', 'Kolom3'])
9     st.table(chart_data)
10 with col2:
11     chart_data = pd.DataFrame(
12         np.random.randn(5, 3),
13         columns=['Kolom1', 'Kolom2', 'Kolom3'])
14     st.area_chart(chart_data)
```

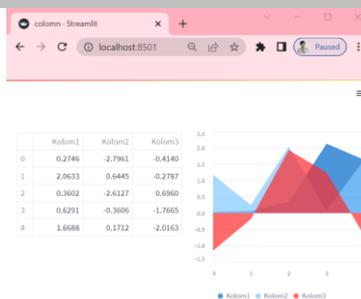


Penjelasan

Fungsi `st.column()` di Streamlit digunakan untuk mengatur tata letak kolom dalam aplikasi web yang dibangun menggunakan Streamlit. manfaat dan penggunaan fungsi `st.column()` memungkinkan Anda untuk membagi tata letak aplikasi web menjadi beberapa kolom secara vertikal. Anda dapat mengelompokkan elemen-elemen antarmuka pengguna menjadi kolom-kolom yang terpisah,



Output



Gambar 2. 40 Tampilan Colomn

c. Perintah st.tab()

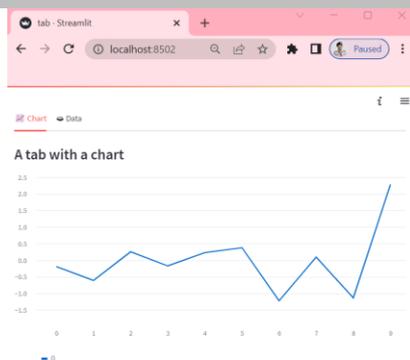
File tab.py

```
tab.py > tab1
1 import streamlit as st
2 import numpy as np
3 tab1, tab2 = st.tabs(["Chart", "Data"])
4 data = np.random.randn(10, 1)
5 tab1.subheader("contoh tab dengan grafik")
6 tab1.line_chart(data)
7 tab2.subheader("contoh tab dengan data ")
8 tab2.write(data)
```

Penjelasan

Fungsi dari st.tabs() adalah elemen navigasi yang memungkinkan pengguna berpindah antar grup konten terkait dengan sangat mudah. Pada baris ketiga kita membuat variabel tab1 yang berisi data grafik dan tab2 hanya berisi data yang diambil dari baris keempat.

Output



Gambar 2. 41 Tampilan Tab

d. Perintah st.expander()



File expander.py

```
expander.py > ...
1 import streamlit as st
2 #cara pertama menggunakan with
3 with st.expander("Klik untuk melihat Expander"):
4     st.write('ini adalah contoh expander')
5     st.bar_chart({"data": [1, 5, 2, 6, 2, 1]})
6 #cara kedua menggunakan objek
7 klik = st.expander("Klik untuk melihat Expander")
8 klik.write('ini adalah contoh expander')
9 klik.bar_chart({"data": [1, 5, 2, 6, 2, 1]})
```

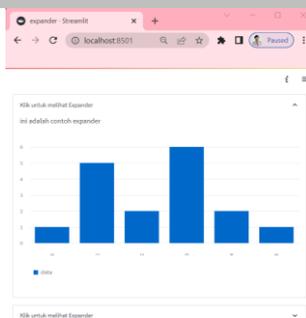


Penjelasan

Fungsi `st.expander()` adalah Menyisipkan wadah ke dalam aplikasi Anda yang dapat digunakan untuk menampung beberapa elemen dan dapat diperluas atau dicituk oleh pengguna. Saat dicituk, yang terlihat hanyalah label yang disediakan. Untuk menambahkan elemen ke wadah yang dikembalikan, Anda dapat menggunakan notasi "with" (lebih disukai) atau cukup panggil metode langsung pada objek yang dikembalikan.



Output



Gambar 2. 42 Tampilan Expander

e. Perintah `st.container()`



File `container.py`

```
container.py > ...
1 import streamlit as st
2 #cara pertama menggunakan with
3 with st.container():
4     st.write("ini didalam container")
5 st.write("ini diluar container")
6 #cara kedua menggunakan objek
7 klik = st.container()
8 klik.write('ini adalah contoh container')
9 klik.bar_chart({"data": [1, 5, 2, 6, 2, 1]})
```

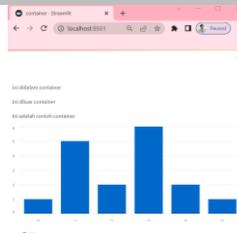


Penjelasan

Fungsi dari `st.container()` adalah Menyisipkan wadah tak terlihat ke dalam aplikasi Anda yang dapat digunakan untuk menampung banyak elemen. Hal ini memungkinkan Anda untuk, misalnya, menyisipkan beberapa elemen ke dalam aplikasi secara tidak berurutan. Untuk menambahkan elemen ke wadah yang dikembalikan, Anda dapat menggunakan notasi "with" atau cukup panggil metode langsung pada objek yang dikembalikan. Lihat contoh di bawah ini



Output



Gambar 2. 43 Tampilan Container

f. Perintah `st.empty()`

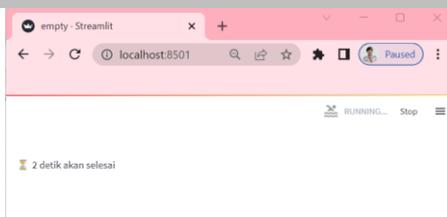
File `empty.py`

```
empty.py > klik
1 import streamlit as st
2 import time
3 #cara pertama menggunakan with
4 with st.empty():
5     for seconds in range(10):
6         st.write(f"⌚ {seconds} detik akan selesai ")
7         time.sleep(1)
8         st.write("✅ 10 detik sudah selesai!")
9 #cara kedua menggunakan objek
10 klik = st.empty()
11 for detik in range(10):
12     klik.write(f"⌚ {detik} detik kedua akan selesai ")
13     time.sleep(1)
14     klik.write("✅ 10 detik kedua sudah selesai!")
```

Penjelasan

Fungsi `st.empty()` adalah Menyisipkan wadah ke dalam aplikasi Anda yang dapat digunakan untuk menampung satu elemen. Ini memungkinkan Anda untuk, misalnya, menghapus elemen kapan saja, atau mengganti beberapa elemen sekaligus. Untuk menyisipkan/mengganti/menghapus elemen pada wadah yang dikembalikan, Anda dapat menggunakan notasi "with" atau cukup memanggil metode langsung pada objek yang dikembalikan.

Output



Gambar 2. 44 Tampilan Empty

2.8. Menampilkan Control Flow

Control flow pada Streamlit digunakan untuk mengontrol jalannya aplikasi web yang dibuat. Control flow dapat digunakan untuk mengatur urutan tampilan komponen, memeriksa input pengguna, mengatur kondisi tertentu, dan melakukan tindakan berdasarkan kondisi yang diberikan. Perintah yang akan digunakan meliputi `st.stop()`, `st.form()`, `st.form_submit_button()`, `st.experimental_rerun()`.

a. Perintah `st.stop()`



File `stop.py`

```
import streamlit as st
st.title("Aplikasi Streamlit")
# Menghentikan eksekusi aplikasi dan menampilkan
pesan
if st.button("Stop Aplikasi"):
    st.warning("Aplikasi dihentikan!")
    st.stop()
# Kode di bawah ini tidak akan dieksekusi jika
st.stop() dijalankan
st.write("Baris ini tidak akan dieksekusi jika
aplikasi dihentikan.")
```

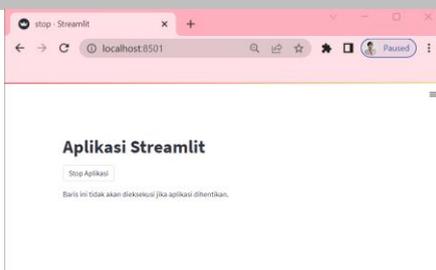


Penjelasan

Fungsi `st.stop()` adalah untuk menghentikan eksekusi aplikasi Streamlit. Berikut adalah tampilan di web browser terkait dengan fungsi `stop()`



Output



Gambar 2. 45 Tampilan perintah stop

Pada contoh di atas, terdapat judul aplikasi "Aplikasi Streamlit" dan sebuah tombol "Stop Aplikasi". Jika tombol tersebut ditekan, maka pesan "Aplikasi dihentikan!" akan ditampilkan, dan eksekusi aplikasi akan dihentikan menggunakan `st.stop()`. Baris kode di bawah `st.stop()` tidak akan dieksekusi jika aplikasi dihentikan.

b. Perintah `st.form()`



File `form.py`

```
import streamlit as st
# Membuat form dengan st.form()
with st.form("Formulir Pengguna"):
    # Menambahkan input teks
    nama = st.text_input("Nama")
    umur = st.number_input("Umur", min_value=0,
max_value=100, step=1)
    alamat = st.text_area("Alamat")

    # Menambahkan tombol submit
    submit_button =
st.form_submit_button(label="Submit")

# Menampilkan hasil form setelah tombol submit
ditekan
if submit_button:
    st.write("Nama:", nama)
    st.write("Umur:", umur)
    st.write("Alamat:", alamat)
```

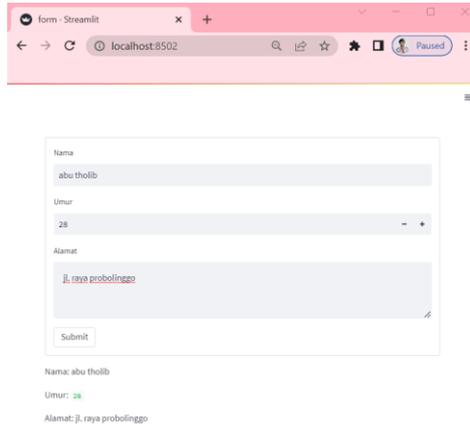


Penjelasan

Fungsi `st.form()` pada Streamlit digunakan untuk membuat form atau formulir di aplikasi Streamlit. Berikut adalah tampilan di web browser penggunaan `st.form()`:



Output



Gambar 2. 46 Tampilan Form

Pada contoh di atas, `st.form()` digunakan untuk membuat form dengan judul "Formulir Pengguna". Di dalam form, terdapat beberapa elemen input seperti `st.text_input()`, `st.number_input()`, dan `st.text_area()`. Kemudian, kita menambahkan tombol submit menggunakan `st.form_submit_button()` dengan label "Submit". Setelah tombol submit ditekan, kita menggunakan kondisi `if submit_button:` untuk menampilkan hasil form di bawahnya menggunakan `st.write()`. Dalam contoh ini, ketika tombol submit ditekan, aplikasi akan menampilkan nama, umur, dan alamat yang diinputkan oleh pengguna. Perhatikan bahwa `st.form()` digunakan untuk mengatur tampilan dan interaksi dengan elemen-elemen form, sementara pemrosesan data yang dikirim oleh form harus ditangani di dalam blok `if submit_button:` atau setelah form telah di-submit.

c. Perintah `st.form_submit_button()`



File `submit.py`

```
import streamlit as st
# Membuat form dengan st.form()
st.title("Aplikasi Dasar Streamlit Formulir")
with st.form("Formulir Data Mahasiswa"):
    # Menambahkan input teks
    col1, col2 = st.columns(2)
    with col1:
```

```

        nama = st.text_input("Nama")
        umur = st.number_input("Umur", min_value=0,
max_value=100, step=1)
        alamat = st.text_area("Alamat")

        # Menambahkan input selectbox
        jurusan = st.selectbox("Jurusan",
["Informatika", "Sistem Informasi", "Teknik
Elektro"])
        with col2:
            # Menambahkan input checkbox
            hobi = st.multiselect("Hobi", ["Olahraga",
"Membaca", "Menulis"])

            # Menambahkan input nilai
            uts = st.number_input("uts", min_value=0,
max_value=100, step=1)
            uas = st.number_input("uas", min_value=0,
max_value=100, step=1)
            tugas = st.number_input("tugas",
min_value=0, max_value=100, step=1)
            na = (uts+uas+tugas)/3
            # Menambahkan tombol submit
            submit_button =
st.form_submit_button(label="Submit")

