Proses analisis sentimen dimulai dari proses pengumpulan Pengumpulan data dapat dilakukan melaui media sosial. Apabila mengumulkan data melalui media sosial. Dataset untuk analisis sentiment dapat dikumpulkan menggunakan hashtag. Selanjutnya dilakukan proses labeling bisa dilakukan berdasarkan komentar yang ada. Misalnya Topping ayamnya melimpah, dapat kita labeli dengan label Positif. Kalimat kedua : Antrinya panjang banget. Tempat agak panas penuh pembeli. Kalimat kedua dapat kita labeli dengan label negatif. Data yang sudah labeling bisa dilanjutkan ke preprocessing data. Preprocessing data meliputi pembersihan data dari karakter-karakter yang tidak relevan seperti tanda baca, stop word, serta stemming atau lemmatization untuk mengubah kata-kata menjadi bentuk dasarnya.

Data hasil preprocessing dilanjutkan proses Ekstraksi Fitur. Proses ekstraksi fitur dapat menggunakan Unigram dan TFIDF. Pengimplementasian proses ekstraksi fitur memanfaatkan class TfidfVectorizer pada library scikit-learn. Pada tahap ini, data akan diklasifikasikan menjadi tiga jenis sentimen yaitu positif, negatif, atau netral. Metode yang digunakan untuk klasifikasi sentimen dapat menggunakan machine learning. Terakhir, hasil analisis sentimen perlu dievaluasi dan disajikan secara visual agar mudah dipahami oleh pengguna.

Analisis sentimen menjadi semakin penting dalam dunia pemasaran karena memungkinkan perusahaan untuk memahami preferensi dan kebutuhan pelanggan secara lebih mendalam. Dengan mengumpulkan data dari media sosial dan platform online lainnya, perusahaan dapat menganalisis opini dan preferensi pelanggan terhadap merek mereka dan produk yang ditawarkan. Analisis sentiment dapat dilakukan salah satunya dengan menggunakan bahasa pemrograman python.

UNIPMA Press
WE GOT IT

Penerbit UNIPMA Press
Universitas PGRI Madiun
Jl. Setia Budi No. 85 Madiun, Jawa Timur, 63118
E-Mail: upress@unipma.ac.id
Website: kwu.unipma.ac.id

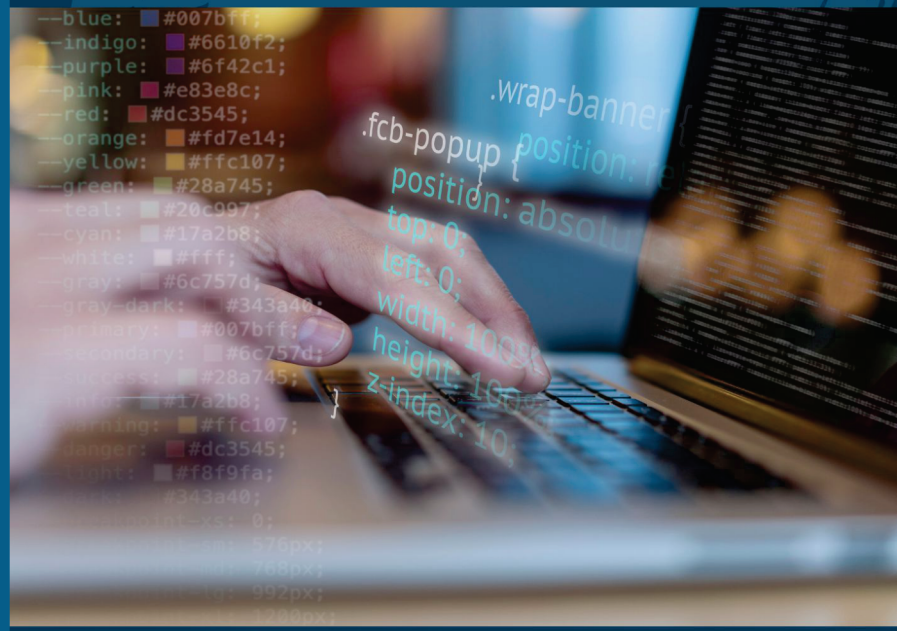
ISBN 978-623-8095-44-5
9 786238 095445

UNIPMA Press
WE GOT IT

Buku Ajar

UNIPMA Press
WE GOT IT

TEORI DAN IMPLEMENTASI ANALISIS SENTIMEN MENGGUNAKAN PYTHON



python™

Yessi Yunitasari., S.Kom.M.Cs

Buku Ajar

TEORI DAN IMPLEMENTASI ANALISIS SENTIMEN MENGGUNAKAN PYTHON

YESSI YUNITASARI., S.KOM.,M.Cs

UNIPMA Press
WE GOT IT

Buku Ajar
TEORI DAN IMPLEMENTASI ANALISIS SENTIMEN MENGGUNAKAN PYTHON

Penulis:
YESSI YUNITASARI

Perancang Sampul:
YESSI YUNITASARI

Penata Letak:
Tim Kreatif UNIPMA PRESS

Cetakan Pertama, November 2023

Diterbitkan Oleh:
UNIPMA Press (Anggota IKAPI)
Universitas PGRI Madiun
JI. Setiabudi No. 85 Madiun Jawa Timur 63118
Telp. (0351) 462986, Fax. (0351) 459400
E-Mail: upress@unipma.ac.id
Website: kwu.unipma.ac.id

ISBN: 978-623-8095-44-5

KATA PENGANTAR

Segala puji bagi Allah, Tuhan Yang Maha Esa atas rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan buku ajar yang berjudul "**TEORI DAN IMPLEMENTASI ANALISIS SENTIMEN MENGGUNAKAN PYTHON".** Tidak lupa juga penulis mengucapkan salawat serta salam kepada Nabi Besar Muhammad SAW, karena berkat beliau, kita mampu keluar dari kegelapan menuju jalan yang lebih terang.

Kami ucapkan juga rasa terima kasih kami kepada pihak-pihak yang mendukung lancarnya buku ajar ini mulai dari proses penulisan hingga proses cetak, yaitu orang tua kami, rekan-rekan kami, penerbit, dan masih banyak lagi yang tidak bisa kami sebutkan satu per satu.

Adapun, buku ajar telah selesai kami buat secara semaksimal dan sebaik mungkin agar menjadi manfaat bagi pembaca. Terimakasih.

Penulis

Yessi Yunitasari

DAFTAR ISI

BAB I
PENDAHULUAN

Pengolahan data merupakan topik yang sangat penting dalam era persaingan bisnis, dimana kita dituntut untuk menyajikan informasi dengan cepat. Informasi bekaitan dengan data untuk penentuan strategi kedepannya dan juga untuk proses bisnis sehingga penggunaan penambangan data disebut sebagai data mining.

Dengan majunya perkembangan zaman, penggunaan internet dapat menghasilkan begitu banyak data yang berjumlah sangat banyak sehingga dapat digunakan untuk mengambil keputusan. Selain itu data dapat di proses dan diolah untuk mendapatkan informasi yang bermanfaat.

A. DATA MINNING

Data mining merukapan sebuah proses penyatuan data penting dan juga informasi dalam jumlah yang banyak atau big data. Terdapat beberapa metode yang dapat digunakan dalam proses pengumpulan informasi dan data, seperti penggunaan matematika, statistika, dan teknologi kecerdasan buatan (AI).

Data mining memiliki isilah lain yang biasanya dikenal sebagai *Knowledge Discovery in Database* (KDD) dan Data Analysis. Proses pengumpulan data terdapat beberapa teknik dan langkah-langkah yaitu mulai dari cleansing atau pembersihan data, integrasi data, pemilihan data dan transformasi data hingga evaluasi pola untuk mendapakan informasi dari data tersebut.

1. Pembersihan data (untuk menghilangkan noise dan data yang tidak berubah-ubah).
2. Integrasi data (Proses penggabungan beberapa data)
3. Pemilihan data (data yang sesuai dengan tugas analisis dapat diambil dari basis data)

4. Transformasi data (data diubah dan digabungkan menjadi bentuk yang sesuai untuk penambangan dengan melakukan operasi ringkasan atau agregasi)
5. Penambangan data (sebuah proses yang penting di mana metode cerdas diterapkan untuk mengekstrak pola data)
6. Evaluasi pola (mengidentifikasi pola yang sangat menarik yang mewakili pengetahuan)
7. Presentasi pengetahuan (suatu teknik visualisasi dan penggambaran pengetahuan yang berguna dalam memaparkan pengetahuan yang diambil)

Suatu data minning menyediakan cara bagi komputer untuk mempelajari membuat keputusan data. Keputusan ini dapat memprediksi cuaca besok, suatu trend di media sosial, ataupun tanggapan masyarakat terhadap kebijakan pemerintah maupun kepuasan terhadap pelayanan publik.

Data Minning adalah bagian dari algoritma, statistik, teknik, optimisasi, dan ilmu Komputer. Data Minning juga menggunakan konsep dan pengetahuan dari bidang lain seperti linguistik, ilmu saraf, atau perencanaan kota. Penerapannya biasanya membutuhkan pengetahuan khusus untuk diintegrasikan dengan suatu algoritma tertentu.

Sebagian besar aplikasi data minning dimulai dengan membuat kumpulan data atau biasa disebut dengan Dataset. Dataset terdiri dari dari dua aspek:

1. Sampel yang merupakan objek di dunia nyata. Ini bisa berupa buku, foto, hewan, orang, daun, atau objek lainnya.

2. Fitur yang merupakan deskripsi sampel dalam dataset kami. Fitur bisa panjang, frekuensi kata tertentu, jumlah kaki, tanggal pembuatannya, dan seterusnya.

B. MANFAAT DATA MINNING

Setiap algoritma data minning memiliki parameter, baik dalam algoritma atau disediakan oleh pengguna. Setelan ini memungkinkan sebuah algoritma untuk mempelajari cara membuat keputusan terhadap data tersebut. Sebagai contoh sederhana, komputer dapat mengkategorikan kecepatan mobil melaju tergolong "cepat" atau "lambat". Sample data set kecepatan Laju Mobil dapat dilihat pada Tabel dibawah ini.

Tabel 1. Tabel Kecepatan Laju Mobil

Data mobil ke -	Kecepatan Melaju	Cepat atau Lambat
Mobil 1	120 km/jam	Cepat
Mobil 2	100 km/jam	Cepat
Mobil 3	45 km/jam	Lambat
Mobil 4	50 km/jam	Lambat
Mobil 5	63 km/jam	Cepat
Mobil 6	65 km/jam	Cepat
Mobil 7	80 km/jam	Cepat
Mobil 8	90 km/jam	Cepat
Mobil 9	40 km/jam	Lambat
Mobil 10	110 km/jam	Cepat

Sebuah contoh algoritma sederhana untuk data set diatas adalah jika kecepatan sebuah mobil melaju diatas 60 km/jam maka mobil tersebut

tergolong melaju dengan cepat. Sedangkan jika sebuah mobil melaju dibawah 60 km/jam maka mobil tersebut tergolong melaju dengan lambat.

C. DATABASE SYSTEM

Sistem basis data atau yang biasa disebut sistem manajemen basis data (DBMS) merupakan sebuah sistem yang terdiri dari kumpulan data terkait, yang disebut basis data, dan seperangkat program perangkat lunak untuk mengelola dan mengakses data. Program dalam perangkat lunak menyediakan mekanisme untuk mendeskripsikan struktur basis data dan penyimpanan data.

Perangkat lunak berfungsi untuk pengelolaan serta mentapkan bersamaan, berbagi, serta akses data terdistribusi, memastikan konsistensi dan juga keamanan sebuah informasi dapat disimpan walaupun terdapat crash pada sistem. Database relasional merupakan kumpulan tabel yang masing - masing diberikan keunikan nama.

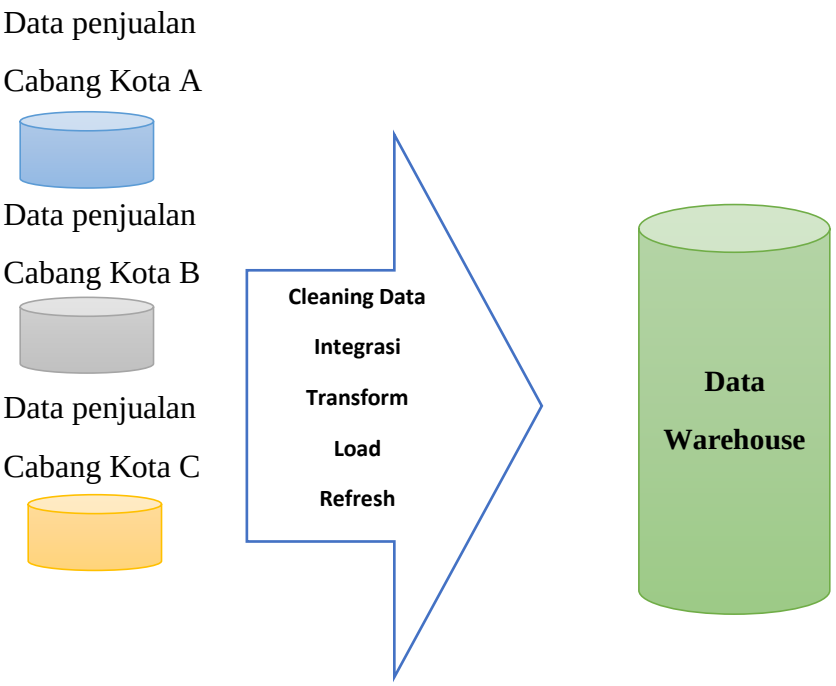
Setiap tabel terdiri dari satu set atribut (kolom atau field), biasanya disimpan dalam satu set tupel besar (record atau baris). Dalam tabel rasional unuk setiap tupel dapat mewakili objek yang diidentifikasi oleh primary key dan dijelaskan oleh satu set nilai atribut.

Model data semantik seperti model data ER atau model data entitas-hubungan, sering kali dibuat untuk database relasional. Model data entitas-hubungan mewakili database sebagai hubungan antar data dan juga satu set entitas.

D. DATA WAREHOUSES

Sebuah gudang informasi yang dikumpulkan dari banyak sumber lalu disimpan dalam skema terpadu dan biasanya berada di satu situs disebut sebagai data warehouse. Data tersebut dibagun dari proses pembersihan data,

integrasi data, transformasi data, pemuatan data, dan pembaruan data secara berkala. Ilustrasi data warehouse dapat dilihat pada Gambar 1.



E. RINGKASAN

Data Minning merupakan proses untuk menemukan pola menarik dari jumlah yang sangat besar data. Sebagai proses penemuan pengetahuan, biasanya melibatkan pembersihan data, integrasi data, pemilihan data, transformasi data, penemuan pola, evaluasi pola, dan presentasi pengetahuan. Fungsi data minning dapat digunakan untuk menentukan jenis pola atau pengetahuan meliputi karakterisasi, pengambilan pola, asosiasi, dan korelasi, klasifikasi dan regresi.

F. **LATIHAN**

1. Apakah data minning itu?
2. Jelaskan langkah-langkah yang terlibat dalam data minning untuk proses penemuan pengetahuan !
3. Apakah perbedaan antara Database dan Data Warehouse?
4. Dimanakah letak kesamaan antara Database dan Data Warehouse?

BAB II
ANALISIS SENTIMEN

Analisis sentimen merupakan sebuah proses pengumpulan dan juga evaluasi data yang akan digunakan untuk mengidentifikasi, mengekstrak, dan memproses informasi dari berbagai sumber untuk menentukan sentimen atau pendapat yang terkait dengan topik tertentu. Adapula yang menyebutkan bahwa analisis sentiment adalah deteksi sikap-sikap (attitude) terhadap objek/orang.

Opini publik dapat menjadi faktor penting dalam keputusan bisnis, politik, dan masyarakat. Namun, seringkali sulit untuk mengukur opini publik secara akurat dan efektif. Inilah mengapa analisis sentimen menjadi begitu penting. Dengan menggunakan algoritma khusus, analisis sentimen dapat membantu kita memahami perasaan dan pandangan orang terhadap suatu topik atau produk tertentu.

Dalam konteks bisnis, analisis sentimen dapat membantu perusahaan dalam memahami umpan balik pelanggan dan menciptakan strategi yang lebih efektif untuk meningkatkan kepuasan pelanggan. Dalam bisnis, analisis sentimen sangat penting karena dapat membantu perusahaan dalam memahami preferensi dan kebutuhan pelanggan.

Analisis sentiment dapat digunakan oleh perusahaan untuk mengidentifikasi isu-isu yang mempengaruhi persepsi publik tentang merek mereka. Selain itu, analisis sentimen juga dapat membantu meningkatkan efektivitas pemasaran dengan memahami preferensi pelanggan dan menyesuaikan strategi pemasaran.

Contoh kasus penggunaan analisis sentimen dalam kehidupan nyata adalah ketika sebuah perusahaan melakukan survei untuk mengetahui bagaimana konsumen merespons produk mereka. Dengan menggunakan analisis sentimen, perusahaan dapat mengetahui apakah umpan balik dari konsumen positif atau negatif. Misalnya, jika banyak konsumen mengeluh tentang kualitas produk, perusahaan dapat memperbaiki produk mereka agar lebih memuaskan pelanggan.

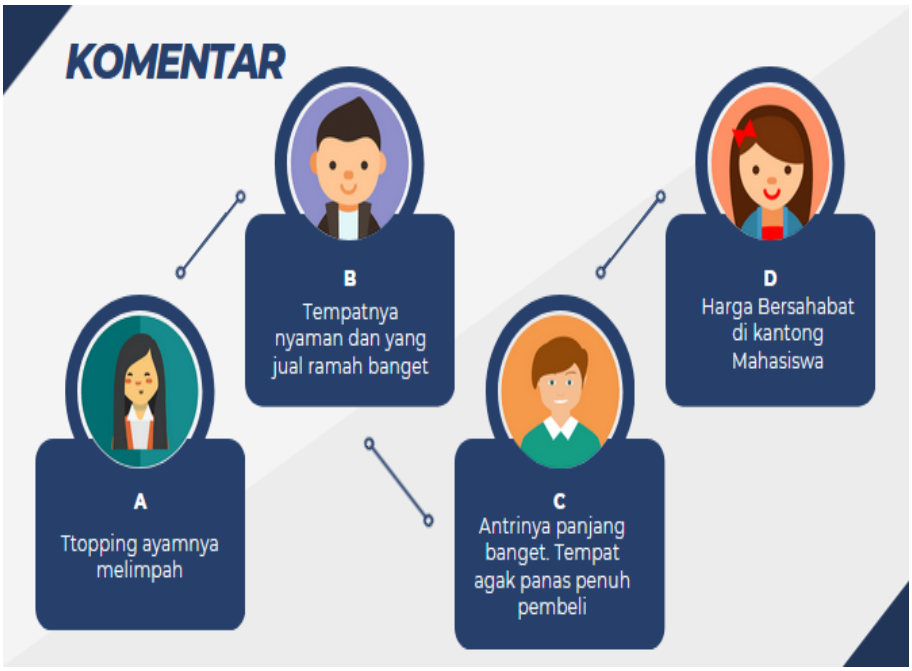
Contoh berikutnya yaitu sebuah perusahaan kosmetik menggunakan analisis sentimen untuk memantau percakapan online tentang produk-produk mereka. Mereka menemukan bahwa banyak pelanggan mengeluh tentang kemasan yang sulit dibuka. Perusahaan kemudian membuat perubahan pada kemasan produk mereka, dan melihat peningkatan signifikan dalam kepuasan pelanggan. Tahapan analisis sentiment dapat dilihat pada gambar 2.1.

Gambar 2.1 Tahapan Analisis Sentimen

A. Pengumpulan Data

Pengumpulan data dapat dilakukan melaui media sosial. Apabila mengumulkan data melalui media sosial. Dataset untuk analisis sentiment dapat dikumpulkan menggunakan hashtag.

Contoh hastag yang dapat digunakan adalah #ProdukPopuler, #Bisnis, #ProdukUMKM apabila kita ingin mengumpulkan dataset terkait produk yang popular, data yang menuat kata bisnis dan data yang memuat kata produk UMKM. Apabila kita ingin mencari data terkait Tempat makan A. Maka contoh data komentar dapat dilihat pada Gambar 2.2.



Gambar 2.2 Contoh Data Komentar yang didapat berdasar hastag tertentu.

2.1. Labeling

Proses labeling bisa dilakukan berdasarkan komentar yang ada. Misalnya Topping ayamnya melimpah, dapat kita labeli dengan label

Positif. Kalimat kedua : Antrinya panjang banget. Tempat agak panas penuh pembeli. Kalimat kedua dapat kita labeli dengan label negatif.

2.2. Preprocessing

Preprocessing data meliputi pembersihan data dari karakter-karakter yang tidak relevan seperti tanda baca, stop word, serta stemming atau lemmatization untuk mengubah kata-kata menjadi bentuk dasarnya.

2.3. Ekstraksi Fitur

Proses Ekstraksi Fitur Menggunakan Unigram dan TFIDF. Pengimplementasian proses ekstraksi fitur memanfaatkan class TfidfVectorizer pada library scikit-learn.

2.4. Training

Pada tahap ini, data akan diklasifikasikan menjadi tiga jenis sentimen yaitu positif, negatif, atau netral. Metode yang digunakan untuk klasifikasi sentimen dapat menggunakan machine learning.

2.5. Testing

Terakhir, hasil analisis sentimen perlu dievaluasi dan disajikan secara visual agar mudah dipahami oleh pengguna.

Analisis sentimen menjadi semakin penting dalam dunia pemasaran karena memungkinkan perusahaan untuk memahami preferensi dan kebutuhan pelanggan secara lebih mendalam.

Dengan mengumpulkan data dari media sosial dan platform online lainnya, perusahaan dapat menganalisis opini dan preferensi pelanggan terhadap merek mereka dan produk yang ditawarkan.

B. RINGKASAN

Menganalisis opini dan preferensi pelanggan, perusahaan dapat meningkatkan layanan pelanggan, reputasi merek, dan efektivitas pemasaran.

Perusahaan dapat mengetahui apa yang sedang dibicarakan oleh pelanggan dan bagaimana mereka merespons produk atau layanan perusahaan. Dengan adanya hal ini dapat mendukung perusahaan dalam pengambilan keputusan yang lebih baik dan akurat untuk kedepannya.

C. LATIHAN

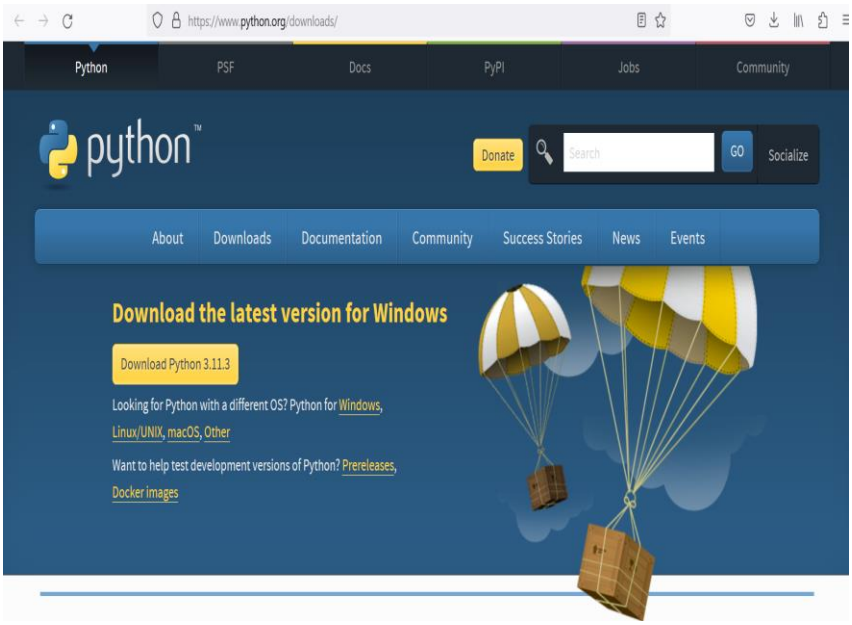
1. Apakah Analisis sentiment itu?
2. Bagaimana cara melakukan analisis sentiment untuk dunia bisnis?
3. Apa yang membedakan ketika kita akan melakukukan analisis sentiment terkait dunia Pendidikan dan terkait dunia bisnis atau Industri?
4. Apa manfaat analisis sentiment untuk produsen?
5. Algoritma apa saja yang dapat digunakan untuk analisis sentiment?

