

PP/018/Sms ke-.../Revisi ke-



LABORATORIUM
TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS AHMAD DAHLAN



PETUNJUK PRAKTIKUM

Pengenalan POLA

Penyusun

Dr. Abdul Fadhil, M.T

Ahmad Azhari, S.Kom., M.Eng

Miftahurrahma Rosyda, S.Kom., M.Eng

2022

KATA PENGANTAR

Puji syukur kami panjatkan kehadirat Allah SWT yang telah melimpahkan rahmat dan karunia-Nya, sehingga dapat diselesaikannya Petunjuk Praktikum ini. Sholawat dan salam semoga terlimpah kepada baginda Nabi Muhammad saw beserta keluarga, shohabat dan para pengikutnya.

Petunjuk Praktikum Pengenalan Pola ini disusun dengan harapan dapat memberikan kemudahan bagi mahasiswa untuk memahami dasar-dasar dalam membangun sebuah sistem pengenalan pola. Berbagai contoh aplikasi diberikan yang meliputi metode ekstraksi ciri maupun teknik-teknik klasifikasi. Ucapkan terima kasih yang sebesar-besarnya diberikan kepada pihak semua pihak yang telah membantu dalam proses penyusunan petunjuk praktikum ini.

Akhirnya, tiada gading yang tak retak, kritik dan saran membangun kami harapkan untuk lebih sempurnanya petunjuk praktikum ini. Semoga ilmu yang diperoleh dari membaca dan mempraktekkan petunjuk praktikum ini dapat bermanfaat dan barokah.

Yogyakarta, Januari 2022

Penyusun

DAFTAR PENYUSUN

Dr. Abdul Fadlil, M.T.

Ahmad Azhari, S.Kom., M.Eng.

Miftahurrahma Rosyda, S.Kom., M.Eng.

HALAMAN PERNYATAAN

Yang bertanda tangan di bawah ini:

Nama : Ahmad Azhari, S.Kom., M.Eng.

NIK/NIY : 60160863

Jabatan : Dosen Pengampu

Menerangkan dengan sesungguhnya bahwa Petunjuk Praktikum ini telah direview dan akan digunakan untuk pelaksanaan praktikum di Semester Genap Tahun Akademik 2019/2020 di Laboratorium Komputasi Dasar, Program Studi Teknik Informatika, Fakultas Teknologi Industri, Universitas Ahmad Dahlan.

Yogyakarta, 22 Januari 2022

Mengetahui,

Ketua Kelompok Keilmuan Sistem Cerdas



Dewi Soyusiawaty, S.T., M.T.

NIK/NIY. 60040497

Kepala Laboratorium Komputasi Dasar



Adhi Prahara, S.Si., M.Cs.

NIK/NIY. 60150841

VISI DAN MISI PRODI TEKNIK INFORMATIKA

VISI

Menjadi Program Studi Informatika yang diakui secara internasional dan unggul dalam bidang Informatika serta berbasis nilai-nilai Islam.

MISI

1. Menjalankan pendidikan sesuai dengan kompetensi bidang Informatika yang diakui nasional dan internasional
2. Meningkatkan penelitian dosen dan mahasiswa dalam bidang Informatika yang kreatif, inovatif dan tepat guna.
3. Meningkatkan kuantitas dan kualitas publikasi ilmiah tingkat nasional dan internasional
4. Melaksanakan dan meningkatkan kegiatan pengabdian masyarakat oleh dosen dan mahasiswa dalam bidang Informatika.
5. Menyelenggarakan aktivitas yang mendukung pengembangan program studi dengan melibatkan dosen dan mahasiswa.
6. Menyelenggarakan kerja sama dengan lembaga tingkat nasional dan internasional.
7. Menciptakan kehidupan Islami di lingkungan program studi.

TATA TERTIB LABORATORIUM TEKNIK INFORMATIKA

DOSEN/KOORDINATOR PRAKTIKUM

1. Dosen harus hadir saat praktikum minimal 15 menit di awal kegiatan praktikum dan menandatangani presensi kehadiran praktikum.
2. Dosen membuat modul praktikum, soal seleksi asisten, pre-test, post-test, dan responsi dengan berkoordinasi dengan asisten dan pengampu mata praktikum.
3. Dosen berkoordinasi dengan koordinator asisten praktikum untuk evaluasi praktikum setiap minggu.
4. Dosen menandatangani surat kontrak asisten praktikum dan koordinator asisten praktikum.
5. Dosen yang tidak hadir pada slot praktikum tertentu tanpa pemberitahuan selama 2 minggu berturut-turut mendapat teguran dari Kepala Laboratorium, apabila masih berlanjut 2 minggu berikutnya maka Kepala Laboratorium berhak mengganti koordinator praktikum pada slot tersebut.

PRAKTIKAN

1. Praktikan harus hadir 15 menit sebelum kegiatan praktikum dimulai, dan dispensasi terlambat 15 menit dengan alasan yang jelas (kecuali asisten menentukan lain dan patokan jam adalah jam yang ada di Laboratorium, terlambat lebih dari 15 menit tidak boleh masuk praktikum & dianggap Inhal).
2. Praktikan yang tidak mengikuti praktikum dengan alasan apapun, wajib mengikuti INHAL, maksimal 4 kali praktikum dan jika lebih dari 4 kali maka praktikum dianggap GAGAL.
3. Praktikan harus berpakaian rapi sesuai dengan ketentuan Universitas, sebagai berikut:
 - a. Tidak boleh memakai Kaos Oblong, termasuk bila ditutupi Jaket/Jas Almamater (Laki-laki / Perempuan) dan Topi harus Dilepas.
 - b. Tidak Boleh memakai Baju ketat, Jilbab Minim dan rambut harus tertutup jilbab secara sempurna, tidak boleh kelihatan di jidat maupun di punggung (khusus Perempuan).
 - c. Tidak boleh memakai baju minim, saat duduk pun pinggang harus tertutup rapat (Laki-laki / Perempuan).
 - d. Laki-laki tidak boleh memakai gelang, anting-anting ataupun aksesoris Perempuan.
4. Praktikan tidak boleh makan dan minum selama kegiatan praktikum berlangsung, harus menjaga kebersihan, keamanan dan ketertiban selama mengikuti kegiatan praktikum atau selama berada di dalam laboratorium (tidak boleh membuang sampah sembarangan baik kertas, potongan kertas, bungkus permen baik di lantai karpet maupun di dalam ruang CPU).
5. Praktikan dilarang meninggalkan kegiatan praktikum tanpa seizin Asisten atau Laboran.
6. Praktikan harus meletakkan sepatu dan tas pada rak/loker yang telah disediakan.
7. Selama praktikum dilarang NGENET/NGE-GAME, kecuali mata praktikum yang membutuhkan atau menggunakan fasilitas Internet.
8. Praktikan dilarang melepas kabel jaringan atau kabel power praktikum tanpa sepengetahuan laboran
9. Praktikan harus memiliki FILE Petunjuk praktikum dan digunakan pada saat praktikum dan harus siap sebelum praktikum berlangsung.

10. Praktikan dilarang melakukan kecurangan seperti mencontek atau menyalin pekerjaan praktikan yang lain saat praktikum berlangsung atau post-test yang menjadi tugas praktikum.
11. Praktikan dilarang mengubah setting software/hardware komputer baik menambah atau mengurangi tanpa permintaan asisten atau laboran dan melakukan sesuatu yang dapat merugikan laboratorium atau praktikum lain.
12. Asisten, Koordinator Praktikum, Kepala laboratorium dan Laboran mempunyai hak untuk menegur, memperingatkan bahkan meminta praktikan keluar ruang praktikum apabila dirasa anda mengganggu praktikan lain atau tidak melaksanakan kegiatan praktikum sebagaimana mestinya dan atau tidak mematuhi aturan lab yang berlaku.
13. Pelanggaran terhadap salah satu atau lebih dari aturan diatas maka Nilai praktikum pada pertemuan tersebut dianggap 0 (NOL) dengan status INHAL.

ASISTEN PRAKTIKUM

1. Asisten harus hadir 15 Menit sebelum praktikum dimulai (konfirmasi ke koordinator bila mengalami keterlambatan atau berhalangan hadir).
2. Asisten yang tidak bisa hadir WAJIB mencari pengganti, dan melaporkan kepada Koordinator Asisten.
3. Asisten harus berpakaian rapi sesuai dengan ketentuan Universitas, sebagai berikut:
 - a. Tidak boleh memakai Kaos Oblong, termasuk bila ditutupi Jaket/Jas Almamater (Laki-laki / Perempuan) dan Topi harus Dilepas.
 - b. Tidak Boleh memakai Baju ketat, Jilbab Minim dan rambut harus tertutup jilbab secara sempurna, tidak boleh kelihatan di jidat maupun di punggung (khusus Perempuan).
 - c. Tidak boleh memakai baju minim, saat duduk pun pinggang harus tertutup rapat (Laki-laki / Perempuan).
 - d. Laki-laki tidak boleh memakai gelang, anting-anting ataupun aksesoris Perempuan.
4. Asisten harus menjaga kebersihan, keamanan dan ketertiban selama mengikuti kegiatan praktikum atau selama berada di laboratorium, menegur atau mengingatkan jika ada praktikan yang tidak dapat menjaga kebersihan, ketertiban atau kesopanan.
5. Asisten harus dapat merapikan dan mengamankan presensi praktikum, Kartu Nilai serta tertib dalam memasukan/Input nilai secara Online/Offline.
6. Asisten harus dapat bertindak secara profesional sebagai seorang asisten praktikum dan dapat menjadi teladan bagi praktikan.
7. Asisten harus dapat memberikan penjelasan/pemahaman yang dibutuhkan oleh praktikan berkenaan dengan materi praktikum yang diasistensi sehingga praktikan dapat melaksanakan dan mengerjakan tugas praktikum dengan baik dan jelas.
8. Asisten tidak diperkenankan mengobrol sendiri apalagi sampai membuat gaduh.
9. Asisten dimohon mengkoordinasikan untuk meminta praktikan agar mematikan komputer untuk jadwal terakhir dan sudah dilakukan penilaian terhadap hasil kerja praktikan.
10. Asisten wajib untuk mematikan LCD Projector dan komputer asisten/praktikan apabila tidak digunakan.
11. Asisten tidak diperkenankan menggunakan akses internet selain untuk kegiatan praktikum, seperti Youtube/Game/Medsos/Streaming Film di komputer praktikan.

LAIN-LAIN

1. Pada Saat Responsi Harus menggunakan Baju Kemeja untuk Laki-laki dan Perempuan untuk Praktikan dan Asisten.
2. Ketidakhadiran praktikum dengan alasan apapun dianggap INHAL.
3. Izin praktikum mengikuti aturan izin SIMERU/KULIAH.
4. Yang tidak berkepentingan dengan praktikum dilarang mengganggu praktikan atau membuat keributan/kegaduhan.
5. Penggunaan lab diluar jam praktikum maksimal sampai pukul 21.00 dengan menunjukkan surat ijin dari Kepala Laboratorium Prodi Teknik Informatika.

Yogyakarta, 22 Januari 2022

Kepala Laboratorium Komputasi Dasar



Adhi Prahara, S.Si., M.Cs.

NIK/NIY. 60150841

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR PENYUSUN	ii
HALAMAN REVISI	iii
HALAMAN PERNYATAAN	iv
VISI DAN MISI PRODI TEKNIK INFORMATIKA	v
TATA TERTIB LABORATORIUM TEKNIK INFORMATIKA	vi
DAFTAR ISI	ix
DAFTAR GAMBAR	x
PRAKTIKUM 1: PENGANTAR MATLAB	1
PRAKTIKUM 2: PEMROSESAN AWAL	18
PRAKTIKUM 3: EKSTRAKSI CIRI	25
PRAKTIKUM 4: KLASIFIKASI DENGAN GARIS LINEAR	28
PRAKTIKUM 5: KLASIFIKASI DENGAN METRIK JARAK	43
PRAKTIKUM 6: SISTEM PENGENALAN WAJAH	48
PRAKTIKUM 7: JARINGAN SYARAF TIRUAN UNTUK PENGENALAN POLA	55
PRAKTIKUM 8: ALGORITMA PERCEPTRON UNTUK PENGENALAN POLA	Error! Bookmark not defined.
PRAKTIKUM 9: ALGORITMA BACKPROPAGATION UNTUK PENGENALAN POLA	0
PRAKTIKUM 10: ALGORITMA LVQ UNTUK PENGENALAN POLA	Error! Bookmark not defined.
DAFTAR PUSTAKA	8

DAFTAR GAMBAR

Gambar 1. 1 Tampilan Command Window	Error! Bookmark not defined.
Gambar 1. 2 Layar Editor	5
Gambar 1. 3. Hasil output coba.m	7
Gambar 1. 4 Hasil skrip kasus1.m	10
Gambar 1. 5. Hasil skrip kasus2.m	12
Gambar 1. 6 Hasil skrip kasus5.m	14
Gambar 1. 7 Hasil Output kasus6.m	15
Gambar 1. 8 Hasil skrip kasus7.m	16
Gambar 3. 1 Citra tesktur makrosutruktur	26
Gambar 3. 2 citra teksutr mikrostruktur	26
Gambar 4. 1 Fungsi keputusan untuk mengklasifikasi dua jenis pola	34
Gambar 6. 1. Blok Diagram Sistem Pengenalan Wajah	49
Gambar 7. 1. Arsitektur Jaringan Syaraf Tiruan Sederhana	56
Gambar 9. 1 Arsitektur Backpropagation	1

PRAKTIKUM 1: PENGENALAN PYTHON DAN IDE

Pertemuan ke : 1

Total Alokasi Waktu : 90 menit

- Materi : 15 menit
- Pre-Test : 15 menit
- Praktikum : 45 menit
- Post-Test : 15 menit

Total Bobot Penilaian : 100%

- Pre-Test : 20 %
- Praktik : 30 %
- Post-Test : 50 %

Pemenuhan CPL dan CPMK:

CPL-03	Deskripsi dari CPL tsb...
CPMK-01	Deskripsi dari CPMK tsb...

1.1. DESKRIPSI CAPAIAN PEMBELAJARAN

Setelah mengikuti praktikum ini mahasiswa diharapkan mampu:

1. Menjelaskan syntax bahasa pemrograman python
2. Mengimplementasikan library pada python (numpy, matplotlib, pandas, sklearn)
3. Menggunakan IDE (collab, JupyterLab) sebagai environment pada bahasa pemrograman python

1.2. INDIKATOR KETERCAPAIAN PEMBELAJARAN

Indikator ketercapaian diukur dengan:

1. Mampu membuat program sederhana dengan python
2. Mampu mengenal dan memahami beberapa library pada python (numpy, matplotlib, pandas, sklearn)
3. Mampu mengenal dan menggunakan IDE (collab, JupyterLab) pada python

CPL-01	CPMK-01	Kemampuan mahasiswa dalam menerapkan ...
--------	---------	------------------------------------------

1.3. TEORI PENDUKUNG

Komputer dan Bahasa Pemrograman

Sebuah program membuat komputer dapat digunakan. Tanpa program, komputer tidak lebih dari sebuah objek.

Komputer mampu melakukan tugas yang sangat kompleks, tetapi kemampuan ini bukan bawaan. Sifat komputer sangat berbeda. Itu hanya dapat menjalankan operasi yang sangat sederhana, misalnya, komputer tidak dapat mengevaluasi nilai fungsi matematika yang rumit dengan sendirinya, meskipun ini tidak melampaui kemungkinan dalam waktu dekat.

Komputer kontemporer hanya dapat mengevaluasi hasil operasi yang sangat n menambah atau membagi, tetapi mereka dapat melakukannya dengan sangat c mengulangi tindakan ini hampir beberapa kali.

Bayangkan Anda ingin mengetahui kecepatan rata-rata yang telah Anda capai selama perjalanan panjang. Anda tahu jarak, Anda tahu waktu, Anda butuh kecepatan.

Secara alami, komputer akan dapat menghitung ini, tetapi komputer tidak menyadari hal-hal seperti jarak, kecepatan, atau waktu. Oleh karena itu, komputer perlu diinstruksikan untuk:

- menerima nomor yang mewakili jarak;
- menerima nomor yang mewakili waktu perjalanan;
- membagi nilai sebelumnya dengan yang terakhir dan menyimpan hasilnya dalam memori;
- menampilkan hasilnya (mewakili kecepatan rata-rata) dalam format yang dapat dibaca.

Empat tindakan sederhana ini membentuk sebuah **program**. Tentu saja, contoh-contoh ini tidak diformalkan, dan sangat jauh dari apa yang dapat dipahami komputer, tetapi cukup baik untuk diterjemahkan ke dalam bahasa yang dapat diterima komputer.

Bahasa adalah kata kuncinya.

Bahasa yang Anda gunakan setiap hari adalah bahasa ibu Anda, yang Anda gunakan untuk mewujudkan keinginan Anda dan untuk berpikir tentang kenyataan. Komputer juga memiliki bahasa mereka sendiri, yang disebut bahasa mesin, yang sangat sederhana.

Sebuah komputer, bahkan yang paling canggih secara teknis, tidak memiliki sedikitpun kecerdasan. Anda dapat mengatakan bahwa itu seperti anjing yang terlatih - ia hanya menanggapi serangkaian perintah yang telah ditentukan sebelumnya.

Perintah yang dikenalnya sangat sederhana. Kita dapat membayangkan bahwa komputer menanggapi perintah seperti "ambil nomor itu, bagi dengan yang lain dan simpan hasilnya".

Satu set lengkap perintah yang diketahui disebut daftar instruksi, kadang-kadang disingkat IL. Jenis komputer yang berbeda dapat bervariasi tergantung pada ukuran IL mereka, dan instruksinya bisa sangat berbeda dalam model yang berbeda.

Catatan: bahasa mesin dikembangkan oleh manusia.

Python sebagai Bahasa Pemrograman

Python adalah bahasa pemrograman yang banyak digunakan, ditafsirkan, berorientasi objek, dan tingkat tinggi dengan semantik dinamis, digunakan untuk pemrograman tujuan umum.

Python bukan bahasa baru. Python dikategorikan dalam bahasa pemrograman "dewasa" berdasarkan sejarah perkembangannya dan dapat dipercaya. Waktu yang dihabiskan untuk mempelajari Python adalah investasi yang sangat bagus.