# Menampilkan hasil form setelah tombol submit
ditekan
if submit_button:
    st.write("Data Mahasiswa:")
    st.write("Nama:", nama)
    st.write("Umur:", umur)
    st.write("Alamat:", alamat)
    st.write("Jurusan:", jurusan)
    st.write("Hobi:", hobi)
    st.write("Nilai Akhir", na)

```

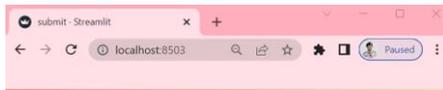


Penjelasan

Fungsi `st.form_submit_button()` adalah untuk membuat tombol submit di dalam form. Tombol submit tersebut diberi label "Submit" menggunakan argumen `label="Submit"`



Output



Aplikasi Dasar Streamlit Formulir

Nama	Hobi
Albi	Menyala
Umur	Sex
20	Male
Alamat	Age
Jl. Raya Pendidikan	30
Jurusan	Height
Informatika	175
<input type="button" value="Submit"/>	

Data Hasilkan:

Nama: Albi
Umur: 20
Alamat: Jl. Raya Pendidikan
Jurusan: Informatika
Hobi: Menyala

Note: Albi is 20 years old

Gambar 2. 47 Tampilan form submit

pada contoh di atas, kita mencoba untuk menggunakan `st.form_submit_button()` bersama dengan beberapa jenis inputan lainnya di dalam form. Inputan yang digunakan mencakup `st.text_input()` untuk input teks, `st.number_input()` untuk input angka, `st.text_area()` untuk input teks area, `st.selectbox()` untuk input seleksi satu opsi, dan `st.multiselect()` untuk input seleksi banyak opsi. Setelah tombol submit ditekan, aplikasi akan menampilkan hasil form, yaitu nama, umur, alamat, jurusan, dan hobi yang diinputkan oleh pengguna. Perhatikan bahwa Anda dapat menyesuaikan jenis inputan dan opsi-opsi yang ada sesuai dengan kebutuhan Anda dalam form tersebut.

d. Perintah `st.experimental_rerun()`.



File sidebar.py

```
import streamlit as st
st.title("Aplikasi Dasar Streamlit ")
# Mengambil input dari pengguna
nama = st.text_input("Nama")
umur = st.number_input("Umur", min_value=0,
```

```

max_value=100, step=1)

# Menambahkan tombol rerun
rerun_button = st.button("Rerun")

# Mengeksekusi ulang aplikasi jika tombol rerun
ditekan
if rerun_button:
    st.experimental_rerun()

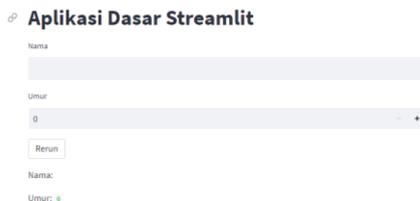
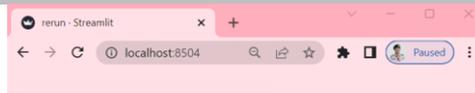
# Menampilkan hasil input
st.write("Nama:", nama)
st.write("Umur:", umur)

```

 **Penjelasan**

Fungsi `st.experimental_rerun()` pada Streamlit digunakan untuk memicu ulang (rerun) eksekusi aplikasi Streamlit

 **Output**



Gambar 2. 48 Tampilan experimental rerun

Pada contoh di atas, kita mencoba untuk menggunakan `st.experimental_rerun()` untuk memicu ulang (rerun) eksekusi aplikasi Streamlit saat tombol rerun ditekan.

Kode yang ingin dijalankan ulang ditempatkan di dalam blok `if rerun_button:`. Dalam contoh ini, jika tombol rerun ditekan, maka aplikasi akan dieksekusi ulang. Setelah tombol rerun ditekan atau saat aplikasi pertama kali dijalankan, aplikasi akan menampilkan input nama dan umur menggunakan `st.text_input()` dan `st.number_input()`. Hasil input tersebut kemudian ditampilkan di bawahnya menggunakan `st.write()`.

Perhatikan bahwa `st.experimental_rerun()` digunakan untuk memicu ulang eksekusi aplikasi Streamlit secara manual. Anda dapat mengatur logika dan kondisi tertentu untuk menentukan kapan harus memicu rerun aplikasi berdasarkan kebutuhan Anda.

BAB 3

Implementasi Algoritma SVM menggunakan Framework Streamlit

Deskripsi Materi:

Pembahasan bab 3 mempelajari materi tentang konsep dasar algoritma support vector machine, dataset, pembersihan data, membuat model, evaluasi model dan pembuatan aplikasi website dengan algoritma support vector machine berbasis web framework streamlit.

3.1. Support Vector Machine

Support Vector Machine pertama kali diperkenalkan dan dikembangkan oleh Vladimir N. Vapnik untuk model linear pada tahun 1963 [13] dan kemudian dikembangkan ke data pelatihan non-linear pada tahun 1995 [14]. Ini adalah algoritma klasifikasi dalam *machine learning* masuk kategori *supervised learning* adalah jenis pembelajaran mesin di mana kita menggunakan data yang memiliki output berlabel. Sedangkan *unsupervised learning* adalah jenis pembelajaran mesin di mana kita memiliki data tanpa label keluaran. Di sini, tujuannya adalah untuk mengidentifikasi pola yang berarti dalam data [15]. yang telah menjadi sangat populer saat ini karena hasil yang sangat efisien. SVM diimplementasikan dengan cara yang sedikit berbeda dari algoritma *machine learning* lainnya. Itu mampu melakukan klasifikasi, regresi dan deteksi outlier. Support Vector Machine adalah pengklasifikasi diskriminatif yang secara formal dirancang oleh hyperplane terpisah. Ini adalah

representasi contoh sebagai titik-titik dalam ruang yang dipetakan sehingga titik-titik dari kategori yang berbeda dipisahkan oleh celah seluas mungkin. Selain itu, SVM juga dapat melakukan klasifikasi non-linear. Mari kita lihat bagaimana Support Vector Machine bekerja. Tujuan utama dari mesin vektor pendukung adalah untuk memisahkan data yang diberikan dengan cara terbaik. Saat pemisahan dilakukan, jarak antara titik terdekat dikenal sebagai margin. Pendekatannya adalah memilih hyperplane dengan margin maksimum yang mungkin antara vektor pendukung dalam kumpulan data yang diberikan. SVM memberikan solusi prediksi berdasarkan bentuk fungsional persamaan[16] berikut :

$$y(x) = \sum_{n=0}^n w_n K(x, x_n) + w_0$$

di mana W_n adalah 'bobot' model dan $K(X, X_n)$ adalah fungsi kernel.

3.2. Dataset

Pada langkah awal kalian membuat file dengan nama **svm.ipynb**. Pada latihan ini kita menggunakan kumpulan data kanker yang sudah tersedia secara public di kaggle.com pada link <https://www.kaggle.com/code/dhainjeamita/breast-cancer-dataset-classification> , dipersilahkan anda download terlebih dahulu selanjutnya file tersebut ditaruh dalam satu folder dengan file svm.ipynb tujuannya agar mudah untuk load file. Tujuan dari latihan ini kami membuat klasifikasi untuk prediksi apakah kanker itu ganas atau jinak. Kita akan memuat file dataset yang sudah ada di library dengan cara berikut :



Coding memuat dataset

```
import pandas as pd
```

```
df = pd.read_csv('breast_cancer.csv')
df.head()
```



Output

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280

5 rows × 9 columns

Gambar 3. 1 Menampilkan Data Kanker

3.3. Pembersihan Data

Untuk memeriksa apakah kumpulan data kita berisi nilai yang hilang dan nanti rata-ratanya berapa, kita dapat menggunakan fungsi **isna().sum()**, Kemudian kita dapat menghitung berapa banyak nilai yang hilang untuk setiap kolom. Pada contoh kasus dataset yang kita pelajari tidak ada nilai yang missing, sehingga semua variabel isinya 0.



Memeriksa Data yang kosong

```
df.isna().sum()
```



Output

```

id                0
diagnosis         0
radius_mean      0
texture_mean     0
perimeter_mean   0
area_mean        0
smoothness_mean  0
compactness_mean 0
concavity_mean   0
concave_points_mean 0
symmetry_mean    0
fractal_dimension_mean 0
radius_se        0
texture_se       0
perimeter_se     0
area_se          0
smoothness_se   0
compactness_se  0
concavity_se     0
concave_points_se 0
symmetry_se      0
fractal_dimension_se 0
radius_worst     0
texture_worst    0
perimeter_worst  0
...
concave_points_worst 0
symmetry_worst     0
fractal_dimension_worst 0
Unnamed: 32        569
dtype: int64

```

Gambar 3. 2 Menampilkan Data yang kosong



Data Reduction

```

#menghapus kolom yang tidak dibutuhkan
df.drop(df.columns[[-1, 0]], axis=1,
inplace=True)
df.info()

```



Output

```

-----
#      Column                Non-Null Count  Dtype
-----
0      diagnosis              569 non-null   object
1      radius_mean              569 non-null   float64
2      texture_mean              569 non-null   float64
3      perimeter_mean            569 non-null   float64
4      area_mean                  569 non-null   float64
5      smoothness_mean           569 non-null   float64
6      compactness_mean          569 non-null   float64
7      concavity_mean             569 non-null   float64
8      concave_points_mean       569 non-null   float64
9      symmetry_mean              569 non-null   float64
10     fractal_dimension_mean     569 non-null   float64
11     radius_se                  569 non-null   float64
12     texture_se                  569 non-null   float64
13     perimeter_se                569 non-null   float64
14     area_se                     569 non-null   float64
15     smoothness_se              569 non-null   float64
16     compactness_se              569 non-null   float64
17     concavity_se                569 non-null   float64
18     concave_points_se          569 non-null   float64
19     symmetry_se                 569 non-null   float64
...
29     symmetry_worst              569 non-null   float64
30     fractal_dimension_worst    569 non-null   float64
dtypes: float64(30), object(1)
memory usage: 137.9+ KB

```

Gambar 3. 3 Menghapus kolom

Sebelum membangun model machine learning, kita perlu menentukan variabel input dan output. Variabel input ini disebut variabel independen. Variabel output adalah variabel target variabel ini disebut variabel dependen. Dalam memilah dataset kita menggunakan 70 % untuk data training yang nantinya kita gunakan untuk membuat model dan 30 % untuk data testing . yang nantinya akan kita coba untuk testing model yang sudah dibuat sebelumnya dengan perintah dibawah ini :



Membagi data X dan y

```

# X adalah variabel input, y adalah variabel
target
semua = list(df.columns[1:31])
X = df.loc[:,semua]
y = df[['diagnosis']]
print(X.shape)
print(y.shape)

```



Output

```
(569, 30)
(569, 1)
```

Langkah selanjutnya adalah memilah dataset



Memilah data training dan testing

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y,
test_size=0.3, random_state=0)
print(X_train.shape, y_train.shape)
print(X_test.shape, y_test.shape)
```



Output

```
(398, 30) (398, 1)
(171, 30) (171, 1)
```

3.4. Membuat Model

Setelah memilah dataset, langkah selanjutnya adalah membuat model dengan cara import algoritma svm dari library scikit-learn seperti pada contoh dibawah ini.



Membuat Model

```
from sklearn import svm
#create a classifier
model = svm.SVC(kernel="linear")
#train the model
modelfit = model.fit(X_train,y_train)
accuracy = modelfit.score(X_test, y_test)
#predict the response
pred = model.predict(X_test)
print(accuracy)
```



Output

```
0.9590643274853801
```

3.5. Evaluasi Model

Kita dapat menggunakan confusion matrix untuk menghitung berbagai performance metrics untuk mengukur kinerja model yang telah dibuat. Pada bagian ini mari kita pahami beberapa performance metrics populer yang umum dan sering digunakan seperti : accuracy, precision, dan recall.

1. Accuracy

Accuracy menggambarkan seberapa akurat model dapat mengklasifikasikan dengan benar. Maka, accuracy merupakan rasio prediksi benar (positif dan negatif) dengan keseluruhan data. Dengan kata lain, accuracy merupakan tingkat kedekatan nilai prediksi dengan nilai aktual (sebenarnya)

2. Precision

Precision menggambarkan tingkat keakuratan antara data yang diminta dengan hasil prediksi yang diberikan oleh model. Maka, *precision* merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif.

3. Recall

Recall menggambarkan keberhasilan model dalam menemukan kembali sebuah informasi. Maka, *recall* merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif.



Mengevaluasi model

```
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
```



Output

	precision	recall	f1-score	support
Kanker ganas	0.92	0.97	0.95	63
Kanker jinak	0.98	0.95	0.97	108
accuracy			0.96	171
macro avg	0.95	0.96	0.96	171
weighted avg	0.96	0.96	0.96	171

Gambar 3. 4 Menampilkan Confusion Metric

3.6. Menyimpan Model

Untuk kemudahan dalam implementasi model yang sudah dibuat sebelumnya diatas, maka kita akan menggunakan **import library pickle**, Pickle adalah sebuah modul di Python yang digunakan untuk meng-serialize atau mengubah objek Python menjadi bentuk yang dapat disimpan dan dipertukarkan dengan aplikasi Python lain atau bahkan dengan aplikasi yang ditulis dalam bahasa pemrograman yang berbeda, sebenarnya ada beberapa cara untuk menyimpan model yang sudah kita buat dengan **import joblib** atau **import h5py**. Namun pada buku ini kita menggunakan **import pickle**

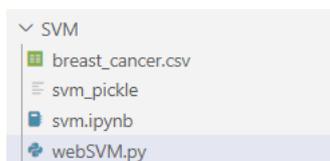
Fungsi utama dari modul pickle di Python adalah untuk menyimpan objek Python ke dalam file atau mengubah objek Python menjadi sebuah string, dan kemudian membaca kembali objek tersebut dari file atau string tersebut menjadi bentuk objek Python yang asli. Pickle berguna ketika kita ingin menyimpan data yang dihasilkan oleh program Python ke dalam file, atau ketika kita ingin mengirim data antara aplikasi Python yang berbeda. bagaimana cara menyimpan data menggunakan pickle di Python. Untuk melakukannya, kita harus mengimpor modul pickle terlebih dahulu dengan

cara import pickle. objek file yang Anda dapatkan dengan membuka file yang diinginkan dalam mode write-binary (wb). Singkatan "**wb**" berarti "**write binary**", yang menunjukkan bahwa file yang dibuka akan ditulis dalam mode biner. Mode ini diperlukan karena saat menggunakan pickle untuk menyimpan (serialize) objek ke dalam file, objek tersebut akan diubah menjadi format biner sebelum ditulis. Kemudian gunakan **pickle.dump()** fungsinya adalah untuk menyimpan data objek ke file, pada contoh dibawah ini kita **pickle.dump(modelfit, r)** cls merupakan variabel yang sudah kita buat sebelumnya diatas, dimana isi dari variabel tersebut adalah Model yang sudah kita buat sebelumnya dari hasil training data.