BAB III
INSTALL PYTHON

A. Instalasi Python

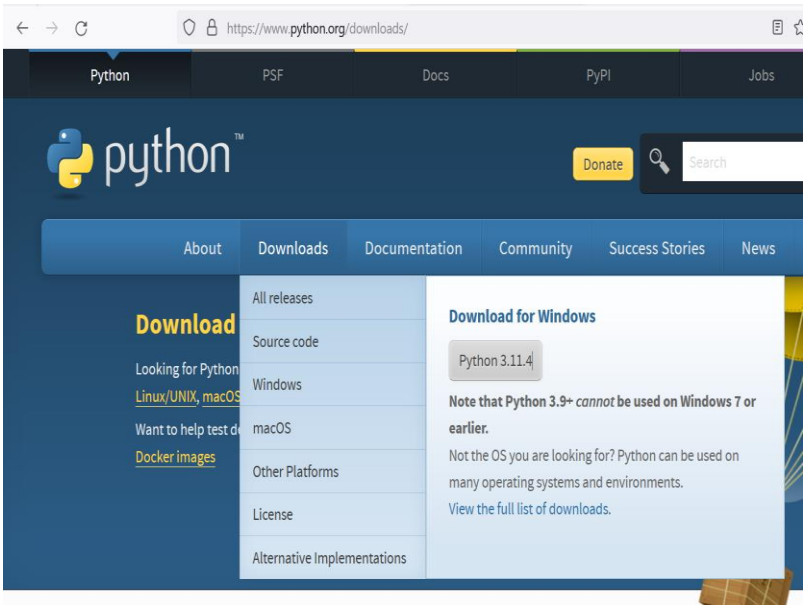
Python ialah salah satu bahasa pemrograman yang sangat populer dan juga banyak digunakan oleh para programmer untuk mengoding. Untuk para pemula juga sangat tertarik menggunakan python karena struktur sintaknya yang rapi dan juga mudah dipahami. Berikut adalah Langkah-langkah untuk install python :

1. Silhakan buka halaman ini **https://www.python.org/downloads/** . **Halaman yang akan ditampilkan adalah seperti ini.**



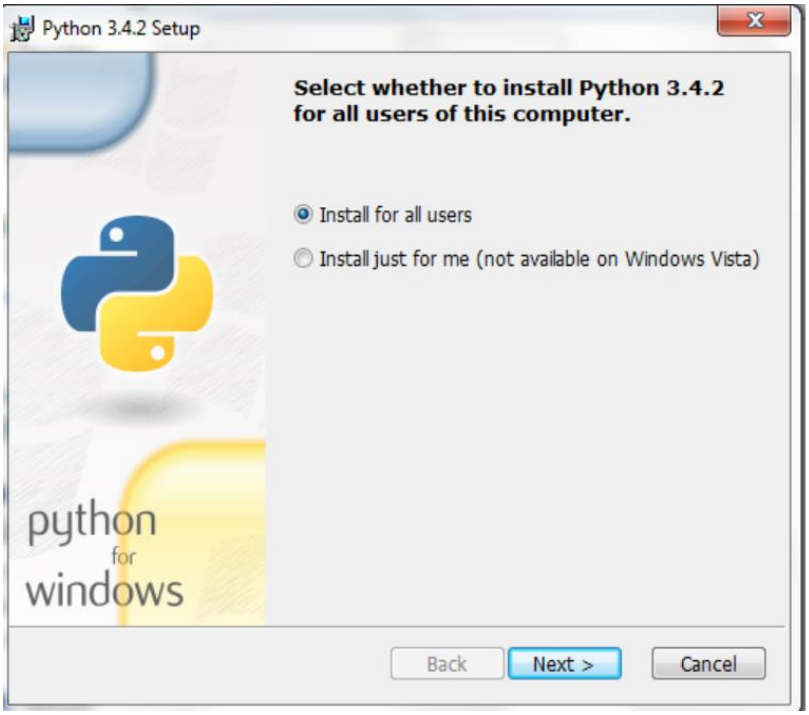
2. Selanjutnya pilihlah jenis sistem operasi sesuai yang digunakan (Windows, UNIX/Linux, Mac OS X, atau lainnya). Jika menggunakan pc atau laptop dengan OS Windows 10 maka klik Windows.

3. Klik salah satu versi Python lalu *scroll* ke bawah hingga muncul pilihan *Files* seperti gambar di atas.
4. Pilih *files* sesuai versi Windows yang digunakan yaitu 32 bit atau 64 bit. Klik file itu dan proses *download* akan mulai berjalan. Walaupun terdapat banyak pilihan/versi, kita tidak perlu bingung memilihnya. Hal yang membedakan antara satu versi dengan lainnya hanyalah fitur yang tersedia karena masing-masing versi memiliki dasar yang sama. Versi Python terbaru memiliki fitur yang paling lengkap.
5. Hal yang perlu kita perhatikan adalah ketika kita ingin mendowload python adalah versi dari OS yang kita gunakan.



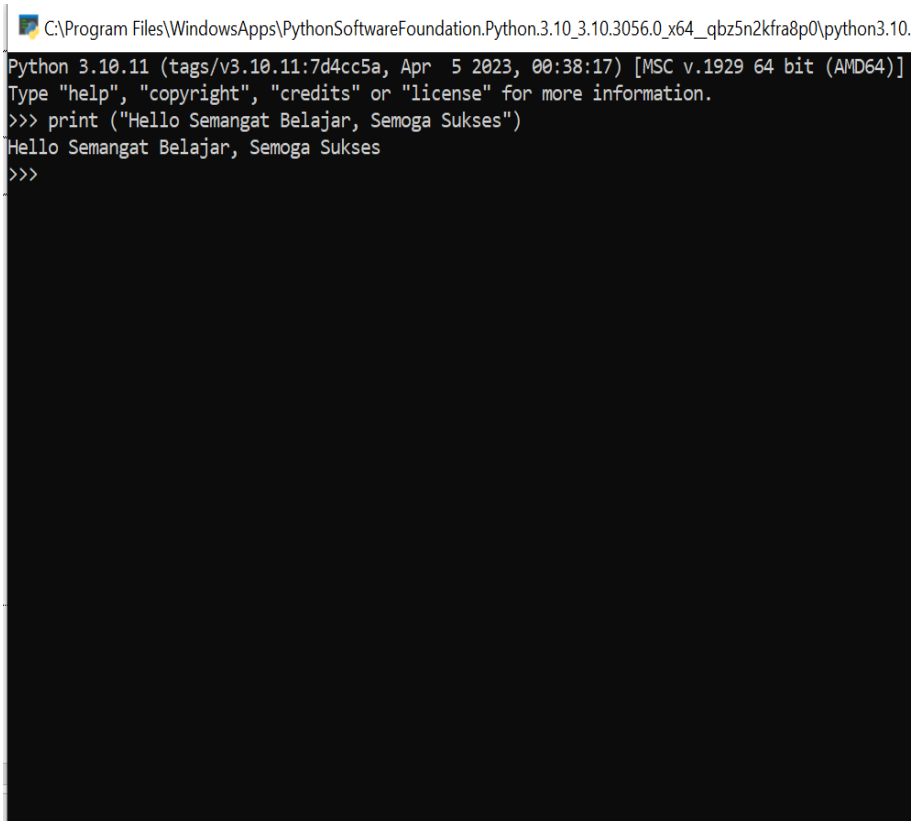
Versi Python 3.9+ tidak dapat digunakan pada Windows 7 atau sebelumnya. Jadi apabila kita masih menggunakan windows 7 silahkan pilih Python versi 3.9 kebawah.

6. Setelah kita download kita dapat open file tersebut sehingga tampilannya akan muncul seperti gambar dibawah ini :



7. Kemudian kita dapat pilih Install for all users. Install for all users dipilih karena semua user computer dapat memakai python nantinya.
8. Tahap selanjutna adalah lokasi Instalasi. Tentukan lokasi dari Python akan diinstall.
9. Proses selanjutnya adalah Kostumisasi. Kostumisasi adalah proses dimana kita menentukan fitur-fitur mana saja yang akan kita install.

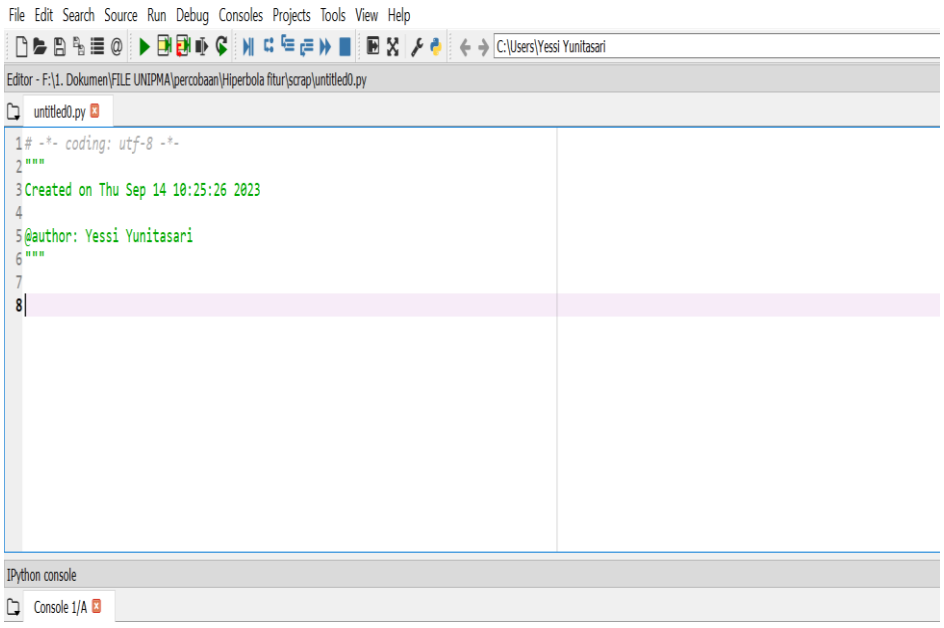
10. Aktifkan Add python.exe to path supaya CMD atau Command Prompt dapat mengenali perintah dari Python. Kemudian klik Finish untuk menyelesaikan.
11. Uji Coba Python. Setelah instalasi berhasil kita dapat uji coba python dengan membuka Python Shell. Pertama kita buka Start Menu kemudian cari ***Python Shell***.
12. Cobalah ketikkan print ("Hello Semangat Belajar, Semoga Sukses"). Jika instalasi sukses maka akan muncul tampilan seperti gambar dibawah:



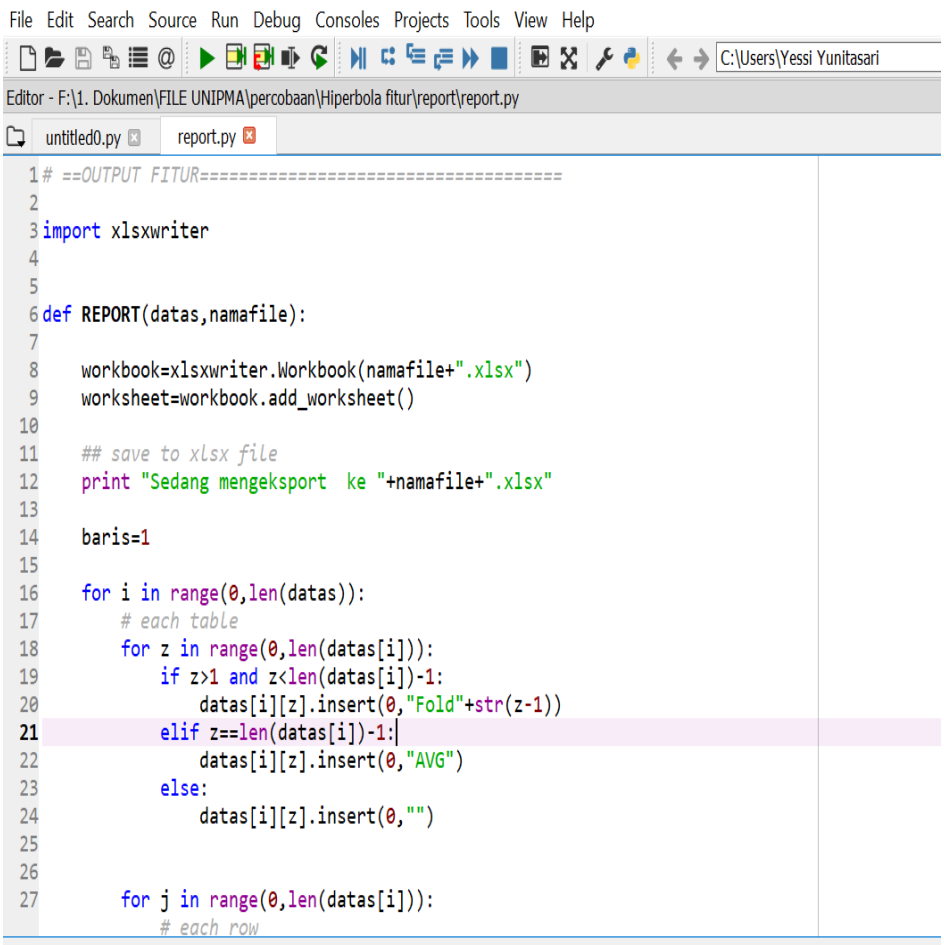
B. Cara menulis Kode Program di Spider

Kode program python dapat kita tuliskan pada spider. Apabila kita sudah melakukan instalasi silahkan cari spider di kompter yang sudah kita install.

Apabila sudah selesai proses open maka akan terbuka tampilan seperti di bawah ini.



Kode program siap dituliskan sesuai dengan algoritma atau sistem yang ingin dibangun. Berikut ini contoh program apabila kita ingin membuat report untuk save data dalam bentuk excel.



C. Cara Menulis Kode Program di Goggle Colab

Apabila kita ingin menuliskan program secara online kita dapat menggunakan Google Colab. Namun sebelumnya dapat kita bahas terlebih dahulu kelebihan dan kekurangan apabila kita menggunakan Google Colab.

Google Colab atau yang biasanya disebut dengan Google Colaboratory merupakan platform yang disediakan oleh Google untuk mengembangkan dan bereksperimen pada proyek-proyek machine learning atau proyek deep learning.

Google Colab merupakan salah satu platform yang menggunakan cloud computing dari google sehingga tidak perlu melakukan instalasi dan konfigurasi pada gadget pengguna. Selain itu Google Colab dapat digunakan tanpa mengeluarkan biaya atau gratis dan juga terintegrasi dengan Google Drive.

1. Kelebihan
    a. Google Colab ialah sebuah platform yang dapat digunakan secara gratis atau tanpa pengeluaran biaya. Platform ini sangat cocok bagi pengembang atau mahasiswa yang ingin belajar machine learning kareana pengguna tidak perlu membayar untuk menggunakan layanan ini.
    b. Google Colab tersambung dengan Google Drive, sehingga pengguna platform ini dapat dengan mudah menyimpan dan mengakses file proyek yang telah dikerjakan secara mudah dan terstruktur.
    c. Penggunan Google Colab sangat efektif karena menggunakan pelayanan cloud computing langsung dari Google, sehingga pengguna tidak perlu repot memasang atau melakukan konfigursi perangkat lunak pada masing-masing komputer untuk menjalankan proyek machine learning.
2. Kekurangan
    a. Google Colab tidak dapat menjamin jenis perangkat keras yang akan diterima. Seperti GPU apa yang akan diterima atau berapa kapasitas Memory yang dapat digunakan secara maksimal.

b. Google Colab tidak menyediakan persistent storage atau penyimpanan permanen (jangka panjang).Sebenarnya Colab sudah menawarkan kemampuan untuk menghubungkan Google Drive dan Colab, namun untuk pengguna yang melakukan pemrograman secara intens atau membuat sistem yang terdiri dari banyak source code file dan data.

c. Hanya terbatas pada format notebook. Walaupun terdapat beberapa trik yang dapat digunakan, secara alami google colab hanya dapat menyediakan format berupa notebook (.ipynb). Sedangkan untuk programmer yang biasa coding dengan script file (.py), atau yang perlu dengan terminal, saat ini belum dapat dilakukan.

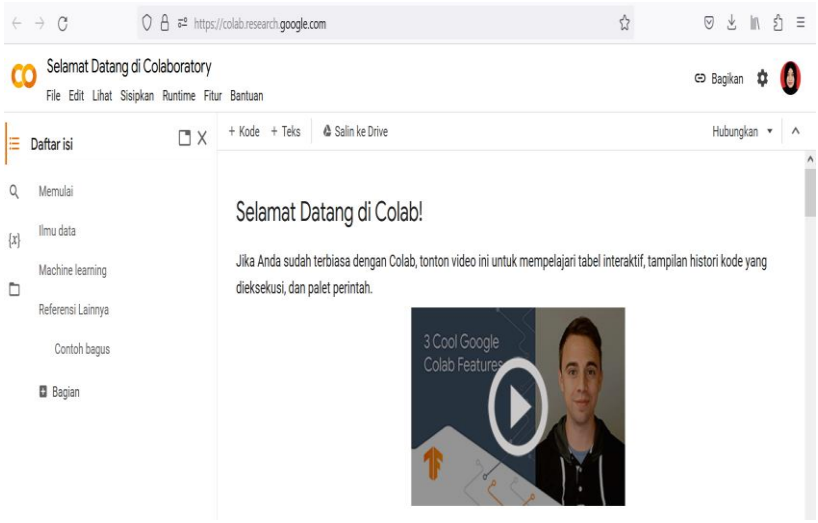
d. Terbatasnya waktu runtime.

   Koneksi bisa diputus jika colab kita tidak aktif dalam beberapa menit. Selain itu running program juga dibatasi maksimal hanya 12 jam, setelahnya ada jeda (sesi cooling down) untuk beberapa waktu baru kita dapat menggunakan Google Colab lagi.

3. Cara menggunakan Google Colab

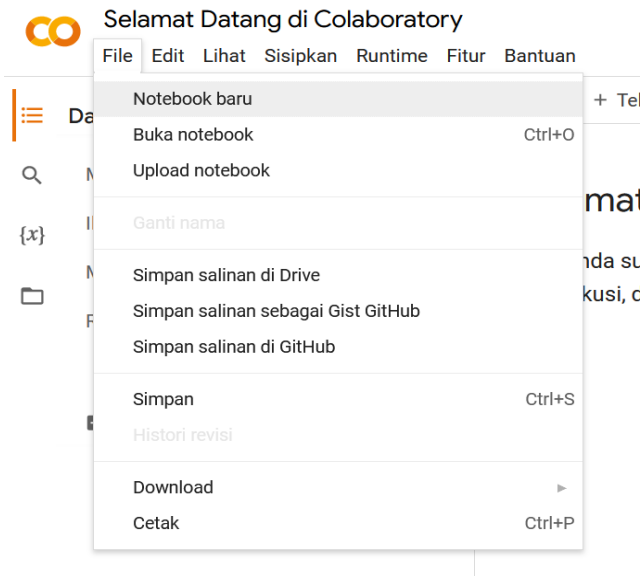
   a. Mengakses Google Colab

      Pengguna dapat mengakses Google Colab dengan cara mengunjungi situs web https://colab.research.google.com/. Selanjutnya, pilih buat notebook baru atau membuka file notebook yang sudah ada.

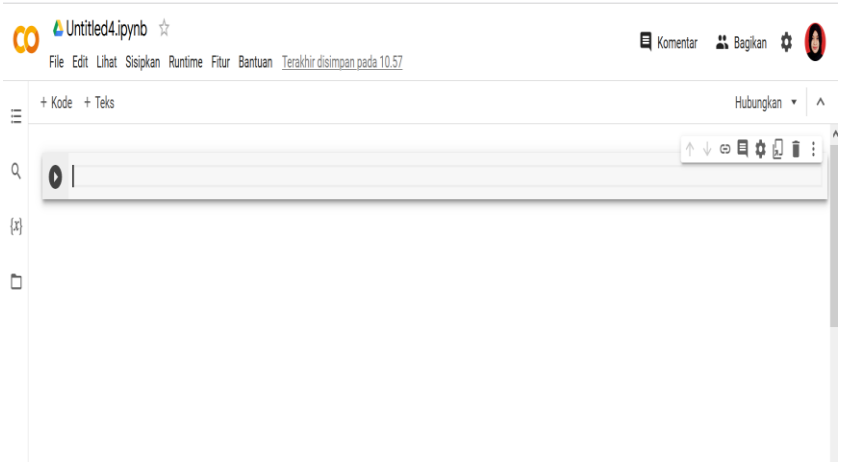


b. Membuat Notebook Baru

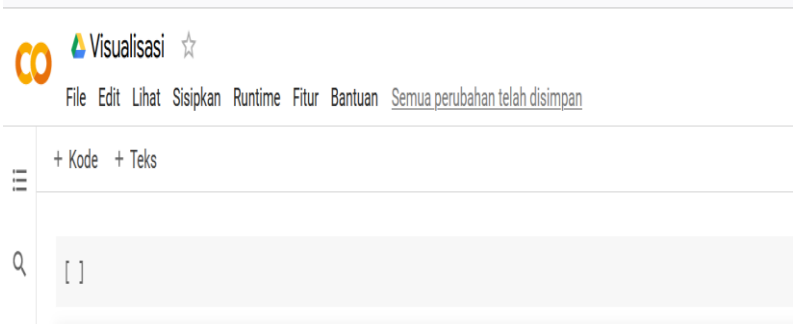
Untuk membuat notebook baru dapat dengan cara mengklik tombol "New Notebook". Selanjutnya kita dapat memilih untuk membuat notebook kosong atau menggunakan salah satu template yang disediakan oleh Google Colab. Jika ingin mmebuat notebook kosong, pilih **File** di pojok kiri atas lalu klik **Notebook baru**.