Python Goals

Pada tahun 1999, Guido van Rossum menetapkan tujuannya untuk Python:

- **Easy and intuitive language**
Bahasa yang mudah dan intuitif sama kuatnya dengan bahasa pesaing utama.
- **Open source**, sehingga siapa saja dapat berkontribusi untuk pengembangannya.
- **Understandable code**
Kode yang dapat dimengerti seperti bahasa Inggris biasa.
- **Suitable for everyday tasks**
Cocok untuk tugas sehari-hari, memungkinkan waktu pengembangan yang singkat.

Python - Sejarah, Penemu, dan Asal Nama

Python dikembangkan oleh Guido van Rossum pada tahun 1990 di CWI, Amsterdam sebagai kelanjutan dari bahasa pemrograman ABC. Tentu saja, Guido van Rossum tidak mengembangkan dan mengembangkan sendiri semua komponen Python.

Python adalah alat, bukan reptil. Nama python bukan diambil dari sosok ular besar, nama bahasa pemrograman Python berasal dari serial sketsa komedi televisi BBC lama yang disebut Monty Python's Flying Circus. Pada puncak kesuksesannya, tim Monty Python menampilkan sketsa mereka kepada penonton langsung di seluruh dunia, termasuk di Hollywood Bowl. Karena Monty Python dianggap sebagai salah satu dari dua nutrisi mendasar bagi seorang programmer (yang lainnya adalah pizza), pencipta Python menamai bahasa tersebut untuk menghormati acara TV tersebut.

Nama Python dipilih oleh Guido sebagai nama bahasa ciptaannya karena kecintaan Guido pada acara televisi Monty Python's Flying Circus. Oleh karena itu seringkali ungkapan-ungkapan khas dari acara tersebut seringkali muncul dalam korespondensi antar pengguna Python.



Gambar 1.1 Creator of Python.

Apa yang membuat Python istimewa?

Beberapa pertanyaan seperti Bagaimana bisa programmer, tua dan muda, berpengalaman dan pemula, ingin menggunakannya? Bagaimana bisa perusahaan besar mengadopsi Python dan mengimplementasikan produk unggulan mereka dengan menggunakannya?

Ada banyak alasan, diantaranya:

- **Mudah dipelajari** - waktu yang dibutuhkan untuk mempelajari Python lebih singkat daripada banyak bahasa lain. Ini berarti memungkinkan untuk memulai pemrograman yang sebenarnya lebih cepat;

- **Mudah untuk mengajar** - beban kerja mengajar lebih kecil daripada yang dibahasa lain; ini berarti bahwa guru dapat lebih menekankan pada yang umur bergantung pada bahasa) teknik pemrograman, tidak membuang-buang energi yang eksotis, pengecualian aneh, dan aturan yang tidak dapat dipahami;
- **Mudah digunakan** untuk menulis perangkat lunak baru - seringkali memungkinkan untuk menulis kode lebih cepat saat menggunakan Python;
- **Mudah dimengerti** - seringkali juga lebih mudah untuk memahami kode orang lain lebih cepat jika ditulis dengan Python;
- **Mudah diperoleh, dipasang, dan disebar** - Python gratis, terbuka, dan multiplatform; tidak semua bahasa bisa membanggakan itu.



Gambar 1.2 Why python?.

Perjalanan Python Anda

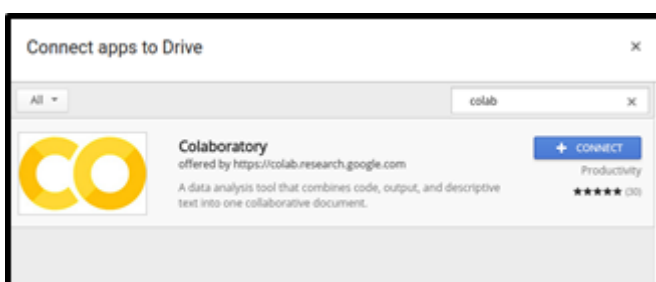
How to get Python and how to get to use it in Google Collab

Google Collaboratory atau Google Collab merupakan tools yang berbasis cloud dan free untuk tujuan penelitian. Google collab dibuat dengan environment jupyter dan mendukung hampir semua library yang dibutuhkan dalam lingkungan pengembangan Artificial Intelligence (AI).

Pada dasarnya google collab sama dengan Jupyter Notebook dan bisa dikatakan google collab adalah jupyter notebook yang dijalankan secara online dan gratis. Berikut adalah beberapa kelebihan dalam menggunakan google collab.

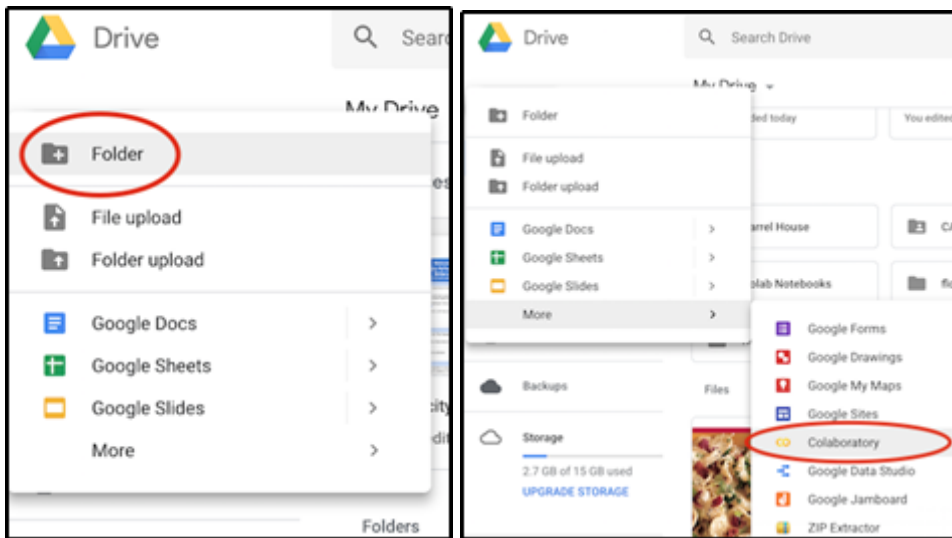
Install collab

Masuk ke dalam google drive kemudian di pojok kiri atas klik **New > More > Connect more apps**. Lalu pada kolom search ketikkan **“collab”** setelah muncul klik **connect**.



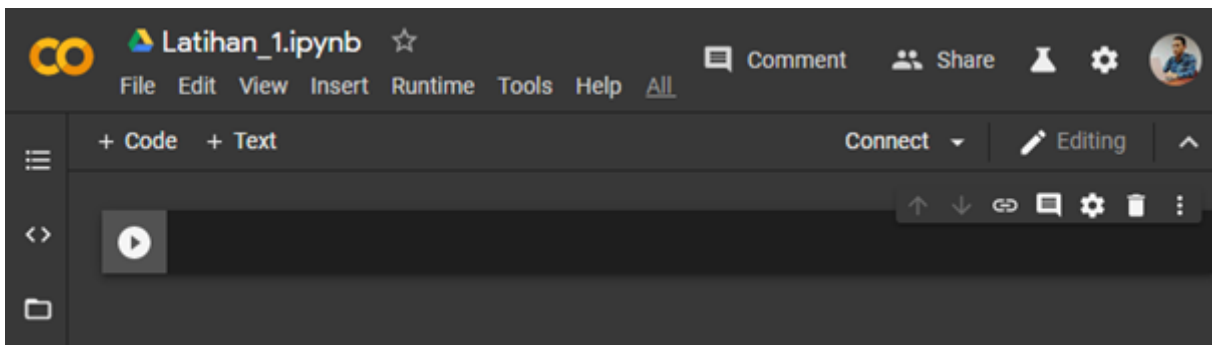
Buat folder project

Setelah berhasil instalasi, buat folder baru pada drive kemudian masuk kedalam folder tersebut. Kemudian **klik kanan > More > Collaboratory**.




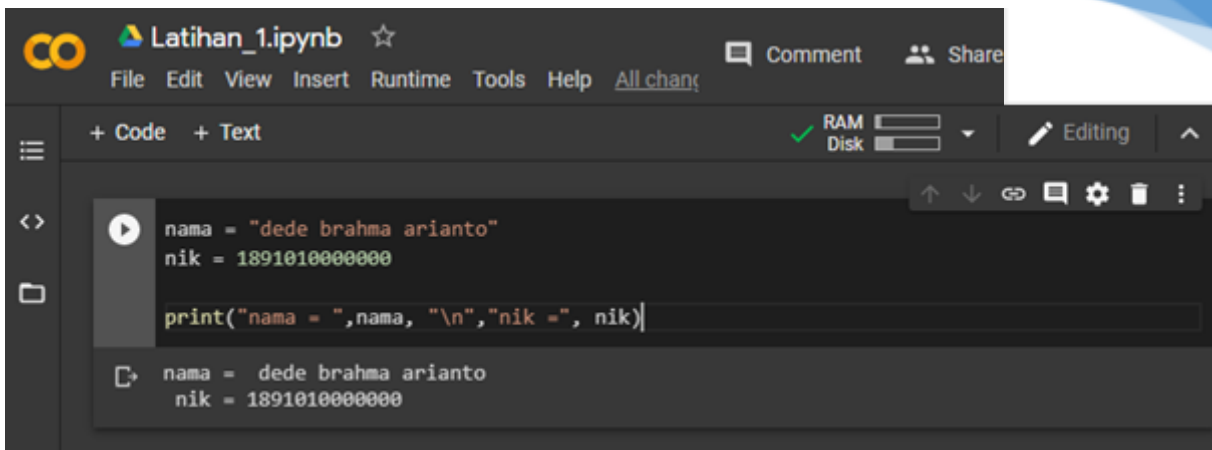
Tampilan collab

Kemudian akan muncul tampilan collab, ubah nama project dengan klik pada **nama project** yang berada di atas. Kemudian kita juga bisa mengatur tampilan atau tema sesuai yang kita inginkan. Pilih **tools > settings > Site**. Kemudian **atur tema dan save**.



Jalankan script

Selanjutnya buat script python sederhana kemudian jalankan. Cara menjalankan script pada google collab bisa dengan klik run . Atau juga dapat menggunakan **Ctrl + Enter**. Jika ingin menambahkan catatan gunakan **+Text** dan jika ingin menambahkan baris coding gunakan **+Code**.



The screenshot shows a Jupyter Notebook titled "Latihan_1.ipynb". The code cell contains the following Python code:

```
nama = "dede brahma arianto"
nik = 1891010000000

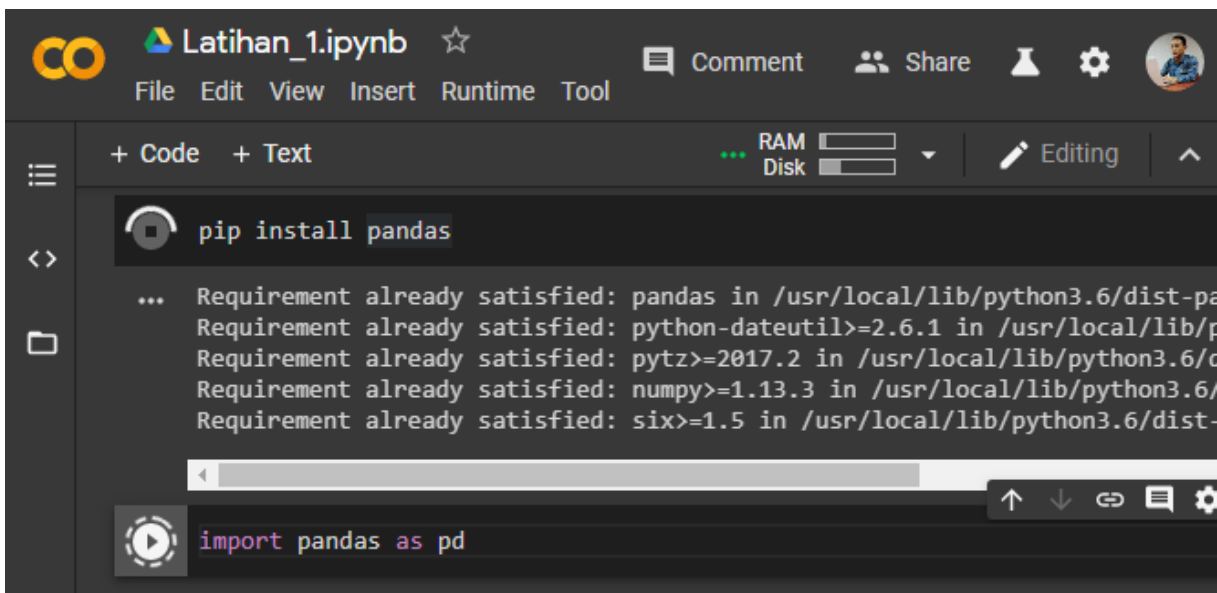
print("nama = ",nama, "\n","nik = ", nik)
```

The output cell shows the result of the print statement:

```
nama = dede brahma arianto
nik = 1891010000000
```

Install library

Jika kita mengalami error pada script yang dijalankan dan itu menggunakan library tertentu, maka pastikan bahwa library tersebut sudah terinstall. Untuk menambahkan library baru pada google collab, gunakan perintah "**pip install (nama_package)**". Kemudian import library yang sudah diinstall.



The screenshot shows a Jupyter Notebook titled "Latihan_1.ipynb". The code cell contains the command:

```
pip install pandas
```

The output cell shows the result of the command:

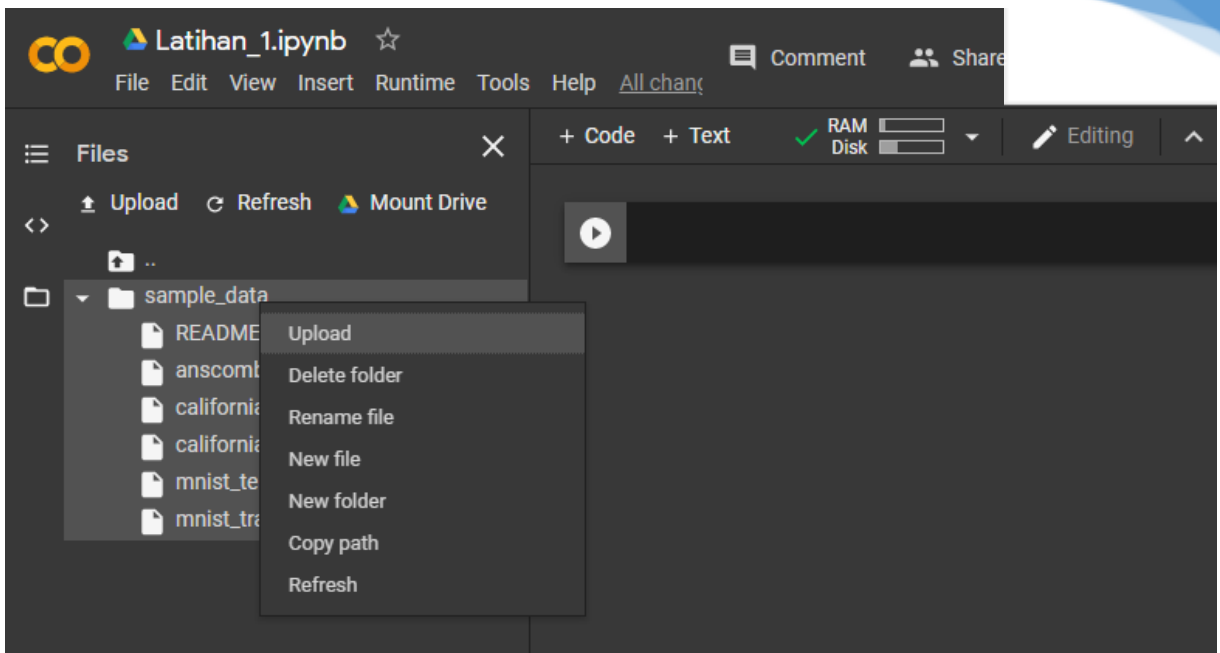
```
... Requirement already satisfied: pandas in /usr/local/lib/python3.6/dist-pa
Requirement already satisfied: python-dateutil>=2.6.1 in /usr/local/lib/p
Requirement already satisfied: pytz>=2017.2 in /usr/local/lib/python3.6/d
Requirement already satisfied: numpy>=1.13.3 in /usr/local/lib/python3.6/
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.6/dist-
```

The code cell below the output contains the import statement:

```
import pandas as pd
```

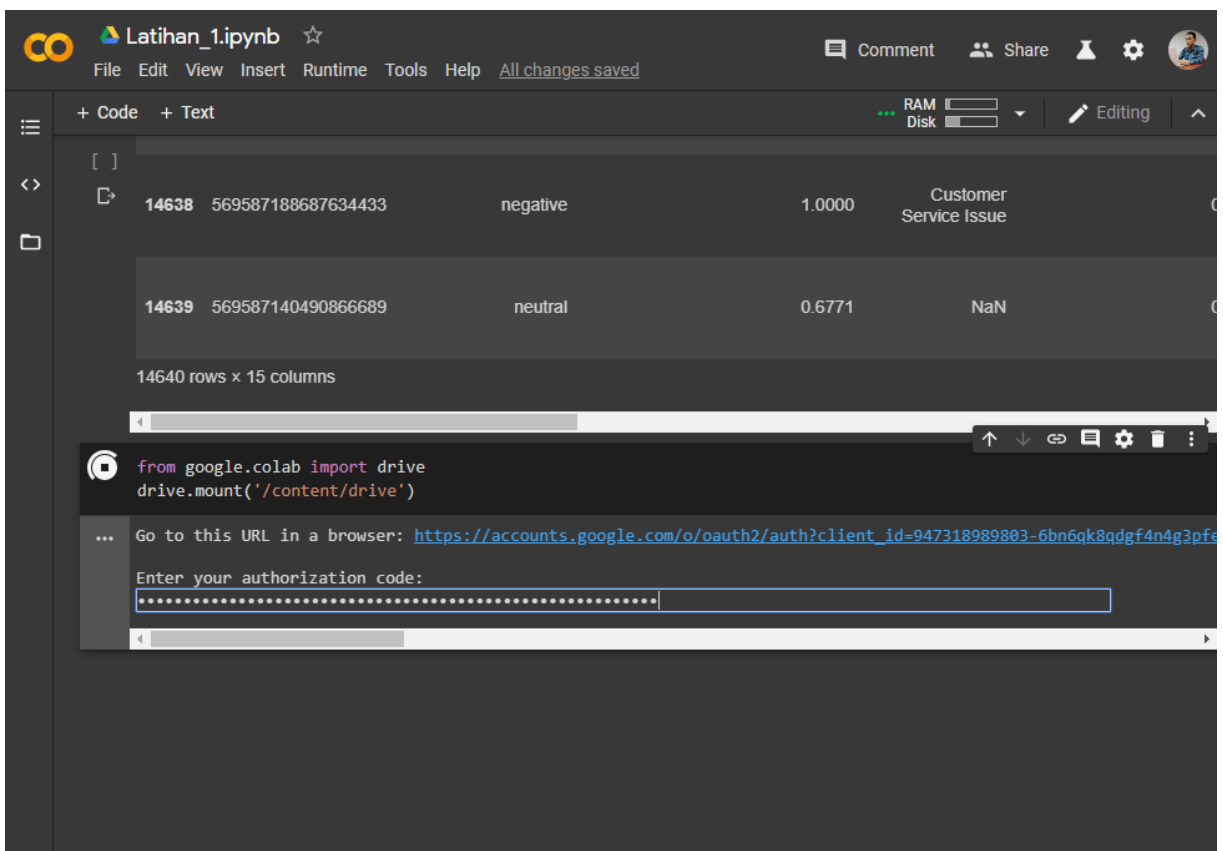
Upload file

Kita bisa upload file dataset yang kita punya ke dalam google collab, caranya klik pada **icon folder** pada sidebar sebelah kiri, kemudian masuk kedalam folder "**content/sample_data**" kemudian pada sample_data **klik kanan > upload**. Jika upload sudah selesai klik **refresh**.