Import model ke File pickle

```
import pickle
with open('svm_pickle','wb') as r:
    pickle.dump(modelfit,r)
```

Dibawah ini contoh file hasil dari import model yang sudah dibuat sebelumnya menjadi nama file svm_pickle yang nantinya akan kita gunakan untuk testing kepada data baru.



Gambar 3. 5 Tampilan susunan folder

3.7. Aplikasi GUI SVM di Streamlit

untuk membaca file svm_pickle yang sudah kita buat sebelumnya, menggunakan contoh coding dibawah ini dengan fungsi **rb (Read Binary)**. Singkatan "**rb**" berarti "read binary", yang menunjukkan bahwa file yang dibuka akan dibaca dalam mode biner. Mode ini diperlukan karena

objek yang di-serialize dengan pickle disimpan dalam format biner. Selanjutnya kita panggil dengan menggunakan fungsi **pickle.load()** dan kita simpan di variabel **model**.



Membaca file svm_pickle

```
with open('svm_pickle','rb') as r:  
    model = pickle.load(r)
```

selanjutnya adalah membuat aplikasi website dengan framework streamlit dengan cara membuat file dengan nama **webSVM.py** di dalam folder SVM sehingga satu folder dengan **svm_pickle**.



Membuat File webSVM.py

```
#import library pickle dan streamlit  
import pickle  
import streamlit as st  
  
# Membaca file svm_pickle  
with open('svm_pickle','rb') as r:  
    model = pickle.load(r)  
  
# Menambahkan pilihan  
menu = ['Halaman Utama', 'Tentang Kami', 'Kontak']  
choice = st.sidebar.selectbox('Navigasi', menu)  
  
# Menampilkan konten sesuai dengan pilihan  
if choice == 'Halaman Utama':  
    st.title('Halaman Utama')  
  
    def welcome():  
        return 'welcome you all'  
  
    def prediction1(x1, x2, x3, x4, x5, x6, x7, x8,  
x9, x10, x11, x12, x13, x14, x15, x16, x17, x18,  
x19, x20, x21, x22, x23, x24, x25, x26, x27, x28,  
x29, x30):  
        prediction1 = model.predict([[x1, x2, x3,  
x4, x5, x6, x7, x8, x9, x10, x11, x12, x13, x14,  
x15, x16, x17, x18, x19, x20, x21, x22, x23, x24,
```

```

x25, x26, x27, x28, x29, x30]])
    print(prediction1)
    return prediction1

def main():
    st.title('Aplikasi Web Machine Learning')
    st.title("Prediksi Kanker Payudara")
    st.title("Menggunakan Algoritma Support
Vector Machine")
    col1, col2, col3 = st.columns(3)
    with col1:
        x1 = st.text_input ("mean radius ", "")
        x2 = st.text_input ("mean texture ", "")
        x3 = st.text_input ("mean perimeter ", "")
        x4 = st.text_input ("mean area ", "")
        x5 = st.text_input ("mean smoothness ", "")
        x6 = st.text_input ("mean compactness ", "")
        x7 = st.text_input ("mean concavity ", "")
        x8 = st.text_input ("mean concave points ",
"")
        x9 = st.text_input ("mean symmetry ", "")
        x10 = st.text_input ("mean fractal dimension
", "")
    with col2:
        x11 = st.text_input ("radius error ", "")
        x12 = st.text_input ("texture error ", "")
        x13 = st.text_input ("perimeter error ", "")
        x14 = st.text_input ("area error ", "")
        x15 = st.text_input ("smoothness error ", "")
        x16 = st.text_input ("compactness error ", "")
        x17 = st.text_input ("concavity error ", "")
        x18 = st.text_input ("concave points error ",
"")
        x19 = st.text_input ("symmetry error ", "")
        x20 = st.text_input ("fractal dimension error
", "")
    with col3:
        x21 = st.text_input ("worst radius", "")
        x22 = st.text_input ("worst texture", "")
        x23 = st.text_input ("worst perimeter", "")
        x24 = st.text_input ("worst area", "")
        x25 = st.text_input ("worst smoothness", "")
        x26 = st.text_input ("worst compactness", "")
        x27 = st.text_input ("worst concavity", "")
        x28 = st.text_input ("worst concave points",
"")
        x29 = st.text_input ("worst symmetry", "")
        x30 = st.text_input ("worst fractal
dimension ", "")

    hasil = " "

```

```

        if st.button ("Predict"):
            hasil = prediction1 (x1, x2, x3, x4, x5,
x6, x7, x8, x9, x10, x11, x12, x13, x14, x15, x16,
x17, x18, x19, x20, x21, x22, x23, x24, x25, x26,
x27, x28, x29, x30)

            st.success ('Hasil Prediksinya adalah
{}'.format(hasil))
            st.write("Hasil Prediksinya adalah",
format(hasil))

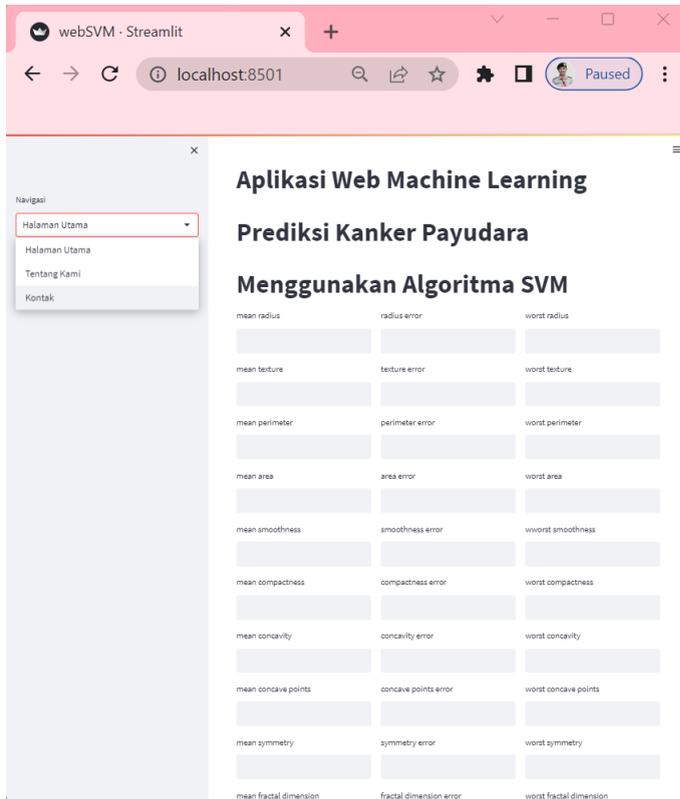
    if __name__== '__main__':
        main()
elif choice == 'Tentang Kami':
    st.title('Tentang Kami')
    st.write('Buku ini dibuat untuk belajar aplikasi
Streamlit')
    st.write('Kami sangat senang anda belajar
machine learning dengan streamlit.')

elif choice == 'Kontak':
    st.title('Kontak')
    st.write('Kalian Bisa Menghubungi kami di
abu@unuja.ac.id')

```



Output



Gambar 3. 6 Tampilan Aplikasi GUI SVM Streamlit

Berdasarkan hasil diagnosis aplikasi web machine learning yang telah kita buat diatas, maka dalam memprediksi kanker payudara menggunakan algoritma SVM terdapat dua keputusan termasuk dalam “**kanker jinak**” atau “**kanker ganas**” berdasarkan data yang telah di inputkan oleh pengguna.

BAB 4

Implementasi Algoritma Decision Tree menggunakan Framework Streamlit

Deskripsi Materi:

Pembahasan bab 4 mempelajari materi tentang konsep dasar algoritma decision tree, dataset, pembersihan data, membuat model, evaluasi model dan pembuatan aplikasi website dengan algoritma decision tree berbasis web framework streamlit.

4.1. Decision Tree(C4.5)

Algoritma Decision Tree C4.5 adalah sebuah metode untuk membangun model Decision Tree yang menggunakan atribut dengan teknik pemilihan atribut berdasarkan rasio keuntungan informasi (ratio information gain). Algoritma ini adalah termasuk algoritma machine learning didasarkan pada pemanfaatan data untuk mengembangkan model statistik, yang kemudian digunakan oleh sistem untuk membuat prediksi masa depan berdasarkan data input sebelumnya atau untuk mempelajari pola dalam data[17], Metode ini dikembangkan oleh Ross Quinlan pada tahun 1993 sebagai pengembangan dari algoritma Decision Tree sebelumnya yang dikenal dengan nama ID3.

Berikut adalah langkah-langkah umum dalam algoritma Decision Tree C4.5:

1. Menghitung atribut dengan nilai Information Gain yang paling tinggi: Atribut dengan Information Gain tertinggi akan dipilih sebagai atribut pemisah pada tahap selanjutnya.
2. Membuat node atau daun baru berdasarkan atribut pemisah: Atribut dengan Information Gain tertinggi pada langkah sebelumnya akan digunakan sebagai atribut pemisah (splitting attribute) pada node atau daun baru dalam pohon keputusan. Setiap nilai unik dari atribut pemisah akan membentuk cabang-cabang yang keluar dari node tersebut.
3. Memisahkan dataset berdasarkan nilai-nilai atribut pemisah: Dataset awal akan dibagi menjadi subset-subset yang lebih kecil berdasarkan nilai-nilai yang mungkin dari atribut pemisah. Setiap subset akan berisi contoh-contoh yang sesuai dengan nilai tertentu dari atribut pemisah.
4. Mengulangi langkah 1 hingga 3 untuk setiap subset: Proses ini akan berlanjut hingga semua contoh dalam subset memiliki kelas target yang sama atau tidak ada atribut lagi yang tersisa untuk dijadikan pemisah.
5. Menentukan label kelas pada node atau daun: Setelah rekursi selesai dan semua cabang telah terbentuk, kita perlu menentukan label kelas untuk setiap node atau daun dalam pohon keputusan.
6. Membangun pohon keputusan: Langkah-langkah di atas akan membentuk struktur pohon keputusan yang lengkap, di mana setiap node internal mewakili atribut pemisah dan setiap daun mewakili label kelas.

Untuk perhitungan nilai entropy [18] dengan persamaan berikut:

$$\text{Entropy}(S) = \sum_{i=1}^n -p_i * \log_2 p_i$$

Keterangan :

S : himpunan kasus

n : jumlah partisi S

pi : proporsi dari Si terhadap S

untuk perhitungan information gain sebagai berikut :

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

Keterangan:

S : himpunan kasus

A : atribut

n : jumlah partisi atribut A

|Si| : jumlah kasus pada partisi ke-i

|S| : jumlah kasus dalam S

4.2. Dataset

Untuk tujuan demonstrasi, saya telah mengambil kumpulan data diabetes dari kaggle.com pada tautan [berikut](#) . Tujuan dari dataset ini adalah untuk memprediksi apakah seorang pasien diabetes atau non-diabetes. Secara pribadi, saya hanya menjelajah dengan kumpulan data berukuran kecil dan menggunakan streamlit



Coding memuat dataset

```
import pandas as pd
df = pd.read_csv('diabetes.csv')
df.head()
```



Output

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

Gambar 4.1 Menampilkan Data Diabetes

4.3. Pembersihan Data

Preprocessing data adalah proses mempersiapkan data mentah dan membuatnya sesuai untuk model machine learning. Ini adalah langkah pertama dan krusial dalam membuat model machine learning. Saat membuat proyek machine learning, tidak selalu kita menemukan data yang bersih. Dan saat melakukan operasi apa pun dengan data, adalah wajib untuk membersihkannya dan sekiranya siap untuk pembuatan model sehingga tidak mengurangi akurasi dari model.