Tampilan yang akan muncul adalah seperti dibawah ini :

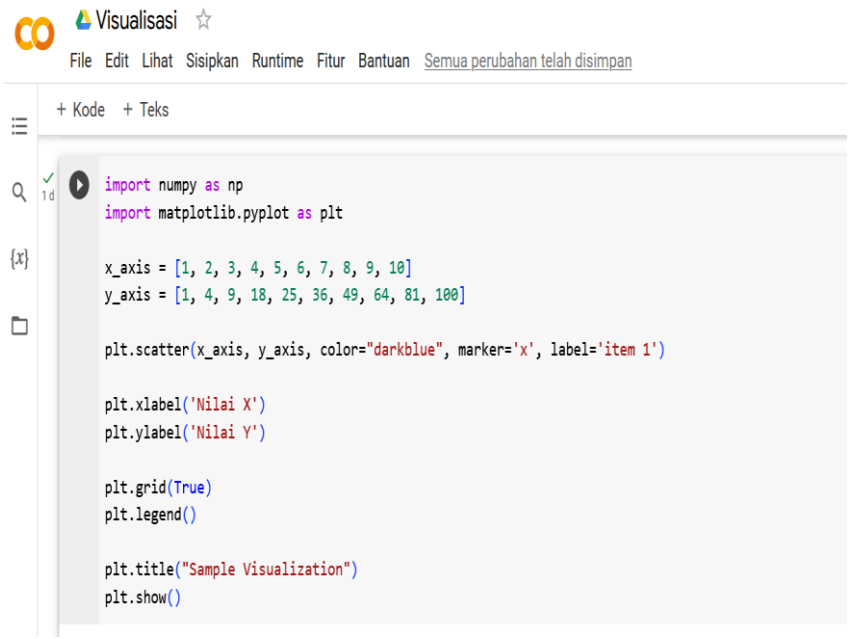


Kita dapat mengubah nama file di Google Colab kita dengan merubah **Untitled4.ipynb** dengan mengetikkan nama file kita di pojok kiri atas. Berikut contoh Google Colab yang sudah dirubah nama filenya menjadi **Visualisasi**.



c. Mengkoding Kode Python

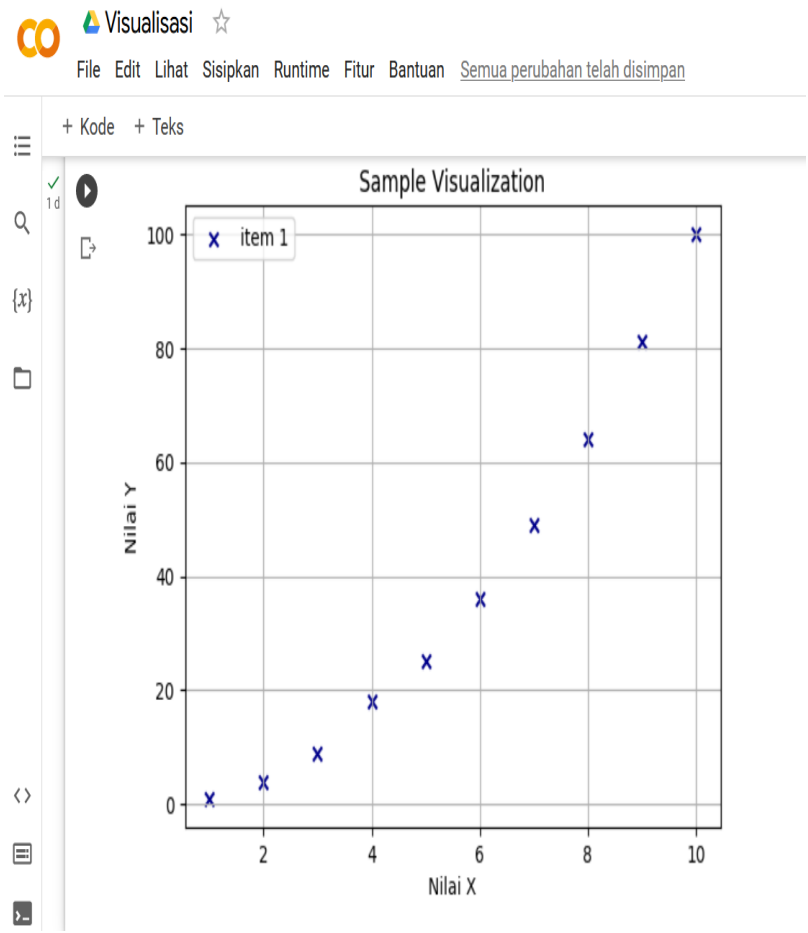
Pengguna dapat mulai mengkoding kode Python di dalam notebook Google Colab. Notebook ini memiliki fitur syntax highlighting dan autocompletion, sehingga pengguna dapat mengkoding dengan cepat dan mudah. Berikut ini adalah contoh kode program untuk melakukan visualisasi terhadap titik x_axis dan y_axis.



d. Menjalankan Kode Python

Setelah menulis kode Python, pengguna dapat menjalankan kode tersebut dengan menekan tombol "Run" atau menggunakan pintasan keyboard. Google Colab akan menjalankan kode Python tersebut dan menampilkan outputnya di bawah kode. Berikut ini adalah hasil run terhadap kode program visualisasi terhadap titik x_axis dan y_axis.

Kita dapat menambahkan teks, gambar, dan grafik ke dalam notebook Google Colab. Untuk menambahkan teks, kita bisa menggunakan markdown. Selain itu untuk menambahkan grafik dan gambar, kita dapat memanfaatkan library seperti Matplotlib atau Seaborn.



D. Latihan

1. Silahkan praktekkan cara menginstall python pada laptop masing-masing dengan memilih versi Python sesuai kebutuhan. Pastikan proses Installasi Python berhasil.
2. Silahkan coba buat program sederhana menggunakan google colab.
3. Tampilkan sebuah data grafik atau visualisasi menggunakan data set sederhana.

BAB IV
CRAWLING DATA PADA SOSIAL MEDIA

Crawling data biasa disebut dengan proses pengambilan data secara online yang dapat diakses oleh masyarakat umum. Crawling data salah satunya dapat diperoleh melalui sosial media. Pada pembahasan kali ini sosial media yang digunakan adalah twitter.

Twitter adalah salah satu jejaring sosial yang sering digunakan sebagai alat komunikasi dan media periklanan. Twitter merupakan jejaring sosial dengan ciri dan format tersendiri, dengan simbol dan aturan khusus.

Seperti biasa, pengguna Twitter hanya bisa mengirim dan melihat pesan blog maksimal 140 karakter. Pesan-pesan ini disebut tweet. Tweet yang diposting oleh pengguna berbeda-beda tergantung keinginan pengguna. Tweet berisi opini, saran, dan kritik tentang topik tertentu.

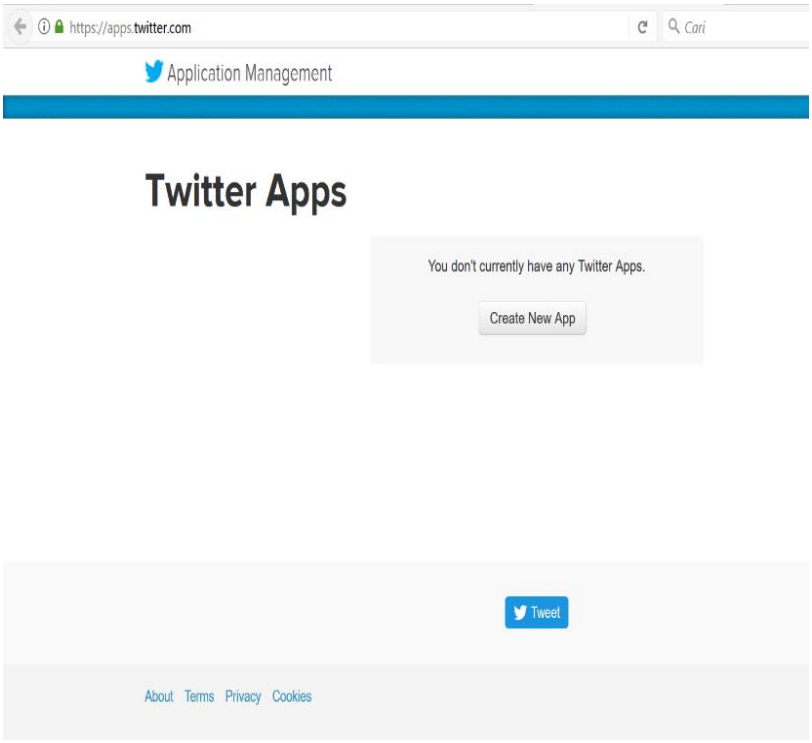
Twitter memungkinkan pengguna untuk mengakses data dari twitter melalui API Twitter, Beberapa jenis API yang disediakan oleh Twitter untuk masalah tweet antara lain :

1. *Stream* API : Jenis API yang dapat digunakan untuk mendapatkan data pencarian melalui kata kunci *tweet* untuk kurun waktu saat ini *(realtime).*
2. REST API : API yang dapat digunakan untuk pengambilan data seperti posting dan pengambilan *tweet* pada akun tertentu *(user id),* status tertentu *(status id),* maupun suatu lokasi tertentu *(geolocation).*

Langkah awal yang harus dilakukan sebelum memulai crawling data adalah membuat Twitter Apps.

A. Langkah Membuat Twitter Apps

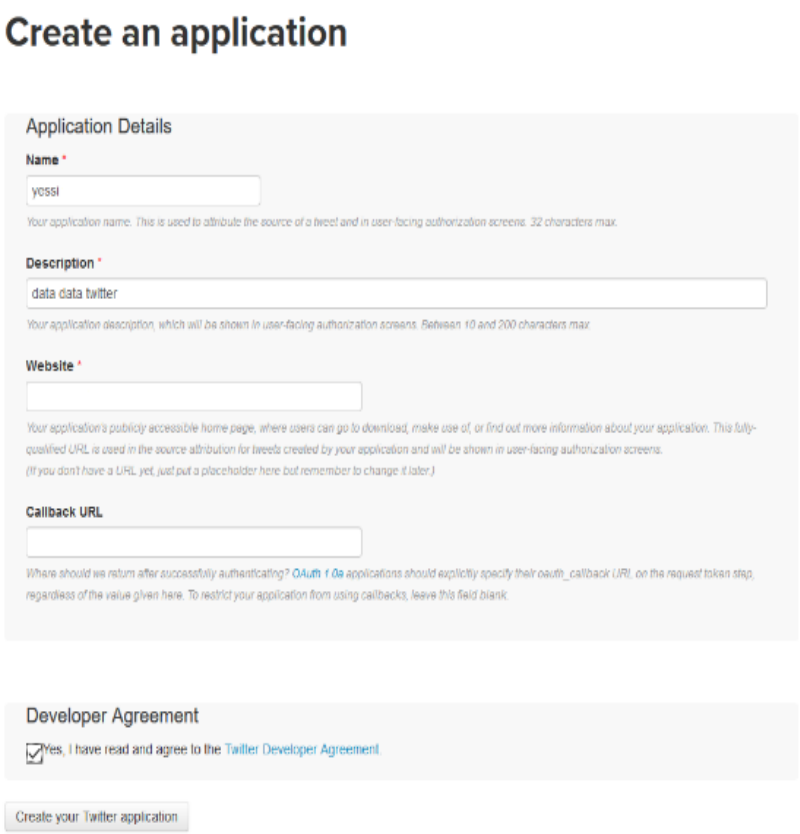
Aplikasi Twitter dibuat sebagai antarmuka multimedia yang menghubungkan aplikasi pengguna dengan server data di Twitter. Kita dapat membuat twitter apps dengan cara register akun media social twitter terlebih dahulu, kemudian menuju twitter application management dengan alamat url https://apps.twitter.com. Halaman yang akan muncul dapat dilihat pada gambar 4.1.



Gambar 4.1 Twitter Application manajement

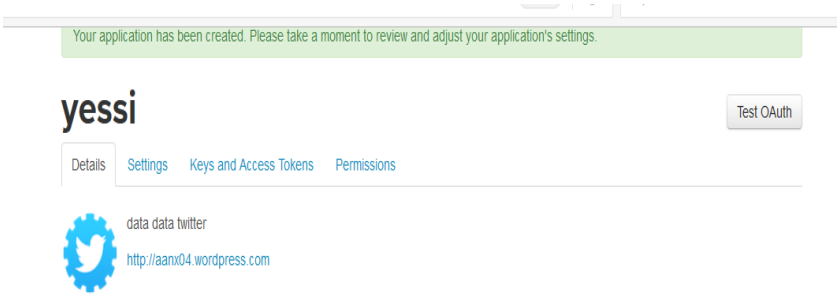
Pembentukan twitter apps dapat dilakukan dengan klik tombol Create New App kemudian akan dibawa ke halaman seperti Gambar 4.2. Isi semua yang bertanda bintang (*) dan jangan sampai lupa untuk memberi tanda centang pada bagian Developer Agreement. Isikan nama pada bagian nama.

Ketikkan deskripsi terkait data yang akan diambil. kemudian isikan alamat website yang dimiliki (dapat berupa blogspot maupun wordpress). Callback URL dapat dikosongkan karena tidak memerlukan server external untuk autentikasi akun.



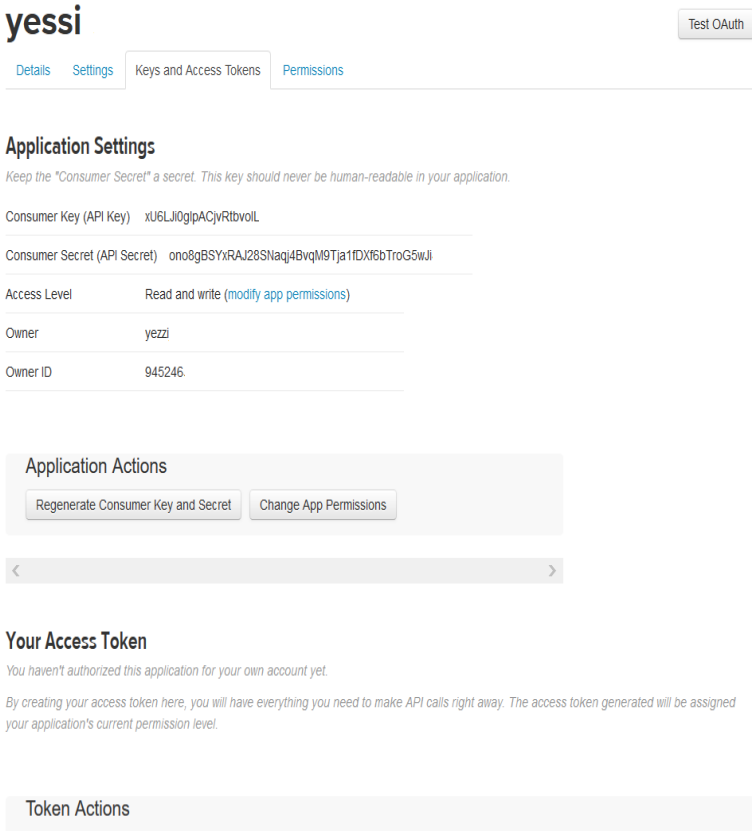
Gambar 4.2 Create New App

Setelah Checklist Developer Agreement selanjutnya pilih tombol Create your Twitter application dan akan berpindah halaman seperti Gambar 4.3. dibawah ini :



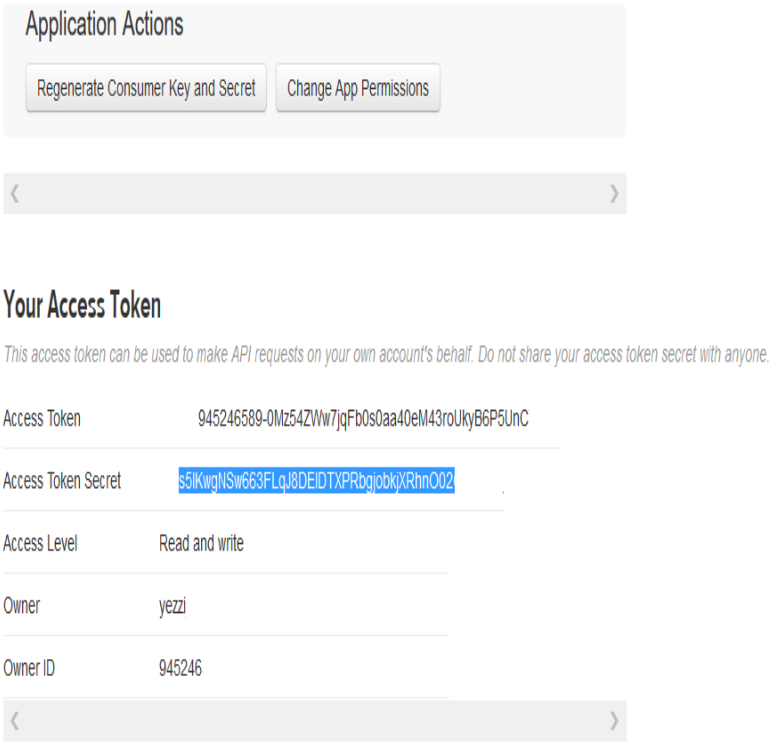
Gambar 4.3. Halaman Application Management

Dari halaman Application Management tersebut pilih tabulasi Keys and Access Tokens untuk mendapatkan kode Consumer Key dan Consumer Secret pada Gambar 4.4.



Gambar 4.4 Halaman Keys and Access Tokens

Selain Consumer Key dan Consumer Secret dibutuhkan Access Token dan Access Token Secret untuk autentikasi penggunakan Twitter App tersebut. Untuk mendapatkan Access Token dan Access Token Secret dapat dilakukan dengan cara klik pada Create my access token. kita bisa melakukan pencabutan token access yang telah tergenerate dengan klik Revoke Token Access. Setelah Access Token dihasilkan, tahap selanjutnya menuju tabulasi Permissons pada radio button pilih Read, Write and Access direct messages dapat ditunjukan seperti gambar dibawah kemudian klik Update Settings untuk menyimpan seperti Gambar 4.5.



Gambar 4.5. Access Token

Tahapan ini Twitter App bisa digunakan sebagai akses pengambilan data tweet. Langkah selanjutnya yaitu membuat Coding Python dengan Library Tweepy.

B. Coding Python dengan Library Tweepy

API Twitter App telah selesai dibuat, kemudian diperlukan source code untuk melakukan akses kedalam API tersebut. Untuk pengambilan data secara streaming, python menyediakan salah satu library khusus yang dapat digunakan yaitu library Tweepy. Berikut merupakan baris code yang dapat dituliskan.

```
import tweepy
import csv
consumer_key = "xU6LJi0glpACjvRtbvoILY3tz"
consumer_secret = "ono8gBSYxRAJ28SNaqj4BvqM9Tja1fDXf6bTroG5wJi4A4tl"
access_token = "945246589-0Mz54ZWw7jqFb0s0aa40eM43roUkyB6P5UnC7A"
access_token_secret = "s5lKwgNSw663FLqJ8DElDTXPRbgjobkjXRhnO02GDq"

auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)
api = tweepy.API(auth,wait_on_rate_limit=True)

csvFile = open('kampus merdeka.csv', 'a')
csvWriter = csv.writer(csvFile)

for tweet in tweepy.Cursor(api.search,q="kampus merdeka",count=1500, lang="
                           since="2022-01-01").items():
    print (tweet.created_at, tweet.text)
    csvWriter.writerow([tweet.created_at, tweet.text.encode('utf-8')])

```

Keterangan :

import tweepy digunakan untuk menggunakan library pengambilan data.

import csv berfungsi sebagai penyimpanan data hasil crawling ke dalam format csv.

Isikan data consumer_key, consumer_secret, access_token, access_token_secret sesuai dengan kode yang telah diperoleh sebelumnya. consumer_secret, access_token, access_token_secret sesuai dengan kode yang telah diperoleh sebelumnya.

Baris kode ini csvFile = open('kampus merdeka.csv', 'a') digunakan untuk memberikan nama file csv, nama file dapat disesuaikan dengan projek yang sedang dikerjakan.

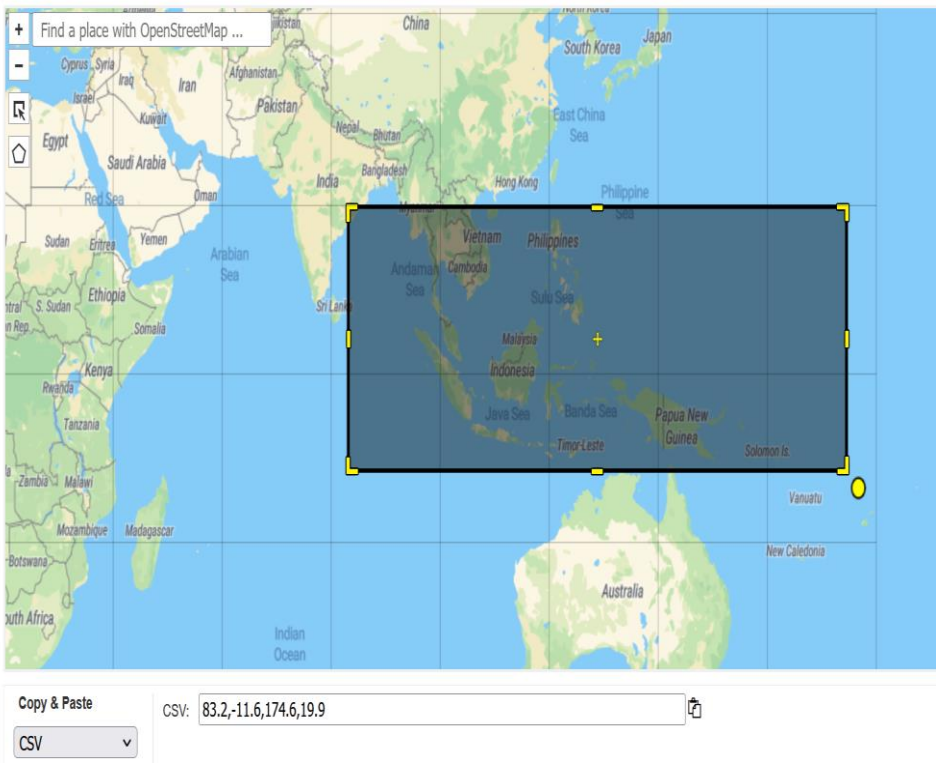
Baris kode ini for tweet in tweepy.Cursor(api.search,q="kampus merdeka",count=1500, lang="id", since="2022-01 01").items():
digunakan untuk mencari tweet berdasarkan kata kunci tertentu, misalnya dengan kata kunci "Kampus Merdeka". Count merukapan jumlah data yang ingin dihasilkan. Jumlah data dapat disesuikan dengan kebutuhan. Waktu tweet juga dapat di set sesuai tanggal yang diinginkan.

C. Filter Berdasarkan Lokasi

Untuk melakukan filter berdasarkan lokasi tertentu, kita dapat menambahkan baris kode berikut :

```
# http://boundingbox.klokantech.com/

# option csv

stream.filter(locations=[105.03,-8.99,115.67,-5.59],track =['Kampu Merdeka'])
```

Titik lokasi dapat di rubah dengan mengcopy paste kode yang ada di kotak csv seperti Gambar 4.6. Pada contoh diterapkan filter lokasi tweet untuk wilayah yang ada di area dikotak hitam. Track yang dimaksud disini adalah kata kunci tweet yang digunakan.



Gambar 4.6. Filter Lokasi

Apabila kita ingin mengcrawling data tweet di sekitaran pulau jawa maka kita dapat merubah seperti Gambar 4.7 dan secara otomatis maka titik pada bagian CSV juga akan berubah. Titik yang digunakan adalah 105.3871,-8.9586,114.4728,-5.8978.

Gambar 4.7 Filter Lokasi Tweet untuk Pulau Jawa

Pengambilan data tweet secara real time dapat dilakukan dengan menambahkan 3 libarary dengan baris kode seperti pada Gambar 4.8.

```
from tweepy.streaming import StreamListener
from tweepy import OAuthHandler
from tweepy import Stream
```

Gambar 4.8 Library untuk crawling Realtime

StreamListener merupakan class tweepy yang digunakan client untuk melakukan listen (hubungan) dengan Twitter API stream. OAuthHandler digunakan untuk penanganan autentikasi consumer key dan consumer secret pada Twitter API. Sedangkan Stream digunakan untuk menyambungkan hasil autentikasi menuju capturing data tweet.

D. **Hasil Proses Crawling**

Hasil dari proses crawling data berdasarkan kata kunci "Kampus Merdeka" dapat dilihat prosesnya dari 1 tweet sampai tweet terakhir pada terminal. Setelah proses crawling selesai sesuai dengan banyaknya data yang dibutuhkan maka data tweet tersebut akan tersimpan pada file format CSV maupun Excel sesuai dengan baris kode yang telah dituliskan.