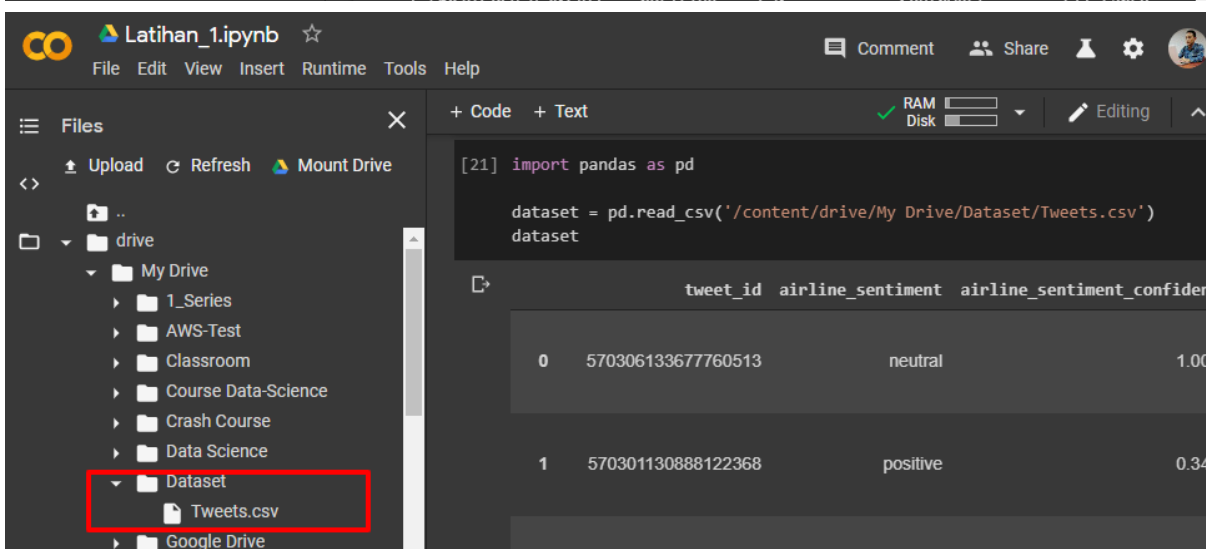
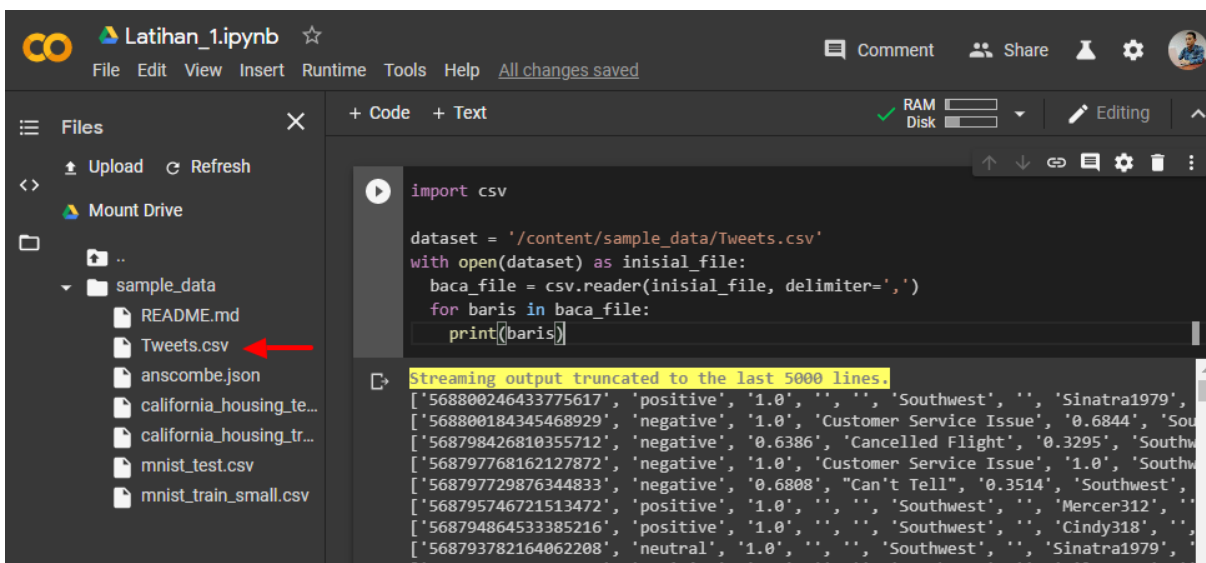
Mounting google drive

Cara lain untuk membaca file adalah dengan mengintegrasikan google drive kita ke dalam google colab. Untuk langkah awal kita perlu **upload file** ke dalam google drive. Kemudian jalankan **script mounting drive**, lalu pada output klik **URL link > izinkan > salin kode > masukkan** ke dalam script output.



Membaca file

Untuk dapat membaca file kita bisa menggunakan library **csv** ataupun menggunakan **path folder** tempat file tersimpan.



Starting your work with Python, Google Collab

Dengan kode yang simple dan mudah diimplementasikan, seorang programmer dapat lebih mengutamakan pengembangan aplikasi yang dibuat, bukan malah mencaai syntax error.

```
print ("Python sangat simple")
```

hanya dengan menuliskan kode print seperti diatas sudah bisa mencetak apapun yang diinginkan di dalam tanda kurung (). Di Bagian akhir kodenya tidak harus mengakhirinya dengan semicolon ;.

```

>>> print ("hello world")

hello word

>>> print "hello world python"
    
```

```
hello world python
```

How to write and run your very first program, Hello World

1.4. HARDWARE DAN SOFTWARE

Hardware dan software yang digunakan dalam praktikum ini yaitu:

1. Komputer.
2. Python 3.x

1.5. PRE-TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-03	CPMK-01	Apa keunggulan bahasa pemrograman python dengan bahasa pemrograman yang lain ?	25
2.			Apa kelebihan dan kelemahan dari IDE google collabs ?	25
3.			Apa kelebihan dan kelemahan dari IDE JupyterLab ?	25
4.			Jelaskan kegunaan dari library numpy, pandas, sklearn dan matplotlib	25

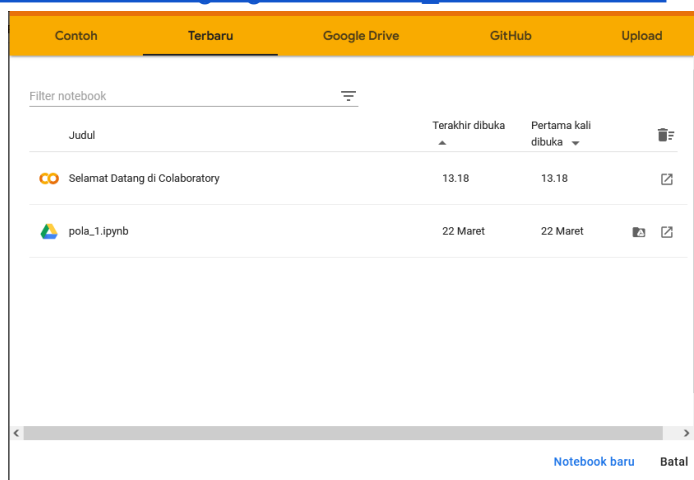
1.6. LANGKAH PRAKTIKUM

Aturan Penilaian (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Dokumen Pendukung	Skor
1.	CPL-03	CPMK-01	Selesaikan langkah praktikum 1 – 6	Hasil praktikum langkah 1 – 6	

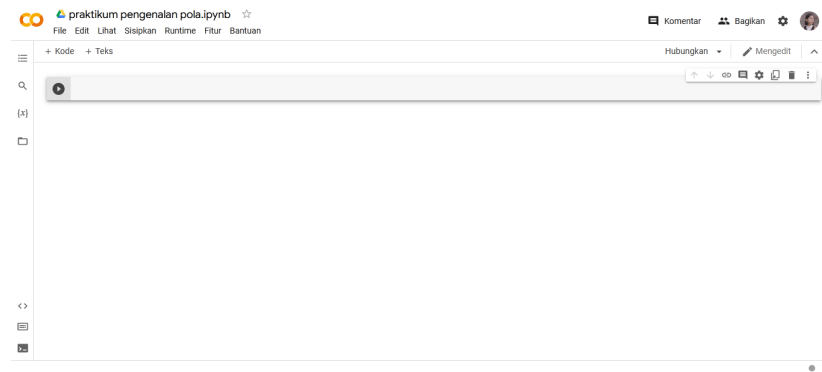
Langkah-Langkah Praktikum:

1. Google colabs.
Menjalankan google collabs dengan cara mengakses link :
https://colab.research.google.com/?utm_source=scs-index



Gambar 1. Halaman utama google collabs

Pada gambar 1 pilih “notebook baru” untuk membuat proyek terbaru

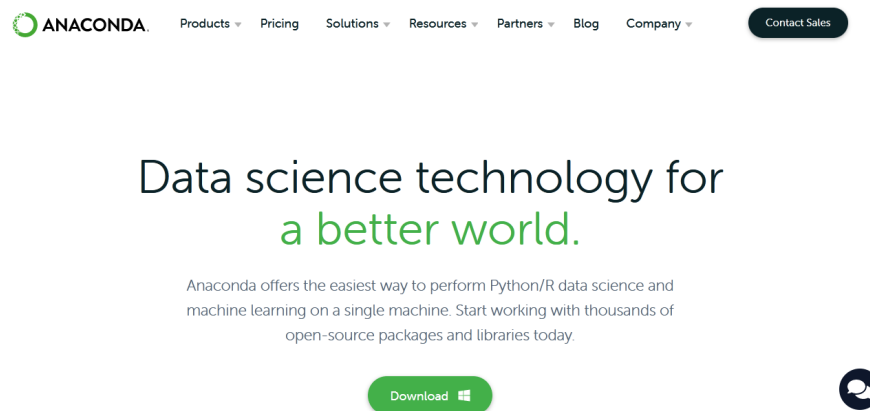


Gambar 2. Halaman notebook baru

Pada gambar 2 setelah masuk ke halaman notebook baru maka selanjutnya kalian dapat menuliskan kode program menggunakan bahasa pemrograman python

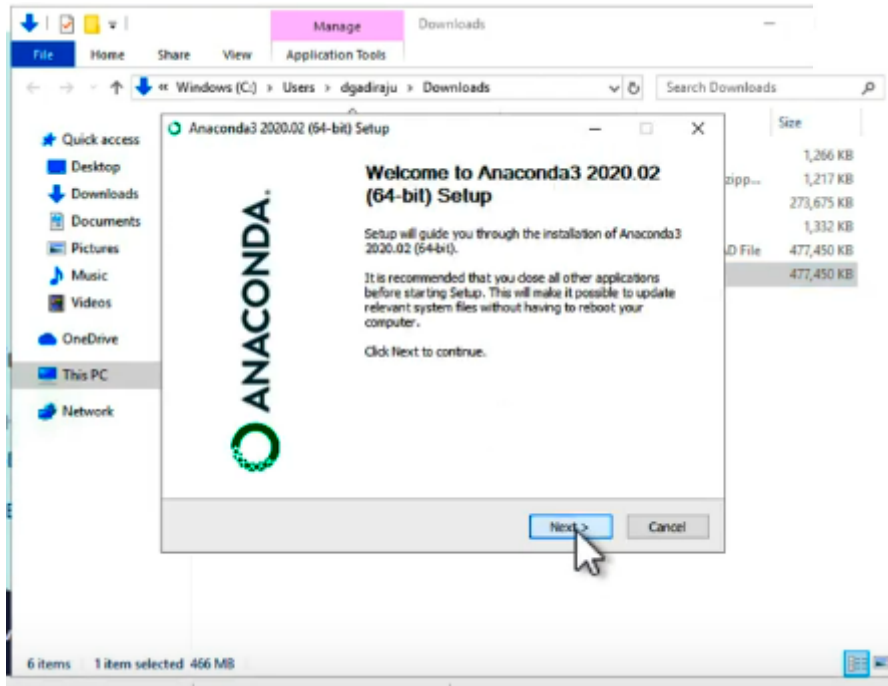
2. jupyterLab

Install anaconda menggunakan link : <https://www.anaconda.com/> lalu pilih menu download



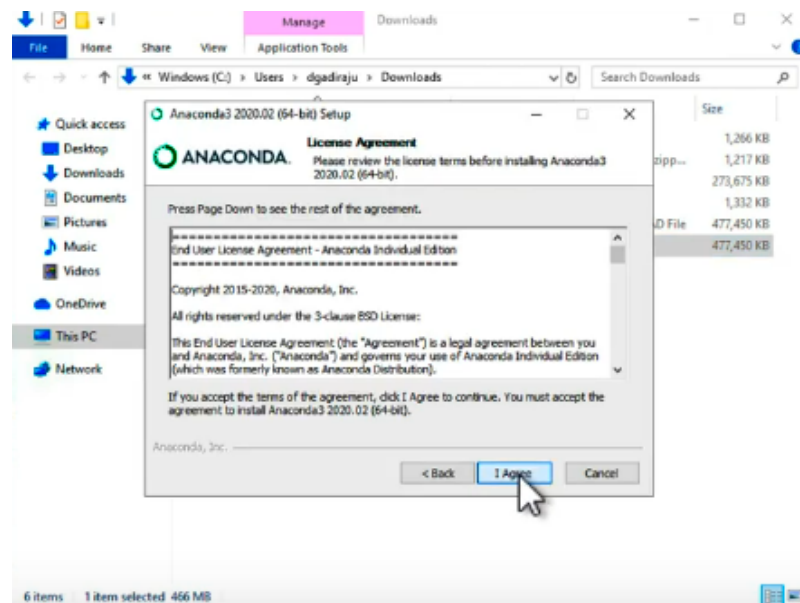
Gambar 3. Halaman download anaconda

Selanjutnya install anaconda pada komputer kalian, klik aplikasi anaconda yang sudah terunduh.