Melihat dataset yang kosong

```
df.isnull().sum()
```


Output

```

Pregnancies      0
Glucose           0
BloodPressure     0
SkinThickness     0
Insulin           0
BMI               0
DiabetesPedigreeFunction  0
Age               0
Outcome           0
dtype: int64
```

Gambar 4. 2 Menampilkan dataset yang kosong


Membagi data X dan Y

```

#Memisahkan antara feature (x) dan label (y)
X = df.drop('Outcome', axis=1)
y = df[['Outcome']]
print(X.shape)
print(y.shape)
```



Output

```
(768, 8)
(768, 1)
```

Langkah selanjutnya adalah Membagi dataset menjadi data training dan testing. salah satu bagian penting dari prapemrosesan data, karena dengan melakukan hal tersebut, kita dapat meningkatkan kinerja model kita dan karenanya memberikan prediktabilitas yang lebih baik



Membagi data training dan testing

```
#split data menjadi data training dan data testing
from sklearn.model_selection import
train_test_split
X_train,X_test,y_train,y_test =
train_test_split(X,y,test_size =
0.2,random_state=0)
print(X_train.shape, y_train.shape)
print(X_test.shape, y_test.shape)
```



Output

```
(614, 8) (614, 1)
(154, 8) (154, 1)
```

4.4. Membuat Model

Model machine learning mirip dengan sebuah perangkat lunak komputer yang dirancang untuk mengenali pola atau perilaku berdasarkan pengalaman atau data sebelumnya. Algoritma pembelajaran menemukan pola dalam data pelatihan, dan menghasilkan model machine learning yang menangkap pola-pola ini dan membuat prediksi pada data baru



Membuat Model

```
# import algoritma Decisiontree from sklearn.tree
import DecisionTreeClassifier
from sklearn.tree import DecisionTreeClassifier
model = DecisionTreeClassifier(criterion='entropy')
modelfit = model.fit(X_train, y_train)
accuracy = modelfit.score(X_test, y_test)
pred = model.predict(X_test)
print(accuracy)
```



Output

0.7402597402597403

4.5. Evaluasi Model

Langkah selanjutnya adalah Mengevaluasi kinerja model Machine learning, salah satu langkah penting dalam membangun model Machine learning yang efektif adalah mengevaluasi kinerja atau kualitas model, evaluasi model ini membantu kita memahami seberapa baik kinerja model kita untuk data yang diberikan



Coding memuat dataset

```
from sklearn import metrics
#accuracy
print("accuracy:",
metrics.accuracy_score(y_test,y_pred=pred))
#precision score
print("precision:",
metrics.precision_score(y_test,y_pred=pred))
#recall score
print("recall" ,
metrics.recall_score(y_test,y_pred=pred))
print(metrics.classification_report(y_test,
y_pred=pred))
```



Output

```

accuracy: 0.7922077922077922
precision: 0.6470588235294118
recall 0.7021276595744681

```

	precision	recall	f1-score	support
0	0.86	0.83	0.85	107
1	0.65	0.70	0.67	47
accuracy			0.79	154
macro avg	0.76	0.77	0.76	154
weighted avg	0.80	0.79	0.79	154

Gambar 4. 3 Hasil Confusion Metric

4.6. Menyimpan Model

Untuk menyimpan model yang sudah kita buat sebelumnya diatas, dimana isi dari variabel `modelfit` tersebut adalah Model yang sudah kita buat sebelumnya dari hasil training data.

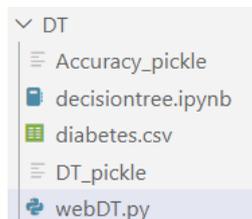
 **Import model ke File pickle**

```

import pickle
with open('DT_pickle','wb') as r:
    pickle.dump(modelfit,r)

```

Fungsi `pickle.dump()` dalam modul `pickle` untuk menyimpan objek Python ke dalam file dengan format serialisasi. Dibawah ini contoh file hasil dari import model yang sudah dibuat sebelumnya menjadi nama file `DT_pickle` yang nantinya akan kita gunakan untuk implementasi model di streamlit.



Gambar 4. 4 Tampilan susunan folder

4.7. Aplikasi GUI Decision Tree di Streamlit

Langkah selanjutnya adalah membuat aplikasi website dengan framework streamlit dengan cara membuat file dengan nama **webDT.py** di dalam folder SVM sehingga satu folder dengan svm_pickle.



Membuat File webDT.py

```
import pickle
import streamlit as st

with open('DT_pickle','rb') as r:
    hasil = pickle.load(r)
# Menambahkan pilihan
menu = ['Halaman Utama', 'Tentang Kami', 'Kontak']
choice = st.sidebar.selectbox('Navigasi', menu)

# Menampilkan konten sesuai dengan pilihan
if choice == 'Halaman Utama':
    st.title('Halaman Utama')

    def welcome():
        return 'welcome you all'

    def
prediction1(Pregnancies,Glucose,BloodPressure,SkinThic
kness,Insulin,BMI,DiabetesPedigreeFunction,Age):
    prediction1 =
hasil.predict([[Pregnancies,Glucose,BloodPressure,Skin
Thickness,Insulin,BMI,DiabetesPedigreeFunction,Age]])

    print(prediction1)
    return prediction1

def main():
    st.title('Aplikasi Web Machine Learning')
    st.title("Algoritma Decision Tree ")
    col1, col2 = st.columns(2)
    with col1:
        Pregnancies = st.text_input ("Pregnancies
", "")
        Glucose = st.text_input ("Glucose ", "")
        BloodPressure = st.text_input
("BloodPressure ", "")
        SkinThickness= st.text_input
("SkinThickness ", "")
```

```

with col2:
    Insulin = st.text_input ("Insulin ", "")
    BMI = st.text_input ("BMI ", "")
    DiabetesPedigreeFunction = st.text_input
("DiabetesPedigreeFunction ", "")
    Age = st.text_input ("Age ", "")
    result = " "

    if st.button ("Predict"):
        result = prediction1
(Pregnancies,Glucose,BloodPressure,SkinThickness,Insul
in,BMI,DiabetesPedigreeFunction,Age)

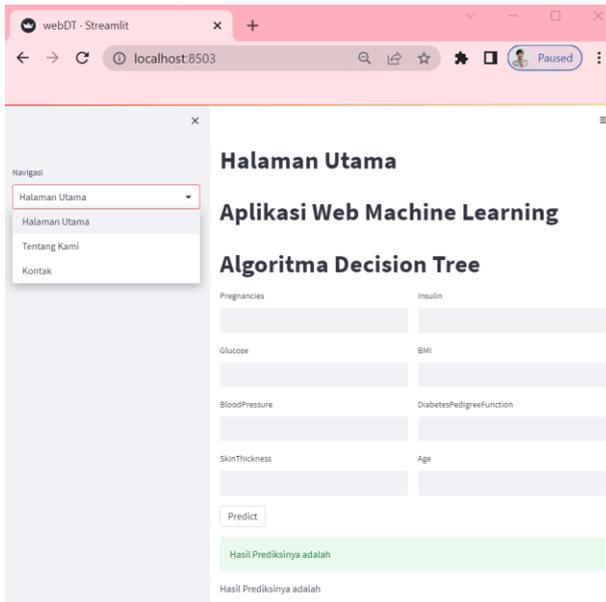
        st.success ('Hasil Prediksinya adalah
{}'.format(result))
        st.write("Hasil Prediksinya adalah",
format(result))
        #st.write(f'Algoritma = {nama_algoritma}')
        if __name__== '__main__':
            main()
elif choice == 'Tentang Kami':
    st.title('Tentang Kami')
    st.write('Buku ini dibuat untuk belajar aplikasi
Streamlit')
    st.write('Kami sangat senang anda belajar machine
learning dengan streamlit.')

elif choice == 'Kontak':
    st.title('Kontak')
    st.write('Kalian Bisa Menghubungi kami di
abu@unuja.ac.id')

```



Output



Gambar 4.5 Tampilan Aplikasi Decision Tree

Berdasarkan hasil 8 variabel input maka Jika **hasil prediksi 0** tidak mengidap diabetes, jika **hasil prediksi 1** maka mengidap diabetes. Machine learning biasanya didasarkan pada data yang dikumpulkan pada saat tertentu. Namun, dunia terus berubah, dan data baru yang relevan terus dihasilkan. Oleh karena itu, pengembangan model machine learning yang mampu belajar secara berkelanjutan dengan mengakomodasi perubahan dan data baru menjadi penting, selamat mencoba dan mengimplementasikan dengan kasus yang berbeda sehingga dapat memberikan kontribusi keilmuan yang lebih berguna.

BAB 5

Implementasi Algoritma Random Forest menggunakan framework streamlit

Pembahasan bab 5 mempelajari materi tentang konsep dasar algoritma random forest, dataset, pembersihan data, membuat model, evaluasi model dan pembuatan aplikasi website dengan algoritma random forest berbasis web framework streamlit.

5.1. Random Forest

Random Forest adalah salah satu algoritma dalam Machine Learning yang digunakan untuk tugas-tugas seperti klasifikasi dan regresi, Random forest, khususnya telah terbukti memiliki tingkat akurasi yang tinggi dalam memprediksi data dari berbagai atribut[19], diperlukan sedikit upaya untuk memahami dengan tepat apa yang diprediksi. Untungnya library dari Scikit-Learn membuatnya cukup mudah untuk menjalankan algoritma ini dan menginterpretasikan hasilnya. Random Forest mudah digunakan dan dipahami oleh pemula karena algoritma ini tidak memerlukan penyetelan parameter yang rumit. Memiliki Kemampuan Seleksi Fitur: Random Forest dapat digunakan untuk memilih fitur yang paling penting dalam dataset, sehingga dapat membantu untuk mengurangi dimensi data dan meningkatkan akurasi model, selanjutnya langsung saja kita praktikkan.

Berikut adalah langkah-langkah umum dalam algoritma Random Forest:

1. Memilih jumlah pohon ($n_{\text{estimators}}$) yang akan dibangun dalam Random Forest.
2. Untuk setiap pohon dalam Random Forest, lakukan langkah-langkah berikut:
 - a. Mengambil sampel acak dengan penggantian (bootstrap) dari dataset pelatihan. Sampel ini digunakan sebagai dataset untuk melatih pohon yang bersangkutan.
 - b. Memilih sejumlah acak atribut (features) yang akan digunakan dalam membangun pohon. Biasanya, jumlah atribut yang dipilih adalah akar kuadrat dari jumlah total atribut.
 - c. Membangun pohon keputusan menggunakan sampel dataset dan atribut yang dipilih.
3. Setelah semua pohon dibangun, hasil prediksi dari masing-masing pohon digabungkan (voting) untuk menghasilkan prediksi akhir dalam kasus klasifikasi, atau dilakukan pemangkasan rata-rata (averaging) dalam kasus regresi.
4. Evaluasi performa model dengan menggunakan metrik evaluasi yang sesuai, seperti akurasi, presisi, recall, atau mean squared error (MSE).
5. Gunakan model Random Forest yang telah dilatih untuk melakukan prediksi pada data baru.

5.2. Dataset

Pada langkah awal kalian membuat file dengan nama **randomforest.ipynb**, selanjutnya anda akan membutuhkan dataset bunga iris bisa di download pada link berikut <https://www.kaggle.com/datasets/uciml/iris> dimana bunga

Iris dataset adalah salah satu dataset yang paling populer digunakan dalam Machine Learning. Dataset ini berisi data mengenai tiga spesies bunga iris yang berbeda, yaitu Iris setosa, Iris versicolor, dan Iris virginica. Setiap spesies memiliki 50 sampel dan memiliki empat fitur yang diukur, yaitu panjang dan lebar sepal (bagian hijau yang menutupi bunga) dan panjang dan lebar petal (bagian warna bunga). Dataset ini sangat sering digunakan untuk tugas-tugas Machine Learning, terutama dalam tugas klasifikasi. Karena dataset ini relatif kecil dan mudah digunakan, sehingga dapat digunakan untuk belajar dasar-dasar Machine Learning dan pemrosesan data. Pada tahap awal kita membutuhkan library pandas untuk manipulasi dataset dengan mengetikkan perintah `import pandas as pd`, pada tahap selanjutnya kita membaca file `iris.csv` yang sudah kita download dengan cara `df = pd.read_csv('Iris.csv')` lalu menampilkan dataset dengan perintah `df.head()`. Agar lebih mudah langsung saja kita praktikkan seperti dibawah ini.

 **Coding memuat dataset**

```
import pandas as pd
#TAHAPAN 1 IMPORT DATA
df = pd.read_csv('Iris.csv')
df.head()
```

 **Output**

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

Gambar 5. 1 Menampilkan dataset iris

5.3. Pembersihan Data

Pembersihan data adalah langkah penting dalam machine learning karena membantu membersihkan, mengubah, dan mengatur data mentah ke dalam format yang siap untuk digunakan sehingga membantu meningkatkan kualitas, akurasi, dan efisiensi algoritma machine learning.