Format hasil penyimpanan dalam bentuk File Excel dapat kita proses dengan mengunakan baris kode dibawah ini :

```
import xlsxwriter

workbook=xlsxwriter.Workbook("Kampus Merdeka.xlsx")

worksheet=workbook.add_worksheet()
```

Data hasil proses crawling akan disimpan dalam format excel. Proses crawling data pada terminal dapat dilihat pada Gambar 4.9 dibawah ini:

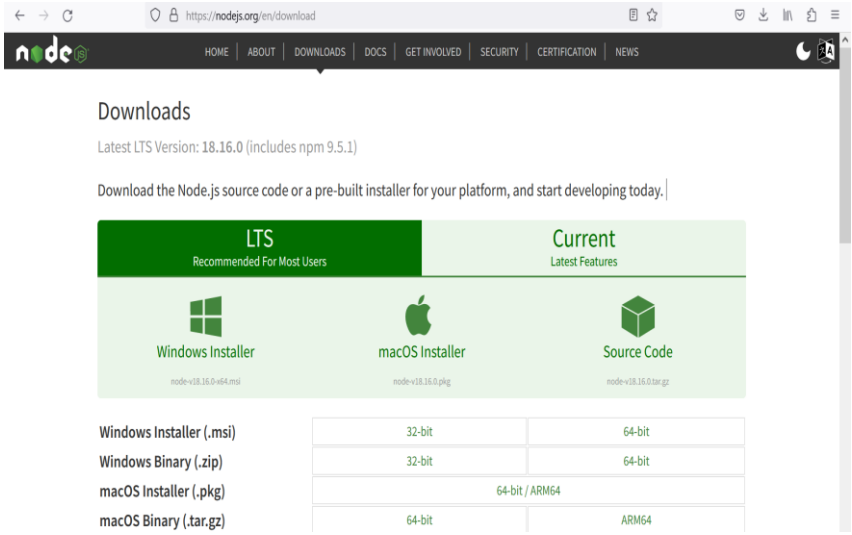
```
2023-03-10 15:19:37 Program Bangkit Kampus Merdeka mewadahi mahasiswa mengem
bangkan keahlian di bidang teknologi. Berikut cerita Fatiha… https://t.co/Su
HscydwZl
2023-03-10 15:03:45 RT @collegemenfess: [CM] Guys kalian daftar magang selai
n dari program Kampus Merdeka sama LinkedIn darimana lagi deh? Apa coba-coba
 email…
2023-03-10 13:44:07 @collegemenfess Gw bener bener paham gara" dulu ikut pro
gram kampus merdeka nder
2023-03-10 13:12:40 Rapat Koordinasi Lanjutan Tim Task Force Program Kompeti
si Kampus Merdeka (PK-KM) Jurusan Manajemen FE UBB dalam ra… https://t.co/2W
QC4nSTGj
2023-03-10 13:06:37 @Askrlfess cari magang yg udh kerjasama kampus merdeka a
ja
2023-03-10 10:48:37 @collegemenfess Sebenrnya gapapa ikut program kampus mer
deka tapi kek dibagi gitu jgn kampus merdeka aja. Apalagi r… https://t.co/kk
kCRcaHBi
2023-03-10 10:47:36 @collegemenfess Pokoknya organisasi kampus, kalau aku se
ndiri suka yg dilingkup fakultas soalnya sekalian belajar h… https://t.co/Jj
TaXxD4U9
2023-03-10 10:42:45 @likeafenol @collegemenfess Iya kemungkinan gini Nder, k
eknya dr kampus merdeka, dia ngeclaim2 gitu
2023-03-10 08:46:46 sebagai Kampus Merdeka berkarakter mulia dan Universitas
 Berkelas Dunia
```

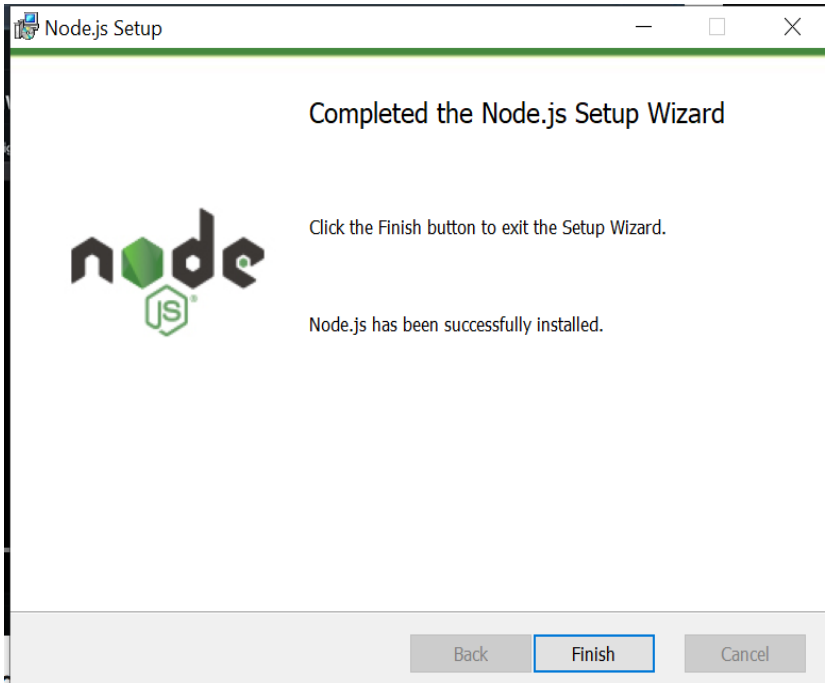
Gambar 4.9 Sample Data Crawling Pada Terminal

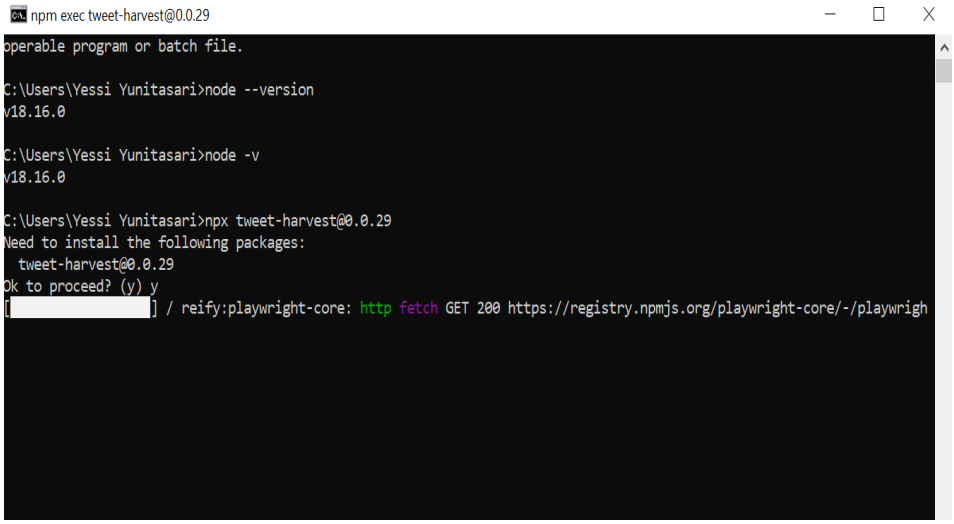
E. Cara Crawling Data Dengan Node Js

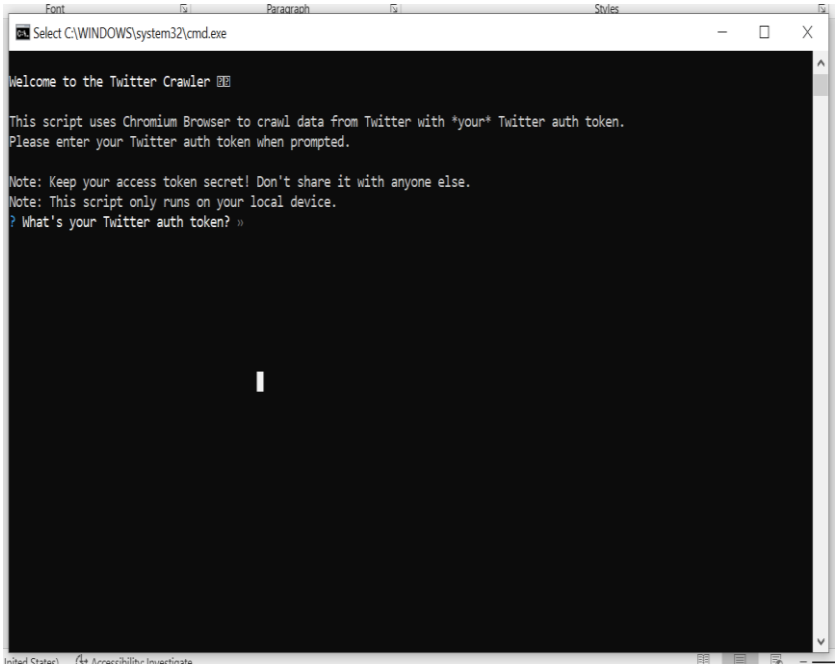
Cara lebih simple dalam crawling data dapat dilakukan dengan mengikuti Langkah dibawah ini.

Pertama kita harus download node Js. Kita buka link https://nodejs.org/en/download. Pilih OS yang kita gunakan.


https://nodejs.org/en/download
HOME
ABOUT
DOWNLOADS
DOCS
GET INVOLVED
SECURITY
CERTIFICATION
NEWS
Downloads
Latest LTS Version: 18.16.0 (includes npm 9.5.1)
Download the Node.js source code or a pre-built installer for your platform, and start developing today.
LTS
Recommended For Most Users
Current
Latest Features
Windows Installer
macOS Installer
Source Code
Windows Installer (.msi)
32-bit
64-bit
Windows Binary (.zip)
32-bit
64-bit
macOS Installer (.pkg)
64-bit / ARM64
macOS Binary (.tar.gz)
64-bit
ARM64

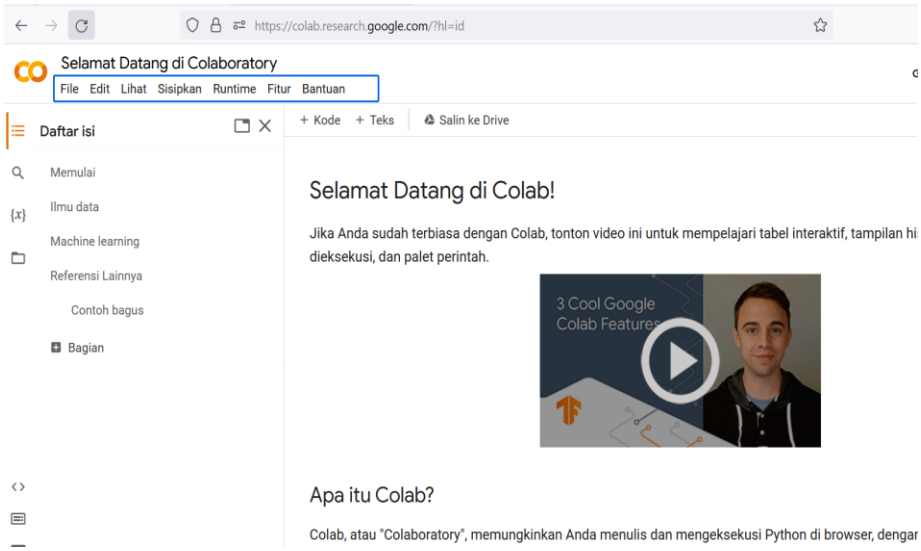

Node.js Setup
Completed the Node.js Setup Wizard
Click the Finish button to exit the Setup Wizard.
Node.js has been successfully installed.
Back
Finish
Cancel





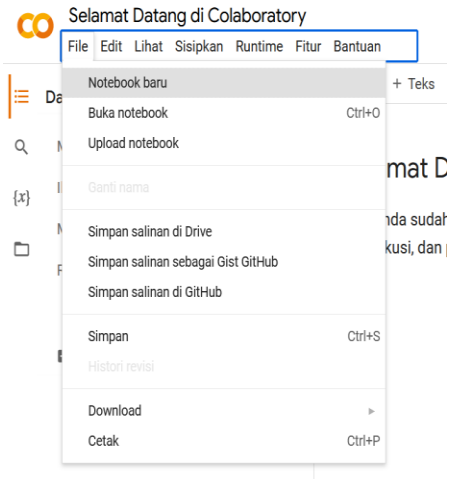
F. Cara crawling yang diaplikasikan di Google Colabs

Pertama kita harus masuk ke halaman google colabs. Berikut link alamatnya : https://colab.research.google.com/?hl=id. Akan muncul halaman sebagai berikut:



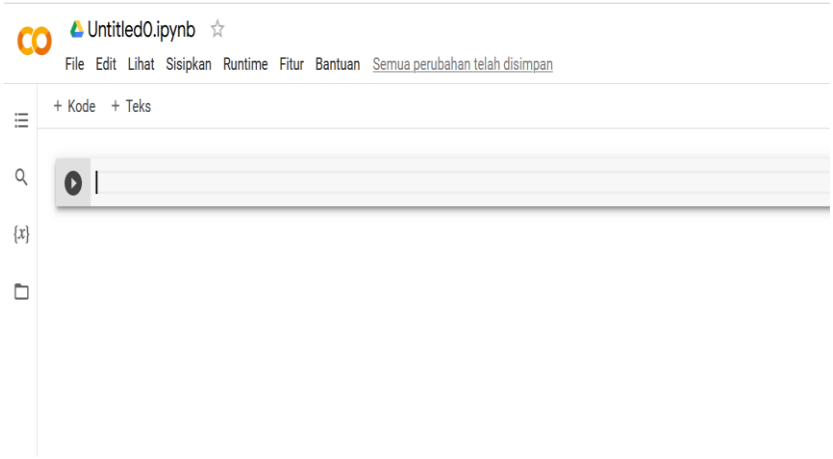
Gambar . Halaman awal google colab

Kemudian pilih File , Nootbook baru seperti gambar dibawah ini.



Gambar. Membuka notebook baru

Apabila sudah tampil halaman pada gambar dibawah ini. Kita dapat mulai menuliskan code untuk crawling data.



Pertama kita harus menginstal library pandas. Perintah yang dapat kita lakukan adalah pip install pandas. Setelah itu kita dapat melakukan run code dengan menekan tombol disebelah pojok kiri. Tuliskan code dibawah ini untuk setting berapa jumlah tweet yang akan diambil dan tanggal tweet tersebut diambil.

```
import os
import datetime

# Batasi jumlah hasil yang diambil
max_results = 1000

# Gunakan Twitter search untuk mencari tweet yang di-favoritkan minimal 10000 kali dan berbahasa Indonesia
twitter_search = "phising lang:id since:2023-01-01 until:2023-06-13"

# Tentukan nama file dengan format "<kueri pencarian>_<tanggal saat ini>.json"
filename = f"{twitter_search.replace(' ', '_').replace(':', '-').replace('#', '')}_{datetime.date.today().strftime('%Y-%m-%d')}.json"

USING_TOP_SEARCH = False

snscrape_params = '--jsonl --max-results'
twitter_search_params = ''
```

```
if USING_TOP_SEARCH:
    twitter_search_params += "--top"

snscrape_search_query = f"snscrape {snscrape_params} {max_results}
twitter-search {twitter_search_params} '{twitter_search}' > {filename}"

print(snscrape_search_query)

os.system(snscrape_search_query)
```

Setelah code diatas dapat di Run. Lanjutkan dengan mengetikkan code dibawah ini.

```
import pandas as pd
import ast
import json

# Membaca file JSON hasil dari perintah CLI sebelumnya dan membuat
dataframe pandas
tweets_df = pd.read_json(filename, lines=True)

NAMA_FILE_CSV = 'phising.csv'

# Membuat kamus untuk mengganti nama kolom
new_columns = {
    'conversationId': 'Conv. ID',
    'url': 'URL',
    'date': 'Date',
    'rawContent': 'Tweet',
    'id': 'ID',
    'replyCount': 'Replies',
    'retweetCount': 'Retweets',
    'likeCount': 'Likes',
    'quoteCount': 'Quotes',
    'bookmarkCount': 'Bookmarks',
    'lang': 'Language',
    'links': 'Links',
    'media': 'Media',
    'retweetedTweet': 'Retweeted Tweet',
    'username': 'Username'
```

```
}

if len(tweets_df) == 0:
   print('Pencarian tidak ditemukan coba ganti keyword lain, keywordmu: ',
twitter_search)
   exit()
else:
 # Memilih kolom yang akan digunakan dan mengganti nama kolom
menggunakan kamus yang telah dibuat
 tweets_df = tweets_df.loc[:, ['url', 'date', 'rawContent', 'id',
                       'replyCount', 'retweetCount', 'likeCount', 'quoteCount',
                       'conversationId', 'lang', 'links',
                       'media', 'retweetedTweet', 'bookmarkCount', 'username']]
 tweets_df = tweets_df.rename(columns=new_columns)

 # Ekstrak fullUrl dari kolom media dan url dari kolom links
 tweets_df['Media'] = tweets_df['Media'].apply(lambda x: x[0]['fullUrl'] if
isinstance(x, list) and x and isinstance(x[0], dict) and 'fullUrl' in x[0] else
None)
 tweets_df['Links'] = tweets_df['Links'].apply(lambda x: x[0]['url'] if
isinstance(x, list) and x and isinstance(x[0], dict) and 'url' in x[0] else None)

 # Menampilkan dataframe tweets_df
 display(tweets_df)

 # Simpan ke csv
 tweets_df.to_csv(NAMA_FILE_CSV, index=False)
```

Hasil dari run code akan mengcrawling data sebanyak 1000 data serta dari crawling tersebut akan disimpan dalam bentuk csv dengan nama file yang dapat kita sesuiakan.

G. **Crawling data pada Playstore**

Proses crawling data dari Google Play Store, kita dapat menggunakan Python bersama dengan beberapa pustaka seperti google-play-scraper untuk mengakses dan mengambil data dari toko aplikasi Google Play. Berikut adalah contoh sederhana bagaimana Anda dapat melakukannya:

1. Install pustaka google-play-scraper menggunakan pip jika Anda belum melakukannya :

```
pip install google-play-scraper
```

2. Berikut adalah contoh kode untuk melakukan crawling data dari Google Play Store menggunakan pustaka tersebut :

```
from google_play_scraper import app

# Contoh penggunaan: mendapatkan data aplikasi WhatsApp
app_info = app('com.whatsapp')

# Menampilkan informasi aplikasi
print("Nama Aplikasi:", app_info['title'])
print("Rating:", app_info['score'])
print("Jumlah Ulasan:", app_info['reviews'])
print("Jumlah Pengunduhan:", app_info['installs'])
print("Perbarui Terakhir:", app_info['updated'])
print("Kategori:", app_info['genre'])
print("Developer:", app_info['developer'])
print("Ikon Aplikasi:", app_info['icon'])
```

print("Nama Aplikasi:", app_info['title'])digunakan untuk menampilkan nama aplikasi yang ingin kita cari informasi.
print("Rating:", app_info['score']) digunakan untuk menampilkan rating aplikasi.
print("Jumlah Ulasan:", app_info['reviews']) digunakan untuk menampilkan jumlah ulasan.
print("Jumlah Pengunduhan:", app_info['installs']) digunakan untuk menampilkan jumlah pengunduhan.

print("Perbarui Terakhir:", app_info['updated'])digunakan untuk informasi pengupdaten aplikasi.
print("Kategori:", app_info['genre']) digunakan untuk informasi kategori aplikasi.
print("Developer:", app_info['developer']) digunakan untuk menampilkan informasi developer.
print("Ikon Aplikasi:", app_info['icon'])digunakan untuk mendapatkan ikon aplikasi.

Hasil running dari source code diatas dapat dilihat pada gambar dibawah ini.

```
Nama Aplikasi: WhatsApp Messenger
Rating: 4.3125806
Jumlah Ulasan: 1906403
Jumlah Pengunduhan: 5,000,000,000+
Perbarui Terakhir: 1693536680
Kategori: Communication
Developer: WhatsApp LLC
Ikon Aplikasi: https://play-lh.googleusercontent.com/bYtqbOcTYOlgc6gqZ2rwb8lptHuwlNE75zYJu6Bn076-hTmvd96HH-6v7S0YUAAJXoJN
```

H. **Latihan**

1. Apakah manfaat dari crawling?
2. Butlah code untuk crawling data dengan ketentuan .
    a. Data yang di crawling terkait objek wisata di kota madiun.
    b. Banyak data 500 data.
    c. Data yang diambil dari bulan januari – juni 2023.
3. Bagaimanakah cara menggunakan filter lokasi untuk crawling?
4. Tuliskan code yang digunakan ketika kita ingin meyimpan file data hasil crawling dalam bentuk excel dan csv.
5. Apakah fungsi dari Stream API?
6. Menurut anda library apa saja yang diperlukan untuk mengambil data dari sosial media?
7. Mengapa kita memerlukan akses token sebelum crawling data?
8. Apabila kita ingin melakukan crawling data terkait opini mahasiswa penerima beasiswa sekolah di luar negeri, kira-kira hastag apa yang dapat kita gunakan?

BAB V
KONSEP PREPROCESSING DATA

Preprocessing atau pemrosesan awal data teks merupakan langkah penting dalam pemrosesan bahasa alami (NLP) dan analisis teks. Pemrosesan awal data teks melibatkan pembersihan data teks mentah dan mengubahnya menjadi format yang sesuai untuk analisis dan pembelajaran mesin. Berikut adalah beberapa teknik prapemrosesan data teks yang umum:

1. Pembersihan Teks:

   Huruf kecil: Ubah semua teks menjadi huruf kecil untuk memastikan konsistensi (misalnya, "Teks" dan "teks" menjadi sama).

   Hapus Tanda Baca: Hapus tanda baca seperti titik, koma, dan tanda kutip.

   Hapus Angka: Hilangkan angka numerik atau ganti dengan token khusus jika diperlukan.

   Hapus Karakter Khusus: Hapus karakter atau simbol non-alfanumerik yang mungkin tidak relevan dengan analisis.
2. Tokenisasi:

   Pisahkan Teks menjadi Token: Tokenisasi memecah teks menjadi kata atau frasa individual (token). Perpustakaan seperti NLTK atau spaCy dapat membantu dalam hal ini.
3. Stopword: Hapus kata-kata umum (stopwords) seperti "dan", "the", "in", yang mungkin tidak memberikan kontribusi banyak pada analisis. NLTK atau spaCy juga dapat membantu dalam hal ini.

4. Stemming dan Lemmatisasi:

   Stemming dan Lemmitisasi : Mengurangi kata-kata ke bentuk akarnya untuk menormalkan teks dan mengubahnya menjadi makna dasar.

   Walaupun memiliki fungsi yang sama namun terdapat perbedaan dalam implementasi stemming dan lemmatization. Pada stemming pengubahan hanya dilakukan dengan memotong/ menghapus imbuhan kata. Sedangkan lemmatization memiliki alur yang lebih kompleks dengan melibatkan kamus bahasa untuk mencari kata dasar (root).
5. Menangani Data yang Hilang:

   Menangani nilai yang hilang dalam data teks, baik dengan menghapus baris dengan teks yang hilang atau memasukkan nilai yang hilang.
6. Menangani Masalah Pengkodean: Mengatasi masalah pengkodean karakter, terutama ketika berhadapan dengan data teks multibahasa.
7. Rekayasa Fitur: Membuat fitur atau representasi baru dari data teks, seperti TF-IDF (Term Frekuensi-Invers Dokumen Frekuensi) atau embeddings kata (misalnya, Word2Vec, GloVe).
8. Menghapus Tag HTML: Jika bekerja dengan data web, hapus tag HTML dan ekstrak konten teks biasa.
9. Menghapus URL dan Alamat Email: Hapus URL, alamat email, atau pola spesifik lainnya yang tidak relevan dengan analisis.
10. Pemeriksaan dan Koreksi Ejaan: Secara opsional, lakukan pemeriksaan ejaan dan koreksi untuk membakukan teks.
11. Normalisasi Teks: Normalisasikan teks dengan mengganti sinonim, singkatan, atau akronim dengan bentuk lengkapnya untuk konsistensi.