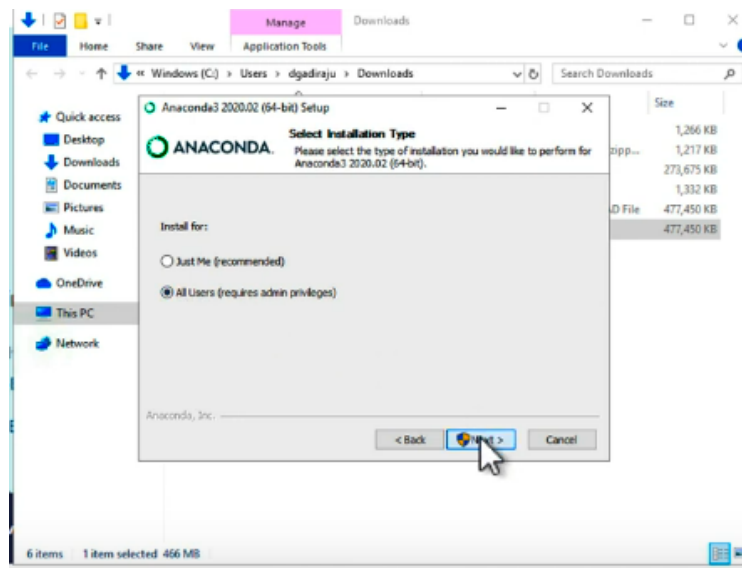
Gambar 4. Instalasi anaconda

Pada gambar 4 halaman utama pada saat ingin menginstall anaconda, pilih next untuk melanjutkan instalasi



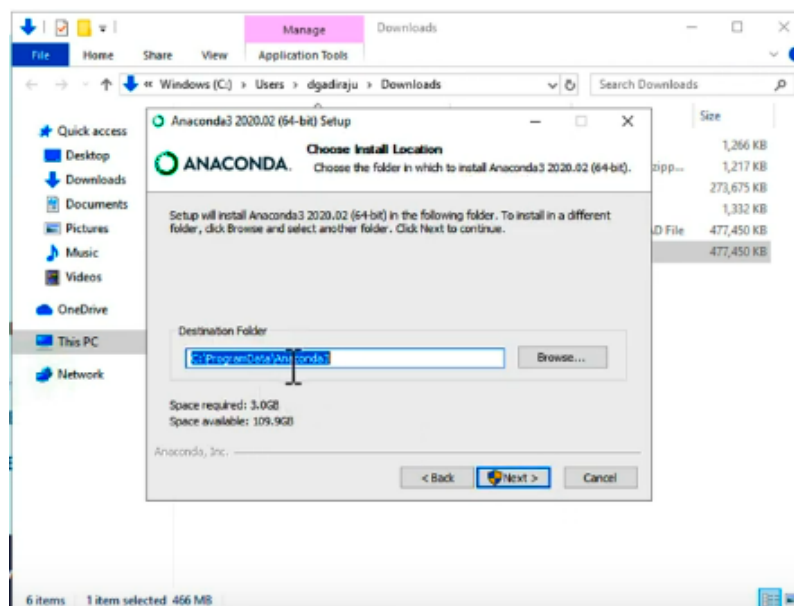
Gambar 5. License Agreement

Pada gambar 5 memilih "I Agree" untuk melanjutkan tahapan instalasi anaconda



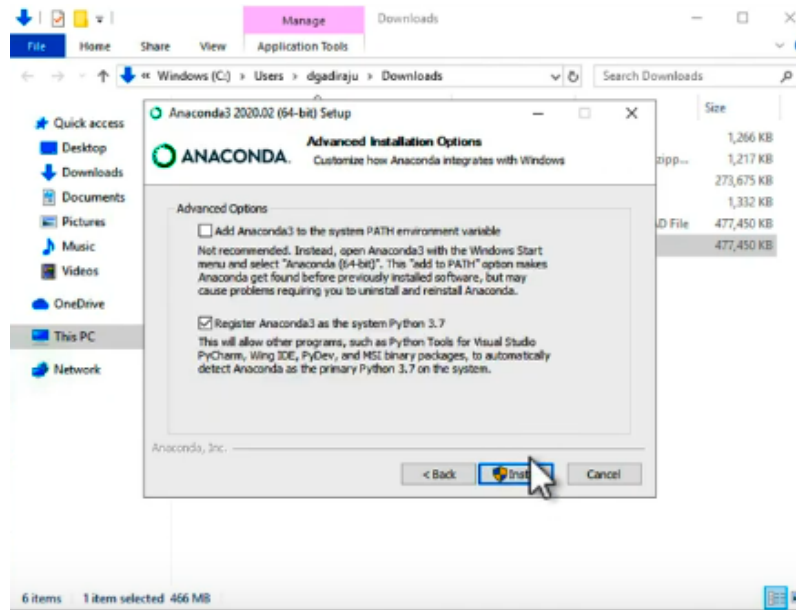
Gambar 6. Tipe Instalasi

Pada gambar 6 memilih tipe instalasi “just me” atau “All users” sesuai keinginan kalian, untuk praktikum kali ini memilih “All users” kemudian pilih next untuk melanjutkan instalasi



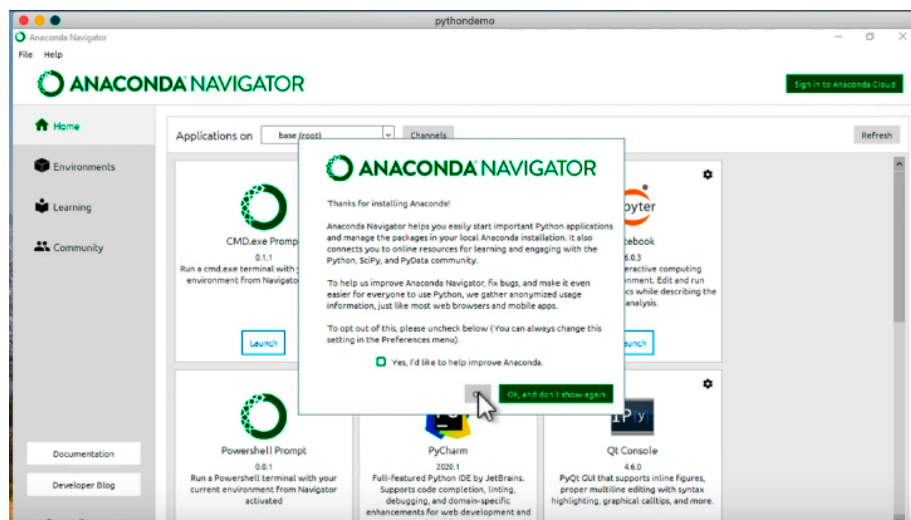
Gambar 7. Lokasi instalasi

Pada gambar 7 yaitu memilih lokasi untuk menyimpan data instalasi anaconda, kemudian pilih next untuk melanjutkan



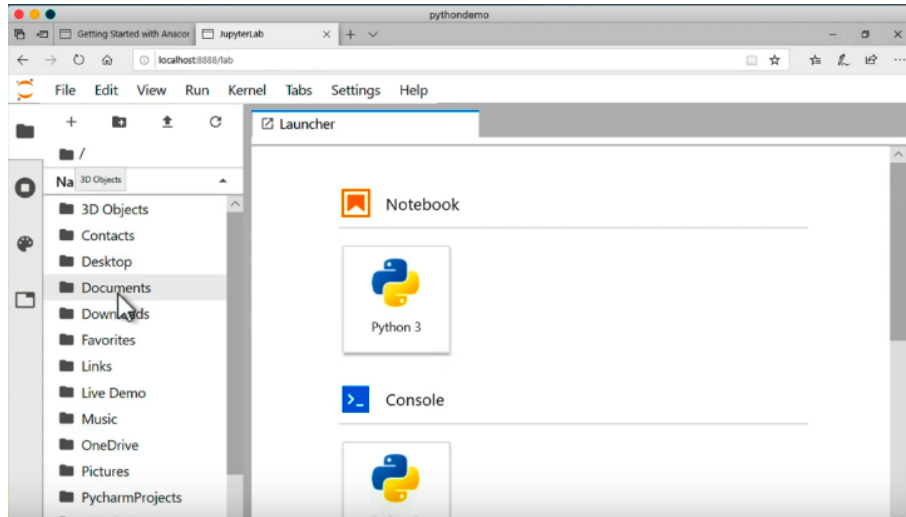
Gambar 8. Advanced Options

Pada gambar 8 yaitu memilih opsi lanjutan untuk instalasi, pada praktikum ini memilih semua opsi lalu klik "install", kemudian tunggu sampai tahap instalasi selesai



Gambar 9. Anaconda Navigator

Pada gambar 9 yaitu halaman utama pada aplikasi anaconda navigator yang terdapat beberapa tools, pada praktikum kali ini tools atau IDE yang digunakan adalah JupyterLab



Gambar 10. Halaman jupyterLab

Pada gambar 10 disini yaitu halaman awal saat membuka jupyterLab dan kalian dapat membuat beberapa proyek menggunakan bahasa pemrograman python

3. Pengenalan Library Pandas

Library pandas yaitu sebuah library di Python yang berlisensi BSD dan open source yang menyediakan struktur data dan analisis data yang mudah digunakan. Pandas biasa digunakan untuk membuat tabel, mengubah dimensi data, mengecek data, dan lain sebagainya

1) Import library pandas

Bertujuan untuk menggunakan package dan module pada library pandas

```
[1] import pandas as pd
```

2) Membuat sebuah dataframe dengan 3 kolom yang didalamnya berisi dua data

```
[2] data_list = list([["Faiz Fadh", "Computer Vision", 15_000_000],
                    ["Intan Desti", "Speech Recognition", 15_500_000]])

df = pd.DataFrame(data_list, columns=["Nama", "Job Desk", "Salary"])
```

3) Menampilkan dataframe yang sudah dibuat. Fungsi head() disini yaitu untuk menampilkan 5 data teratas.


```
[3] df.head()
```

	Nama	Job Desk	Salary
0	Faiz Fadh	Computer Vision	15000000
1	Intan Desti	Speech Recognition	15500000

4. Pengenalan Library Numpy

NumPy merupakan salah satu library Python yang berfungsi untuk proses komputasi numerik. NumPy memiliki kemampuan untuk membuat objek N-dimensi array. Array merupakan sekumpulan variabel yang memiliki tipe data yang sama.

1) Import library numpy

```
[ ] import numpy as np
```

2) Membuat sebuah data array 2 dimensi

```
[ ] data_list = list([[1, 2, 3, 4, 5], [6, 7, 8, 9, 10]])
np_data = np.array(data_list)
```

3) Menampilkan data list pada variabel data_list

```
[ ] data_list
[[1, 2, 3, 4, 5], [6, 7, 8, 9, 10]]
```

4) Menampilkan data list yang diubah kedalam array

```
[ ] np_data
array([[ 1,  2,  3,  4,  5],
       [ 6,  7,  8,  9, 10]])
```

5) Contoh perhitungan pada data array

```
[ ] np_data * 2
array([[ 2,  4,  6,  8, 10],
       [12, 14, 16, 18, 20]])
```

6) Menjadikan data array menjadi 1 dimensi

```
[ ] np_data.flatten()
array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10])
```

7) Membuat contoh array lainnya

```
[ ] c_array = np.array([[1.5,2,3], (4,5,6)],[(3,2,1), (4,5,6)]], dtype = float)
c_array
array([[1.5, 2. , 3. ],
       [4. , 5. , 6. ]],
      [[3. , 2. , 1. ],
       [4. , 5. , 6. ]])
```

8) Membuat matrix ordo 3x4 yang berisi nilai 0

```
[ ] np.zeros((3,4)) # Membuat array matriks 2 x 4 yang berisi nilai 0
array([[0., 0., 0., 0.],
       [0., 0., 0., 0.],
       [0., 0., 0., 0.]])
```

9) Membuat sebuah 2 matrix dengan ordo 3x4 yang berisi nilai 1

```
[ ] np.ones((2,3,4),dtype=np.int16) # Membuat array 2 buah matriks 3 x 4 yang berisi nilai 1
array([[[1, 1, 1, 1],
       [1, 1, 1, 1],
       [1, 1, 1, 1]],
      [[1, 1, 1, 1],
       [1, 1, 1, 1],
       [1, 1, 1, 1]]], dtype=int16)
```

10) Membuat sebuah array constant dengan ordo 2x2

```
[ ] e = np.full((2,2),7) # Create a constant array
e
array([[7, 7],
       [7, 7]])
```

11) Membuat array berisikan nilai biner dengan ordo 2x2

```
[ ] f = np.eye(2) # Create a 2X2 identity matrix
f
array([[1., 0.],
       [0., 1.]])
```

5. Pengenalan Library Matplotlib

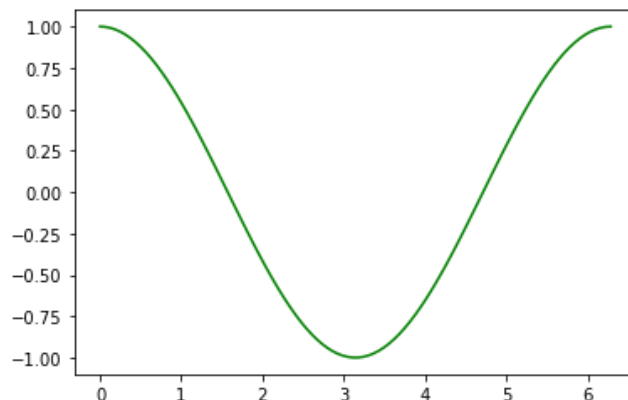
Matplotlib adalah Library berbasis Python untuk kepentingan visualisasi data . Dengan matplotlib kita dapat membuat plot, histogram, diagram batang dan jenis diagram lainnya hanya dengan beberapa baris kode.

1) Import library matplotlib

```
[ ] import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
```

2) Membuat sebuah contoh grafik

```
X = np.linspace(0, 2*np.pi, 100)
Y = np.cos(X)
fig, ax = plt.subplots()
ax.plot(X, Y, color="green")
fig.savefig("figure.pdf")
fig.show()
```



3) Import library matplotlib untuk membuat contoh grafik lainnya

```
[ ] import matplotlib.pyplot as plt
import numpy as np
from matplotlib import colors
from matplotlib.ticker import PercentFormatter
```

4) Contoh lain membuat grafik pada python

```
# Create a random number generator with a fixed seed for reproducibility
rng = np.random.default_rng(19680801)

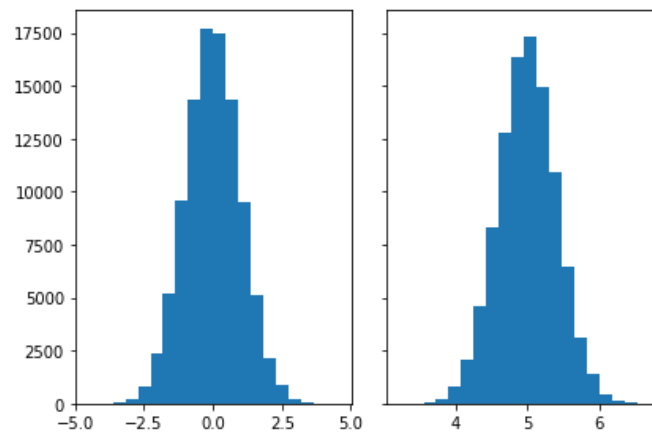
N_points = 100000
n_bins = 20

# Generate two normal distributions
dist1 = rng.standard_normal(N_points)
dist2 = 0.4 * rng.standard_normal(N_points) + 5

fig, axs = plt.subplots(1, 2, sharey=True, tight_layout=True)

# We can set the number of bins with the *bins* keyword argument.
axs[0].hist(dist1, bins=n_bins)
axs[1].hist(dist2, bins=n_bins)
```

```
(array([2.0000e+00, 2.1000e+01, 5.1000e+01, 2.3500e+02, 7.8100e+02,
        2.1000e+03, 4.5730e+03, 8.3390e+03, 1.2758e+04, 1.6363e+04,
        1.7345e+04, 1.4923e+04, 1.0920e+04, 6.4830e+03, 3.1070e+03,
        1.3810e+03, 4.5300e+02, 1.2200e+02, 3.6000e+01, 7.0000e+00]),
 array([3.20889223, 3.38336526, 3.55783829, 3.73231132, 3.90678435,
        4.08125738, 4.25573041, 4.43020344, 4.60467647, 4.7791495 ,
        4.95362253, 5.12809556, 5.30256859, 5.47704162, 5.65151465,
        5.82598768, 6.00046071, 6.17493374, 6.34940677, 6.5238798 ,
        6.69835283]),
 <a list of 20 Patch objects>)
```



6. Pengenalan Library SKlearn

Scikit-learn atau sklearn merupakan sebuah module dari bahasa pemrograman Python yang dibangun berdasarkan NumPy, SciPy, dan Matplotlib. Fungsi dari module ini adalah untuk membantu melakukan processing data ataupun melakukan training data untuk kebutuhan machine learning atau data science.

1) Import library sklearn

```
[ ] from sklearn.datasets import load_boston
```

2) Mengambil dataset boston dari python

```
[ ] boston_dataset = load_boston()
    boston_dataset
```

3) Menampilkan dataset boston

```
[ ] boston_dataset.target

array([24. , 21.6, 34.7, 33.4, 36.2, 28.7, 22.9, 27.1, 16.5, 18.9, 15. ,
       18.9, 21.7, 20.4, 18.2, 19.9, 23.1, 17.5, 20.2, 18.2, 13.6, 19.6,
       15.2, 14.5, 15.6, 13.9, 16.6, 14.8, 18.4, 21. , 12.7, 14.5, 13.2,
       13.1, 13.5, 18.9, 20. , 21. , 24.7, 30.8, 34.9, 26.6, 25.3, 24.7,
       21.2, 19.3, 20. , 16.6, 14.4, 19.4, 19.7, 20.5, 25. , 23.4, 18.9,
       35.4, 24.7, 31.6, 23.3, 19.6, 18.7, 16. , 22.2, 25. , 33. , 23.5,
       19.4, 22. , 17.4, 20.9, 24.2, 21.7, 22.8, 23.4, 24.1, 21.4, 20. ,
       20.8, 21.2, 20.3, 28. , 23.9, 24.8, 22.9, 23.9, 26.6, 22.5, 22.2,
       23.6, 28.7, 22.6, 22. , 22.9, 25. , 20.6, 28.4, 21.4, 38.7, 43.8,
       33.2, 27.5, 26.5, 18.6, 19.3, 20.1, 19.5, 19.5, 20.4, 19.8, 19.4,
       21.7, 22.8, 18.8, 18.7, 18.5, 18.3, 21.2, 19.2, 20.4, 19.3, 22. ,
       20.3, 20.5, 17.3, 18.8, 21.4, 15.7, 16.2, 18. , 14.3, 19.2, 19.6,
       23. , 18.4, 15.6, 18.1, 17.4, 17.1, 13.3, 17.8, 14. , 14.4, 13.4,
       15.6, 11.8, 13.8, 15.6, 14.6, 17.8, 15.4, 21.5, 19.6, 15.3, 19.4,
```

4) Membagi dataset menjadi data training dan data testing dengan perbandingan 70% untuk data training dan 30% untuk data testing

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(boston_dataset.data,
                                                  boston_dataset.target, test_size=0.3)
```

5) Menampilkan besaran dimensi datanya. Contoh untuk X_train terdapat 354 baris dan 13 kolom