**Melihat Nilai yang kosong**

```
#melihat missing value
df.isnull().sum()
```

**Output**

Id	0
SepalLengthCm	0
SepalWidthCm	0
PetalLengthCm	0
PetalWidthCm	0
Species	0
dtype:	int64

Gambar 5. 2 Tampilan Data yang bersih

Pada bagian data reduction kita menghapus variabel id dan species untuk dimasukkan ke variabel X, selanjutnya untuk variabel species kita masukkan ke variabel y dengan fungsi sebagai class/variabel target

**Data Reduction**

```
#memisahkan variabel dan class
X = df.drop(['Id', 'Species'], axis=1)
y = df[['Species']]
X.head()
```

**Output**

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2

Gambar 5. 3 Data Reduction

5.4. Membuat Model

Memilah dataset menjadi data training dan data testing menjadi dua bagian dengan tujuan. Data training sebanyak 70% dari data digunakan untuk melatih model, sedangkan data testing sebanyak 30 dari data digunakan untuk menguji performa model.



Memilah Data Training dan Testing

```
# SPLIT DATA TRAINING = 70 %, DATA TESTING = 30 %
from sklearn.model_selection import
train_test_split
X_train,X_test,y_train,y_test = train_test_split(X,
y, test_size = 0.3, random_state=0)
print(X_train.shape, y_train.shape)
print(X_test.shape, y_test.shape)
```



Output

```
(105, 4) (105, 1)
(45, 4) (45, 1)
```



Membuat Model Random Forest

```
# TAHAPAN 3 ADALAH TRAINING MODEL
from sklearn.ensemble import RandomForestClassifier
#create a classifier
model = RandomForestClassifier(n_estimators=40)
#train the model
modelfit = model.fit(X_train, y_train)
accuracy = modelfit.score(X_test, y_test)
#predict the response
y_pred = model.predict(X_test)
```

```
print(accuracy)
```



Output :

```
0.9777777777777777
```

5.5. Evaluasi Model

Kita dapat menggunakan confusion matrix untuk menghitung berbagai performance metrics untuk mengukur kinerja model yang telah dibuat. performance metrics populer yang umum dan sering digunakan seperti : accuracy, precision, f1-score dan support.



Evaluasi Model Machine Learning

```
#TAHAPAN 4 ADALAH EVALUASI MODEL DENGAN CONFUSION MATRIX  
from sklearn.metrics import classification_report  
print(classification_report(y_test, y_pred))
```



Output :

	precision	recall	f1-score	support
Iris-setosa	1.00	1.00	1.00	16
Iris-versicolor	1.00	0.94	0.97	18
Iris-virginica	0.92	1.00	0.96	11
accuracy			0.98	45
macro avg	0.97	0.98	0.98	45
weighted avg	0.98	0.98	0.98	45

Gambar 5. 4 Hasil Evaluasi Model

Pada gambar 5.4 dapat kita lihat bahwasanya akurasi yang dihasilkan dari algoritma random forest untuk data diatas adalah 98 %

5.6. Menyimpan Model

Untuk menyimpan model yang sudah kita buat sebelumnya diatas, dimana isi dari variabel **modelfit** tersebut adalah Model yang sudah kita buat sebelumnya dari hasil training data maka kita menggunakan koding dibawah ini.

```
Import model ke File pickle

import pickle
with open('RF_pickle','wb') as r:
    pickle.dump(modelfit,r)
```

Fungsi **pickle.dump()** dalam modul pickle untuk menyimpan objek Python ke dalam file dengan format serialisasi. Dibawah ini contoh file hasil dari import model yang sudah dibuat sebelumnya menjadi nama file DT_pickle yang nantinya akan kita gunakan untuk implementasi model di streamlit.



Gambar 5. 5 Tampilan Susunan Folder Random Forest

5.7. Aplikasi GUI Random Forest di streamlit

selanjutnya untuk membaca file **RF_pickle** yang sudah kita buat sebelumnya, menggunakan contoh coding dibawah ini dengan fungsi **rb (Read Binary)**. Selanjutnya kita panggil dengan menggunakan fungsi **pickle.load()** dan kita simpan di variabel **classifier1**.

```
Membaca file km_pickle

with open('RF_pickle','rb') as r:
    classifier1 = pickle.load(r)
```

untuk membuat aplikasi website dengan framework streamlit dengan cara membuat file dengan nama

webSVM.py upayakan bisa satu folder dengan file **svm_pickle** yang isinya adalah model yang sudah dibuat sebelumnya diatas.



Membuat File webRF.py

```
import pickle
import streamlit as st

with open('RF_pickle','rb') as r:
    classifier1 = pickle.load(r)

# Menambahkan pilihan
menu = ['Halaman Utama', 'Tentang Kami', 'Kontak']
choice = st.sidebar.selectbox('Navigasi', menu)

# Menampilkan konten sesuai dengan pilihan
if choice == 'Halaman Utama':
    st.title('Halaman Utama')
    def welcome():
        return 'welcome you all'

    def prediction1(sepal_length1, sepal_width1,
petal_length1, petal_width1):
        prediction1 =
classifier1.predict([[sepal_length1, sepal_width1,
petal_length1, petal_width1]])
        print(prediction1)
        return prediction1

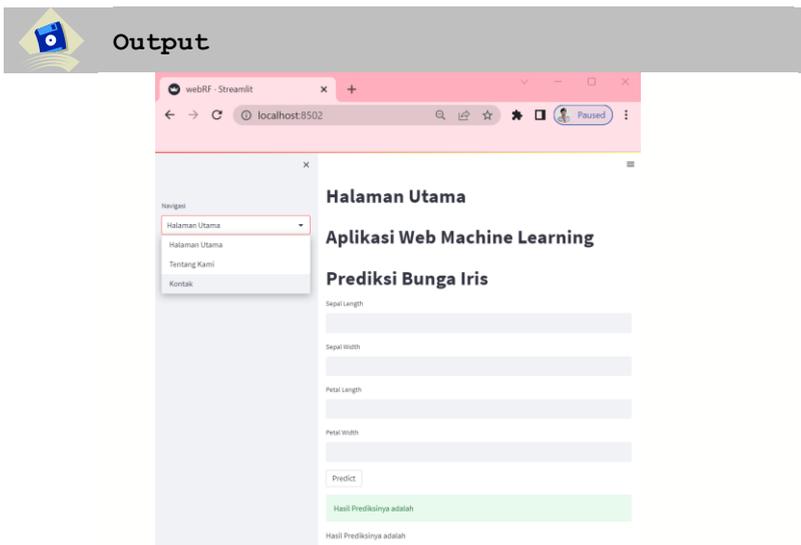
    def main():
        st.title('Aplikasi Web Machine Learning')
        st.title("Prediksi Bunga Iris")
        sepal_length1 = st.text_input ("Sepal Length
", "")
        sepal_width1 = st.text_input ("Sepal Width
", "")
        petal_length1 = st.text_input ("Petal Length
", "")
        petal_width1 = st.text_input ("Petal Width
", "")
        result = " "

        if st.button ("Predict"):
            result = prediction1 (sepal_length1,
sepal_width1, petal_length1, petal_width1)
```

```

if (result[0]==1):
    result = 'Bunga iris Setosa'
elif (result[0]==2):
    result = 'Iris-versicolor'
elif (result[0]==3):
    result = 'Iris-virginica'
#st.success ('Hasil Prediksinya adalah
{}'.format(result))
st.success ('Hasil Prediksinya adalah
{}'.format(result))
st.write("Hasil Prediksinya adalah",
format(result))
#st.write(f'Algoritma = {nama_algoritma}')
if __name__ == '__main__':
    main()
elif choice == 'Tentang Kami':
    st.title('Tentang Kami')
    st.write('Buku ini dibuat untuk belajar aplikasi
Streamlit')
    st.write('Kami sangat senang anda belajar
machine learning dengan streamlit.')
elif choice == 'Kontak':
    st.title('Kontak')
    st.write('Kalian Bisa Menghubungi kami di
abu@unuja.ac.id')

```



Gambar 5. 6 Tampilan Aplikasi Random Forest

Berdasarkan penerapan model pada aplikasi web machine learning yang telah kita buat diatas, maka dalam memprediksi bunga iris menggunakan algoritma rando forest terdapat tiga keputusan termasuk dalam bunga '**Iris-setosa**' jika hasil outputnya adalah 1, selanjutnya akan masuk bunga '**Iris-versicolor**' jika hasil outputnya adalah 2, dan masuk bunga '**Iris-virginica**' jika outputnya adalah 3. Meskipun model machine learning yang telah dibuat dapat menghasilkan prediksi yang akurat, kita harapkan pengembangan model dan dataset yang digunakan nantinya bisa di sesuaikan dengan studi kasus yang berkembang saat ini.

BAB 6

Implementasi Algoritma K-Means Clustering menggunakan framework streamlit

Deskripsi Materi:

Pembahasan bab 6 mempelajari materi tentang konsep dasar algoritma random forest, dataset, pembersihan data, membuat model, evaluasi model dan pembuatan aplikasi website dengan algoritma random forest berbasis web framework streamlit.

6.1. K-Means Clustering

K-Means Clustering adalah salah satu algoritma dalam analisis data yang digunakan untuk mengelompokkan data ke dalam kelompok-kelompok (clusters) berdasarkan kemiripan atribut yang dimiliki oleh setiap data. Beberapa algoritma clustering telah di implementasikan seperti K-Means, Nearest Neighbors dan DBSCAN [20] dalam sistem cerdas. Algoritma ini termasuk dalam kategori unsupervised learning karena tidak memerlukan label atau kelas yang sudah diketahui sebelumnya. Pada algoritma K-Means Clustering, jumlah kelompok yang akan dibentuk (K) harus didefinisikan sebelumnya. Kemudian, algoritma akan secara iteratif menemukan centroid untuk setiap kelompok dan mengelompokkan setiap data ke kelompok yang memiliki centroid terdekat. Proses ini dilakukan berulang-ulang hingga tidak ada lagi perubahan dalam pengelompokkan data.

Cara kerja K-Means Clustering didasarkan pada prinsip bahwa data yang mirip cenderung berada dalam kelompok

yang sama. Oleh karena itu, tujuan utama dari algoritma ini adalah meminimalkan jarak antara setiap data dan centroid kelompoknya, sehingga membentuk kelompok-kelompok yang homogen.

Langkah-langkah clustering dengan algoritma K-Means Clustering[21] sebagai berikut :

1. Menentukan nilai K sebagai klaster yang akan dibentuk
2. Menentukan pusat titik dari setiap klaster
3. Menghitung jarak data input pada masing-masing centroid menggunakan rumus (Euclidean Distance) sampai ditemukan jarak yang terdekat dari setiap data dengan centroid

Berikut adalah persamaan Euclidian Distance :

$$D(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$$

Keterangan :

D = Jarak

X = Data

Y = Centroid

4. Klasifikasi data dengan dasar kedekatan dengan centroid
5. Menghitung kembali pusat cluster dengan anggota cluster. Pusat dari cluster adalah nilai rata-rata dari semua data objek dalam cluster
6. Hitung lagi setiap objek memakai pusat cluster yang baru, jika pusat cluster tidak berubah maka proses klustering selesai.

6.2. Dataset

Pada langkah awal untuk mempelajari latihan ini anda membuat file dengan nama **kmeans.ipynb**. selanjutnya setelah selesai kita akan menggunakan data perumahan di

California yang diambil dari data publik kaggle tinggal klik tautan dibawah ini untuk download file <https://www.kaggle.com/datasets/camnugent/california-housing-prices?resource=download> pada latihan ini menggunakan data lokasi serta nilai rata-rata rumah. Dan bagaimana harga rumah berfluktuasi di seluruh California. File csv yang dipanggil **'housing.csv'** pada latihan ini harus satu folder dengan file **kmeans.ipynb**



Coding memuat dataset

```
import pandas as pd

df = pd.read_csv('housing.csv', usecols =
['longitude', 'latitude', 'median_house_value'])
df.head()
```



Output

	longitude	latitude	median_house_value
0	-122.23	37.88	452600.0
1	-122.22	37.86	358500.0
2	-122.24	37.85	352100.0
3	-122.25	37.85	341300.0
4	-122.25	37.85	342200.0

Gambar 6. 1 Menampilkan Data Rumah

Data tersebut mencakup 3 variabel yang telah kami pilih menggunakan `usecols` parameter:

- **longitude** : Nilai yang menunjukkan seberapa jauh kearah barat sebuah rumah. Nilai yang lebih tinggi mewakili rumah yang lebih jauh ke Barat.
- **Latitude** : Nilai yang menunjukkan seberapa jauh kearah utara sebuah rumah. Nilai yang lebih tinggi mewakili rumah yang lebih jauh ke utara.

- **median_house_value** : Harga rata-rata rumah dalam satu blok diukur dalam USD.

6.3. Pembersihan Data

Pada tahapan ini kita melakukan normalisasi data, pertama kita menyiapkan pemisahan data training dan testing menggunakan `train_test_split` from `sklearn`.