12. Menangani Duplikat: Hapus entri teks duplikat jika ada di kumpulan data.
13. Penyesuaian Panjang Teks: Pad atau potong teks ke panjang tetap, jika perlu, untuk model NLP tertentu.
14. Pemisahan Data: Pisahkan data yang telah diproses sebelumnya menjadi set pelatihan, validasi, dan pengujian untuk tugas pembelajaran mesin.

Langkah-langkah pra-pemrosesan spesifik yang perlu Anda lakukan mungkin berbeda-beda bergantung pada data Anda dan tugas yang ada. Penting untuk menyesuaikan alur prapemrosesan dengan persyaratan proyek Anda.

Selain itu, Anda mungkin perlu menggunakan pustaka seperti NLTK Python, spaCy, scikit-learn, atau pustaka NLP khusus seperti Hugging Face Transformers, bergantung pada kebutuhan dan keahlian Anda.

A. Membuka Data

Proses preprocessing diawali dengan membuka file data kita. File data yang berisikan komentar harus kita buka terlebih dahulu. Berikut ini adalah contoh kode program untuk membuka file data dengan bentuk CSV.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.feature_extraction.text import CountVectorizer
import nltk
import string
import re
from Sastrawi.StopWordRemover.StopWordRemoverFactory import StopWordRemoverFactory
```

```
df = pd.read_csv('datatry5.csv', encoding = 'unicode_escape')
df.head()
```

Hasil dari kode progam diatas akan menampilkan file data csv yang berisi komentar/ sentiment terhadap suatu topik tertentu. Beserta label sentiment yang diberikan. Judul file csv yang digunakan adalah “datatry5”.

Out[54]:

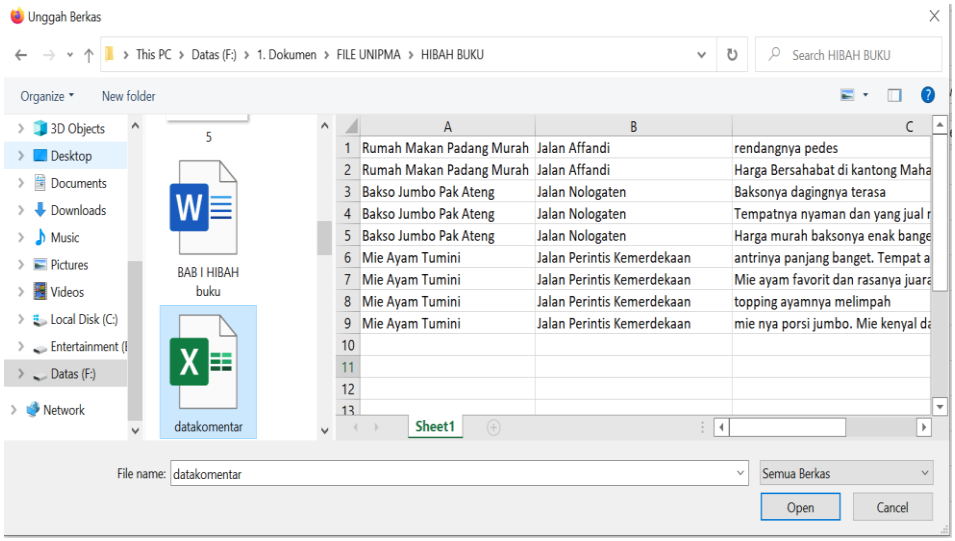
	Tanggal	Isi	Sentimen
0	12/16/2021 4:10	TEMU ALUMNI DAN SEMINAR PENDIDIKAN FTK UIN MAT...	0
1	12/16/2021 4:03	ya mayan sih suruh ikut kkn mbkm ribet bgt 6 b...	1
2	12/16/2021 1:41	Alur Pendaftaran MBKM Fakultas Ilmu Budaya Uni...	0
3	12/16/2021 0:19	kajur ruwet tim mbkm ruwet semua ruwet	-1
4	12/15/2021 14:09	tp jujur font yg laporan mbkm tuh enak bgt di ...	1

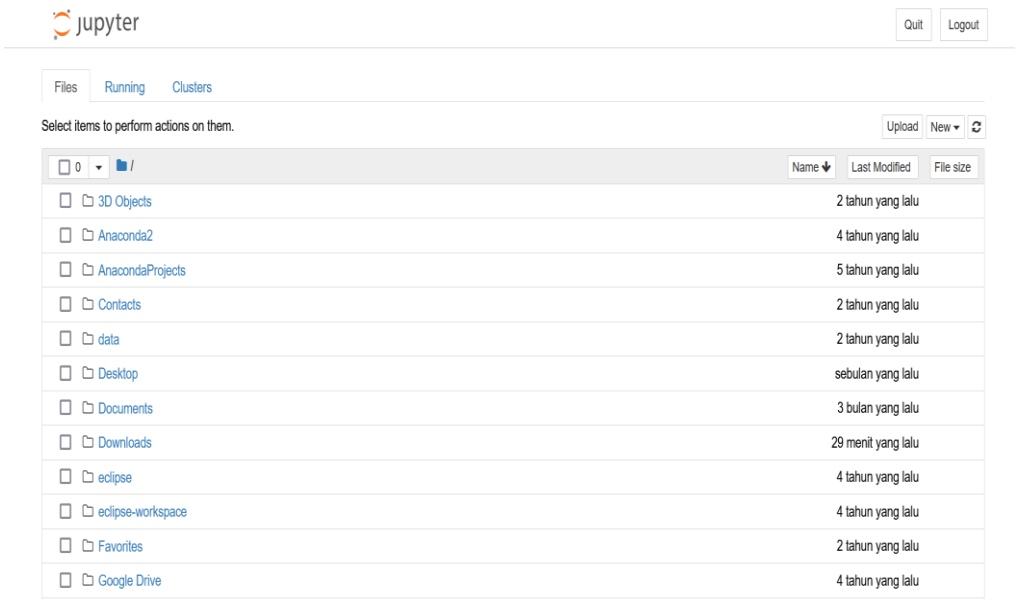
Hasil output diatas menampilkan sentiment ke 0 sampai data ke 4. Tanggal yang tertera adalah tanggal penuliasan komentar yang dibuat. Sedangkan kode 0, 1, -1 yang ada pada tabel sentimen disebut sebagai label. 0 menandakan komentar tersebut bersifat netral, 1 menandakan label komentar tersebut bersifat positif dan label komentar bersifat negative ditandai dengan -1.

Library openpyxl digunakan untuk membaca file Excel. Syntax yang dapat kita gunakan adalah from openpyxl import load_workbook**.** Berikut ini adalah contoh kode program untuk membuka data dari file excel.

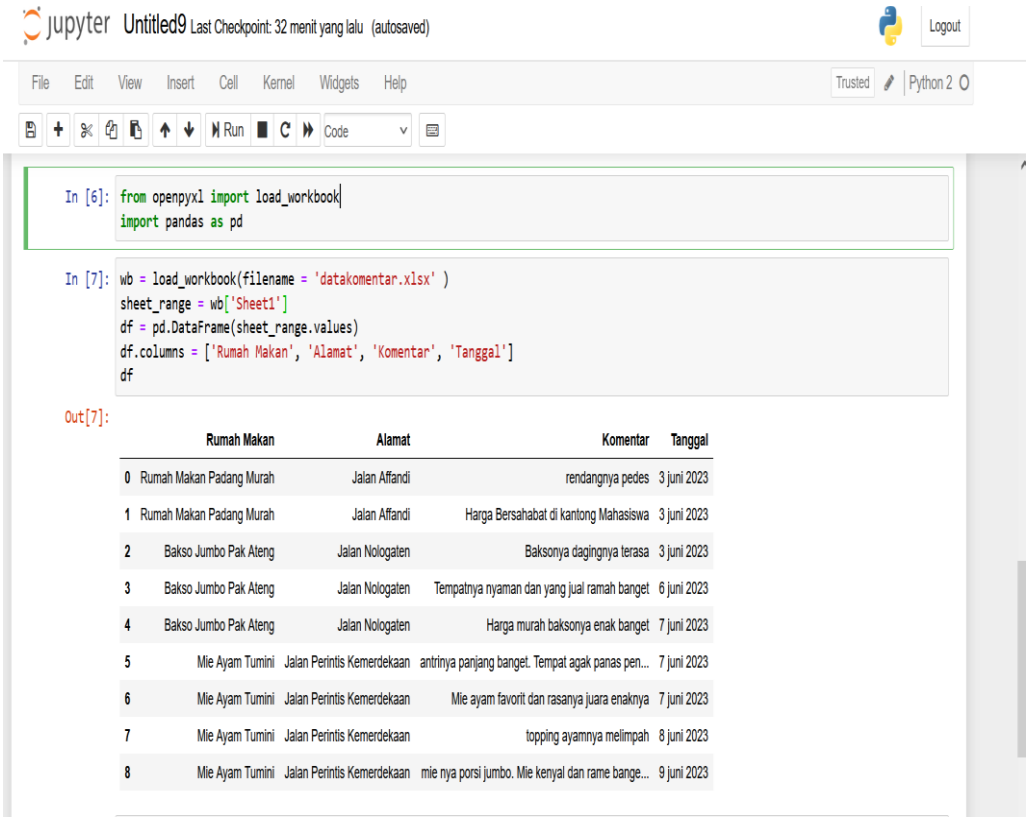
import pandas as pd

```
In [6]: from openpyxl import load_workbook
        import pandas as pd
```






```
wb = load_workbook(filename = 'datakomentar.xlsx' )
sheet_range = wb['Sheet1']
df = pd.DataFrame(sheet_range.values)
df.columns = ['Rumah Makan', 'Alamat', 'Komentar', 'Tanggal']
df
```



	Rumah Makan	Alamat	Komentar	Tanggal
0	Rumah Makan Padang Murah	Jalan Affandi	rendangnya pedes	3 juni 2023
1	Rumah Makan Padang Murah	Jalan Affandi	Harga Bersahabat di kantong Mahasiswa	3 juni 2023
2	Bakso Jumbo Pak Ateng	Jalan Nologaten	Baksonya dagingnya terasa	3 juni 2023
3	Bakso Jumbo Pak Ateng	Jalan Nologaten	Tempatnya nyaman dan yang jual ramah banget	6 juni 2023
4	Bakso Jumbo Pak Ateng	Jalan Nologaten	Harga murah baksonya enak banget	7 juni 2023
5	Mie Ayam Tumini	Jalan Perintis Kemerdekaan	antrinya panjang banget. Tempat agak panas pen...	7 juni 2023
6	Mie Ayam Tumini	Jalan Perintis Kemerdekaan	Mie ayam favorit dan rasanya juara enaknya	7 juni 2023
7	Mie Ayam Tumini	Jalan Perintis Kemerdekaan	topping ayamnya melimpah	8 juni 2023
8	Mie Ayam Tumini	Jalan Perintis Kemerdekaan	mie nya porsi jumbo. Mie kenyal dan rame bange...	9 juni 2023

```
In [6]: from openpyxl import load_workbook
        import pandas as pd
```

```
In [7]: wb = load_workbook(filename = 'datakomentar.xlsx' )
        sheet_range = wb['Sheet1']
        df = pd.DataFrame(sheet_range.values)
        df.columns = ['Rumah Makan', 'Alamat', 'Komentar', 'Tanggal']
        df
```

Out[7]:

	Rumah Makan	Alamat	Komentar	Tanggal
0	Rumah Makan Padang Murah	Jalan Affandi	rendangnya pedes	3 juni 2023
1	Rumah Makan Padang Murah	Jalan Affandi	Harga Bersahabat di kantong Mahasiswa	3 juni 2023
2	Bakso Jumbo Pak Ateng	Jalan Nologaten	Baksonya dagingnya terasa	3 juni 2023
3	Bakso Jumbo Pak Ateng	Jalan Nologaten	Tempatnya nyaman dan yang jual ramah banget	6 juni 2023
4	Bakso Jumbo Pak Ateng	Jalan Nologaten	Harga murah baksonya enak banget	7 juni 2023
5	Mie Ayam Tumini	Jalan Perintis Kemerdekaan	antrinya panjang banget. Tempat agak panas pen...	7 juni 2023
6	Mie Ayam Tumini	Jalan Perintis Kemerdekaan	Mie ayam favorit dan rasanya juara enaknya	7 juni 2023
7	Mie Ayam Tumini	Jalan Perintis Kemerdekaan	topping ayamnya melimpah	8 juni 2023
8	Mie Ayam Tumini	Jalan Perintis Kemerdekaan	mie nya porsi jumbo. Mie kenyal dan rame bange...	9 juni 2023

df ['Komentar']

```
In [8]: df ['Komentar']
```

```
Out[8]: 0                                  rendangnya pedes
        1                Harga Bersahabat di kantong Mahasiswa
        2                            Baksonya dagingnya terasa
        3          Tempatnya nyaman dan yang jual ramah banget
        4                      Harga murah baksonya enak banget
        5     antrinya panjang banget. Tempat agak panas pen...
        6            Mie ayam favorit dan rasanya juara enaknya
        7                              topping ayamnya melimpah
        8     mie nya porsi jumbo. Mie kenyal dan rame bange...
        Name: Komentar, dtype: object
```

df [['Komentar']]

```
In [9]: df [['Komentar']]
```

Out[9]:

	Komentar
0	rendangnya pedes
1	Harga Bersahabat di kantong Mahasiswa
2	Baksonya dagingnya terasa
3	Tempatnya nyaman dan yang jual ramah banget
4	Harga murah baksonya enak banget
5	antrinya panjang banget. Tempat agak panas pen...
6	Mie ayam favorit dan rasanya juara enaknya
7	topping ayamnya melimpah
8	mie nya porsi jumbo. Mie kenyal dan rame bange...

```
In [10]: df [:5]
```

Out[10]:

	Rumah Makan	Alamat	Komentar	Tanggal
0	Rumah Makan Padang Murah	Jalan Affandi	rendangnya pedes	3 juni 2023
1	Rumah Makan Padang Murah	Jalan Affandi	Harga Bersahabat di kantong Mahasiswa	3 juni 2023
2	Bakso Jumbo Pak Ateng	Jalan Nologaten	Baksonya dagingnya terasa	3 juni 2023
3	Bakso Jumbo Pak Ateng	Jalan Nologaten	Tempatnya nyaman dan yang jual ramah banget	6 juni 2023
4	Bakso Jumbo Pak Ateng	Jalan Nologaten	Harga murah baksonya enak banget	7 juni 2023

df.iloc[2]

```
In [5]: df.iloc[2]

Out[5]: Rumah Makan        Bakso Jumbo Pak Ateng
        Alamat                   Jalan Nologaten
        Komentar       Baksonya dagingnya terasa
        Tanggal                      3 juni 2023
        Name: 2, dtype: object
```

huruf_kecil = ("TEMPAT MAKAN DENGAN PEMANDANGAN ALAM YANG INDAH").lower()

huruf_kapital = ("Tempat Makan dengan Pemandangan Alam yang Indah").upper()

print huruf_kecil

print huruf_kapital

```
huruf_kecil = ("TEMPAT MAKAN DENGAN PEMANDANGAN ALAM YANG INDAH").lower()
huruf_kapital = ("Tempat Makan dengan Pemandangan Alam yang Indah").upper()
print huruf_kecil
print huruf_kapital
```

```
tempat makan dengan pemandangan alam yang indah
TEMPAT MAKAN DENGAN PEMANDANGAN ALAM YANG INDAH
```

B. Stopword Removal

Penerapan *stopword removal* tersebut bertujuan untuk menghilangkan noise terhadap kata yang akan di proses ke dalam tahap analisis.

```
daftar_stopword = ["dengan", "yang", "ke", "di", "dan"]
```

```
sentence = ("Resto di kota Klaten memiliki view yang indah, apalagi di saat senja")
words_tokenize = []
words = sentence.split()
for word in words :
    check = word in daftar_stopword
    if not check :
        words_tokenize.append(word)

print words_tokenize
```

```
['Resto', 'kota', 'Klaten', 'memiliki', 'view', 'indah,', 'apalagi', 'saat', 'senja']
```

```

```

sentence = ("Resto di kota Klaten memiliki view yang indah, apalagi di saat senja")

words_tokenize = []

words = sentence.split()

for word in words :

check = word in daftar_stopword

if not check :

words_tokenize.append(word)

print words_tokenize.

C. Penerapan stopword menggunakan corpus.

```
#stopword
stopword = nltk.corpus.stopwords.words('indonesian')

def remove_stopwords(text):
    text = [word for word in text if word not in stopword]
    return text

df ['STOP_REMOVAL'] = df['TOKENIZATION'].apply(lambda x: remove_stopwords(x))
df.head()
```

Hasil dari code program diatas adalah :

	Tanggal	Isi	Sentimen
0	12/16/2021 4:10	TEMU ALUMNI DAN SEMINAR PENDIDIKAN FTK UIN MAT...	0
1	12/16/2021 4:03	ya mayan sih suruh ikut kkn mbkm ribet bgt 6 b...	1
2	12/16/2021 1:41	Alur Pendaftaran MBKM Fakultas Ilmu Budaya Uni...	0
3	12/16/2021 0:19	kajur ruwet tim mbkm ruwet semua ruwet	-1
4	12/15/2021 14:09	tp jujur font yg laporan mbkm tuh enak bgt di ...	1

STOP_REMOVAL
[temu, alumni, seminar, pendidikan, ftk, uin, ...
[ya, mayan, sih, suruh, kkn, mbkm, ribet, bgt,...
[alur, pendaftaran, mbkm, fakultas, ilmu, buda...
[kajur, ruwet, tim, mbkm, ruwet, ruwet]
[tp, jujur, font, yg, laporan, mbkm, tuh, enak...

D. Case Folding

Case Folding digunakan untuk menyeeragamkan bentuk huruf ke lowercase. Berikut ini adalah kode program yang dapat digunakan untuk mengubah atau menyeragamkan sentiment ke bentuk lowercase.

```
# case folding
df['ISI'] =df['Isi'].str.lower()
df.head()
```

Hasil dari code program diatas adalah sebagai berikut :

	Tanggal	Isi	Sentimen	ISI
0	12/16/2021 4:10	TEMU ALUMNI DAN SEMINAR PENDIDIKAN FTK UIN MAT...	0	temu alumni dan seminar pendidikan ftk uin mat...
1	12/16/2021 4:03	ya mayan sih suruh ikut kkn mbkm ribet bgt 6 b...	1	ya mayan sih suruh ikut kkn mbkm ribet bgt 6 b...
2	12/16/2021 1:41	Alur Pendaftaran MBKM Fakultas Ilmu Budaya Uni...	0	alur pendaftaran mbkm fakultas ilmu budaya uni...
3	12/16/2021 0:19	kajur ruwet tim mbkm ruwet semua ruwet	-1	kajur ruwet tim mbkm ruwet semua ruwet
4	12/15/2021 14:09	tp jujur font yg laporan mbkm tuh enak bgt di ...	1	tp jujur font yg laporan mbkm tuh enak bgt di ...

E. Tokenize

Teknik pemecahan kalimat ke dalam kata menggunakan split.

sentence = ("Resto di pinggir danau memiliki view yang indah, apalagi di saat senja")

words_tokenize = []

words = sentence.split()

```
sentence = ("Resto di pinggir danau memiliki view yang indah, apalagi di saat senja")
words_tokenize = []
words = sentence.split()
```

```
print words
```

```
['Resto', 'di', 'pinggir', 'danau', 'memiliki', 'view', 'yang', 'indah,', 'apalagi', 'di', 'saat', 'senja']
```

F. Memecah komentar menjadi per-kalimat.

Split diatas melakukan pemecahan atau pemenggalan kalimat berdasarkan tanda baca “titik”.

sentence = ("Resto di malang memiliki fasilitas petik strawberry gratis. Resto juga memiliki menu yang variatif")

split = sentence.split(".")

print split

```
sentence = ("Resto di malang memiliki fasilitas petik strawberry gratis. Resto juga memiliki menu yang variatif")
split = sentence.split(".")
print split
```

```
['Resto di malang memiliki fasilitas petik strawberry gratis', ' Resto juga memiliki menu yang variatif']
```

G. Memecah komentar Menggunakan Regular Expresion.

Library re adalah class reguler expression yang umumnya digunakan untuk string matching. Sedangkan split digunakan untuk *spliting* kata dari hasil pencarian *regular expression* yang dihasilkan.

import re

sentence = ("Kota malang memiliki banyak resto yang unik, begitu juga kota Bandung.")

split = sentence.split(",")

print split

```
import re
```

```
sentence = ("Kota malang memiliki banyak resto yang unik, begitu juga kota Bandung.")
split = sentence.split(",")
print split
```

```
['Kota malang memiliki banyak resto yang unik', ' begitu juga kota Bandung.']
```

Penerapan Proses Cleansing dan Tokenize untuk data dari file CSV.

```
#Cleaning Text
def remove_punct(text):
    text = re.sub(r'[^a-zA-z0-9]', '', str(text))
    text = re.sub(r'\b\w(1,2)\b','', text)
    text = re.sub(r'\s\s+', '', text)

df['ISI'] = df['Isi'].apply(lambda x: remove_punct(x))
```

```
#Tokeninize
def tokenization(text):
    text = re.split('\W+', text)
    return text

df['TOKENIZATION'] = df['Isi'].apply(lambda x : tokenization(x.lower()))
df.head()
```

	Tanggal	Isi	Sentimen	ISI	TOKENIZATION
0	12/16/2021 4:10	TEMU ALUMNI DAN SEMINAR PENDIDIKAN FTK UIN MAT...	0	None	[temu, alumni, dan, seminar, pendidikan, ftk, ...
1	12/16/2021 4:03	ya mayan sih suruh ikut kkn mbkm ribet bgt 6 b...	1	None	[ya, mayan, sih, suruh, ikut, kkn, mbkm, ribet...
2	12/16/2021 1:41	Alur Pendaftaran MBKM Fakultas Ilmu Budaya Uni...	0	None	[alur, pendaftaran, mbkm, fakultas, ilmu, buda...
3	12/16/2021 0:19	kajur ruwet tim mbkm ruwet semua ruwet	-1	None	[kajur, ruwet, tim, mbkm, ruwet, semua, ruwet]
4	12/15/2021 14:09	tp jujur font yg laporan mbkm tuh enak bgt di ...	1	None	[tp, jujur, font, yg, laporan, mbkm, tuh, enak...

H. Stemming

Istilah *stemming* sering diartikan merubah kata berimbuhan menjadi kata dasar.

Stemming kata berbahasa Indonesia (Sastrawi Stemming)

1. Sastrawi Stemming

Untuk menginstall Sastrawi *stemming* dapat menggunakan pip dengan instruksi sebagai berikut: pip install Sastrawi.