```
[ ] X_train.shape

(354, 13)
```

```
[ ] X_test.shape

(152, 13)
```

```
[ ] y_train.shape
(354,)
```

```
[ ] y_test.shape
(152,)
```

1.7. POST TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-03	CPMK-01	buatlah pandas dataframe berupa biodata anggota kelompok kalian (nama, nim, alamat, makanan favorit, minuman favorit dan cita-cita) serta buatlah fungsi export ke csv dan excel	50
2.			buatlah perkalian matrix dari dataset yang diberikan asisten	50

1.8. HASIL CAPAIAN PRAKTIKUM

Diisi oleh asisten setelah semua assessment dinilai.

No	Bentuk Assessment	CPL	CPMK	Bobot	Skor (0-100)	Nilai Akhir (Bobot x Skor)
1.	Pre-Test	CPL-03	CPMK-01	20%		
2.	Praktik	CPL-03	CPMK-01	30%		
3.	Post-Test	CPL-03	CPMK-01	50%		
Total Nilai						

LEMBAR JAWABAN PRE-TEST DAN POST-TEST PRAKTIKUM

<p>Nama : NIM :</p>	<p>Asisten: Paraf Asisten:</p>	<p>Tanggal: Nilai:</p>
---------------------------------------	--------------------------------------------------	------------------------------------------

- Pertemuan ke** : 2
- Total Alokasi Waktu** : 90 menit
- Materi : 15 menit
 - Pre-Test : 15 menit
 - Praktikum : 45 menit
 - Post-Test : 15 menit

- Total Bobot Penilaian** : 100%
- Pre-Test : 20 %
 - Praktik : 30 %
 - Post-Test : 50 %

Pemenuhan CPL dan CPMK:

CPL-03	Deskripsi dari CPL tsb...
CPMK-01	Deskripsi dari CPMK tsb...

2.1. DESKRIPSI CAPAIAN PEMBELAJARAN

Setelah mengikuti praktikum ini mahasiswa diharapkan mampu:

1. Melakukan preprocessing untuk text, gambar dan numerik
2. Menentukan teknik yang tepat untuk melakukan preprocessing

2.2. INDIKATOR KETERCAPAIAN PEMBELAJARAN

Indikator ketercapaian diukur dengan:

CPL-01	CPMK-01	Kemampuan mahasiswa dalam menerapkan preprocessing pada dataset yang dimiliki
--------	---------	-------------------------------------------------------------------------------

2.3 TEORI PENDUKUNG

Data Preprocessing yaitu proses mengubah dari data mentah atau raw data menjadi data yang lebih efisien sehingga dapat di olah ke tahap selanjutnya. Data Preprocessing yaitu proses yang penting dan harus dilakukan karena data mentah sering kali memiliki format yang tidak teratur. Data preprocessing dapat dilakukan pada data text, numeric dan image.

Manfaat Data Preprocessing yaitu sebagai berikut:

- 1) membuat data lebih mudah dibaca
- 2) mengurangi beban representasi data
- 3) mengurangi data mining secara signifikan
- 4) mempermudah proses analisis dalam Machine Learning

Setelah mengetahui tentang apa itu data preprocessing, ada beberapa step yang ketika akan melakukan data preprocessing. Berikut ini beberapa tahapannya:

1. Data cleaning

Tahap pertama yang perlu dilakukan ketika akan preprocessing data adalah data cleaning atau membersihkan data. Artinya, data mentah yang telah Anda peroleh perlu diseleksi kembali. Kemudian, hapus atau hilangkan data-data yang tidak lengkap, tidak relevan, dan tidak akurat. Dengan melakukan tahap ini, Anda akan menghindari kesalahpahaman ketika menganalisis data tersebut.

Ada dua hal yang harus Anda perhatikan ketika melakukan data cleaning, yakni pastikan data-data yang dikumpulkan tidak lagi mengandung data dengan missing values. Lalu, Anda juga harus memastikan bahwa data-data tersebut seluruhnya diperlukan saat proses analisis data. Dengan demikian, data yang Anda kumpulkan telah disesuaikan dan tidak mubazir.

2. Data Integration

Karena data preprocessing akan menggabungkan beberapa data dalam suatu dataset, maka Anda harus mengecek data-data yang datang dari berbagai sumber tersebut supaya memiliki format yang sama. Proses ini menjadi salah satu step penting dalam proses ini.

Beberapa permasalahan bisa muncul ketika melakukan data integration. Misalnya, Anda ingin menggabungkan data dari beberapa sumber. Anda harus mengetahui bahwa data pada sumber pertama dimiliki oleh si A, dan data pada sumber kedua juga terkait dengan si A. Kelihatannya seperti hal mudah, padahal dua sumber tersebut memiliki format yang berbeda. Itulah yang membuat data integration sedikit lebih rumit.

3. Transformasi data

Proses berikutnya yang harus dilakukan adalah transformasi data. Seperti yang telah dijelaskan di atas, data akan diambil dari berbagai sumber yang kemungkinan memiliki perbedaan format. Anda harus menyamakan seluruh data yang terkumpul supaya dapat mempermudah proses analisis data.

Misalnya, Anda akan mengambil data karyawan pada sumber pertama yang menggunakan format DD/MM/YYYY. Kemudian, pada sumber berikutnya, data karyawan menggunakan format MM/DD/YYYY. Ketika akan mengumpulkan data, keduanya tentu perlu diubah dan diseragamkan dalam satu format yang sama.

4. Mengurangi Data

Tahap terakhir yang perlu dilakukan adalah mengurangi jumlah data (data reduction). Maksudnya adalah Anda harus mengurangi sampel data yang diambil, tetapi dengan catatan, tidak akan mengubah hasil analisis data.

Ada tiga teknik yang bisa diterapkan saat melakukan pengurangan data, yakni dimensionality reduction (pengurangan dimensi), numerosity reduction (pengurangan jumlah), dan data compression (kompresi data). Ketiga teknik tersebut bisa disesuaikan dengan kebutuhan; apakah data yang diolah besar, sedang, atau perlu dikompresi dan berisiko merugikan.

2.4. HARDWARE DAN SOFTWARE

Hardware dan software yang digunakan dalam praktikum ini yaitu:

1. Komputer.
2. IDE Python (Google collab, Jupyter Notebook)
3. Python 3.x

2.5. PRE-TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-03	CPMK-01	Sebutkan teknik preprocessing pada data text !	
2.	CPL-03	CPMK-01	Sebutkan teknik preprocessing pada data gambar !	
3.	CPL-03	CPMK-01	Sebutkan teknik preprocessing pada data numerik !	
4.	CPL-03	CPMK-01	Apa pengaruh dari penggunaan preprocessing dibandingkan tidak menggunakan preprocessing ?	

2.6. LANGKAH PRAKTIKUM

Aturan Penilaian (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Dokumen Pendukung	Skor
1.	CPL-03	CPMK-01	Selesaikan langkah praktikum 1 – 3	Hasil praktikum langkah 1 – 3	20
2.	CPL-03	CPMK-01	Kerjakan soal pada link berikut :		80

Langkah-Langkah Praktikum:

1. Text Preprocessing

Text preprocessing adalah suatu proses untuk menyeleksi data text agar menjadi lebih terstruktur lagi dengan melalui serangkaian tahapan.

1) Import library

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn import svm
```

2) Mengimport dataset pesan 'spam.csv' untuk dilakukan Text Preprocessing

```
# import data spam.csv
spam = pd.read_csv('spam.csv')
```

3) Membagi data menjadi dua versi yaitu untuk v1 adalah data fitur dan v2 data class/label

```
z = spam['v2'] # data feature
y = spam["v1"] # data class / label
# membagi menjadi 4 bagian, dengan data train 80% dan test 20%
z_train, z_test, y_train, y_test = train_test_split(z, y, test_size = 0.2)
```

- 4) Melakukan proses ekstraksi fitur menggunakan Count Vectorizer

```
cv = CountVectorizer() # membuat driver untuk ekstraksi fitur
features = cv.fit_transform(z_train)
```

- 5) Menghitung distribusi kelas dengan mencari jumlah ham (pesan yang diinginkan) dan jumlah spam (pesan yang tidak diinginkan) pada dataset

```
data = spam[['v1', 'v2']] # mengambil kolom v1 dan v2 pada variabel spam dan disimpan ke dalam variabel data
data['v1'].value_counts() # menghitung distribusi kelas nya (ham dengan spam)

ham      4825
spam     747
Name: v1, dtype: int64
```

- 6) Import library untuk manipulasi string dengan menampilkan tanda baca

```
import string
string.punctuation

'!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~'
```

- 7) Membuat sebuah fungsi untuk menghapus tanda baca pada pesan

```
def remove_punctuation(text):
    punctuationfree="".join([i for i in text if i not in string.punctuation])
    return punctuationfree

data['clean_msg'] = data['v2'].apply(lambda x: remove_punctuation(x))
data.head()
```

Hasil menghilangkan pesan disimpan pada kolom 'clean_msg'

	v1	v2	clean_msg
0	ham	Go until jurong point, crazy.. Available only ...	Go until jurong point crazy Available only in ...
1	ham	Ok lar... Joking wif u oni...	Ok lar Joking wif u oni
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	Free entry in 2 a wkly comp to win FA Cup fina...
3	ham	U dun say so early hor... U c already then say...	U dun say so early hor U c already then say
4	ham	Nah I don't think he goes to usf, he lives aro...	Nah I dont think he goes to usf he lives aroun...

- 8) Membuat setiap baris kalimat pada pesan menjadi huruf kecil dengan menggunakan fungsi lambda dan disimpan pada kolom 'msg_lower'

```
data['msg_lower'] = data['clean_msg'].apply(lambda x: x.lower());
```

- 9) Membuat fungsi untuk melakukan proses tokenisasi pada kalimat di dalam kolom 'msg_lower' yang akan disimpan pada kolom 'msg_tokenied'

```
import re
def tokenization(text):
    tokens = re.split('W+',text)
    return tokens

data['msg_tokenied']= data['msg_lower'].apply(lambda x: tokenization(x))
```

- 10) Import library untuk proses stopwords

```
import nltk
nltk.download('stopwords')

stopwords = nltk.corpus.stopwords.words('english')
stopwords[0:10]

['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're"]
```

- 11) Membuat sebuah fungsi untuk menghapus stopwords pada data pesan

```
def remove_stopwords(text):
    output= [i for i in text if i not in stopwords]
    return output
```

- 12) Melakukan proses menghapus stopwords pada kolom 'msg_tokenied' yang hasilnya disimpan pada kolom 'no_stopwords'

```
data['no_stopwords']= data['msg_tokenied'].apply(lambda x:remove_stopwords(x))
```

- 13) Import library untuk proses stemming

```
from nltk.stem.porter import PorterStemmer

porter_stemmer = PorterStemmer()
```

- 14) Melakukan proses stemming pada kolom dari proses stopwords yang hasilnya disimpan pada kolom 'msg_stemmed'

```
def stemming(text):
    stem_text = [porter_stemmer.stem(word) for word in text]
    return stem_text

data['msg_stemmed'] = data['no_stopwords'].apply(lambda x: stemming(x))
```

Menampilkan data dari hasil teks preprocessing yang dilakukan

```
data.head()
```

	v1	v2	clean_msg	msg_lower	msg_tokenied	no_stopwords	msg_stemmed
0	ham	Go until jurong point, crazy.. Available only in ...	Go until jurong point crazy Available only in ...	go until jurong point crazy available only in ...	[go until jurong point crazy available only in...	[go until jurong point crazy available only in...	[go until jurong point crazy available only in...
1	ham	Ok lar... Joking wif u oni...	Ok lar Joking wif u oni	ok lar joking wif u oni	[ok lar joking wif u oni]	[ok lar joking wif u oni]	[ok lar joking wif u oni]
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	Free entry in 2 a wkly comp to win FA Cup fina...	free entry in 2 a wkly comp to win fa cup fina...	[free entry in 2 a wkly comp to win fa cup fin...	[free entry in 2 a wkly comp to win fa cup fin...	[free entry in 2 a wkly comp to win fa cup fin...
3	ham	U dun say so early hor... U c already then say...	U dun say so early hor U c already then say	u dun say so early hor u c already then say	[u dun say so early hor u c already then say]	[u dun say so early hor u c already then say]	[u dun say so early hor u c already then say]
4	ham	Nah I dont think he goes to usf, he lives aro...	Nah I dont think he goes to usf he lives aroun...	nah i dont think he goes to usf he lives aroun...	[nah i dont think he goes to usf he lives arou...	[nah i dont think he goes to usf he lives arou...	[nah i dont think he goes to usf he lives arou...

15) Import library untuk proses lemitisasi

```

nlTK.download('wordnet')
from nlTK.stem import WordNetLemmatizer

wordnet_lemmatizer = WordNetLemmatizer()

```

16) Melakukan proses lemitisasi

```

def lemmatizer(text):
    lemm_text = [wordnet_lemmatizer.lemmatize(word) for word in text]
    return lemm_text

data['msg_lemmatized']=data['no_stopwords'].apply(lambda x:lemmatizer(x))

```

Hasil dari teks preprocessing yang sudah dilakukan

```
data.head()
```

	v1	v2	clean_msg	msg_lower	msg_tokenied	no_stopwords	msg_stemmed	msg_lemmatized
0	ham	Go until jurong point, crazy.. Available only in ...	Go until jurong point crazy Available only in ...	go until jurong point crazy available only in ...	[go until jurong point crazy available only in...	[go until jurong point crazy available only in...	[go until jurong point crazy available only in...	[go until jurong point crazy available only in...
1	ham	Ok lar... Joking wif u oni...	Ok lar Joking wif u oni	ok lar joking wif u oni	[ok lar joking wif u oni]	[ok lar joking wif u oni]	[ok lar joking wif u oni]	[ok lar joking wif u oni]
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	Free entry in 2 a wkly comp to win FA Cup fina...	free entry in 2 a wkly comp to win fa cup fina...	[free entry in 2 a wkly comp to win fa cup fin...	[free entry in 2 a wkly comp to win fa cup fin...	[free entry in 2 a wkly comp to win fa cup fin...	[free entry in 2 a wkly comp to win fa cup fin...
3	ham	U dun say so early hor... U c already then say...	U dun say so early hor U c already then say	u dun say so early hor u c already then say	[u dun say so early hor u c already then say]	[u dun say so early hor u c already then say]	[u dun say so early hor u c already then say]	[u dun say so early hor u c already then say]
4	ham	Nah I dont think he goes to usf, he lives aro...	Nah I dont think he goes to usf he lives aroun...	nah i dont think he goes to usf he lives aroun...	[nah i dont think he goes to usf he lives arou...	[nah i dont think he goes to usf he lives arou...	[nah i dont think he goes to usf he lives arou...	[nah i dont think he goes to usf he lives arou...

2. Numeric Preprocessing

Numeric Preprocessing yaitu tahapan untuk menyiapkan data mentah atau biasa dikenal dengan raw data yang dikumpulkan dari berbagai sumber menjadi informasi yang lebih bersih dan bisa digunakan untuk pengolahan selanjutnya.

1) Import library

```
import pandas as pd
import scipy
import numpy as np
from sklearn.preprocessing import MinMaxScaler
```

2) Import dataset diabetes 'diabetes.csv'

```
dataframe = pd.read_csv('diabetes.csv')
array = dataframe.copy()
array.head()
```

Menampilkan 5 data pertama pada dataset

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

3) Membagi data dimana X berarti data fitur dan Y yaitu data class/label

```
X = array(['Pregnancies', 'Glucose', 'BloodPressure',
          'SkinThickness', 'Insulin', 'BMI',
          'DiabetesPedigreeFunction', 'Age'])
Y = array(['Outcome'])
```

4) Melakukan proses normalisasi yaitu dengan mengubah data X menjadi berskala rentang 0 sampai 1

```
scaler = MinMaxScaler(feature_range=(0, 1))
rescaledX = scaler.fit_transform(X)
```

5) Melakukan presisi data menjadi 3 bilangan dibelakang koma (titik)

```
np.set_printoptions(precision=3)
print(rescaledX[0:5,:])
```

```
[[0.353 0.744 0.59  0.354 0.    0.501 0.234 0.483]
 [0.059 0.427 0.541 0.293 0.    0.396 0.117 0.167]
 [0.471 0.92  0.525 0.    0.    0.347 0.254 0.183]
 [0.059 0.447 0.541 0.232 0.111 0.419 0.038 0.   ]
 [0.    0.688 0.328 0.354 0.199 0.642 0.944 0.2  ]]
```

3. Image Processing

Image Preprocessing adalah memanipulasi gambar untuk menyempurnakan atau mengekstrak informasi darinya, sebelum digunakan untuk pelatihan model.

1) Import library

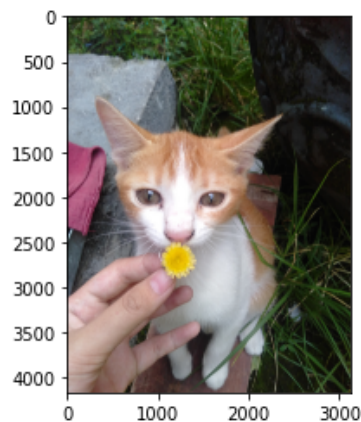
```
import os
import glob
from skimage import io
import random
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

2) Menampilkan gambar dari file yang di import

```
# mengakses file gambar dngan library skimage
image = io.imread('gambar.jpg')

# melakukan plotting / menampilkan gambar dari file yang dimpor
i, (iml) = plt.subplots(1) # membuat subplot
i.set_figwidth(15) # membuat panjang dari figure
iml.imshow(image) # tampilkan gambar
```

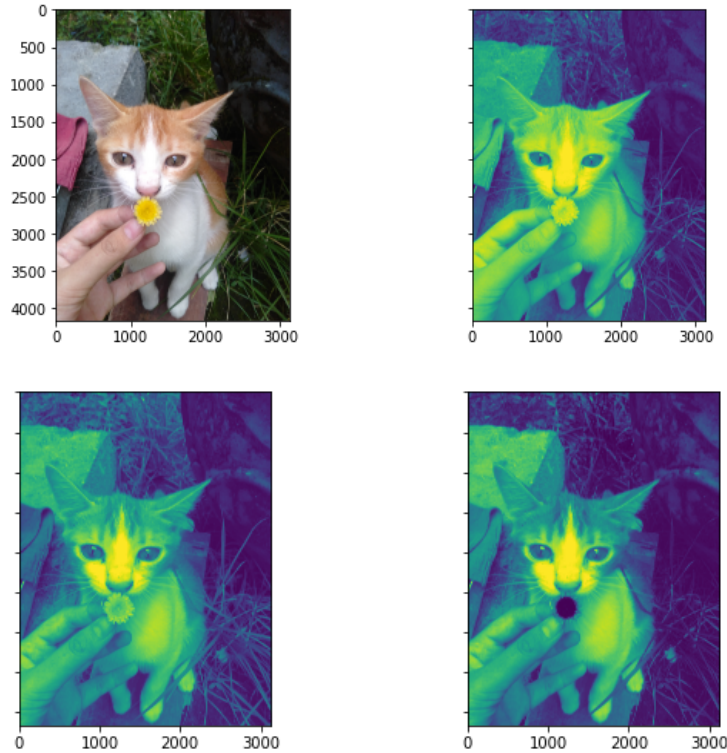
<matplotlib.image.AxesImage at 0x7f50b7f8b250>



3) Menampilkan gambar berdasarkan nilai rgb

```
i, (im1, im2, im3, im4) = plt.subplots(1, 4, sharey=True) # membuat subplot sebanyak 4 kolom pada 1 baris
i.set_figwidth(20) #mengatur figure size pada width nya

im1.imshow(image) #Original image
im2.imshow(image[:, :, 0]) #Red
im3.imshow(image[:, :, 1]) #Green
im4.imshow(image[:, :, 2]) #Blue
i.suptitle('Original & RGB image channels')
```



4) Melakukan proses gray scale