Pemisahan data

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test =
train_test_split(df[['latitude', 'longitude']],
df[['median_house_value']], test_size=0.3,
random_state=0)
print(X_train.shape, y_train.shape)
print(X_test.shape, y_test.shape)
```



Output

```
(14448, 2) (14448, 1)
(6192, 2) (6192, 1)
```



Normalisasi Data

```
from sklearn import preprocessing

X_train_norm = preprocessing.normalize(X_train)
X_test_norm = preprocessing.normalize(X_test)
print(X_train_norm.shape)
print(X_test_norm.shape)
```



Output

```
(14448, 2)
(6192, 2)
```

6.4. Membuat Model

Untuk iterasi pertama, secara acak memilih sejumlah cluster (disebut sebagai k) dari 3. Membangun dan

memasang model `sklearn` sangat sederhana. Kami membuat instance dari K-Means, menentukan jumlah cluster menggunakan `n_clusters` atribut, dan akan menyetelnya `random_state` ke 0 sehingga mendapatkan hasil yang sama setiap kali kita menjalankan kode. kemudian dapat menyesuaikan model dengan data pelatihan yang dinormalisasi menggunakan `fit()`



Training Model

```
from sklearn.cluster import KMeans

kmeans = KMeans(n_clusters = 3, random_state = 0,
                n_init='auto')
model_fit = kmeans.fit(X_train_norm)
```

setelah data sesuai, kita dapat mengakses label dari atribut `labels_`. Di bawah ini, kami memvisualisasikan data yang baru saja kami sesuaikan dengan perintah `sns.scatterplot()`.

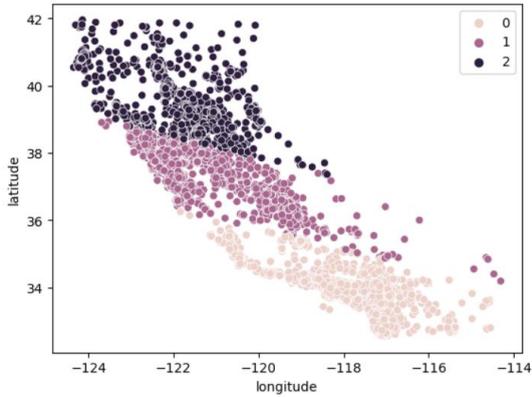


Menampilkan scatterplot

```
sns.scatterplot(data = X_train, x = 'longitude', y = 'latitude', hue = kmeans.labels_)
```



Output



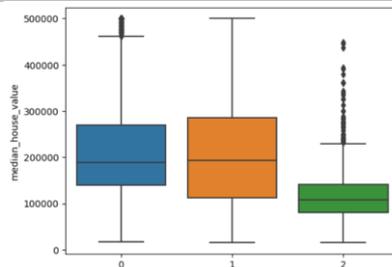
Gambar 6. 2 Grafik Scater 3 cluster

Kami melihat bahwa data sekarang jelas dibagi menjadi 3 kelompok berbeda (California Utara, California Tengah, dan California Selatan). Kita juga bisa melihat distribusi harga rata-rata rumah di 3 kelompok ini dengan menggunakan `sns.boxplot()`. dapat kita lihat pada output bahwasanya (cluster 0 dan 1) memiliki harga rumah yang lebih tinggi, sedangkan cluster 2 memiliki harga rumah yang lebih rendah.

Menampilkan Boxplot

```
sns.boxplot(x = kmeans.labels_, y =
y_train['median_house_value'] )
```

Output



Gambar 6. 3 Tampilan Grafik Box 3 Cluster

6.5. Evaluasi Model

Untuk mengevaluasi kinerja algoritma clustering menggunakan `silhouette_score` yang merupakan bagian di `sklearn.metrics` dimana skor yang lebih rendah menunjukkan kecocokan yang lebih baik



Coding evaluasi model

```
from sklearn.metrics import silhouette_score

silhouette_score(X_train_norm, kmeans.labels_,
metric='euclidean')
```



Output

```
0.7491654371657605
```

Kelemahan k-means clustering adalah kita tidak mengetahui berapa banyak cluster yang kita butuhkan dengan hanya menjalankan model. Kita perlu menguji rentang nilai dan membuat keputusan tentang nilai k terbaik. Selanjutnya membuat loop di bawah ini untuk menguji dan menyimpan hasil model yang berbeda sehingga dapat membuat keputusan tentang jumlah cluster terbaik.



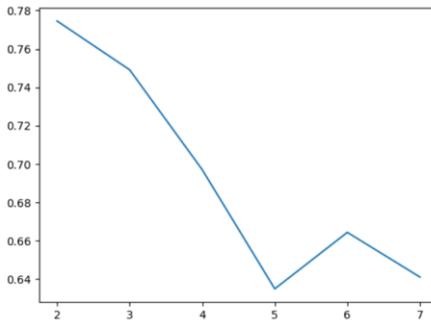
Memilih jumlah cluster terbaik

```
K = range(2, 8)
fits = []
score = []
for k in K:
    # melatih model untuk nilai k saat ini pada data
    # pelatihan
    model = KMeans(n_clusters = k, random_state = 0,
n_init='auto').fit(X_train_norm)

    # menambahkan model agar sesuai
    fits.append(model)
    # menambahkan silhouette score
    score.append(silhouette_score(X_train_norm,
model.labels_, metric='euclidean'))
sns.lineplot(x = K, y = score)
```



Output



Gambar 6. 4 Grafik uji coba range 2 sampai 8

Berdasarkan hasil experiment dengan $k = \text{range}(2, 8)$ maka peningkatan kinerja mulai mendatar atau memburuk. melihat kluster $k = 5$ mungkin yang terbaik dari range 2 sampai 8 yang bisa kami lakukan tanpa overfitting.

6.6. Menyimpan Model

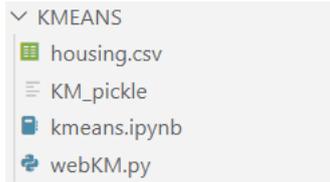
Sebelum membuat web dengan streamlit kita membutuhkan library pickle untuk menyimpan model yang sudah kita buat diatas menjadi sebuah file dengan perintah **pickle.dump()** fungsinya adalah untuk menyimpan data objek ke file, pada contoh dibawah ini ketikkan **import** pickle untuk memanggil library pickle di python, selanjutnya kita ketikkan **with open('KM_pickle','wb') as r:** dimana fungsinya adalah 'wb' write binary membaca file sehingga menjadi KM_pickle sebagai r. dimana isi dari variabel tersebut adalah Model dari Algoritma K-Means Clustering yang sudah kita buat sebelumnya dari hasil training data.



Import model ke File pickle

```
import pickle
with open('KM_pickle','wb') as r:
    pickle.dump(modelfit,r)
```

Dibawah ini contoh file hasil dari import model yang sudah dibuat sebelumnya menjadi nama file `KM_pickle` yang nantinya akan kita gunakan untuk testing kepada data baru.



Gambar 6. 5 susunan folder k-means

6.7. Implementasi Algoritma K-Means di streamlit

pada tahapan untuk membaca file `KM_pickle` yang sudah kita buat sebelumnya, menggunakan contoh coding dibawah ini dengan fungsi **rb (Read Binary)**. Selanjutnya kita panggil dengan menggunakan fungsi `pickle.load()` dan kita simpan di variabel `classifier1`.

Membaca file `km_pickle`

```
with open('KM_pickle','rb') as r:  
    classifier1 = pickle.load(r)
```

selanjutnya adalah membuat aplikasi website dengan framework streamlit dengan cara membuat file dengan nama `webKM.py` lalu ketikkan kode dibawah ini.

Membuat File `webKM.py`

```
import pickle  
import streamlit as st  
  
with open('KM_pickle','rb') as r:  
    classifier1 = pickle.load(r)  
  
# Menambahkan pilihan  
menu = ['Halaman Utama', 'Tentang Kami', 'Kontak']  
choice = st.sidebar.selectbox('Navigasi', menu)  
  
# Menampilkan konten sesuai dengan pilihan tab
```

```

if choice == 'Halaman Utama':
    st.title('Halaman Utama')
    def welcome():
        return 'welcome you all'
    def prediction1(longitude, latitude):
        prediction1 =
classifier1.predict([[longitude, latitude]])
        print(prediction1)
        return prediction1
    def main():
        st.title('Aplikasi Web Machine Learning')
        st.title("Klaster Rumah")
        longitude = st.text_input ("longitude ", "")
        latitude = st.text_input ("latitude ", "")

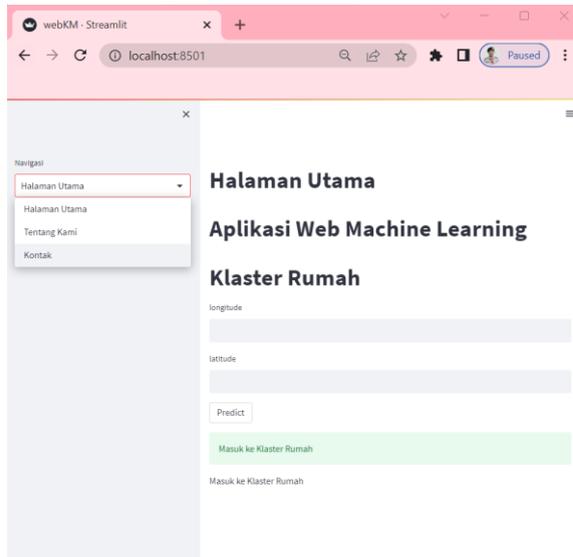
        result = " "
        if st.button ("Predict"):
            result = prediction1(longitude, latitude)
            if (result[0]==0):
                result = 'Cukup Tinggi'
            elif (result[0]==1):
                result = 'Cukup Tinggi'
            elif (result[0]==2):
                result = 'Lebih Rendah'
            st.success ('Masuk ke Klaster Rumah
format(result))
            st.write("Masuk ke Klaster Rumah",
format(result))
            if __name__ == '__main__':
                main()
elif choice == 'Tentang Kami':
    st.title('Tentang Kami')
    st.write('Buku ini dibuat untuk belajar
aplikasi Streamlit')
    st.write('Kami sangat senang anda belajar
machine learning dengan streamlit.')

elif choice == 'Kontak':
    st.title('Kontak')
    st.write('Bisa Menghubungi kami abu@unuja.ac.id')

```



Output



Gambar 6. 6 Halaman utama kluster rumah

Berdasarkan penerapan model pada aplikasi web *machine learning* yang telah kita buat diatas, maka dalam penerapan kluster rumah menggunakan algoritma K-Means Clustering terdapat tiga kluster. termasuk dalam kluster harga ‘**Cukup Tinggi**’ jika hasil outputnya adalah 0 atau 1, selanjutnya termasuk dalam kluster harga ‘**Lebih Rendah**’ jika hasil outputnya adalah 2.

Tentunya pengembangan kedepan sangat penting karena *Machine learning* akan semakin terintegrasi dengan teknologi lain seperti Internet of Things (IoT), augmented reality (AR), virtual reality (VR), dan robotika. Smartphone Apple Iphone telah tersedia bagi para pengembang untuk mengintegrasikan alur kerja *machine learning* ke dalam aplikasi [22], Kolaborasi antara domain ini akan membuka peluang baru dalam berbagai aplikasi seperti otomatisasi industri, kendaraan otonom, perawatan kesehatan personal, dan pengalaman pengguna yang ditingkatkan. Kita tinggal menunggu waktu saja informasi yang terus akan memberikan

perubahan yang signifikan pada semua segi kehidupan terkait pengembangan dari *machine learning*.

BAB 7

Implementasi Algoritma Naïve Bayes menggunakan streamlit

Deskripsi Materi:

Pembahasan bab 7 mempelajari materi tentang konsep dasar algoritma naïve bayes, dataset, pembersihan data, membuat model, evaluasi model dan pembuatan aplikasi website dengan algoritma naïve bayes berbasis web framework streamlit.

7.1 Naïve Bayes

Naive Bayes adalah termasuk teknik klasifikasi berdasarkan Teorema Bayes[23]. Ini adalah salah satu algoritma *supervised learning* yang paling sederhana. Naive Bayes adalah algoritma yang cepat, akurat, dan handal. mengasumsikan bahwa efek dari fitur tertentu dalam sebuah kelas tidak bergantung pada fitur lainnya. Misalnya, seorang pemohon pinjaman diinginkan atau tidak tergantung pada pendapatannya, pinjaman sebelumnya dan riwayat transaksi, usia, dan lokasi. Sekalipun fitur-fitur ini saling bergantung, fitur-fitur ini tetap dianggap independen. Asumsi ini menyederhanakan perhitungan, Teorema Naive Bayes terdiri dari tiga jenis, yaitu Gaussian, Multinomial, dan Benoulli. Kami akan menjelaskan teorema Gaussian Naive Bayes dengan penggunaan rumus :

$$P(h|D) = \frac{P(D|h) * P(h)}{P(D)}$$

1. $P(h|D)$: probabilitas kelas target (h) diberikan fitur (D)

2. $P(D)$: probabilitas prior untuk kelas target (y)
3. $P(D|h)$: probabilitas mengamati fitur-fitur (X) jika kelas target (y) benar.
4. $P(D)$: probabilitas sebelumnya dari fitur

7.2 Dataset

Pada langkah awal untuk mempelajari latihan ini anda membuat file dengan nama **bayes.ipynb**. selanjutnya setelah selesai kita tinggal klik tautan dibawah ini untuk download https://drive.google.com/file/d/1sv99NOMwF3T3r_CZqp4L0523cl3FVYFa/view?usp=share_link pada latihan ini menggunakan data pasien yang memiliki penyakit jantung. File csv yang dipanggil ‘heart.csv’ pada latihan ini harus satu folder dengan file **bayes.ipynb**

Coding memuat dataset