```
pip install Sastrawi
```

```
Collecting Sastrawi
  Downloading Sastrawi-1.0.1-py2.py3-none-any.whl (209 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 209.7/209.7 kB 4.5 MB/s eta 0:00:00
Installing collected packages: Sastrawi
Successfully installed Sastrawi-1.0.1
```

2. Proses Stemming

Import library stemming sastrawi seperti berikut:

```
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
```

```
factory = StemmerFactory()
stemmer = factory.create_stemmer()
print ("mendatangi")
```

I. Import Hasil Preprocessing.

Setelah semua proses preprocessing dilakukan kita dapat mendowload hasil preprocessing menggunakan kode program di bawah ini :

```
import csv
df.to_csv('hasil_preprocessing4.csv')
```

Hasil preprocessing di beri nama file “Hasil_preprocessing4”. File hasil preprocessing di download dalam bentuk file csv.

Kode program untuk menampilkan hasil preprocesing tanpa harus mendownload dapat dilihat seperti contoh di bawah ini :

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

Hasil dari code program diatas adalah :

Tanggal	Isi	Sentimen	ISI	TOKENIZATION	STOPWORD
12/16/2021 4:10	TEMU ALUMNI DAN SEMINAR PENDIDIKAN FTK UIN MAT...	0	temu alumni dan seminar pendidikan ftk uin mat...	['temu', 'alumni', 'dan', 'seminar', 'pendidik...	['temu', 'alumni', 'seminar', 'pendidikan', 'f...
12/16/2021 4:03	ya mayan sih suruh ikut kkn mbkm ribet bgt 6 b...	1	ya mayan sih suruh ikut kkn mbkm ribet bgt 6 b...	['ya', 'mayan', 'sih', 'suruh', 'ikut', 'kkn',...	['ya', 'mayan', 'sih', 'suruh', 'kkn', 'mbkm',...
12/16/2021 1:41	Alur Pendaftaran MBKM Fakultas Ilmu Budaya Uni...	0	alur pendaftaran mbkm fakultas ilmu budaya uni...	['alur', 'pendaftaran', 'mbkm', 'fakultas', 'i...	['alur', 'pendaftaran', 'mbkm', 'fakultas', 'i...
12/16/2021 0:19	kajur ruwet, tim mbkm ruwet semua ruwet	-1	kajur ruwet, tim mbkm ruwet semua ruwet	['kajur', 'ruwet', 'tim', 'mbkm', 'ruwet', 'se...	['kajur', 'ruwet', 'tim', 'mbkm', 'ruwet', 'ru...
12/15/2021 14:09	tp jujur font yg laporan mbkm tuh enak bgt di ...	1	tp jujur font yg laporan mbkm tuh enak bgt di ...	['tp', 'jujur', 'font', 'yg', 'laporan', 'mbkm...	['tp', 'jujur', 'font', 'yg', 'laporan', 'mbkm...

J. Latihan

1. Apakah tujuan melakukan proses preprocessing?
2. Library apa yang kita butuhkan untuk proses preprocessing?
3. Jelaskan beberapa jenis preprocessing dan kapan kita harus menggunakannya?
4. Buat kumpulan stopword seperti dibawah ini !

```
daftar_stopword = ["jika","bila","dengan", "yang", "ke", "di", "dan"]
```

5. Cobalah buat sample beberapa komentar menggunakan stop word removal diatas !

BAB VI
EKSTRAKSI FITUR

Ekstrasi fitur adalah proses pengambilan informasi yang relevan atau karakteristik penting dari data mentah, seperti gambar, teks, suara, atau data lainnya. Tujuan dari ekstraksi fitur adalah untuk mengurangi dimensi data dan menggambarkan informasi yang paling relevan dalam bentuk yang lebih sederhana atau terstruktur. Fitur ini bermanfaat dalam berbagai aplikasi contohnya pemrosesan gambar, pengenalan pola, analisis teks, dan pembelajaran mesin. Berikut adalah beberapa teknik ekstraksi fitur yang umum digunakan dalam berbagai konteks:

1. **Ekstraksi Fitur Gambar:**
    - **Histogram Warna:** Menghitung sebaran warna dalam gambar.
    - **Pengenal tepi:** Mendeteksi tepi dan batas dalam gambar.
    - **Piramida Gabor:** Digunakan dalam pengenalan pola.
    - **Pengurangan Dimensi (PCA, LDA):** Mereduksi dimensi data gambar untuk menghilangkan noise dan menyoroti fitur utama.
2. **Ekstraksi Fitur Teks:**
    - **TF-IDF (Term Frequency-Inverse Document Frequency):** Mengukur pentingnya kata-kata dalam dokumen.
    - **Word Embeddings:** Mengubah kata-kata menjadi vektor dalam ruang fitur semantik.
    - **N-grams:** Membuat representasi teks dengan menggabungkan kata-kata berurutan.

3. **Ekstraksi Fitur Audio:**
    - **MFCC (Mel-Frequency Cepstral Coefficients):** Mengukur ciri akustik dalam sinyal suara.
    - **Chroma Feature:** Mengukur ciri nada dalam audio.
    - **Spectral Contrast:** Mengukukr kontras spektral dalam suara.
4. **Ekstraksi Fitur dalam Pembelajaran Mesin:**
    - **Principal Component Analysis (PCA):** Mengurangi dimensi dalam data.
    - **Linear Discriminant Analysis (LDA):** Meningkatkan pemisahan kelas dalam data.
    - **Autoencoders:** Jaringan saraf tiruan untuk ekstraksi fitur otomatis.
5. **Ekstraksi Fitur dalam Visi Komputer:**
    - **Deteksi Wajah:** Mengidentifikasi wajah dalam gambar.
    - **Deteksi Kantong Mata:** Mendeteksi mata dalam gambar.
    - **Deteksi tepi, sudut, dan garis:** Mengidentifikasi elemen-elemen visual penting.

Teknik ekstraksi fitur yang tepat tergantung pada jenis data dan masalah yang ingin dipecahkan. Setelah fitur-fitur relevan diekstraksi, data dapat digunakan untuk pelatihan model pembelajaran mesin, analisis data, dan berbagai aplikasi lainnya.

A. Ekstraksi Fitur Teks Menggunakan TF-IDF

Term Frequency atau TF ialah sebuah nilai frekuensi kemunculan token atau kata dalam sebuah dokumen. Misalnya terdapat 5 dokumen/kalimat seperti berikut.

Komentar
rendangnya pedes
Harga Bersahabat di kantong Mahasiswa
Baksonya dagingnya terasa
Tempatnya nyaman dan yang jual ramah banget
Harga murah baksonya enak banget

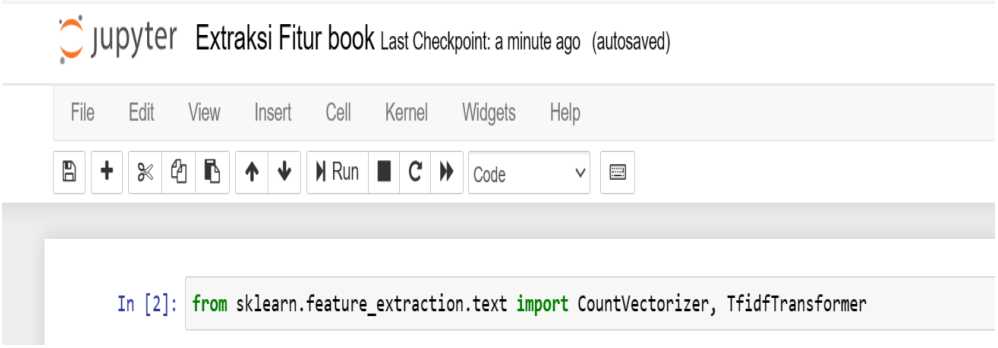
Untuk nilai *term frequency* dapat dilihat pada tabel dibawah ini:

No	Term	D1	D2	D3	D4	D5
1	Rendang	1	0	0	0	0
2	Pedas	1	0	0	0	0
3	Harga	0	1	0	0	1
4	Sahabat	0	1	0	0	0
5	Kantong	0	1	0	0	0
6	Mahasiswa	0	1	0	0	0
7	Bakso	0	0	1	0	1
8	Daging	0	0	1	0	0
9	Terasa	0	0	1	0	0
10	Tempat	0	0	0	1	0
11	Nyaman	0	0	0	1	0
12	Jual	0	0	0	1	0
13	Ramah	0	0	0	1	0
14	Murah	0	0	0	0	1
15	Enak	0	0	0	0	1

1. **Import library Sklearn.**

   Kode untuk melakukan import library sklearn :

   from sklearn.feature_extraction.text import CountVectorizer, TfidfTransformer



2. **Menuliskan list komentar yang akan digunakan**

   List komentar dapat dituliskan dengan cara :

   list_komentar = ['rendangnya pedes', 'Harga Bersahabat di kantong Mahasiswa','Baksonya dagingnya terasa','Tempatnya nyaman dan yang jual ramah banget','Harga murah baksonya enak banget']

3. **Menghitung jumlah term yang unik dalam list_komentar.**

   Baris kode program yang dapat digunakan untuk menghitung term yang dinilai unik yang terdapat dalam list_komentar adalah :

   CV = CountVectorizer ()

term_fit = CV.fit (list_komentar)

print len(term_fit.vocabulary_)

Hasil dari baris program diatas adalah :

```
In [5]: CV = CountVectorizer ()
        term_fit = CV.fit (list_komentar)
        print len(term_fit.vocabulary_)

        19
```

Jumlah term yang unik dalam list_komentar ada 19 term.

4. **Menampilkan term frequency**

   Baris kode yang digunakan untuk term frequency secara keseluruhan adalah sebagai berikut :

   term_frequency_all= term_fit.transform(list_komentar)

   print term_frequency_all

   Transform() digunakan untuk *Transform vocabulary* documents kedalam matrik *term frequency*

   Hasil dari baris program diatas adalah :

```
In [9]: term_frequency_all = term_fit.transform(list_komentar)

In [10]: print term_frequency_all
```

```
  (0, 13)       1
  (0, 15)       1
  (1, 2)        1
  (1, 5)        1
  (1, 7)        1
  (1, 9)        1
  (1, 10)       1
  (2, 0)        1
  (2, 3)        1
  (2, 17)       1
  (3, 1)        1
  (3, 4)        1
  (3, 8)        1
  (3, 12)       1
  (3, 14)       1
  (3, 16)       1
  (3, 18)       1
  (4, 0)        1
  (4, 1)        1
  (4, 6)        1
  (4, 7)        1
  (4, 11)       1
```

5. **Menampilkan TF untuk dokumen atau kalimat tertentu.**

Jika kita perlu menampilkan term frequency untuk sebuah dokumen tertentu, kita dapat menggunakan baris kode berikut:

komentar_tf = list_komentar[0]

print komentar_tf

Baris kode kode diatas digunakan ketika kia ingin menampilkan isi komentar pada list indek ke 0. Hasil output baris kode diatas adalah sebagai berikut :

```
In [13]: komentar_tf = list_komentar[0]

In [14]: print komentar_tf

         rendangnya pedes
```

Hasil dari output menampilkan isi komentar pada list indek ke 1 – indeks ke 4 adalah sebagai berikut :

```
In [15]: komentar_tf = list_komentar[1]

In [16]: print komentar_tf

         Harga Bersahabat di kantong Mahasiswa

In [17]: komentar_tf = list_komentar[2]

In [18]: print komentar_tf

         Baksonya dagingnya terasa

In [19]: komentar_tf = list_komentar[3]

In [20]: print komentar_tf

         Tempatnya nyaman dan yang jual ramah banget

In [21]: komentar_tf = list_komentar[4]

In [22]: print komentar_tf

         Harga murah baksonya enak banget
```

Selanjutnya kita perlu untuk memperlihatkan *term frequency* pada dokumen yang disimpan dalam variable komentar_tf.

term_frequency = term_fit.transform ([komentar_tf])

print term_frequency

Hasil dari baris code diatas adalah :

```
In [11]: term_frequency = term_fit.transform ([komentar_tf])
         print term_frequency

           (0, 13)       1
           (0, 15)       1
```

Menampilkan *term* atau *feature* tertentu berdasarkan indeks *term vocabulary*. Komentar yang digunakan adalah “Bakso dagingnya terasa”.

```
In [17]: komentar_tf = list_komentar[2]

In [18]: print komentar_tf

         Baksonya dagingnya terasa
```

print term_fit.get_feature_names () [0]

Hasil dari indeks *term vocabulary* ke [0] dari komentar “Bakso dagingnya terasa” adalah term “Baksonya”. Hasil dari running baris kode diatas adalah sebagai berikut :

```
In [15]: print term_fit.get_feature_names () [0]

         baksonya
```

Langkah berikutnya adalah menghitung nilai dari TF-IDF untuk dokumen atau kalimat tertentu.

komentar_term = term_fit.transform (list_komentar)

tfidf_transformer = TfidfTransformer ().fit(komentar_term)

tfidf = tfidf_transformer.transform (term_frequency)

print tfidf

Hasil dari baris code diatas adalah :

```
In [17]: komentar_term =  term_fit.transform (list_komentar)
         tfidf_transformer = TfidfTransformer ().fit(komentar_term)
         tfidf = tfidf_transformer.transform (term_frequency)
         print tfidf

  (0, 15)       0.7071067811865475
  (0, 13)       0.7071067811865475
```

TfidfTransformer() berguna untuk *Transform vocabulary* documents kedalam matrik *term frequency* atau TFIDF.

Setelah melakukan Transform vocabulary documents kedalam matrik term frequency langkah selanjunya menghitung nilai IDF untuk dokumen atau kalimat tertentu.

print tfidf_transformer.idf_

Idf_ digunakan untuk menampilkan atribut IDF dengan hasil seperti dibawah ini :

```
In [18]: print tfidf_transformer.idf_

[1.69314718 1.69314718 2.09861229 2.09861229 2.09861229 2.09861229
 2.09861229 1.69314718 2.09861229 2.09861229 2.09861229 2.09861229
 2.09861229 2.09861229 2.09861229 2.09861229 2.09861229 2.09861229
 2.09861229]
```

Ketika kita membutuhkan perhitungan nilai IDF untuk sebuah kalimat atau dokumen tertentu kita dapat menggunakan baris kode program seperti ini :

print tfidf_transformer.idf_[term_fit.vocabulary_['baksonya']]

Hasil nilai idf dari term ‘bakso’ adalah sepert dibawah ini :

```
In [23]: print tfidf_transformer.idf_[term_fit.vocabulary_['baksonya']]

1.6931471805599454
```

Apabila kita ingin menghitung nilai TF-IDF dari semua data yang kita miliki adalah dengan menuliskan baris kode dibawah ini :

komentar_tfidf = tfidf_transformer.transform (komentar_term)

print komentar_tfidf

Nilai TF-IDF dari semua dokumen seperti dibawah ini.

```
In [25]: komentar_tfidf = tfidf_transformer.transform (komentar_term)
         print komentar_tfidf

  (0, 15)       0.7071067811865475
  (0, 13)       0.7071067811865475
  (1, 10)       0.4636932227319092
  (1, 9)        0.4636932227319092
  (1, 7)        0.3741047724501572
  (1, 5)        0.4636932227319092
  (1, 2)        0.4636932227319092
  (2, 17)       0.6141889663426562
  (2, 3)        0.6141889663426562
  (2, 0)        0.49552379079705033
  (3, 18)       0.38775666010579296
  (3, 16)       0.38775666010579296
  (3, 14)       0.38775666010579296
  (3, 12)       0.38775666010579296
  (3, 8)        0.38775666010579296
  (3, 4)        0.38775666010579296
  (3, 1)        0.3128396318588854
  (4, 11)       0.5029796600534002
  (4, 7)        0.40580082271361156
  (4, 6)        0.5029796600534002
  (4, 1)        0.40580082271361156
  (4, 0)        0.40580082271361156
```

B. Ekstraksi Fitur Teks Menggunakan N-Gram

Selain menggunakan TF-IDF, unuk proses ekstraksi fitur kita juga dapat meggunkan N-Gram. Sebelumnya kita harus *import library* NLTK yang akan digunakan.

1. Import Library Nltk terlebih dahulu.

```
[28] from nltk import ngrams
```

```
[29] from nltk.util import ngrams
```

```
[30] import nltk
```

N-gram karakter/huruf

1. Menggunakan *Library* NLTK dengan fungsi tokenize.

```
from nltk.corpus.reader.tagged import word_tokenize

text = "Kota malang memiliki banyak resto yang unik, begitu juga kota Bandung."
tokenize = word_tokenize(text)
n = 2
bigrams = ngrams (tokenize,n)
for gram in bigrams:
  print (gram)
```

```
('Kota', 'malang')
('malang', 'memiliki')
('memiliki', 'banyak')
('banyak', 'resto')
('resto', 'yang')
('yang', 'unik')
('unik', ',')
(',', 'begitu')
('begitu', 'juga')
('juga', 'kota')
('kota', 'Bandung')
('Bandung', '.')
```

2. N-gram karakter huruf menggunakan Library NLTK.

```
text_ngrams = ngrams (sentence, n)
for text in text_ngrams:
  print (text)
```

```
('g', ' ', 'm')
(' ', 'm', 'e')
('m', 'e', 'm')
('e', 'm', 'i')
('m', 'i', 'l')
('i', 'l', 'i')
('l', 'i', 'k')
('i', 'k', 'i')
('k', 'i', ' ')
('i', ' ', 'b')
(' ', 'b', 'a')
('b', 'a', 'n')
('a', 'n', 'y')
('n', 'y', 'a')
('y', 'a', 'k')
('a', 'k', ' ')
('k', ' ', 'r')
(' ', 'r', 'e')
('r', 'e', 's')
('e', 's', 't')
('s', 't', 'o')
('t', 'o', ' ')
('o', ' ', 'y')
(' ', 'y', 'a')
```

3. Ngram menggunakan fungsi Split.

```
sentence = "Kota malang memiliki banyak resto yang unik, begitu juga kota Bandung."
n = 3
treegrams = ngrams(sentence.split(),n)
for grams in treegrams:
  print (grams)
```

```
('Kota', 'malang', 'memiliki')
('malang', 'memiliki', 'banyak')
('memiliki', 'banyak', 'resto')
('banyak', 'resto', 'yang')
('resto', 'yang', 'unik,')
('yang', 'unik,', 'begitu')
('unik,', 'begitu', 'juga')
('begitu', 'juga', 'kota')
('juga', 'kota', 'Bandung.')
```

4. N-Gram tanpa menggunakan library NLTK.

```
sentence = "Angkutan umum yang nyaman dan aman"
n = 6

[sentence[i:i+n]for i in range(len(sentence)-n+1)]
```

Hasilnya :

```
['Angkut',
 'ngkuta',
 'gkutan',
 'kutan ',
 'utan u',
 'tan um',
 'an umu',
 'n umum',
 ' umum ',
 'umum y',
 'mum ya',
 'um yan',
 'm yang',
 ' yang ',
 'yang n',
 'ang ny',
 'ng nya',
 'g nyam',
 ' nyama',
 'nyaman',
 'yaman ',
 'aman d',
 'man da',
 'an dan',
 'n dan ',
 ' dan a',
 'dan am',
 'an ama',
 'n aman']
```

C. Latihan

1. Bagaimana cara untuk ekstraksi fitur?
2. Buat simulasi sederhana untuk menghitung jumlah term yang unik dalam list_komentar?

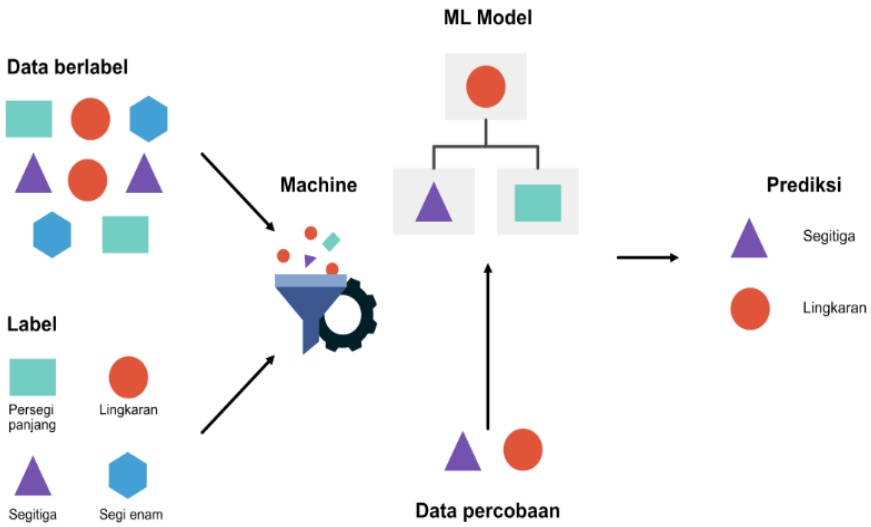
```
CV = CountVectorizer ()
term_fit = CV.fit (list_komentar)
print len(term_fit.vocabulary_)
```

3. Buat program sederhana untuk proses ekstraksi future dengan menggunakan TFIDF!

BAB VII
KLASIFIKASI DENGAN PYTHON

Klasifikasi merupakan sebuah teknik dan bagian dari *supervised learning* yang dapat digunakan untuk menentukan atau memperkirakan kelas suatu objek berdasarkan atribut data yang ada. Klasifikasi ini dapat diterapkan pada banyak sektor termasuk: perbankan, kesehatan, industri dan bahkan perdagangan.

Di beberapa bidang, taksonomi digunakan sebagai alat untuk medukung pengambilan keputusan dengan cepat sehingga dapat memecahkan masalah yang kompleks dan BigData. Ini adalah salah satu contoh penggunaan klasifikasi dalam machine learning.



A. Machine Learning dengan Python

Machine learning umumnya menggunakan Python. Khusus untuk klasifikasi, kita bisa menggunakan *library scikit-learn*. Untuk memulai dapat mengikuti langkah langkah berikut ini :

1. Install *scikit-learn* melalui terminal.

```
pip install scikit-learn
```

2. Install scikit-learn melalui JupyterLab atau Notebook.

```
!pip install scikit-learn
```

B. Melatih Machine Learning dengan Kumpulan Data

Machine learning dapat bekerja seperti otak manusia yang tidak langsung pintar tetapi harus diberi pembelajaran dahulu. *Machine learning* membutuhkan pmbelajaran berupa pemrosesan berbagai jenis data. Kumpulan data ibarat informasi yang akan diproses oleh manusia.

1. Mengimpor Library

   Langkah pertama yang harus dilakukan yaitu memasukkan modul atau library pendukung yang berfungsi untuk melakukan klasifikasi yaitu :

   - Library Pandas yang memiliki fungsi dalam mengelola dataset.
   - Matplotlib yang berperan dalam memvisualisasi data.
   - Seaborn yang berfungsi untuk visualisasi data.
   - Train atau Test Split berfungsi untuk melatih data.

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
```

2. Mengelola Dataset

Setelah berhasil mengimpor modul - modul ialah mengelola dataset. Langkah selanjutnya yang perlu dilakukan yaitu memuat dataset. Kali ini akan menggunakan dataset penguin dan fitur-fitur yang ada. Seperti script dibawah ini :

```
penguins_df = pd.read_csv("penguins.csv")
print(penguins_df)
```

	Unnamed: 0	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex
0	0	Adelie	Torgersen	39.1	18.7	181.0	3750.0	Male
1	1	Adelie	Torgersen	39.5	17.4	186.0	3800.0	Female
2	2	Adelie	Torgersen	40.3	18.0	195.0	3250.0	Female
3	3	Adelie	Torgersen	NaN	NaN	NaN	NaN	NaN
4	4	Adelie	Torgersen	36.7	19.3	193.0	3450.0	Female
...	...	...	...	...	...	...	...	...
339	339	Gentoo	Biscoe	NaN	NaN	NaN	NaN	NaN
340	340	Gentoo	Biscoe	46.8	14.3	215.0	4850.0	Female
341	341	Gentoo	Biscoe	50.4	15.7	222.0	5750.0	Male
342	342	Gentoo	Biscoe	45.2	14.8	212.0	5200.0	Female
343	343	Gentoo	Biscoe	49.9	16.1	213.0	5400.0	Male

344 rows × 8 columns

Label yang akan digunakan pada kasus ini adalah label spesies.

```
[in]:
print(penguins_df["species"].unique())

[out]:
['Adelie' 'Chinstrap' 'Gentoo']
```

Menghapus kategori selain label spesies.