```
import skimage

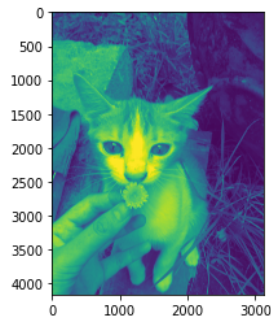
gray_image = skimage.color.rgb2gray(image) #melakukan gray scale
plt.imshow(gray_image, cmap = 'gray') # menampilkan data dengan cmap nya gray
```



5) Melakukan proses normalisasi pada gambar

```
norm_image = (gray_image - np.min(gray_image)) / (np.max(gray_image) - np.min(gray_image))
plt.imshow(norm_image)
```

<matplotlib.image.AxesImage at 0x7f50b3746d10>



4.

2.7. POST TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-03	CPMK-01	Carilah dataset berupa teks, gambar dan numerik !	40
2.	CPL-93	CPMK-01	Lakukan preprocessing pada dataset tersebut !	60

2.8. HASIL CAPAIAN PRAKTIKUM

Diisi oleh asisten setelah semua assessment dinilai.

No	Bentuk Assessment	CPL	CPMK	Bobot	Skor (0-100)	Nilai Akhir (Bobot x Skor)
1.	Pre-Test	CPL-03	CPMK-01	20%		
2.	Praktik	CPL-03	CPMK-01	30%		
3.	Post-Test	CPL-03	CPMK-01	50%		
Total Nilai						

LEMBAR JAWABAN PRE-TEST DAN POST-TEST PRAKTIKUM

Nama : NIM :	Asisten: Paraf Asisten:	Tanggal: Nilai:
-------------------------------	------------------------------------------	----------------------------------

PRAKTIKUM 3: EKSTRAKSI CIRI

Pertemuan ke : 3

Total Alokasi Waktu : 90 menit

- Materi : 15 menit
- Pre-Test : 15 menit
- Praktikum : 45 menit
- Post-Test : 15 menit

Total Bobot Penilaian : 100%

- Pre-Test : 20 %
- Praktik : 30 %
- Post-Test : 50 %

Pemenuhan CPL dan CPMK:

CPL-03	Deskripsi dari CPL tsb...
CPMK-01	Deskripsi dari CPMK tsb...

3.1. DESKRIPSI CAPAIAN PEMBELAJARAN

Setelah mengikuti praktikum ini mahasiswa diharapkan mampu:

3. Melakukan preprocessing untuk text, gambar dan numerik
4. Menentukan teknik yang tepat untuk melakukan preprocessing

3.2. INDIKATOR KETERCAPAIAN PEMBELAJARAN

Indikator ketercapaian diukur dengan:

CPL-01	CPMK-01	Kemampuan mahasiswa dalam menerapkan preprocessing pada dataset yang dimiliki
--------	---------	-------------------------------------------------------------------------------

3.3 TEORI PENDUKUNG

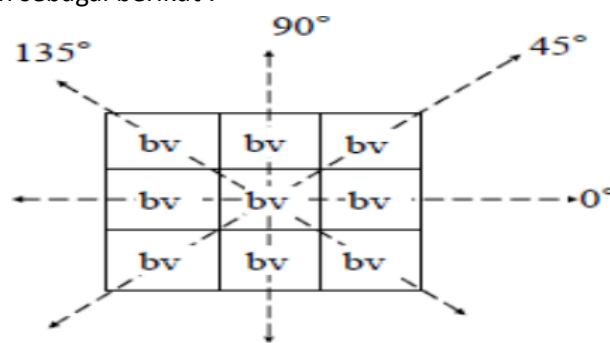
Ekstraksi Ciri adalah proses pengambilan informasi/ciri yang terkandung dalam suatu objek yang kemudian informasi/ciri tersebut yang ingin dikenali atau dibedakan dengan objek lainnya. proses ekstraksi ciri dapat dilakukan pada data image/citra, data teks, dan data numeric.

Tujuan ekstraksi ciri adalah untuk mereduksi data sebenarnya dengan melakukan pengukuran terhadap ciri tertentu yang membedakan pola masukan (input) satu dengan yang lainnya (Salambue, 2006).

Ekstraksi Ciri pada citra yaitu proses pengambilan ciri/informasi dari suatu objek di dalam citra yang ingin dikenali/dibedakan dengan ciri objek lainnya.

Tekstur merupakan keteraturan pola-pola tertentu yang terbentuk dari susunan piksel-piksel dalam citra digital. Sehingga informasi tekstur dapat digunakan untuk membedakan sifat-sifat permukaan suatu benda dalam citra yang berhubungan dengan kasar dan halus, juga sifat-sifat spesifik dari kekasaran dan kehalusan permukaan, yang sama sekali terlepas dari warna permukaan tersebut (Dinar dkk,2013). Ekstraksi ciri tekstur yang digunakan adalah Gray Level Cooccurrence Matrix (GLCM).

Gray level co-occurrence matrix (GLCM) atau matriks ko-okurensi adalah salah satu metode statistik yang dapat digunakan untuk analisis tekstur. Matriks ko-okurensi merupakan matriks yang menggambarkan hubungan ketetanggaan antar piksel dalam suatu citra dengan arah orientasi dan jarak tertentu. Untuk ukuran windows 3x3 misalnya (terdiri dari 9 piksel), fungsi sudut dan jarak tersebut dapat digunakan sebagai berikut :



Gambar 1 MatrikCo-occurrence

GLCM merupakan metode statistikal yang dibagi menjadi dua yaitu metode statistik orde pertama dan metode statistik orde kedua. Sebelum dilakukannya proses Gray level co-occurrence matrix (GLCM) dilakukan terlebih dahulu konversi citra RGB ke citra Grayscale. Citra Grayscale merupakan citra digital yang hanya memiliki satu nilai kanal pada setiap pixelnya, dengan kata lain nilai bagian RED = GREEN = BLUE. Nilai tersebut digunakan untuk 11-3 menunjukkan tingkat intensitas. Warna yang dimiliki adalah warna hitam, keabuan dan putih. Tingkatan keabuan disini merupakan warna abu dengan berbagai tingkatan dari hitam hingga mendekati putih. Citra Grayscale memiliki kedalaman warna 8 bit yakni 256 kombinasi warna keabuan (Putra, 2010).

- a. **Ekstraksi Ciri Orde Pertama** merupakan metode pengambilan ciri yang didasarkan pada karakteristik histogram citra (Yogiarto et al.2011). Pada beberapa kasus ciri orde pertama tidak bisa digunakan untuk mengenali perbedaan antar citra. Pada kasus seperti ini, kita membutuhkan pengambilan ciri orde dua (Munir, 2004).
- b. **Ekstraksi Ciri Orde Kedua** Salah satu teknik untuk memperoleh ciri statistik orde dua adalah dengan menghitung probabilitas hubungan ketetanggaan antara dua piksel pada jarak dan orientasi sudut tertentu. Pendekatan ini bekerja dengan membentuk sebuah matriks ko-ocurrence dari data citra, dilanjutkan dengan menentukan ciri sebagai fungsi dari matriks antara tersebut.

3.4. HARDWARE DAN SOFTWARE

Hardware dan software yang digunakan dalam praktikum ini yaitu:

4. Komputer.
5. IDE Python (Google collab, Jupyter Notebook)
6. Python 3.x

3.5. PRE-TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.			Apa dampak yang terjadi jika tidak menerapkan teknik ekstraksi ciri ?	25
2.	CPL-03	CPMK-01	Sebutkan teknik ekstraksi ciri pada gambar !	25
3.	Dst		Sebutkan teknik ekstraksi ciri pada teks !	25
4.			Sebutkan teknik ekstraksi ciri pada Sinyal !	25

3.6. LANGKAH PRAKTIKUM

Aturan Penilaian (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Dokumen Pendukung	Skor
1.	CPL-03	CPMK-01	Selesaikan langkah praktikum 1 – 3	Hasil praktikum langkah 1 – 3	
2.	dst				

Langkah-Langkah Praktikum:

1. Ekstraksi Ciri Citra

Ekstraksi Ciri pada citra yaitu proses pengambilan ciri/informasi dari suatu objek di dalam citra yang ingin dikenali/dibedakan dengan ciri objek lainnya.

- 1) Import library

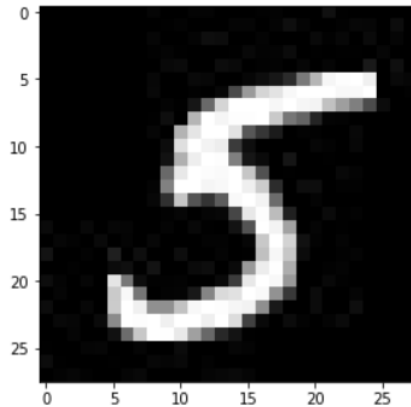
Library yang dibutuhkan untuk proses ekstraksi ciri pada citra

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
from skimage.io import imread, imshow
```

2) Menampilkan gambar dari dataset

```
image = imread('/content/img_264.jpg', as_gray=True)
imshow(image)
```

<matplotlib.image.AxesImage at 0x7f60a1167210>



3) Melihat data shape pada gambar untuk mengetahui besaran data panjang dan tinggi

```
#checking image shape
image.shape
```

(28, 28)

4) Menampilkan data pada setiap cellnya adalah representasi dari pixel 0 adalah warna hitam dan 255 adalah warna putih

```
print(image)
```

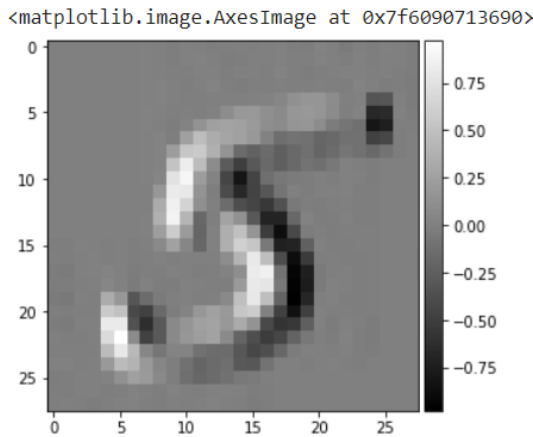
```
[[ 0  0  0  0  0  0  0  0  5  0  0  4  4  0  0  0  2  3
   0  0  0 16  1  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  1  3  0  0  0  5  5  0  4
 14  4  0  0  2  7  6  0 12  0]
 [ 0  0  0  0  0  0  0  0  6  0  0  4 14 12  0  0  4  0
  0  0  9  2  7  0 10  0  5  0]
 [ 0  0  0  0  0  0  0  0  2  4  1  0  0  0  1  8  3  4
 11  0  1  0  1  1  0  6  0  0]
 [ 0  0  0  0  0  0  0  0  0  8 10  0  0 12  9  0  2  3
  0  6  0  5  0  0 25  0 15  0]
 [ 0  0  0  0  0  0  0  0  7  0  0  9  7  0  0  6 13 23
 43 122 181 246 253 255 240  9  6]
 [ 0  0  0  0  0  0  0  0  8  0  0  2  0  7 73 154 213 227
248 255 255 255 248 249 255  0  0  2]
 [ 0  0  0  0  0  0  0  0  0  4  1 22 106 213 255 255 255 247
255 246 243 193 158 127 75 12  0  0]
 [ 0  0  0  0  0  0  0  0  1 14  0 181 255 245 253 253 255 201
```

- 5) Mencoba merubah nilai menjadi abu-abu dan menampilkan tepian objek agar lebih nampak

```
from skimage.filters import prewitt_h, prewitt_v

#calculating horizontal edges using prewitt kernel
edges_prewitt_horizontal = prewitt_h(image)
#calculating vertical edges using prewitt kernel
edges_prewitt_vertical = prewitt_v(image)

imshow(edges_prewitt_vertical, cmap='gray')
```



2. Ekstraksi Ciri Teks

Ekstraksi ciri pada teks penting pada tahapan identifikasi atau klasifikasi untuk mengubah format tekstual yang tidak terstruktur menjadi terstruktur

- 1) Import library

```
import pandas as pd
```

- 2) Load dataset vaksin astrazeneca dan menampilkan dataset dari 5 teratas

```
TWEET_DATA = pd.read_csv('vaksin_astra_df.csv')
TWEET_DATA.head()
```

	Label	Text	clean_msg	msg_lower	token	spell	no_stopwords	msg_stemmed
0	Negative	Mak ada saran gak bagi yg terpaksa covid? Sepu...	Mak ada saran gak bagi yg terpaksa covid Sepup...	mak ada saran gak bagi yg terpaksa covid sepup...	['mak', 'ada', 'saran', 'gak', 'bagi', 'yg', '...]	['ibu', 'ada', 'saran', 'tidak', 'bagi', 'yang...]	['ibu', 'saran', 'terpaksa', 'covid', 'sepupu'...]	['ibu', 'saran', 'paksa', 'covid', 'sepupu', '...]
1	Negative	Sebenarnya gw bingung korea selatan nih kasusn...	Sebenarnya gw bingung korea selatan nih kasusn...	sebenarnya gw bingung korea selatan nih kasusn...	['sebenarnya', 'gw', 'bingung', 'korea', 'sela...]	['sebenarnya', 'aku', 'bingung', 'korea', 'sel...]	['sebenarnya', 'bingung', 'korea', 'selatan', ...]	['benar', 'bingung', 'korea', 'selatan', 'kasu...]
2	Negative	Soalnya kerabat dekat ku banyak yg gak bisa da...	Soalnya kerabat dekat ku banyak yg gak bisa da...	soalnya kerabat dekat ku banyak yg gak bisa da...	['soalnya', 'kerabat', 'dekat', 'ku', 'banyak'...]	['soalnya', 'kerabat', 'dekat', 'ku', 'banyak'...]	['soalnya', 'kerabat', 'dekat', 'banyak', 'ast...]	['soal', 'kerabat', 'dekat', 'banyak', 'astraz...]
3	Negative	Hari ini aku vaksin astra dosis . Alhamdulillah...	Hari ini aku vaksin astra dosis Alhamdulillah...	hari ini aku vaksin astra dosis alhamdulillah...	['hari', 'ini', 'aku', 'vaksin', 'astra', 'dos...]	['hari', 'ini', 'aku', 'vaksin', 'astrazeneca'...]	['hari', 'vaksin', 'astrazeneca', 'dosis', 'al...]	['hari', 'vaksin', 'astrazeneca', 'dosis', 'al...]

- 3) Mendeklarasikan model TF-IDF Vectorizer untuk melakukan feature extraction / ekstraksi ciri pada teks

```
from sklearn.feature_extraction.text import TfidfVectorizer

max_features = 1000

tf_idf = TfidfVectorizer(max_features=max_features, binary=True)

tfidf_mat = tf_idf.fit_transform(TWEET_DATA["msg_string"]).toarray()
```

- 4) Menampilkan term kata dengan probabilitas katanya pada keseluruhan dokumen dan urutkan katanya berdasarkan tingkat probabilitasnya

```
terms = tf_idf.get_feature_names()

sums = tfidf_mat.sum(axis=0)

data = []
for col, term in enumerate(terms):
    data.append((term, sums[col] ))

ranking = pd.DataFrame(data, columns=['term', 'rank'])
ranking.sort_values('rank', ascending=False)
```

	term	rank
105	efek	13.991127
23	astrazeneca	13.986412
429	vaksin	13.924015
327	samping	8.993687
16	apa	7.972279
...

3. Ekstraksi Ciri Numeric

Ekstraksi ciri dilakukan untuk proses pengambilan ciri atau objek yang dikenali begitupun pada numeric, dimana pada praktikum ini metode untuk ekstraksi ciri pada numeric menggunakan ekstraksi ciri statistik ordo pertama.

1) Import library

```
import pandas as pd
import numpy as np
from scipy.stats import kurtosis, skew
```


- 2) Load dataset pada kasus eeg/gelombang otak dan menampilkan datanya dari 5 teratas

```
data = pd.read_csv('Insan.csv')
data.head()
```

	'Time'	'VidTime'	'Signal Level'	'Blink'	'Attention'	'Meditation'	'Zone'
0	2021/03/22 14:40:03.000	07:00:00.000	100	2	38	57	47
1	2021/03/22 14:40:04.000	07:00:00.000	100	2	37	64	50
2	2021/03/22 14:40:05.000	07:00:00.000	100	2	47	63	55
3	2021/03/22 14:40:06.000	07:00:00.000	87	2	47	63	55
4	2021/03/22 14:40:07.000	07:00:00.000	87	2	47	63	55

- 3) Melakukan ekstraksi dengan metode statistik ordo pertama yaitu dengan mencari nilai mean, median, standar deviasi, skewness dan kurtosis

```
mean_of_attention = np.mean(data["Attention"])
median_of_attention = np.median(data["Attention"])
std_of_attention = np.std(data["Attention"])
skew_of_attention = skew(data["Attention"])
kurtosis_of_attention = kurtosis(data["Attention"])
```

- 4) Menampilkan hasil dari metode statistik ordo pertama

```
print(mean_of_attention)
print(median_of_attention)
print(std_of_attention)
print(skew_of_attention)
print(kurtosis_of_attention)
```

```
43.44528710725894
44.0
22.30581273320855
-0.09536720083174428
-0.07376163244453338
```

3.7. POST TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-03	CPMK-01	lakukan ekstraksi ciri dari dataset yang kamu peroleh pada pertemuan sebelumnya	100

3.8 HASIL CAPAIAN PRAKTIKUM

Diisi oleh asisten setelah semua assessment dinilai.

No	Bentuk Assessment	CPL	CPMK	Bobot	Skor (0-100)	Nilai Akhir (Bobot x Skor)
1.	Pre-Test	CPL-03	CPMK-01	20%		
2.	Praktik	CPL-03	CPMK-01	30%		
3.	Post-Test	CPL-03	CPMK-01	50%		
Total Nilai						

LEMBAR JAWABAN PRE-TEST DAN POST-TEST PRAKTIKUM



Nama : NIM :	Asisten: Paraf Asisten:	Tanggal: Nilai:
-------------------------------	------------------------------------------	----------------------------------

PRAKTIKUM 4: REGRESI

Pertemuan ke : 4

Total Alokasi Waktu : 90 menit

- Materi : 15 menit
- Pre-Test : 15 menit
- Praktikum : 45 menit
- Post-Test : 15 menit

Total Bobot Penilaian : 100%

- Pre-Test : 20 %
- Praktik : 30 %
- Post-Test : 50 %

Pemenuhan CPL dan CPMK:

CPL-03	Deskripsi dari CPL tsb...
CPMK-01	Deskripsi dari CPMK tsb...