```
import pandas as pd
df = pd.read_csv("heart.csv")
df.head()
```

Output

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1

Gambar 7. 1 Menampilkan Data penyakit jantung

7.3 Pembersihan Data

Pada tahapan ini kita melakukan pembersihan data sebelum kita menggunakan dataset untuk membuat model *machine learning*, kita pastikan dataset sudah tidak ada nilai yang kosong, sehingga nantinya model yang dibuat dapat meningkat akurasi dan efisien model yang dibuat. Perintah `df.info()` adalah melihat nama colomn dan type data yang digunakan.



Melihat kolom dan type data

```
df.info()
```



Output

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   age         303 non-null    int64
1   sex         303 non-null    int64
2   cp          303 non-null    int64
3   trestbps    303 non-null    int64
4   chol        303 non-null    int64
5   fbs         303 non-null    int64
6   restecg     303 non-null    int64
7   thalach     303 non-null    int64
8   exang       303 non-null    int64
9   oldpeak     303 non-null    float64
10  slope       303 non-null    int64
11  ca          303 non-null    int64
12  thal        303 non-null    int64
13  target      303 non-null    int64
dtypes: float64(1), int64(13)
memory usage: 33.3 KB
```

Gambar 7. 2 Menampilkan Info kolom dan type datanya

Pada tahapan selanjutnya anda bisa menggunakan perintah `df.isnull().sum()` untuk melihat data yang kosong, apabila setiap kolom menampilkan angka 0 maka tidak ada nilai yang kosong.



Perintah cek nilai yang kosong

```
df.isnull().sum()
```



Output

```

age          0
sex          0
cp           0
trestbps    0
chol        0
fbs         0
restecg     0
thalach     0
exang       0
oldpeak     0
slope       0
ca          0
thal        0
target      0
dtype: int64

```

Gambar 7.3 memeriksa nilai kosong

Pada tahapan selanjutnya anda akan memecah data menjadi Variabel input (X) ini disebut variabel independen. Variabel output (Y) adalah variabel target variabel ini disebut variabel dependen.



Memisahkan Variabel X dan Y

```

x = df[['age', 'sex', 'cp', 'trestbps', 'chol',
        'fbs', 'restecg', 'thalach',
        'exang', 'oldpeak', 'slope', 'ca', 'thal']]
#Kolom target pasien memiliki penyakit jantung
(ditunjukkan dengan 1) atau tidak (ditunjukkan
dengan 0)
y = df['target']
#print(x)
print(y)

```



Output

```

0      1
1      1
2      1
3      1
4      1
..
298    0
299    0
300    0
301    0
302    0
Name: target, Length: 303, dtype: int64

```

Gambar 7. 4 Hasil output variabel target

Dalam memilah dataset kita menggunakan 80 % untuk data training yang nantinya kita gunakan untuk membuat model dan 20 % untuk data testing yang nantinya bisa digunakan untuk menguji model yang sudah kita buat sebelumnya. Perintah code yang digunakan seperti pada contoh dibawah ini. Kita tetap menggunakan library scikit-learn seleksi model dengan import train_test_split

Split Data Training dan Testing

```

# splitting the dataset into training and testing
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test =
train_test_split(x,y,test_size=0.2, random_state = 0)
print(x_train.shape)
print(x_test.shape)

```

Output

```

(242, 13)
(61, 13)

```

7.4 Membuat Model

Pada tahapan pembuatan model kita membutuhkan algoritma naïve bayes dengan cara import library dari scikit learn import GaussianNB selanjutnya kita terapkan algoritma

tersebut dengan menjadikannya sebuah variabel dengan nama model. Lalu kita menggunakan Fungsi model.fit dalam pembelajaran mesin (*machine learning*) untuk melatih (train) model pada data pelatihan (training data). Fungsi ini memungkinkan model untuk belajar dari data yang diberikan dengan mengoptimalkan parameter model menggunakan algoritma pelatihan yang ditentukan.



Membuat Model Naïve Bayes

```
from sklearn.naive_bayes import GaussianNB
model = GaussianNB()
model.fit(x_train,y_train)
y_pred = model.predict(x_test)
acc = model.score(x_train, y_train)
#model.score(x_test, y_test)
print(acc)
```



Output

```
0.8347107438016529
```

7.5 Evaluasi Model

Kita dapat menggunakan confusion matrix untuk menghitung berbagai performance metrics untuk mengukur kinerja model yang telah dibuat. performance metrics populer yang umum dan sering digunakan seperti : accuracy, precision, f1-score dan support.



Coding memuat dataset

```
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
```



Output

	precision	recall	f1-score	support
0	0.88	0.78	0.82	27
1	0.84	0.91	0.87	34
accuracy			0.85	61
macro avg	0.86	0.84	0.85	61
weighted avg	0.85	0.85	0.85	61

Gambar 7. 5 Hasil Akurasi dan Presisi

Pada gambar 5.4 dapat kita lihat bahwasanya akurasi yang dihasilkan dari algoritma naïve bayes untuk data diatas adalah 85 %

7.6 Menyimpan Model

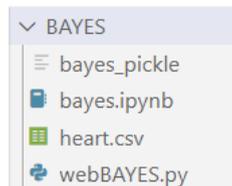
Untuk menyimpan model yang sudah kita buat sebelumnya diatas, dimana isi dari variabel **modelfit** tersebut adalah Model yang sudah kita buat sebelumnya dari hasil training data maka kita menggunakan koding dibawah ini.

```


Import model ke File pickle
import pickle
with open('bayes_pickle','wb') as r:
    pickle.dump(modelfit,r)

```

Fungsi **pickle.dump()** dalam modul pickle untuk menyimpan objek Python ke dalam file dengan format serialisasi. Dibawah ini contoh file hasil dari import model yang sudah dibuat sebelumnya menjadi nama file bayes_pickle yang nantinya akan kita gunakan untuk implementasi model di streamlit.



Gambar 7. 6 Susunan Folder Naïve Bayes

7.7 Implementasi Algoritma Naïve Bayes di streamlit

pada tahapan selanjutnya untuk membaca file **bayes_pickle** yang sudah kita buat sebelumnya, menggunakan contoh coding dibawah ini dengan fungsi **rb** (**Read Binary**). Selanjutnya kita panggil dengan menggunakan fungsi **pickle.load()** dan kita simpan di variabel **hasil**.



Membaca file km_pickle

```
with open('bayes_pickle','rb') as r:
    hasil = pickle.load(r)
```

Untuk membuat aplikasi website dengan framework streamlit dengan cara membuat file dengan nama **webKM.py** lalu ketikkan kode dibawah ini.



Membuat File webBAYES.py

```
#import library pickle dan streamlit
import pickle
import streamlit as st

# Membaca file svm_pickle
with open('bayes_pickle','rb') as r:
    hasil = pickle.load(r)

# Menambahkan pilihan
menu = ['Halaman Utama', 'Tentang Kami', 'Kontak']
choice = st.sidebar.selectbox('Navigasi', menu)

# Menampilkan konten sesuai dengan pilihan
if choice == 'Halaman Utama':
    st.title('Aplikasi Web Machine Learning')
    def welcome():
        return 'welcome you all'

    def
prediction1(age, sex, cp, trestbps, chol, fbs, restecg, thal
ach, exang, oldpeak, slope, ca, thal):
    prediction1 =
hasil.predict([[age, sex, cp, trestbps, chol, fbs, restecg,
thalach, exang, oldpeak, slope, ca, thal]])
    print(prediction1)
    return prediction1
```

```

def main():

    st.title("Algoritma Naive Bayes ")
    col1, col2 = st.columns(2)
    with col1:
        age = st.number_input ("age",
min_value=0, max_value=100, step=0)
        sex = st.number_input ("sex ",
min_value=0, max_value=100, step=0)
        cp = st.number_input ("cp", min_value=0,
max_value=100, step=0)
        trestbps= st.number_input ("trestbps",
min_value=0, max_value=200, step=0)
        chol= st.number_input ("chol",
min_value=0, max_value=300, step=0)
        fbs= st.number_input ("fbs", min_value=0,
max_value=100, step=0)
    with col2:
        restecg = st.number_input ("restecg ",
min_value=0, max_value=100, step=0)
        thalach = st.number_input ("thalach ",
min_value=0, max_value=200, step=0)
        exang = st.number_input ("exang ",
min_value=0, max_value=100, step=0)
        oldpeak = st.number_input ("oldpeak ",
step=0.1)
        slope = st.number_input ("slope ",
min_value=0, max_value=100, step=0)
        ca = st.number_input ("ca ", min_value=0,
max_value=100, step=0)
        thal = st.number_input ("thal ",
min_value=0, max_value=100, step=0)
        result = " "

    if st.button ("Predict"):
        result =
prediction1(age,sex,cp,trestbps,chol,fbs,restecg,thal
ach,exang,oldpeak,slope,ca,thal)
        if(result[0]==0):
            result = 'Tidak Punya Penyakit
Jantung'
        elif(result[0]==1):
            result = 'Punya penyakit jantung'
        st.success ('Hasil Prediksinya adalah
{}'.format(result))
        st.write("Hasil Prediksinya adalah",
format(result))

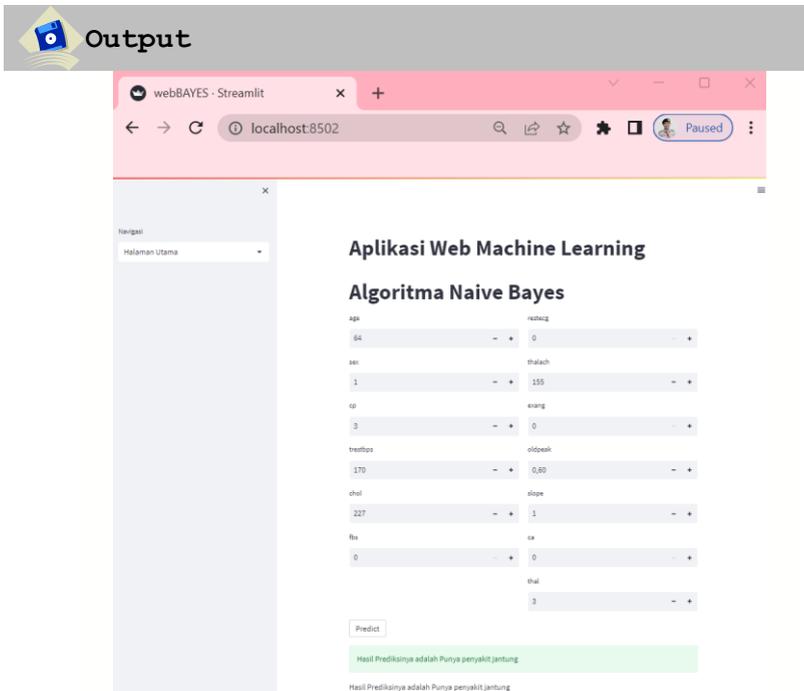
```

```

if __name__ == '__main__':
    main()
elif choice == 'Tentang Kami':
    st.title('Tentang Kami')
    st.write('Buku ini dibuat untuk belajar aplikasi Streamlit')
    st.write('Kami sangat senang anda belajar machine learning dengan streamlit.')

elif choice == 'Kontak':
    st.title('Kontak')
    st.write('Kalian Bisa Menghubungi kami di abu@unuja.ac.id')

```



Gambar 7. 7 Hasil ujicoba algoritma naïve bayes di streamlit

Kolom target menunjukkan apakah pasien memiliki penyakit jantung (ditunjukkan dengan 1) atau tidak (ditunjukkan dengan 0)

DAFTAR PUSTAKA

- [1] Iwona Otol, Marlena Grabowska, BUSINESS MODELS Innovation, Digital Transformation, and Analytics. CRC Press, 2020.
- [2] Minh-Quang Tran, Hoang-Phuong Doan, Viet Q. Vu, Lien T. Vu, “Machine learning and IoT-based approach for tool condition monitoring: A review and future prospects,” Measurement, 2022, doi: <https://doi.org/10.1016/j.measurement.2022.112351>.
- [3] Budi susanto, azminuddin I S aziz, Zohrahayaty, Machine learning & reasoning fuzzy logic algoritma, manual, matlab & rapid miner. Deepublish, 2020.
- [4] Punia, S., Singh, S. P., & Madaan, J. K, “From predictive to prescriptive analytics: A data-driven multi-item newsvendor model,” Decis. Support Syst., pp. 136, 113340., 2020.
- [5] Wu, J., Zhang, Z., & Zhou, S. X., “Credit rating prediction through supply chains: a machine learning approach,” Prod. Oper. Manag., pp. 1613-1629., 2022.
- [6] Foster Provost, Tom Fawcett, Data Science For Business. O’Reilly Media, Inc., 2013.