Disini hanya akan mengelola data-data numerik berdasarkan label spesies. Untuk memudahkan melihat dataset, kategori selain spesies dapat dihapus.

```
penguins_df = penguins_df.drop("Unnamed: 0", axis=1)
penguins_df = penguins_df.drop("island", axis=1)
penguins_df = penguins_df.drop("sex", axis=1)
```

Mengelola nilai data NaN

Dari tampilan dataset penguin, masih ada nilai NaN pada baris-baris data. NaN merupakan sebuah missing value. Dengan adanya missing value dapat berpengaruh dalam proses klasifikasi. Sehingga, perlu adanya pengelolaan nilai dengan fungsi dropna ().

```
penguins_df = penguins_df.dropna()
```

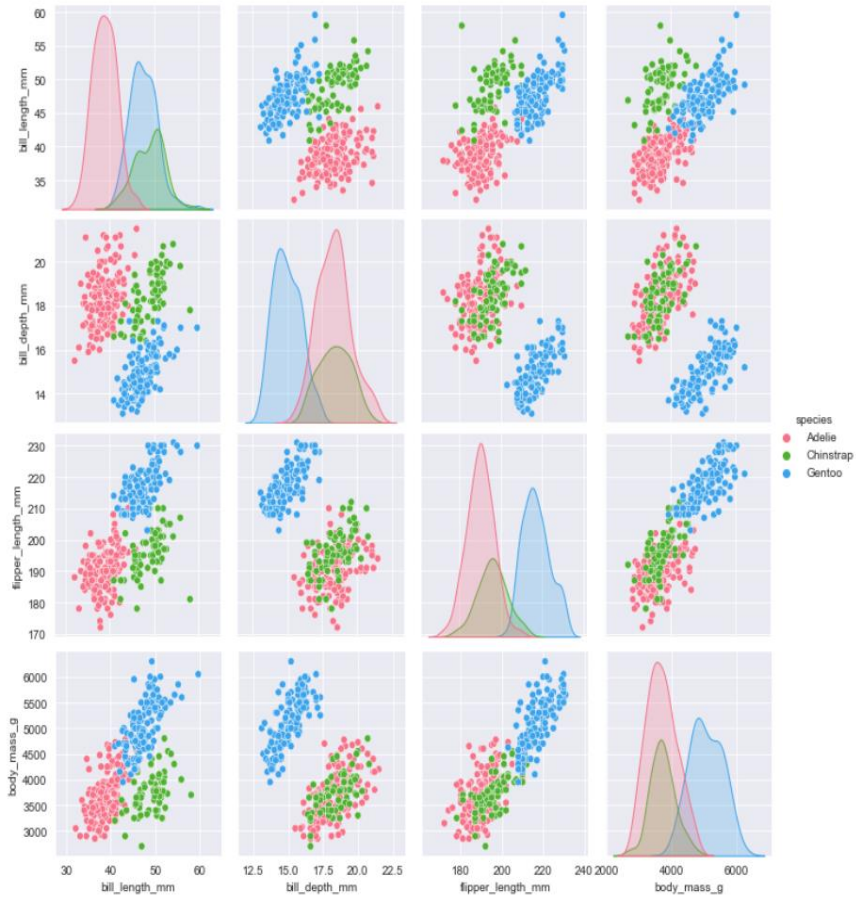
3. Visualisasi Data

   Langkah selanjutnya membuat visualisasi data untuk mempermudah melihat persebaran data fitur-fitur penguin berdasarkan label spesies. Untuk visualisasi data, dapat menggunakan seaborn pairplot ().

   Susunan kodenya seperti dibawah ini :

```
# style background grafik
sns.set_style("darkgrid")
# fungsi pairplot() berisi KDE dan scatterplot
sns.pairplot(penguins_df, hue = "species", diag_kind = "kde", kind = "scatter
# fungsi menampilkan grafik
plt.show()
```

Berikut tampilannya :



4. Melatih Data

   Selanjutya untuk melatih data dapat menggunakan sebuah metode yang disebut dengan metode train atau test split.

Metode tersebut bisa membagi dataset menjadi dua bagian yaitu train dan test. Disini untuk train berfungsi untuk fit model machine learning, sedangkan test untuk evaluasi hasil dari fit model.

Kodenya seperti dibawah ini :

```
# deklarasi y sebagai label
y = penguins_df["species"]
# deklarasi X untuk dataset tanpa label
X = penguins_df.drop("species", axis=1)
# penerapan train/test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.01, ran
```

Menurut susunan beberapa kode diatas terdapat parameter yang perlu dimasukkan ke dalam sebuah fungsi train_test_split().

C. Algoritma dan Model Klasifikasi

Proses selanjutnya ialah menentukan algoritma serta model klasifikasi yang tepat dan paling akurat. Ada banyak model Algoritma klasifikasi dalam Python mulai dari *Logistic Regression* sampai *Random Forest Classifier*. Tingkat akurasi untuk setiap model pasti berbeda-beda tergantung dari jumlah dataset yang ada. Sehingga dalam memilih model pada dataset penguin, perlu dilihat dari tingkat akurasi masing-masing model dengan fungsi score().

1. Logistic Regression

   Logistic Regression merupakan salah satu algoritma klasifikasi yang berfungsi untuk mencari hubungan antara fitur (input) diskrit atau kontinu dengan probabilitas hasil output diskrit tertentu. Ada beberapa tipe *logistic regression* yaitu :

a. Binary Logistic Regression: Sebuah tipe *logistic regression* yang hanya memiliki dua output (klasifikasi pada 2 kelas yang berbeda)
   Contoh : Obesitas – Tidak Obesitas, Positif dan Negatif
b. Multinomial Logistic Regression: Sebuah tipe *logistic regression* yang memiliki dua output atau lebih (klasifikasi pada 2 kelas yang berbeda
   Contoh : Hasil kelas Sentimen Analisis (Positif, Negatif dan Netral)
c. Ordinal Logistic Regression: Sebuah tipe *logistic regression* yang memiliki dua output atau lebih tetapi memperhatikan urutannya.
   Contoh : Membagi range nilai IPK mahasiswa

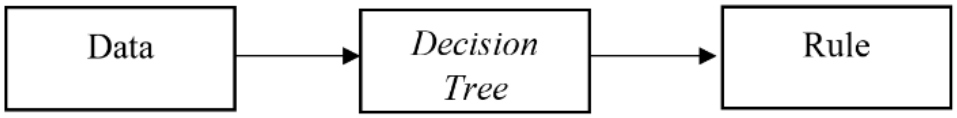
Selanjutnya untuk gambaran tingkat akurasi model *logistic regression* dengan fungsi score ().

```
[in]:
# deklarasi fungsi logistic regression
logreg = LogisticRegression()
# pengaplikasian logistic regression
logreg.fit(X_train, y_train)
# menampilkan hasil akurasi
print('Akurasi train set dengan Logistic Regression: {:.2f}'.format(logreg.sc
print('Akurasi test set dengan Logistic Regression: {:.2f}'.format(logreg.sco

[out]:


Akurasi train set dengan Logistic Regression: 0.99
Akurasi test set dengan Logistic Regression: 0.99
```

2. Decision Tree (Pohon Keputusan)

   Dapat membagi dataset berdasarkan kondisi bercabang yang menghasilkan keputusan akhir. *Decision tree* memiliki 2 jenis algoritma yang terkenal yaitu C4.5 dan *Random Forest*.

Pada algoritma C4.5 merupakan pengembangan dari algoritma ID3 dengan berbagai peningkatan. Sedangkan untuk algoritma *Random Forest* ialah sebuah algoritma yang dapat membentuk satu set pohon saat proses training data set. Pada Algoritma C4.5 terdapat beberapa langkah dalam membuat pohon keputusan seperti gambar di bawah ini :



a. Untuk menyiapkan data training dapat dilakukan dengan cara mengambil data history yang sudah terjadi atau data dari masa lalu dan data tersebut sudah di kelompokkan dalam kelas – kelas tertentu.
b. Menghitung akar pohon. Akar diambil dari atribut yang dipilih dengan menghitung nilai gain masing-masing atribut, nilai gain tertinggi adalah akar pertama. Sebelum menghitung nilai gain atribut, terlebih dahulu menghitung nilai entropinya.

$$Entropy(S) = \sum_{i=1} -pi * log2pi$$

S = Himpunan Kasus A = Fiturn = Jumlah partisi S

pi = Proporsi dari Si terhadap S

c. Hitung nilai gain menggunakan persamaan.

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \epsilon Valus(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

S = Himpunan kasusA = Fitur

n = Jumlah partisi atribut A

|Si| = Proporsi Si terhadap S

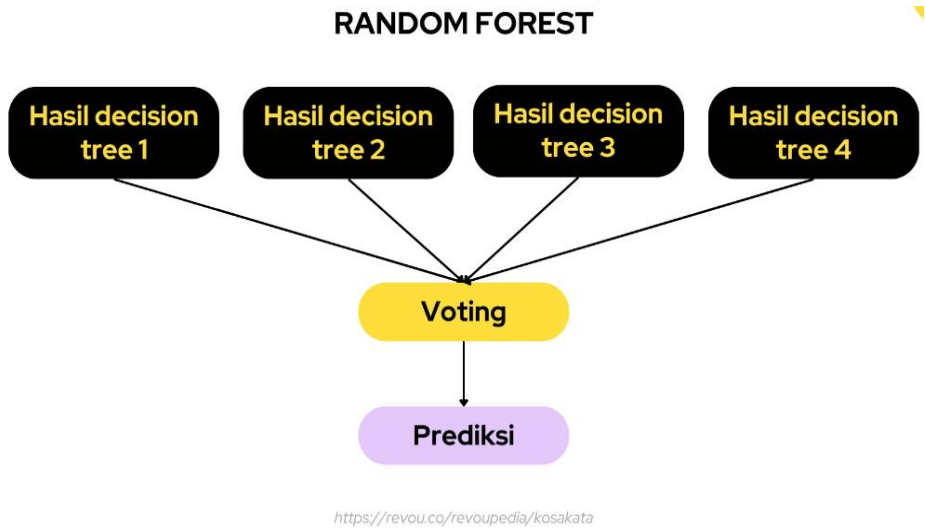
|S| = jumlah kasus dalam S

d. Ulangi langkah ke-2 dan ke-3 hingga semua *record* terpartisi.

e. Dalam proses mempartisi pohon keputusan berhenti ketika semua *record* dalam simpul N mendapat kelas yang sama dan sudah tidak ada atribut di dalam record yang dipartisi lagi.

3. Random Forest (RF)

Random Forest ialah salah satu algoritma yang dapat digunakan untuk klasifikasi data dalam jumlah besar. Selain itu Random Forest juga merupakan hasil improvisasi dari decision tree lalu mengambil rata-rata beberapa decision tree yang diturunkan dari subset data train. Dalam proses kalsifikasi *Random Forest* dapat dilakukan dengan mengambil hasil keputusan yang dominan pada pohon yang terbentuk.



Bentuk random forest

Semakin banyak hasil pohon keputusan maka semakin tinggi keakuratannya, apalagi setiap pohon tidak berkorelasi satu sama lain.

Namun, hal ini tidak berarti bahwa hasil dari random forest selalu benar. Oleh karena itu, ada dua prasyarat untuk hasil yang lebih baik:

- Variabel fitur dalam kumpulan data harus memiliki nilai sebenarnya tertentu agar pengklasifikasi dapat memprediksi hasil yang akurat.
- Prediksi dari setiap pohon seharusnya memiliki korelasi yang sangat rendah.

Dilihat dari beberapa sumber, pengoperasian *random forest* (RF) dibagi menjadi 2 tahap. Langkah awal atau yang pertama adalah menggabungkan hasil (N) dari beberapa pohon keputusan untuk membuat *random forest*. Langkah kedua ialah membuat prediksi untuk setiap pohon yang dibuat pada langkah pertama. Dibawah ini untuk beberapa tahapan proses kerja *random forest*:

Langkah 1: Menentukan titik data K secara acak dari training set.

Langkah 2: Membuat *decision tree* untuk setiap sampel dengan titik data yang dipilih.

Langkah 3: Memilih berapa banyak jawaban *decision tree* yang ingin dilibatkan dalam membuat *random forest*.

Langkah 4: Mempertimbangkan hasil akhir dari rata-rata terbanyak.

Untuk implementasi *random forest* lebih sering digunakan oleh banyak data *scientist* yang bergerak dalam bidang perdagangan, perbankan, medis ,saham, dan e-commerce.

Memiliki tujuan untuk membantu operasional bisnis agar berjalan secara efisien. Sedangkan dalam dunia medis penggunaan random forest biasanya digunakan untuk mengidentifikasi penyakit yang diderita, menganalisis riwayat kesehatan pasien, dan menentukan kombinasi pengobatan yang tepat.

4. K-Nearest Neighbor (KNN)

   Algoritma k-nearest neighbour (k-NN atau KNN) merupakan salah satu metode untuk melakukan klasifikasi berdasarkan jarak dan kedekatan titik data dengan titik-titik lainnya, serta mengurutkan data berdasarkan kemiripan (similarity) atau kedekatannya dengan data lain. Adapun beberapa kelebihan dan kekurangan dari algoritma KNN. Untuk kelebihan algoritma KNN yaitu :

   - Kemudahan dalam implementasi dan dipahami

     KNN terkenal dengan kesederhanaannya. Algoritme ini mudah dipahami dan diterapkan, sehingga cocok untuk pemula dalam dunia data science dan machine learning.
   - Kemampuan dalam beradaptasi

     KNN dapat mudah beradaptasi dengan perubahan pada dataset. Saat menambahkan sampel pelatihan baru, KNN segera mempertimbangkan data baru ini, karena semua data pelatihan disimpan di memori.
   - Jumlah *Hyperparameter* yang sedikit

     KNN hanya memiliki dua *hyperparameter* utama: nilai K (jumlah tetangga terdekat yang dipertimbangkan) dan metrik jarak yang akan digunakan. Hal ini membuat proses parameterisasi model menjadi lebih sederhana dan tidak rumit dibandingkan dengan beberapa algoritma lainnya.

   Sedangkan kekurangan algoritma KNN yaitu :

   - Tidak direkomendasikan untuk dataset yang berukuran besar

     Salah satu kelemahan terbesar KNN adalah ketidakcocokannya untuk dataset yang besar. Algoritma ini memerlukan perhitungan jarak antara titik baru dan seluruh titik dalam dataset, sehingga biaya komputasinya sangat tinggi dan dapat

menurunkan kinerja algoritma secara signifikan untuk dataset yang besar.

- Tidak direkomendasikan untuk dimensi tinggi

  KNN tidak efisien untuk data berdimensi tinggi. Seiring bertambahnya jumlah dimensi, algoritma akan menghadapi masalah penghitungan jarak yang semakin kompleks dan membutuhkan lebih banyak data untuk melakukan penghitungan yang akurat.
- Penskalaan fitur sangat diperlukan

  Sebelum menerapkan KNN ke dalam dataset, penting untuk melakukan penskalaan fitur. Tanpa penskalaan yang baik (standarisasi dan normalisasi), KNN dapat memberikan prediksi yang salah karena beberapa fitur memiliki skala yang dominan.
- Sensitif pada noise, missing value, dan outlier

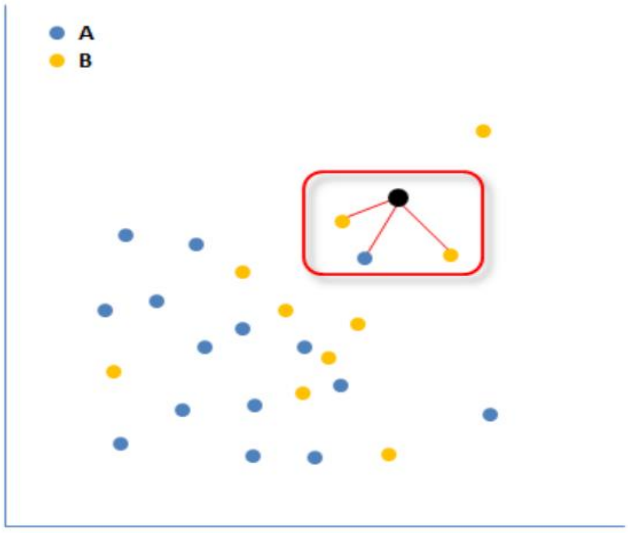
  KNN biasanya sensitif terhadap noise pada dataset. Artinya sebelum menggunakan KNN, kita perlu mengolah data secara cermat, termasuk menangani nilai yang hilang serta mendeteksi dan menangani outlier.

Algoritma KNN bekerja dengan mengambil sejumlah K data terdekat (tetangganya) sebagai acuan untuk menentukan kelas dari data baru.

Proses kerja algoritma KNN secara umum seperti berikut :

1. Menentukan banyaknya tetangga (K) yang berfungsi untuk memperkirakan penentuan kelas.
2. Menghitung jarak dari data baru ke setiap titik data (data poin) dalam dataset.
3. Mengambil beberapa K data dengan jarak paling dekat kemudian menentukan kelas dari data baru.

Lihatlah gambar ilustrasi di bawah ini:



Gambar diatas mempunyai beberapa titik data (data poin) yang terbagi menjadi 2 kelas yaitu kelas A atau yang berwarna biru dan kelas B yang berwarna kuning. Misalkan ada data baru yang berwarna hitam dan kelasnya akan diprediksi menggunakan algoritma KNN. Berdasarkan contoh di atas, nilai K yang digunakan ialah 3. Setelah dihitung jarak titik hitam dengan setiap titik data, diperoleh 3 titik terdekat yang terdiri dari 2 titik kuning dan satu titik biru, seperti terlihat pada kotak berwarna merah. maka kelas data baru yaitu titik hitam adalah B yang berwarna kuning.

Pada K-Nearest Neighbor (KNN) data point yang berada tetapi berdekatan biasanya disebut "neighbor/tetangga". Sedangkan untuk perhitungan jarak menggunakan konsep *Euclidean*.

Untuk jarak *Euclidean* berguna untuk menguji ukuran yang digunakan sebagai interpretasi kedekatan jarak antara 2 objek yang direpresentasikan sebagai berikut ini:

$$dist = \sum_{i=1}^{p} \sqrt{(x2 - x1)^2}$$

Keterangan :

dist = Jarak

x1 = Data *Training*

x2 = Data *testing*

i = Variable Data

p = Jumlah Atribut

Rumus di atas dapat digunakan jika hanya terdapat satu variabel bebas, dan bila terdapat lebih dari satu maka dapat dirumuskan sebagai berikut :

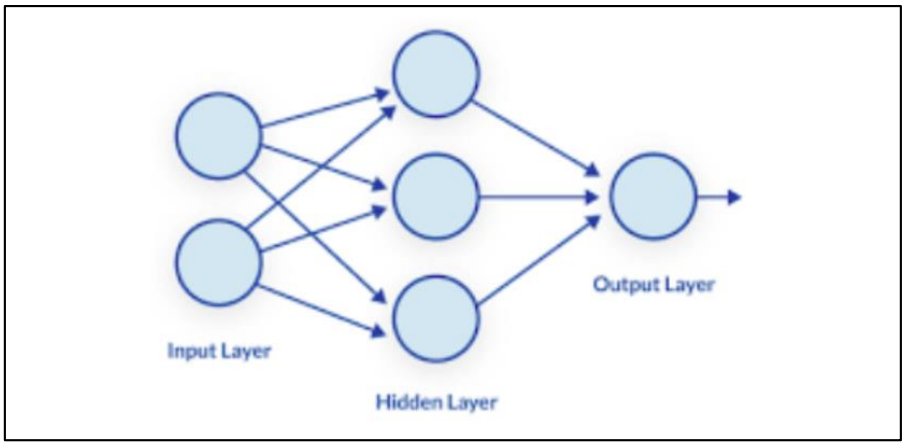
$$dis = \sqrt{\sum_{i=0}^{n} (x_{1i} - x_{2i})^2 + (y_{1i} - y_{2i})^2 + \cdots}$$

5. Multi-Layer Perceptron Classifier (MLP)

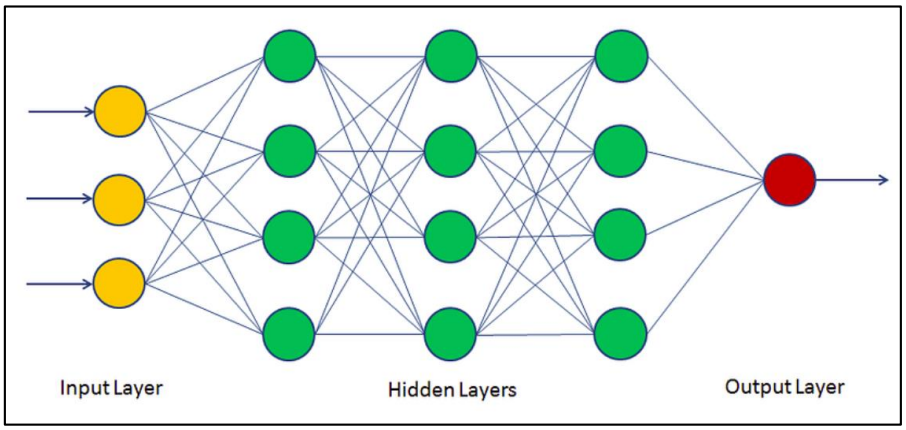
   Metode klasifikasi Multilayer Perceptron (MLP) ialah salah satu jenis algoritma jaringan saraf tiruan yang mengadopsi cara kerja jaringan saraf tiruan pada makhluk hidup.

   Klasifikasi ini bergantung pada neural network sebagai pembuat keputusan serta dapat dipercaya karena pembelajaran dilakukan secara terkontrol dengan memperbarui bobot balik (backpropagation). Penentuan bobot optimal akan memberikan hasil klasifikasi yang tepat.

MLP terdiri dari sistem sederhana serta jaringan atau node yang saling berhubungan. Dibawah ini gambaran ilustrasinya:



Frank Rosenblatt memperkenalkan Perceptron pada tahun 1950. Ia memiliki kemampuan untuk mempelajari hal-hal kompleks seperti otak manusia. Jaringan perceptron terdiri dari 3 unit yaitu : *Sensory* Unit (Input Unit), *Associator* Unit (Hidden Unit), dan *Response* Unit (Output Unit).



Perceptron terbentuk dari *input* layer dan *output* layer yang terhubung sepenuhnya. MLP memiliki input layer dan output layer yang sama, tetapi mungkin ada beberapa lapisan tersembunyi (*hidden* layer) di antara keduanya.

Perceptron Multi-Lapisan melatih model secara berulang. Pada setiap iterasi, turunan parsial dari fungsi kerugian digunakan untuk memperbarui parameter. Kita juga dapat menggunakan regularisasi fungsi kerugian untuk mencegah overfitting pada model.