4.1. DESKRIPSI CAPAIAN PEMBELAJARAN

Setelah mengikuti praktikum ini mahasiswa diharapkan mampu:

1. Melakukan proses regresi pada dataset yang diterima

4.2. INDIKATOR KETERCAPAIAN PEMBELAJARAN

Indikator ketercapaian diukur dengan:

CPL-01	CPMK-01	Kemampuan mahasiswa dalam menerapkan regresi pada dataset yang dimiliki
--------	---------	-------------------------------------------------------------------------

4.3 TEORI PENDUKUNG

Regresi adalah suatu metode analisis yang biasa digunakan untuk melihat pengaruh antara dua atau banyak variabel. Umumnya, analisis regresi digunakan untuk melakukan prediksi atau ramalan. Sedangkan, hubungan variabel tersebut bersifat fungsional yang diwujudkan dalam suatu model matematis.

Analisis regresi termasuk beberapa variasi, yaitu linear, linear majemuk dan non linear. model regresi yang paling sering digunakan adalah linear dan linear majemuk. untuk model non linear biasa dipakai untuk kelompok data yang lebih kompleks, karena hubungan antar variabel tidak sejalan.

Rumus Regresi

berikut yaitu rumus analisis regresi untuk model linear dan linear majemuk:

1. Rumus Regresi Linear

Regresi linear merupakan model yang menunjukkan hubungan antara satu dependen variabel dengan satu independen variabel. Inilah rumus yang menyatakan model regresi linear:

$$Y = a + bX + \epsilon$$

Keterangan :

Y = Variabel dependen

X = Variabel Independen

a = Konstanta

b = Koefisien regresi

ϵ = Error atau residu

2. Rumus Regresi Linear Majemuk

Regresi linear majemuk pada dasarnya hampir sama dengan model linear sederhana, tetapi jumlah variabel independennya lebih dari satu. Berikut representasi matematis dari model linear majemuk:

$$Y = a + bX_1 + cX_2 + dX_3 + \epsilon$$

Keterangan :

Y = Variabel dependen

X₁, X₂, X₃ = Variabel independen

a = Konstanta

b, c, d = Koefisien regresi

ϵ = Error atau residu

4.4. HARDWARE DAN SOFTWARE

Hardware dan software yang digunakan dalam praktikum ini yaitu:

7. Komputer.
8. IDE Python (Google collab, Jupyter Notebook)
9. Python 3.x

4.5. PRE-TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-03	CPMK-01	Sebutkan jenis-jenis teknik regresi !	30
2.	Dst		Kapan metode regresi digunakan pada dataset kita?	70

4.6. LANGKAH PRAKTIKUM

Aturan Penilaian (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Dokumen Pendukung	Skor
1.	CPL-03	CPMK-01	Selesaikan langkah praktikum 1 – 7	Hasil praktikum langkah 1 – 7	
2.	dst				

Langkah-Langkah Praktikum:

1. Regresi

Regresi adalah suatu metode analisis yang biasa digunakan untuk melihat pengaruh antara dua atau banyak variabel. Umumnya, analisis regresi digunakan untuk melakukan prediksi atau ramalan. Sedangkan, hubungan variabel tersebut bersifat fungsional yang diwujudkan dalam suatu model matematis.

1) Import Library

```
[ ] # Impor library yang dibutuhkan
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

2) Load dataset, lalu dataset dibagi menjadi 2 bagian yaitu x dan y dimana x sebagai feature dan y sebagai classnya

```
[ ] # Impor dataset
dataset = pd.read_csv('Daftar_gaji.csv')
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, 1].values
```

3) Lakukan pembagian untuk data testing dan data training

```
[ ] # Membagi data menjadi Training Set dan Test Set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 1/3,
                                                    random_state = 0)
```

- 4) Membuat model linear regression dengan bantuan library sklearn juga kemudian lakukan fitting dari data X_{train} dan y_{train} untuk menemukan nilai slope dan intercept nya

```
# Fitting Simple Linear Regression terhadap Training set
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)
```

```
LinearRegression()
```

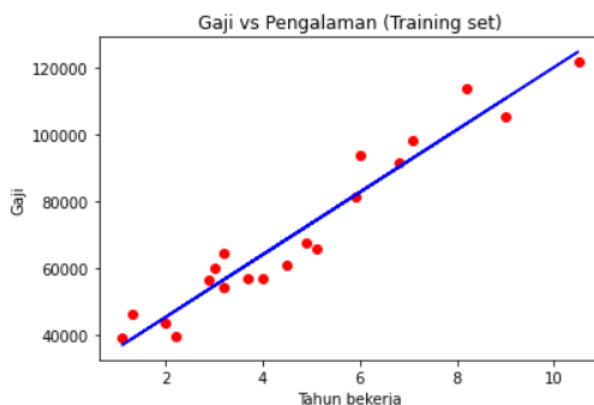
- 5) Melakukan prediksi pada data X_{test}

```
[ ] # Memprediksi hasil Test Set
y_pred = regressor.predict(X_test)
```

- 6) Melakukan plot untuk menampilkan sebaran data dari dataset di atas titik merah diperoleh dari method scatter dimana dari data X dan y. Garis linear berwarna biru diperoleh dari nilai X dan ditarik garis dari hasil prediksi linear regression

```
[ ] # Visualisasi hasil Training Set
plt.scatter(X_train, y_train, color = 'red')
plt.plot(X_train, regressor.predict(X_train), color = 'blue')
plt.title('Gaji vs Pengalaman (Training set)')
plt.xlabel('Tahun bekerja')
plt.ylabel('Gaji')
plt.show()
```

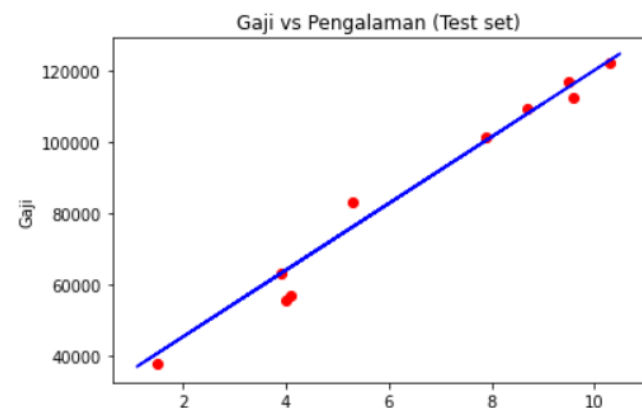
Hasil yang didapat



- 7) Lakukan hal yang serupa dengan data test nya di sini hasilnya cukup baik karena garis biru mendekati nilai X dan y nya (scatter)

```
# Visualisasi hasil Test Set
plt.scatter(X_test, y_test, color = 'red')
plt.plot(X_train, regressor.predict(X_train), color = 'blue')
plt.title('Gaji vs Pengalaman (Test set)')
plt.xlabel('Tahun bekerja')
plt.ylabel('Gaji')
plt.show()
```

Hasil yang didapat



4.7. POST TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-03	CPMK-01	Lakukan regresi dari dataset yang diberikan asisten	100

4.8. HASIL CAPAIAN PRAKTIKUM

Diisi oleh asisten setelah semua assessment dinilai.

No	Bentuk Assessment	CPL	CPMK	Bobot	Skor (0-100)	Nilai Akhir (Bobot x Skor)
1.	Pre-Test	CPL-03	CPMK-01	20%		
2.	Praktik	CPL-03	CPMK-01	30%		
3.	Post-Test	CPL-03	CPMK-01	50%		
Total Nilai						

LEMBAR JAWABAN PRE-TEST DAN POST-TEST PRAKTIKUM



Nama : NIM :	Asisten: Paraf Asisten:	Tanggal: Nilai:
-------------------------------	------------------------------------------	----------------------------------

--

- Pertemuan ke** : 5
- Total Alokasi Waktu** : 90 menit
- Materi : 15 menit
 - Pre-Test : 15 menit
 - Praktikum : 45 menit
 - Post-Test : 15 menit

- Total Bobot Penilaian** : 100%
- Pre-Test : 20 %
 - Praktik : 30 %
 - Post-Test : 50 %

Pemenuhan CPL dan CPMK:

CPL-03	Deskripsi dari CPL tsb...
CPMK-01	Deskripsi dari CPMK tsb...

5.1. DESKRIPSI CAPAIAN PEMBELAJARAN

Setelah mengikuti praktikum ini mahasiswa diharapkan mampu:

1. Mengetahui pengetahuan terhadap beberapa metode menghitung jarak
2. Menentukan teknik yang tepat untuk melakukan menghitung jarak

5.2. INDIKATOR KETERCAPAIAN PEMBELAJARAN

Indikator ketercapaian diukur dengan:

CPL-01	CPMK-01	Kemampuan mahasiswa dalam menerapkan preprocessing pada dataset yang dimiliki
--------	---------	-------------------------------------------------------------------------------

5.3 TEORI PENDUKUNG

Klasifikasi Matrik Jarak adalah metode untuk mencari kecocokan antar matrik misalnya seperti kumpulan pixel pada gambar. Fungsi jarak yaitu sebagai alat pengukuran kemiripan atau ketidakmiripan dimana perbedaan antara pola satu dengan pola-pola yang lain dapat diukur nilai kemiripannya. Salah satu ukuran kemiripan adalah dengan menentukan jarak matrik.

Metode-metode umum yang sering dipakai pada Klasifikasi Matrik Jarak adalah Manhattan, Euclidean, dll.

Manhattan Distance bisa disebut juga sebagai “city block distance” yaitu merupakan jumlah jarak dari semua attribute. Untuk dua data titik x dan y dalam d-ruang dimensi, Manhattan distance antara titik tersebut didefinisikan sebagai berikut:

$$d(x, y) = \sum_{i=1}^n |x_i - y_i|$$

Rumus ini mencari jarak hanya dengan menjumlahkan semua selisih jarak x_i dan y_i .

Mungkin idenya dari menghitung jarak dari 5 ke 2 yaitu 3 karena $|5-2|=3$. Apa bedanya dengan euclidean distance ?

Manhattan mengurangi per elemen antar 2 variabel, memutlakan lalu menjumlahkannya, sedangkan euclidean distance menghitung jarak antara 2 titik dengan konsep pythagoras, yang kalau divisualisasikan berupa garis miring pada segitiga siku-siku.

Euclidean Distance bisa disebut juga Jarak Metode perhitungan jarak terdekat adalah metode perhitungan jarak yang paling sering digunakan untuk menghitung kesamaan dua buah vektor. Rumus Euclidean Distance yaitu akar dari kuadrat perbedaan 2 vektor. Jarak Euclidean Distance adalah jarak antar titik dalam garis lurus. Metode jarak ini menggunakan Teorema Pythagoras dan merupakan perhitungan jarak yang paling sering digunakan dalam proses machine learning. Rumus Euclidean Distance adalah hasil dari akar kuadrat perbedaan dua buah vektor.

$$d(x, y) = \sqrt{\sum_{i=1}^n (y_i - x_i)^2}$$

Keterangan :

x, y = dua titik dalam ruang euclidean

x_i, y_i = vektor euclidean, dimulai dari asal ruang

Ide rumus euclidean adalah dari rumus pythagoras yaitu

$$c = \sqrt{a^2 + b^2}$$

Dimana $d(x, y)$ dibaca distance antara x dan y.

5.4. HARDWARE DAN SOFTWARE

Hardware dan software yang digunakan dalam praktikum ini yaitu:

10. Komputer.
11. IDE Python (Google collab, Jupyter Notebook)
12. Python 3.x

5.5. PRE-TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-03	CPMK-01	Sebutkan metode apa saja yang ada pada matrik jarak !	40
2.	Dst		Apa yang dihasilkan dari perhitungan matrik jarak ?	60

5.6. LANGKAH PRAKTIKUM

Aturan Penilaian (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Dokumen Pendukung	Skor
1.	CPL-03	CPMK-01	Selesaikan langkah praktikum 1 – 9	Hasil praktikum langkah 1 – 9	
2.	dst				

Langkah-Langkah Praktikum:

1. Membuat dua variabel yang berisikan data array

```
point1 = [2, 4, 4, 6]
point2 = [5, 5, 7, 8]
```

2. Membuat beberapa fungsi untuk menghitung beberapa metode menghitung jarak

```
# Calculating Manhattan Distance from Scratch
def manhattan_distance(point1, point2):
    return sum(abs(value1 - value2) for value1, value2 in zip(point1, point2))
```

```
# Calculating Euclidean Distance from Stratch
def euclidean_distance(point1, point2):
    return sum((value1 - value2) ** 2 for value1, value2 in zip(point1, point2))
```

```
# Calculating Canberra Distance from Scratch
def canberra_distance(point1, point2):
    return sum((abs(value1 - value2) / (value1 + value2)) for value1, value2 in zip(point1, point2))
```

- Menampilkan hasil dari perhitungan

```

result_manhattan_distance = manhattan_distance(point1, point2)
print(result_manhattan_distance)

result_euclidean_distance = euclidean_distance(point1, point2)
print(result_euclidean_distance)

result_canberra_distance = canberra_distance(point1, point2)
print(result_canberra_distance)

```

5.7. POST TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-03	CPMK-01		
2.	dst			

5.8. HASIL CAPAIAN PRAKTIKUM

Diisi oleh asisten setelah semua assessment dinilai.

No	Bentuk Assessment	CPL	CPMK	Bobot	Skor (0-100)	Nilai Akhir (Bobot x Skor)
1.	Pre-Test	CPL-03	CPMK-01	20%		
2.	Praktik	CPL-03	CPMK-01	30%		
3.	Post-Test	CPL-03	CPMK-01	50%		
Total Nilai						

LEMBAR JAWABAN PRE-TEST DAN POST-TEST PRAKTIKUM



Nama : NIM :	Asisten: Paraf Asisten:	Tanggal: Nilai:
-------------------------------	------------------------------------------	----------------------------------

--

PRAKTIKUM 6: KLASIFIKASI K-NEAREST NEIGHBOR (KNN)

Pertemuan ke : 6

Total Alokasi Waktu : 90 menit

- Materi : 15 menit
- Pre-Test : 15 menit
- Praktikum : 45 menit
- Post-Test : 15 menit

Total Bobot Penilaian : 100%

- Pre-Test : 20 %
- Praktik : 30 %
- Post-Test : 50 %

Pemenuhan CPL dan CPMK:

CPL-03	Deskripsi dari CPL tsb...
CPMK-01	Deskripsi dari CPMK tsb...

6.1. DESKRIPSI CAPAIAN PEMBELAJARAN

Setelah mengikuti praktikum ini mahasiswa diharapkan mampu:

1. Memahami teori KNN
2. Menentukan teknik yang tepat untuk melakukan klasifikasi

6.2. INDIKATOR KETERCAPAIAN PEMBELAJARAN

Indikator ketercapaian diukur dengan:

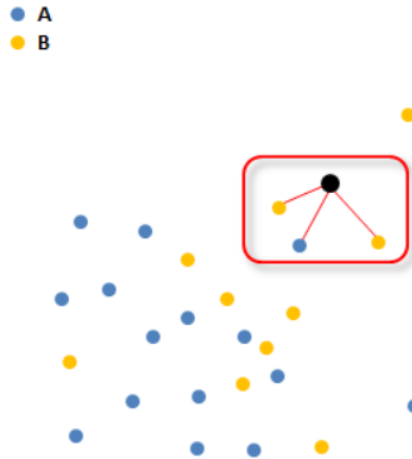
CPL-01	CPMK-01	Kemampuan mahasiswa dalam menerapkan preprocessing pada dataset yang dimiliki
--------	---------	-------------------------------------------------------------------------------

6.3 TEORI PENDUKUNG

Algoritma KNN (K-Nearest Neighbor) yaitu merupakan algoritma klasifikasi yang bekerja dengan mengambil sejumlah K data terdekat sebagai acuan untuk menentukan kelas dari data baru. Algoritma ini mengklasifikasi data berdasarkan kemiripan atau kedekatannya terhadap data lainnya.

Secara umum, **cara kerja algoritma KNN** adalah sebagai berikut :

1. Tentukan jumlah tetangga (K) yang akan digunakan untuk pertimbangan penentuan kelas.
2. Hitung jarak dari data baru ke masing-masing data point di dataset.
3. Ambil sejumlah K data dengan jarak terdekat, kemudian tentukan kelas dari data baru tersebut.



Pada gambar di atas yaitu terdapat sebuah data point yang dibagi dua kelas dimana point yang berwarna biru adalah kelas A dan point yang berwarna kuning adalah kelas B. Misal ada sebuah data baru yang masuk yaitu point hitam yang akan kita prediksi kelasnya menggunakan algoritma KNN. Menurut dari gambar di atas yaitu nilai K adalah 3 dimana dapat dihitung dari jarak terdekat point hitam ke masing-masing poin lainnya didapatkan 3 titik yang terdekat, dimana 2 titik kuning dan 1 titik biru, maka kelas untuk data baru yaitu point hitam adalah kelas B atau point kuning.

Untuk menghitung jarak antara dua titik pada algoritma KNN digunakan yaitu metode euclidean distance yang dapat digunakan pada 1 dimensional space, 2 dimensional space atau multi dimensional space. Dimana 1 dimensional space berarti perhitungan jarak hanya menggunakan satu variabel bebas, 2 dimensional berarti dua variabel bebas, dan multi-dimensional berarti ada lebih dari dua variabel.

Contoh cara kerja algoritma KNN

Terdapat sebuah data dengan 3 kolom yaitu berat badan, tinggi badan dan kelas. Dimana dapat dilihat pada tabel seperti di bawah ini.

Berat Badan	Tinggi Badan	Kelas
52	157	Kecil
60	162	Sedang
55	165	kecil
65	170	Sedang
60	168	Sedang
62	173	Kecil
57	160	?

Dari tabel di atas ada 6 data yang sudah ada kelasnya dan ada 1 data baru yang harus diprediksi bahwa data itu termasuk kelas apa ?

Menentukan nilai K

Pertama kita harus menentukan nilai K terlebih dahulu. Penentuan nilai K tidak ada rumus pasti, tetapi ada satu tips untuk menentukan pertimbangan nilai K yaitu dengan cara jika kelas berjumlah genap maka nilai K sebaiknya ganjil dan sebaliknya jika kelas berjumlah ganjil maka sebaiknya nilai K genap. Misal pada contoh di atas yang memiliki dua kelas jika kita menggunakan nilai K genap maka kemungkinan hasil prediksi jarak terdekat bisa seri begitupun sebaliknya. Pada kasus di atas nilai K yang digunakan adalah 3.

Hitung jarak antara data baru dengan data yang lainnya

Dilakukan proses menghitung jarak menggunakan metode euclidean distance sebagai berikut:

$$\text{Data 1} = \sqrt{(57 - 52)^2 + (160 - 157)^2} = 5,83$$

$$\text{Data 2} = \sqrt{(57 - 60)^2 + (160 - 162)^2} = 3,60$$

$$\text{Data 3} = \sqrt{(57 - 55)^2 + (160 - 165)^2} = 5,38$$

$$\text{Data 4} = \sqrt{(57 - 65)^2 + (160 - 170)^2} = 12,81$$

$$\text{Data 5} = \sqrt{(57 - 60)^2 + (160 - 168)^2} = 10,63$$

$$\text{Data 6} = \sqrt{(57 - 62)^2 + (160 - 173)^2} = 5,83$$

Urutkan data dari yang terdekat

Dari perhitungan euclidean distance di atas dapat dirangkum dari jarak terdekat hasilnya seperti di bawah ini.

Berat Badan	Tinggi Badan	Kelas	jarak data terbaru
60	162	Sedang	3,60
55	165	kecil	5,38
52	157	Kecil	5,83
62	173	Kecil	5,83
60	168	Sedang	10,63
65	170	Sedang	12,81

Dari hasil di atas dengan nilai $K = 3$, dapat disimpulkan bahwa data baru tersebut masuk ke dalam kelas kecil karena dari 3 kelas terdekat ada dua yang masuk ke dalam kelas kecil dan ada satu yang masuk ke kelas sedang.

6.4. HARDWARE DAN SOFTWARE

Hardware dan software yang digunakan dalam praktikum ini yaitu:

13. Komputer.
14. IDE Python (Google collab, Jupyter Notebook)
15. Python 3.x

6.5. PRE-TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-03	CPMK-01		
2.	Dst			

6.6. LANGKAH PRAKTIKUM

Aturan Penilaian (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Dokumen Pendukung	Skor
1.	CPL-03	CPMK-01	Selesaikan langkah praktikum 1 – 9	Hasil praktikum langkah 1 – 9	
2.	dst				

Langkah-Langkah Praktikum:

1. Import library

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

2. Mengimport dataset