- [7] Kiran R, P Kumar, B Bhasker, “DNNRec: A novel deep learning based hybrid recommender system,” 2020, doi: <https://doi.org/10.1016/j.eswa.2019.113054>.
- [8] Mohammad Khorasani, Mohamed Abdou, Javier Hernández Fernández, *Web Application Development With Streamlit Streamlit: Develop and Deploy Secure and Scalable Web Applications to the Cloud Using a Pure Python Framework*. New York: Springer, 2022.
- [9] Tyler Richards, *Getting Started with Streamlit for Data Science*. Packt Publishing, 2021.
- [10] “<https://docs.streamlit.io/>.”
- [11] David Dietrich, Barry Heller, Beibei Yang, *Data Science and Big Data Analytics Discovering, Analyzing, Visualizing and Presenting Data*. John Wiley & Sons, Inc., 2015.
- [12] “<https://code.visualstudio.com/>.”
- [13] Vapnik, V., Lerner, A., “Pattern recognition using generalized portrait method.,” *Autom. Rem. Contr*, 1963.
- [14] Cortes, C., Vapnik, V., Support-vector networks. In: Saitta, L. (Ed.), *Machine Learning*. Boston: Kluwer Academic Publishers, 1995.
- [15] M. Wasim Nawaz, *Data Science Crash Course for Beginner : Fundamental and practices with python*. AI Publishing.

- [16] Tipping, M.E., “Sparse bayesian learning and the relevance vector machine.,” *Mach Learn Res*, pp. 211–244, 2001, doi: <https://doi.org/10.1162/15324430152748236>.
- [17] Fathorazi Nur Fajri, Abu Tholib, Wiwin Yuliana, “Application of Machine Learning Algorithm for Determining Elective Courses in Informatics Study Program,” *J. Tek. Inform. Dan Sist. Inf.*, vol. 8, p. 3, 2022.
- [18] Ade Izyuddin, Setyawan Wibisono, “APLIKASI PREDIKSI PENJUALAN AC MENGGUNAKAN DECISION TREE DENGAN ALGORITMA C4.5,” *MISI J. Manaj. Inform. Sist. Inf.*, vol. 3, no. 2, 2020.
- [19] Pietha Tiara L D, Favian Dewanta, Muhammad Arief N, “Implementasi Machine Learning Model Deployment Pada Website Pemantauan Kondisi Sungai Citarum Menggunakan PlatformAs-A-Service,” *E-Proceeding Eng.*, vol. 8, no. 6, p. 3064, 2022.
- [20] Wang, Zhang, Xue, Lu & Na, “E-commerce personalized recommendation analysis by deeply-learned clustering,” 2019, doi: <https://doi.org/10.1016/j.jvcir.2019.102735>.
- [21] Normah, Siti Nurajizah, Arinda Salbinda, “Penerapan Data Mining Metode K-Means Clustering Untuk Analisa

Penjualan Pada Toko Fashion Hijab Banten,” J. Tek. Komput. AMIK BSI, vol. 7, p. 2, 2021, doi: 10.31294/jtk.v4i2.

[22] Jesús Rogel-Salazar, *Advanced Data Science and Analytics with Python*. CRC Press, 2020.

[23] Gypsy Nandi, Rupam Kumar Sharma, *Data Science Fundamentals and Practical Approaches Understand Why Data Science is the Next*. BPB Publications, 2020.

GLOSARIUM

Algoritma, adalah prosedur langkah demi langkah atau seperangkat aturan untuk memecahkan masalah tertentu atau menyelesaikan tugas tertentu. Algoritma adalah urutan instruksi atau operasi yang terdefinisi dengan baik dan terbatas, yang jika diikuti dengan urutan tertentu, akan menghasilkan hasil yang diinginkan atau menyelesaikan masalah yang diberikan.

Machine learning, adalah subbidang kecerdasan buatan (AI) yang berfokus pada pengembangan algoritma dan model yang memungkinkan komputer untuk belajar dan membuat prediksi atau keputusan tanpa diprogram secara eksplisit. Hal ini melibatkan penggunaan teknik statistik dan model komputasi untuk memungkinkan mesin belajar dari data, mengidentifikasi pola, dan membuat prediksi atau keputusan berdasarkan data.

Framework streamlit, adalah kerangka kerja Python yang open source dengan menyederhanakan pengembangan aplikasi web interaktif untuk tugas-tugas data science dan machine learning. Framework ini memungkinkan para ilmuwan data dan pengembang untuk dengan cepat membangun dan menggunakan aplikasi berbasis web tanpa harus berurusan dengan kerumitan pengembangan web.

Deep learning, adalah bagian dari machine learning yang berfokus pada jaringan saraf tiruan dan kemampuannya untuk belajar dan membuat keputusan dengan memodelkan abstraksi tingkat tinggi dalam data. Terinspirasi dari struktur dan

fungsi otak manusia, khususnya jaringan neuron yang saling terhubung

Data scientist, adalah seorang profesional yang memiliki perpaduan keterampilan dalam berbagai disiplin ilmu, termasuk matematika, statistik, pemrograman dan database serta menerapkan keterampilan ini untuk menganalisis dan mengekstrak wawasan dari kumpulan data yang besar dan kompleks. Ilmuwan data menggunakan keahlian mereka untuk mengungkap pola, tren, dan korelasi dalam data, dan kemudian menerjemahkan temuan tersebut menjadi wawasan yang dapat ditindaklanjuti atau solusi untuk masalah bisnis atau penelitian

Flask, adalah framework web python yang ringan dan fleksibel yang mengikuti pendekatan minimalis. Framework ini menyediakan struktur yang sederhana dan modular, sehingga memungkinkan pengembang untuk memiliki kontrol lebih besar atas arsitektur dan komponen aplikasi. Flask dikenal dengan kesederhanaan dan kemudahan penggunaannya, menjadikannya pilihan populer untuk proyek-proyek kecil hingga menengah atau ketika dibutuhkan penyesuaian tingkat tinggi.

Django, adalah framework web python berfitur lengkap yang mengikuti filosofi "baterai disertakan". Framework ini menyediakan seperangkat alat dan fitur yang komprehensif, sehingga cocok untuk aplikasi web berskala besar dan kompleks. Django memberlakukan struktur dan konvensi tertentu, yang dapat membantu pengembang memulai dengan cepat dan mempertahankan basis kode yang konsisten

Numpy, adalah pustaka Python yang mendasar untuk komputasi ilmiah. Numpy menyediakan dukungan untuk array dan matriks, serta berbagai fungsi matematika untuk mengoperasikan array ini secara efisien.

Pandas, adalah pustaka Python yang menyediakan alat manipulasi dan analisis data. Pandas memperkenalkan dua struktur data utama, Series (1 dimensi) dan DataFrame (2 dimensi), yang dirancang untuk menangani dan memproses data terstruktur secara efisien. Pandas banyak digunakan dalam preprocessing data, pembersihan data, eksplorasi, dan tugas-tugas yang berhubungan dengan data

Matplotlib, adalah pustaka untuk membuat visualisasi statis, animasi, dan interaktif dalam Python. Ini menyediakan berbagai fungsi dan alat plotting untuk membuat berbagai jenis plot, bagan, dan grafik. Matplotlib sangat mudah dikustomisasi, memungkinkan pengguna untuk mengontrol setiap aspek tampilan plot

Scikit-learn, adalah library machine learning yang populer di Python, menyediakan berbagai macam algoritma dan alat untuk penggalian data, analisis data, dan pemodelan prediktif. Library ini dibangun di atas NumPy, SciPy, dan Matplotlib dan menyediakan API yang terpadu dan mudah digunakan untuk berbagai tugas di machine learning maupun deep learning.

Python, adalah bahasa pemrograman tingkat tinggi yang ditafsirkan yang dikenal karena kesederhanaan dan keterbacaannya. Bahasa ini menekankan pada keterbacaan kode dan sintaks yang bersih, sehingga mudah untuk dipelajari dan ditulis.

Python mendukung berbagai paradigma pemrograman, termasuk pemrograman prosedural, berorientasi objek, dan fungsional. Python memiliki pustaka standar yang besar dan ekosistem yang luas dari paket dan pustaka pihak ketiga, membuatnya serbaguna dan cocok untuk berbagai aplikasi, termasuk pengembangan web, analisis data, komputasi ilmiah, machine learning, dan banyak lagi

Visual studio code, adalah editor kode yang dikembangkan oleh Microsoft. Ini menyediakan lingkungan yang ringan namun kuat untuk menulis, mengedit, dan men-debug kode di berbagai bahasa pemrograman, termasuk Python. VS Code mendukung fitur-fitur seperti penyorotan sintaksis, penyelesaian kode, pemfaktoran ulang kode, debugging, integrasi kontrol versi, dan ekstensi untuk fungsionalitas tambahan. VS Code sangat mudah disesuaikan dan menawarkan pengalaman pengembangan yang mulus untuk Python dan bahasa pemrograman lainnya

Web browser, adalah aplikasi perangkat lunak yang memungkinkan pengguna untuk mengakses dan menavigasi World Wide Web. Aplikasi ini mengambil dan menampilkan halaman web, menginterpretasikan HTML, CSS, dan JavaScript, serta menampilkannya secara visual bagi pengguna. Browser web memungkinkan pengguna untuk berinteraksi dengan situs web, mengakses layanan online, melihat konten multimedia, dan melakukan berbagai tugas yang berhubungan dengan web. Browser web yang populer termasuk Google Chrome, Mozilla Firefox, Microsoft Edge, Safari, dan Opera

Command Prompt, adalah penerjemah baris perintah yang tersedia di sistem operasi Windows. Ini menyediakan antarmuka berbasis teks untuk menjalankan perintah dan berinteraksi dengan sistem operasi

Support vector machine, adalah algoritma machine learning termasuk supervised learning yang digunakan untuk tugas klasifikasi dan regresi. SVM bertujuan untuk menemukan hyperplane optimal yang memisahkan titik-titik data dari kelas-kelas yang berbeda dengan margin maksimum.

Decision Tree C4.5, adalah algoritma machine learning termasuk supervised learning yang populer digunakan untuk tugas-tugas klasifikasi. Algoritma ini membangun model keputusan seperti pohon

Random forest, adalah algoritma pembelajaran ensemble yang menggabungkan beberapa pohon keputusan untuk membuat prediksi. Algoritma ini bekerja dengan membuat subset acak dari data pelatihan dan secara acak memilih fitur di setiap node pohon keputusan. Dengan merata-ratakan prediksi dari beberapa pohon, random forest mengurangi risiko overfitting dan meningkatkan kemampuan generalisasi model

K-Means Clustering, adalah algoritma machine learning termasuk unsupervised learning yang digunakan untuk mengelompokkan titik-titik data yang serupa. Algoritma ini mempartisi data ke dalam K cluster, di mana K adalah parameter yang ditentukan pengguna. algoritma dimulai dengan memilih K centroid secara acak, dan kemudian menetapkan setiap titik data ke centroid terdekat berdasarkan jarak (biasanya jarak Euclidean)



BIODATA PENULIS

Abu Tholib, M.Kom, adalah Dosen Fakultas Teknik Universitas Nurul Jadid Paiton Probolinggo pengampu mata kuliah data mining dan jaringan syaraf tiruan. pernah mendapatkan dana hibah penelitian dari ristekdikti 3 tahun berturut-

turut pada tahun 2018, 2019, 2020. Selanjutnya terkait dengan pendidikan tinggi di S1 Sekolah Tinggi Teknologi Nurul Jadid Jurusan Teknik Informatika, sedangkan untuk pendidikan S2 di Universitas Dian Nuswantoro Semarang lulus pada tahun 2013 dengan mengambil jurusan yang sama yaitu Teknik Informatika. Pada saat ini proses pendidikan S3 di Universitas Malang. Penulis juga, selain aktif mengajar Penulis juga aktif di organisasi LTN NU Kab. Probolinggo. Penulis bisa dihubungi pada Email: abu@unuja.ac.id



Streamlit



Perkembangan teknologi kecerdasan buatan dalam beberapa dekade terakhir sungguh luar biasa pesat. Salah satu cabangnya adalah Machine Learning yang berfokus pada sebuah mesin/komputer yang mampu belajar dari data yang tersedia dan mengubah data tersebut menjadi informasi dan keputusan. Pendekatan machine learning dapat digunakan untuk menyelesaikan permasalahan yang kompleks di berbagai bidang yang tidak dapat di selesaikan oleh program konvensional. Implementasi Streamlit sebagai framework Python yang digunakan untuk membangun aplikasi web dengan antarmuka pengguna interaktif untuk proyek-proyek data science dan machine learning akan sangat mudah digunakan.



PUSTAKA NURJA

ISBN 978-623-6757-74-1 (PDF)



9 786236 757741

Abu Tholib, M.Kom