Selanjutnya mencoba menerapkan MPL pada sebuah kasus seperti melakukan prediksi churn karyawan menggunakan MPL sehingga dapat membantu menciptakan rencana retensi karyawan yang lebih baik dan meningkatkan kepuasan karyawan. Contoh penggunaan MPL dengan *python*:

1. Memuat Kumpulan Data

   langkah pertama memuat kumpulan data HR yang diperlukan menggunakan fungsi read CSV pandas.

```python
import numpy as np
import pandas as pd

# Load data
data=pd.read_csv('HR_comma_sep.csv')

data.head()
```

   Keluaran:

	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_company	Work_accident	left	promotion_last_5years	Departments	salary
0	0.38	0.53	2	157	3	0	1	0	sales	low
1	0.80	0.86	5	262	6	0	1	0	sales	medium
2	0.11	0.88	7	272	4	0	1	0	sales	medium
3	0.72	0.87	5	223	5	0	1	0	sales	low
4	0.37	0.52	2	159	3	0	1	0	sales	low

2. Proses Awal dalam Pengkodean Label

Mengkodekan data ini untuk memetakan setiap nilai ke sebuah angka. Dalam sklearn dapat menggunakan Label Encoder.

```
# Import LabelEncoder
from sklearn import preprocessing

# Creating labelEncoder
le = preprocessing.LabelEncoder()

# Converting string labels into numbers.
data['salary']=le.fit_transform(data['salary'])
data['Departments ']=le.fit_transform(data['Departments '])
```

Di sini untuk mengimpor modul preprocessing dan membuat objek Label Encoder. Objek Label Encoder ini memungkinkan Anda untuk menyesuaikan dan mengonversi kolom "gaji" dan "departemen" menjadi kolom numerik.

3. Memisahkan Kumpulan Data

   Untuk mengevaluasi performa model, perlu membagi kumpulan data menjadi set pelatihan dan set pengujian. Membagi kumpulan data menggunakan fungsi train_test_split(). Pada dasarnya harus melewati 3 parameter fitur, target dan ukuran test_set.

```
Python
# Spliting data into Feature and
X=data[['satisfaction_level', 'last_evaluation', 'number_project', 'average_montly_hours
y=data['left']

# Import train_test_split function
from sklearn.model_selection import train_test_split

# Split dataset into training set and test set
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42
```

4. Membangun Model Klasifikasi

   Langkah awalnya mengimpor modul MLP Classifier dan buat objek MLP Classifier menggunakan fungsi MLP Classifier(). Lalu sesuaikan model pada set kereta menggunakan fit() dan lakukan prediksi pada set pengujian menggunakan prediksi().

```
# Import MLPClassifer
from sklearn.neural_network import MLPClassifier

# Create model object
clf = MLPClassifier(hidden_layer_sizes=(6,5),
                    random_state=5,
                    verbose=True,
                    learning_rate_init=0.01)

# Fit data onto the model
clf.fit(X_train,y_train)
```

5. Melakukan Prediksi dan Evaluasi Model

```
# Make prediction on test dataset
ypred=clf.predict(X_test)

# Import accuracy score
from sklearn.metrics import accuracy_score

# Calcuate accuracy
accuracy_score(y_test,ypred)
```

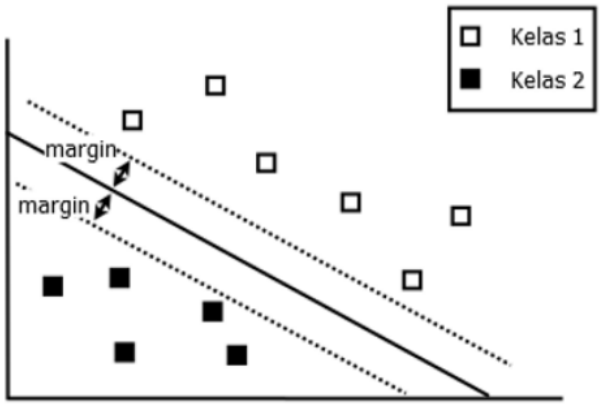
   Keluaran:

   0,9386666666666666

   Hasilnya mendapatkan tingkat klasifikasi sebesar 93,8%, yang dianggap sebagai akurasi yang baik.

6. Support Vector Machine

Metode SVM ialah salah satu jenis model vector berbasis classifier yang dapat merubah teks menjadi vector sebelum dilakukan klasifikasi. Tujuan dari metode SVM adalah untuk menemukan hyperlane yang maksimal dari setiap titik data. Hyperplane pada metode Support Vector Machine dapat dilihat pada gambar dibawah ini:



Metode klasifikasi Support Vector Machine dapat dirumuskan dalam persamaan berikut ini:

$$x_i * w + b \geq +1$$

$$x_i * w + b \leq -1$$

Sedangkan untuk menentukan nilai Lagrange Multiplier a dengan persamaan ini :

$$max_\alpha \; L_D = \sum_{i=1}^{n} a_i - \frac{1}{2} \sum_{i=1}^{n} a_i \; a_j \; y_i \; y_j \; x_i \; x_j$$

$$Subject\ to \sum_{i=1}^{n} a_i \; y_i = 0, a_i \geq 0$$

Persamaan berikutnya merupakan fungsi pemisahan optimal:

$$f(x) = sign(\sum_{i=1}^{m} a_i \; y_i \; K(x, x_i)) + b$$

Keterangan :

w : parameter hyperplane yang dicari

x : titik data masukan Support Vector Machine

b : parameter hyperplane yang dicari (nilai bias)

a_i : nilai bobot pada setiap titik data

$K\ (x, x_i)$: fungsi kernel

7. Naive Bayes

Naive Bayes merupakan salah satu algoritma Supervised Learning yang paling sederhana sertaa sebuah teknik klasifikasi statistik berdasarkan Teorema Bayes. Teorema Bayes merupakan persamaan matematika yang digunakan pada probabilitas dan statistik dalam perhitungan probabilitas bersyarat. Dibawah ini merupakan rumus sederhana teorema Bayes:

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

Diketahui bahwa P(A) dan P(B) ialah dua kejadian yang saling bebas dan (B) tidak sama dengan nol.

Keterangan:

- P (A | B): adalah peluang bersyarat suatu kejadian A terjadi jika B benar.
- P (B | A): adalah peluang bersyarat suatu peristiwa B terjadi jika hal itu A benar.
- P(A)dan P(B): adalah probabilitas Adan Bterjadinya secara independen satu sama lain atau probabilitas marjinal.

Terdapat beberapa jenis klasifikasi Naive Bayes yaitu :

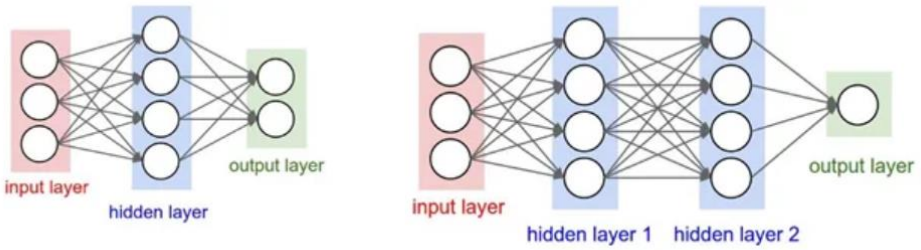
- Metode Multinomial Naive Bayes adalah metode pembelajaran Bayesian yang umum untuk pemrosesan bahasa alami. Dengan penggunaan teorema Bayes sebuah program bisa memperkirakan tag suatu teks, seperti email atau koran. Program ini dapat menilai kemungkinan setiap tag Naive Bayes multinomial untuk sampel tertentu dan mengembalikan tag dengan kemungkinan tertinggi.
- Bernoulli Naive Bayes ialah sebuah bagian dari Naive Bayes dan membutuhkan nilai biner. Terdapat beberapa fitur yang mungkin ada, namun masing-masing diasumsikan sebagai variabel biner (Bernoulli, Boolean).
- Gaussian Naive Bayes adalah varian dari Naive Bayes yang mengikuti distribusi Gaussian secara normal dan mendukung data kontinu. Untuk membangun model sederhana menggunakan Gaussian Naive Bayes, data dapat diasumsikan dengan dicirikan dengan distribusi Gaussian tanpa kovarians (dimensi independen) antar parameter. Model ini dapat diterapkan eorema Bayes untuk menghitung mean dan deviasi standar titik-titik dalam setiap label. Dapat diasumsikan data dicirikan oleh distribusi Gaussian tanpa kovarian antar parameter (dimensi independen). Model ini dapat disesuaikan dengan menerapkan teorema Bayes untuk menghitung mean dan deviasi standar skor setiap label.

8. Neural Network

   Jaringan syaraf tiruan atau Artificial Neural Network (ANN) merupakan jaringan yang terdiri dari sekelompok unit pemroses kecil yang dimodelkan berdasarkan perilaku jaringan syaraf manusia. ANN dapat memodelkan hubungan linear maupun non linear. Jaringan syaraf tiruan terdiri dari elemen-elemen untuk pemrosesan informasi yang disebut neuron, unit, sel atau node. Setiap neuron terhubung ke neuron lain melalui koneksi yang diwakili oleh bobot/*weight*. Cara menentukan nilai bobot disebut dengan training, learning, atau algoritma. Setiap neuron menggunakan fungsi aktivasi pada net input (jumlah masukan tertimbang) untuk menentukan prediksi output. Neural Network disusun dalam grup yang disebut dengan layer (lapis). Secara keseluruhan terdapat tiga lapis yang memebentuk jaringan syaraf tiruan :

   - Lapisan input: Jaringan saraf tiruan menerima data dalam lapis input. Jumlah node atau neuron pada lapis input bergantung pada jumlah masukan dalam model serta setiap masukan menentukan satu neuron.
   - Hidden layer atau lapisan tersembunyi: Lapisan tersembunyi terletak di antara lapisan masukan dan keluaran, yang terdiri dari beberapa lapisan tersembunyi.
   - Lapisan keluaran atau lapis output: lapisan yang telah melalui proses training, network bereaksi terhadap input baru untuk menghasilkan output yang merupakan hasil prediksi.

Secara sederhana arsitektur *neural network* dapat digambarkan sebagai berikut:



D. Latihan

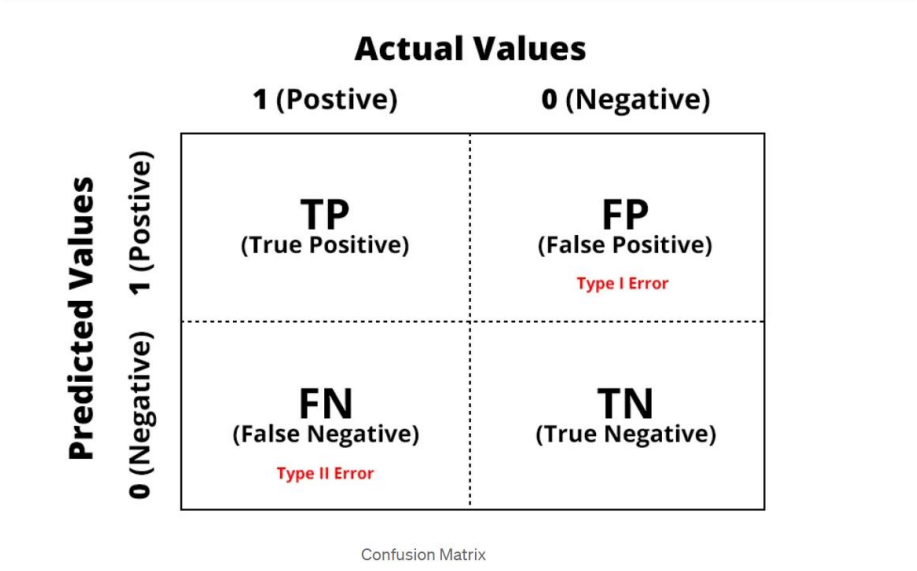
1. Bagaimana cara untuk klasifikasi menggunakan python?
2. Dalam sektor apa saja kalasifikasi dapat diterapkan?
3. Buat simulasi sederhana untuk melatih machine learning dengan kumpulan data!
4. Buat program sederhana untuk proses klasifikasi dengan menggunakan salah satu model klasifikasi!

BAB VIII
METRIK UNTUK MENGEVALUASI PERFORMA PENGKLASIFIKASI

Akurasi klasifikasi merupakan jumlah prediksi yang benar dibagi dengan jumlah total prediksi. Akurasi mungkin saja dapat "menyesatkan", pada kasus tertentu seperti ketidakseimbangan kelasnya besar (*large class imbalance*). Model klasifikasi mampu memprediksi nilai pada kelas terbesar dan memberikan akurasi yang tinggi, dan tentunya model yang dihasilkan dapat memprediksi nilai yang salah, sehingga diperlukan metrik evaluasi lain untuk mengukur kinerja model klasifikasi yang dibuat.

Metrik yang dimaksudkan disini adalah *Precision*, *Recall* dan *Confusion Matrix*. Selain itu ada juga metrik lain yang dapat digunakan, tetapi 3 jenis metrik ini cukup untuk langkah awal. *Binary Classification Problem* atau klasifikasi biner digunakan untuk mengklasifikasikan elemen dalam suatu himpunan menjadi dua kelompok (masing-masing disebut kelas) berdasarkan aturan klasifikasi. Nilai kelas dapat direpresentasikan sebagai positif atau negatif; 0 atau 1; benar (true) atau salah (false).

Membandingkan nilai aktual dengan nilai prediksi merupakan salah satu cara yang dapat digunakan untuk mengukur performa model klasifikasi. *Error Matrik* (*Confusion Matrix*) merupakan sebuah proses pengukuran kinerja untuk masalah klasifikasi *machine learning* yang memiliki dua kelas atau lebih sebagai keluaran. Biasanya *confusion matrix* berbentuk tabel yang berisi 4 kombinasi berbeda antara nilai prediksi dan nilai actual seperti gabar dibawah ini :



Confusion Matrix

a. True Positive (TP): sebuah data positif yang di prediksi benar.
b. True Negative (TN): data negatif yang diprediksi benar.
c. False Postive (FP) — Type I Error: data negatif namun diprediksi sebagai data positif.
d. False Negative (FN) — Type II Error: Beberapa data positif namun diprediksi sebagai data negatif.

Terdapat beberapa cara agar kita mudah dalam mengingatnya :

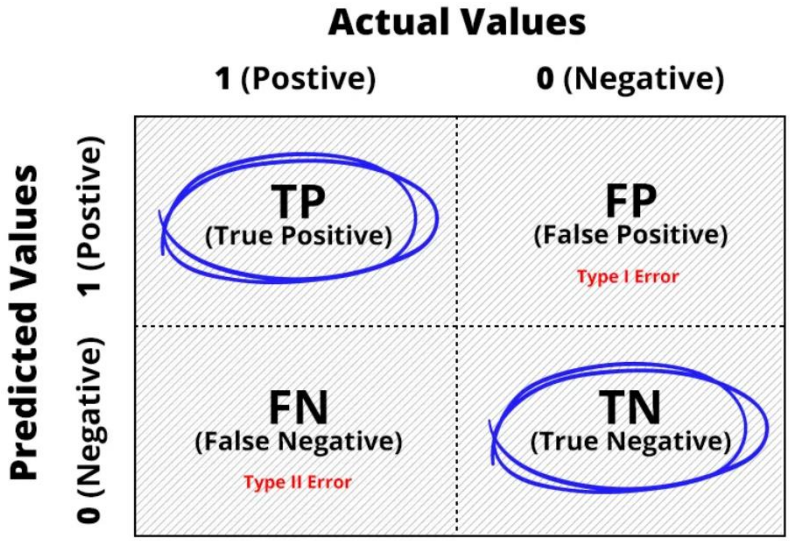
- Jika terdapat awalan True berarti prediksi tersebut benar, walaupun diprediksi terjadi atau tidak terjadi.
- Jika False berada diawal berarti prediksi tersebut salah.
- Positif dan negatif ialah sebuah hasil prediksi dari model.

Confusion matrix dapat digunakan dalam menghitung berbagai performance *metrics* untuk mengukur kinerja model yang dihasilkan. Bagian ini membahas beberapa performance metrics yang umum digunakan seperti accuracy, precission, dan recall. Tetapi jika terjadi *tradeoff* di antara presisi dan recall yang artinya ketika nilai recall tinggi dan presisi sangat rendan dan sebaliknya.

Maka kita tidak dapat menggunakan skor diantara keduanya melainkan menggunakan F1-Score.

A. *Accuracy* (Akurasi)

Akurasi merupakan metrik evaluasi yang dapat mengukur seberapa baik model membuat prediksi yang benar dari semua prediksi yang dibuatnya. Dalam konteks klasifikasi, akurasi memberikan gambaran seberapa sering suatu model memprediksi kelas yang benar, baik berupa kelas positif atau negatif.



Confusion matrix yang menggambarkan nilai accuracy.

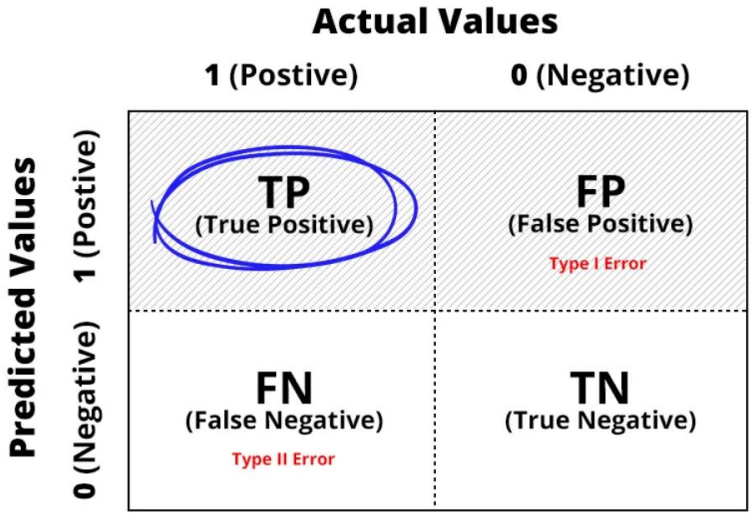
$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Persamaan nilai accuracy.

B. Precision atau Presisi (Positive Predictive Value)

Presisi adalah metrik evaluasi yang mengukur seberapa baik model membuat prediksi yang benar terhadap kelas positif dari semua prediksi

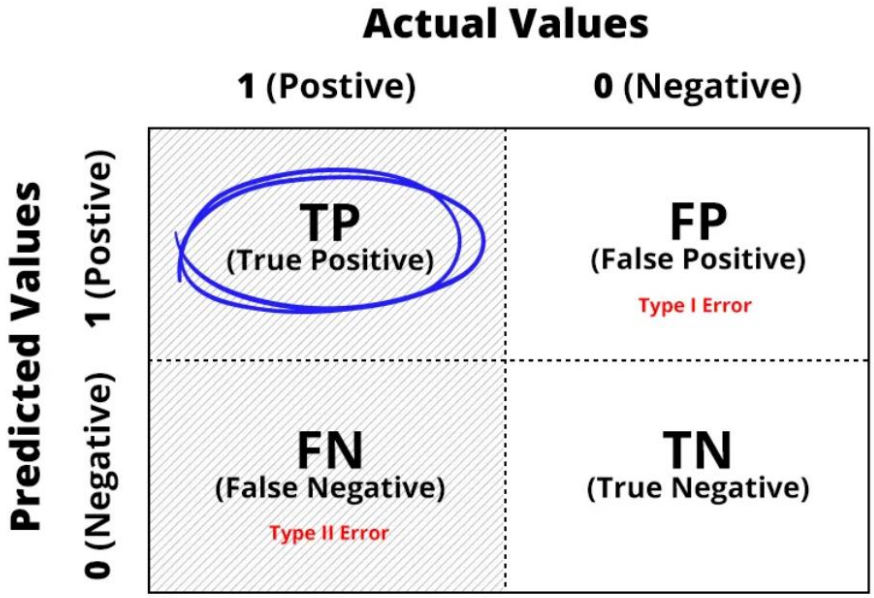
positif yang dibuatnya. Dari semua kelas positif yang diprediksi dengan benar, berapa banyak data yang benar-benar positif.



$$precision = \frac{TP}{TP + FP}$$

C. Recall atau Sensitivity (True Positive Rate)

Recall menggambarkan keberhasilan model dalam pengambilan informasi. Jadi, recall adalah rasio prediksi benar positif yang dibandingkan dengan semua data yang benar positif.



Confusion matrix yang menggambarkan nilai recall.

$$recall = \frac{TP}{TP + FN}$$

D. F1-Score

F1-Score dapat didefinisikan sebagai harmonic mean dari precision dan recall. Secara Matematika rumusannya dapat ditulis sebagai berikut :

$$\frac{1}{F1} = \frac{1}{2}\left(\frac{1}{precision} + \frac{1}{recall}\right)$$

F1-Score memiliki nilai terbaik yaitu 1.0 dan nilai terburuknya 0. Jika nilai F1-Score baik maka dapat diartikan bahwa model klasifikasi tersebut memiliki precision dan recall yang baik.

E. Latihan

1. Bagaimana cara untuk mencari Akurasi menggunakan python?
2. Bagaimana cara untuk mencari Presisi menggunakan python?
3. Bagaimana cara untuk mencari Recall menggunakan python?
4. Buat simulasi cara membuat model sederhana untuk prediksi dan menampilkan confusion matrixnya !
5. Buat program sederhana untuk proses prediksi menggunakan *Confusion Matrix*!

DAFTAR PUSTAKA

A. Deshwal and S. K. Sharma, "*Twitter sentiment analysis using various classification algorithms*," 2016 5th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 2016, pp. 251-257, doi: 10.1109/ICRITO.2016.7784960.

Ankit, and Nabizath Saleena. 2018. "*An Ensemble Classification System for Twitter Sentiment Analysis.*" Procedia Computer Science 132: 937–46.

De Mendonça, Ángel Héctor, Alba Ester Iribarne, and Alan Yurkina. 2017. 8 RInCE: Revista de Investigaciones del Departamento de Ciencias Económicas *Data Mining for the Masses*.

Engelbrecht, Andries P. 2007. "Comp_Intelligence.Pdf." : 630.

François Chollet. 2017. "Deep Learning with Python & Keras." *Manning Publications* 80(1): 453. http://www.ncbi.nlm.nih.gov/pubmed/20608803.

G. Gautam and D. Yadav, "*Sentiment analysis of twitter data using machine learning approaches and semantic analysis*," 2014 Seventh International Conference on Contemporary Computing (IC3), Noida, India, 2014, pp. 437-442, doi: 10.1109/IC3.2014.6897213.

Géron, Aurélien. 2019. O'Reilly Media, Inc. *Hands-on Machine Learning with Scikit-Learn, Keras, and Tensorflow, 2nd Edition*. https://books.google.com/books/about/Hands_on_Machine_Learning_with_Scikit_Le.html?hl=el&id=OCS1twEACAAJ.

Grus, Joel. 2019. *Data Science from Scratch from First Edition*. O'Reilly Media. http://oreilly.com/catalog/errata.csp?isbn=9781492041139 for%0Ahttp://safaribooksonline.com.

Gupta, Bhumika et al. 2017. "*Study of Twitter Sentiment Analysis Using Machine Learning Algorithms on Python.*" International Journal of Computer Applications 165(9): 29–34.

Ha, Jiawei, Micheline Kambe, and Jian Pe. 2011. Data Mining: Concepts and Techniques *Data Mining: Concepts and Techniques*.

Layton, Robert. 2015. Packt Publishing *Learning Data Mining with Python*.

M. Bouazizi and T. Ohtsuki, "*Sentiment analysis: From binary to multi-class classification: A pattern-based approach for multi-class sentiment analysis in Twitter,*" 2016 IEEE International Conference on Communications (ICC), Kuala Lumpur, Malaysia, 2016, pp. 1-6, doi: 10.1109/ICC.2016.7511392.

Rezwanul, Mohammad, Ahmad Ali, and Anika Rahman. 2017. "*Sentiment Analysis on Twitter Data Using KNN and SVM.*" International Journal of Advanced Computer Science and Applications 8(6): 19–25.

Yunitasari, Yessi, Aina Musdholifah, and Anny Kartika Sari. 2019. "Sarcasm Detection For Sentiment Analysis in Indonesian Tweets." *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)* 13(1): 53.