```
dataset = pd.read_csv('Iklan_sosmed.csv')
X = dataset.iloc[:, [2, 3]].values
y = dataset.iloc[:, 4].values
```

3. Membagi dataset menjadi dua yaitu training set dan testing set

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 0)
```

4. Feature Scaling

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

5. Membuat model KNN

```
from sklearn.neighbors import KNeighborsClassifier
classifier = KNeighborsClassifier(n_neighbors = 5, metric = 'minkowski', p = 2)
classifier.fit(X_train, y_train)
```

6. Memprediksi data testing set

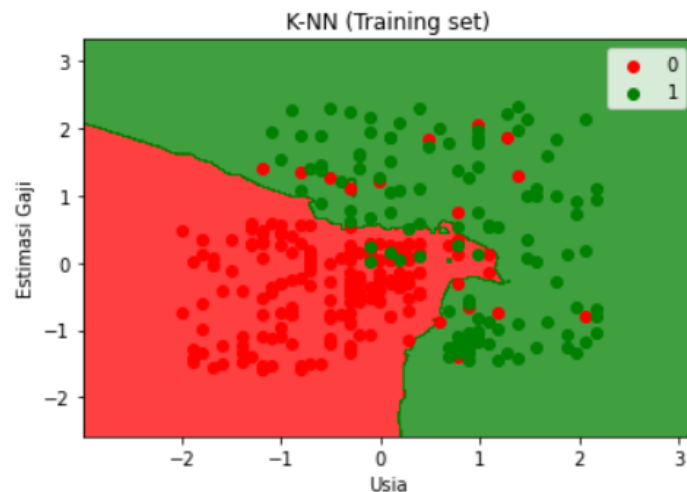
```
y_pred = classifier.predict(X_test)
```

7. Membuat Confusion Matrix

```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
```

8. Visualisasi Training set

```
from matplotlib.colors import ListedColormap
X_set, y_set = X_train, y_train
X1, X2 = np.meshgrid(np.arange(start = X_set[:, 0].min() - 1, stop = X_set[:, 0].max() + 1, step = 0.01),
                    np.arange(start = X_set[:, 1].min() - 1, stop = X_set[:, 1].max() + 1, step = 0.01))
plt.contourf(X1, X2, classifier.predict(np.array([X1.ravel(), X2.ravel()]).T).reshape(X1.shape),
            alpha = 0.75, cmap = ListedColormap(('red', 'green')))
plt.xlim(X1.min(), X1.max())
plt.ylim(X2.min(), X2.max())
for i, j in enumerate(np.unique(y_set)):
    plt.scatter(X_set[y_set == j, 0], X_set[y_set == j, 1],
                c = ListedColormap(('red', 'green'))(i), label = j)
plt.title('K-NN (Training set)')
plt.xlabel('Usia')
plt.ylabel('Estimasi Gaji')
plt.legend()
plt.show()
```

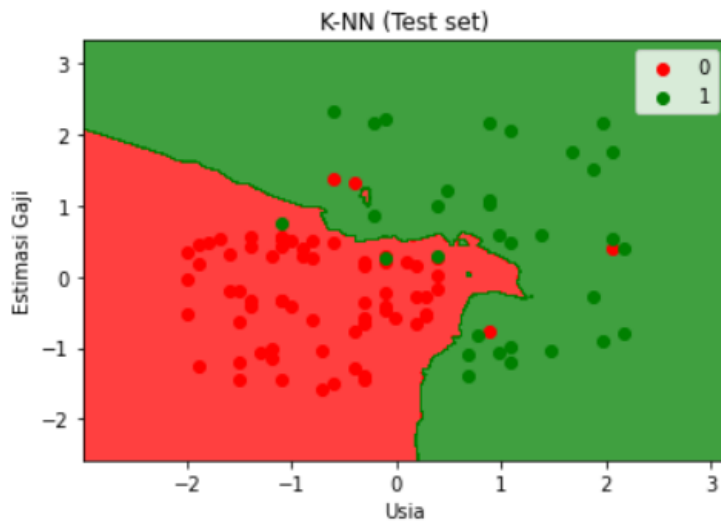


9. Visualisasi Testing set

```

from matplotlib.colors import ListedColormap
X_set, y_set = X_test, y_test
X1, X2 = np.meshgrid(np.arange(start = X_set[:, 0].min() - 1, stop = X_set[:, 0].max() + 1, step = 0.01),
                    np.arange(start = X_set[:, 1].min() - 1, stop = X_set[:, 1].max() + 1, step = 0.01))
plt.contourf(X1, X2, classifier.predict(np.array([X1.ravel(), X2.ravel()]).T).reshape(X1.shape),
            alpha = 0.75, cmap = ListedColormap(('red', 'green')))
plt.xlim(X1.min(), X1.max())
plt.ylim(X2.min(), X2.max())
for i, j in enumerate(np.unique(y_set)):
    plt.scatter(X_set[y_set == j, 0], X_set[y_set == j, 1],
                c = ListedColormap(('red', 'green'))(i), label = j)
plt.title('K-NN (Test set)')
plt.xlabel('Usia')
plt.ylabel('Estimasi Gaji')
plt.legend()
plt.show()

```



6.7. POST TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-03	CPMK-01		
2.	dst			

6.8. HASIL CAPAIAN PRAKTIKUM

Diisi oleh asisten setelah semua assessment dinilai.

No	Bentuk Assessment	CPL	CPMK	Bobot	Skor (0-100)	Nilai Akhir (Bobot x Skor)
1.	Pre-Test	CPL-03	CPMK-01	20%		
2.	Praktik	CPL-03	CPMK-01	30%		
3.	Post-Test	CPL-03	CPMK-01	50%		
Total Nilai						

LEMBAR JAWABAN PRE-TEST DAN POST-TEST PRAKTIKUM



Nama : NIM :	Asisten: Paraf Asisten:	Tanggal: Nilai:
-----------------	----------------------------	--------------------

--

PRAKTIKUM 7: KLASIFIKASI JARINGAN SYARAF TIRUAN (JST)

Pertemuan ke : 7

Total Alokasi Waktu : 90 menit

- Materi : 15 menit
- Pre-Test : 15 menit
- Praktikum : 45 menit
- Post-Test : 15 menit

Total Bobot Penilaian : 100%

- Pre-Test : 20 %
- Praktik : 30 %
- Post-Test : 50 %

Pemenuhan CPL dan CPMK:

CPL-03	Deskripsi dari CPL tsb...
CPMK-01	Deskripsi dari CPMK tsb...

7.1. DESKRIPSI CAPAIAN PEMBELAJARAN

Setelah mengikuti praktikum ini mahasiswa diharapkan mampu:

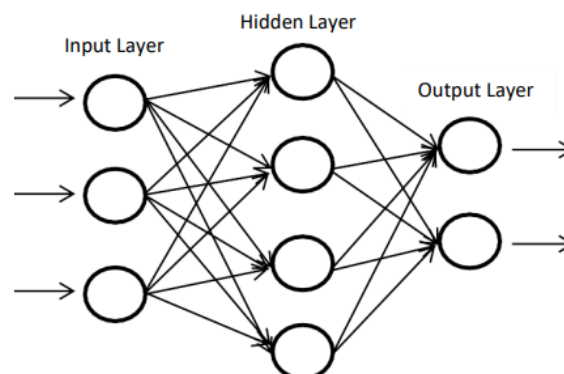
3. Melakukan preprocessing untuk text, gambar dan numerik
4. Menentukan teknik yang tepat untuk melakukan preprocessing

7.2. INDIKATOR KETERCAPAIAN PEMBELAJARAN

Indikator ketercapaian diukur dengan:

CPL-01	CPMK-01	Kemampuan mahasiswa dalam menerapkan preprocessing pada dataset yang dimiliki
--------	---------	-------------------------------------------------------------------------------

7.3 TEORI PENDUKUNG



Pada gambar di atas yaitu merupakan Arsitektur Jaringan Syaraf Tiruan (JST) dimana Jaringan Syaraf Tiruan (JST) yaitu sebuah representasi buatan yang menirukan otak manusia untuk mencoba mensimulasikan proses pembelajaran pada otak manusia tersebut. Maksud dari representasi buatan yaitu implementasi jaringan syaraf ini menggunakan program komputer yang mampu melakukan perhitungan sehingga mirip dengan cara kerja otak manusia (Kusumaningtyas & Asmara, 2016).

Arsitektur JST pada gambar di atas memiliki beberapa lapisan yaitu lapisan input (input layer), lapisan tersembunyi (hidden layer), lapisan output (output layer). Lapisan input layer yaitu lapisan untuk menginput data, lapisan tersembunyi yaitu lapisan yang digunakan untuk proses komputasi, terdiri dari beberapa lapisan atau satu lapisan dan juga bisa tidak memiliki lapisan tersembunyi dan pada lapisan output Hidden Layer Input Layer Output Layer 18 layer yaitu lapisan yang menghasilkan output dari proses input layer dan hidden layer. ANN memiliki beberapa jenis arsitektur diantaranya single layer (lapisan tunggal), multilayer (memiliki banyak lapisan) dan competitive layer (lapisan kompetitif).

Multi-layer Perceptron (MLP) yaitu jaringan saraf tiruan (JST) feed-forward yang terdiri dari satu atau lebih hidden layer. Dalam MLP terdapat fungsi pada setiap layernya yaitu input layer berfungsi menerima masukan vektor/sinyal dari luar mendistribusikan ke semua neuron pada hidden layer . output layer menerima sinyal keluaran dari hidden layer dan memunculkannya kelas/sinyal/nilai keluaran dari keseluruhan jaringan.

Pada JST proses pembelajaran atau pelatihan untuk menentukan bobot optimal untuk digunakan dalam proses pengujian data (Mentari, 2014). Algoritma pelatihan yang paling populer yaitu algoritma Backpropagation, yang mana cara pelatihannya sama dengan perception, sejumlah data latih sebagai data masukan diberikan pada jaringan. Kemudian jaringan dihitung sebagai keluaran, jika hasilnya error atau ada perbedaan antara target keluaran yang diinginkan dengan nilai keluaran yang didapat maka bobot pada jaringan akan diperbaharui untuk mengurangi error tersebut.

Keuntungan lainnya dari penggunaan Neural Network termasuk:

- Pembelajaran adaptif: Kemampuan untuk belajar dalam melakukan tugas-tugas berdasarkan data yang diberikan
- Self-Organization: Sebuah Neural Network dapat membangun representasi dari informasi yang diterimanya selama proses pembelajaran secara mandiri
- Operasi Real-Time: Penghitungan Neural Network dapat dilakukan secara paralel, sehingga proses komputasi menjadi lebih cepat.

7.4. HARDWARE DAN SOFTWARE

Hardware dan software yang digunakan dalam praktikum ini yaitu:

16. Komputer.
17. IDE Python (Google collab, Jupyter Notebook)
18. Python 3.x

7.5. PRE-TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-03	CPMK-01	Apa kekurangan dan kelebihan menggunakan metode neural network ?	50
2.	Dst		Sebutkan dan jelaskan aktivasi yang ada pada Jaringan Syaraf Tiruan!	50

7.6. LANGKAH PRAKTIKUM

Aturan Penilaian (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Dokumen Pendukung	Skor
1.	CPL-03	CPMK-01	Selesaikan langkah praktikum 1 – 13	Hasil praktikum langkah 1 – 13	
2.	dst				

Langkah-Langkah Praktikum:

1. Import Library

```
[3] # Mengimpor library yang diperlukan
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

2. Mengimport Dataset

```
dataset = pd.read_csv('Data_bank_churn.csv')
X = dataset.iloc[:, 3:13].values
y = dataset.iloc[:, 13].values
```

3. Mengkonversi data kategori jenis kelamin menjadi numerik

```
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
from sklearn.compose import ColumnTransformer
labelencoder = LabelEncoder()
X[:, 2] = labelencoder.fit_transform(X[:, 2])
X[:, 2]
array([0, 0, 0, ..., 0, 1, 0], dtype=object)
```

4. Membuat data dummy variabel untuk kolom negara

Data dummy yaitu data palsu yang dibuat dengan tujuan sebagai data contoh atau data sampel. Data dummy biasanya digunakan untuk data percobaan pada sistem

```
columnTransformer = ColumnTransformer([('encoder', OneHotEncoder(),
                                       [1])], remainder='passthrough')
X = np.array(columnTransformer.fit_transform(X), dtype = np.float)
X
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:5: Deprecation
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/
"""
```

```
array([[1.0000000e+00, 0.0000000e+00, 1.0000000e+00, ..., 1.0000000e+00,
        1.0000000e+00, 1.0134888e+05],
       [1.0000000e+00, 0.0000000e+00, 1.0000000e+00, ..., 0.0000000e+00,
        1.0000000e+00, 1.1254258e+05],
       [1.0000000e+00, 0.0000000e+00, 1.0000000e+00, ..., 1.0000000e+00,
        0.0000000e+00, 1.1393157e+05],
       ...,
       [1.0000000e+00, 0.0000000e+00, 1.0000000e+00, ..., 0.0000000e+00,
        1.0000000e+00, 4.2085580e+04],
       [0.0000000e+00, 1.0000000e+00, 0.0000000e+00, ..., 1.0000000e+00,
        0.0000000e+00, 9.2888520e+04],
       [1.0000000e+00, 0.0000000e+00, 1.0000000e+00, ..., 1.0000000e+00,
        0.0000000e+00, 3.8190780e+04]])
```

5. Menghilangkan 1 dummy variabel di kolom pertama

```
X = X[:, 1:]
X
```

```
array([[0.0000000e+00, 1.0000000e+00, 1.0000000e+00, ..., 1.0000000e+00,
        1.0000000e+00, 1.0134888e+05],
       [0.0000000e+00, 1.0000000e+00, 0.0000000e+00, ..., 0.0000000e+00,
        1.0000000e+00, 1.1254258e+05],
       [0.0000000e+00, 1.0000000e+00, 1.0000000e+00, ..., 1.0000000e+00,
        0.0000000e+00, 1.1393157e+05],
       ...,
       [0.0000000e+00, 1.0000000e+00, 1.0000000e+00, ..., 0.0000000e+00,
        1.0000000e+00, 4.2085580e+04],
       [1.0000000e+00, 0.0000000e+00, 0.0000000e+00, ..., 1.0000000e+00,
        0.0000000e+00, 9.2888520e+04],
       [0.0000000e+00, 1.0000000e+00, 1.0000000e+00, ..., 1.0000000e+00,
        0.0000000e+00, 3.8190780e+04]])
```

6. Membagi menjadi dua yaitu data training dan data testing

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2,
                                                    random_state = 0)
```

7. Feature Scaling

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

8. Mengimport library keras dan turunannya

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
```

9. Menginisialisasi Model JST

Model Jaringan Syaraf Tiruan (JST) memiliki 2 hidden layer yang masing-masingnya -

```
MesinKlasifikasi = Sequential()
# Menambah input layer dan hidden layer pertama
MesinKlasifikasi.add(Dense(units = 6, kernel_initializer = 'uniform', activation = 'relu', input_dim = 11))
# Menambah hidden layer kedua
MesinKlasifikasi.add(Dense(units = 6, kernel_initializer = 'uniform', activation = 'relu'))
# Menambah output layer
MesinKlasifikasi.add(Dense(units = 1, kernel_initializer = 'uniform', activation = 'sigmoid'))
```

10. Detail Model JST

```
MesinKlasifikasi.summary()
```

Model: "sequential_2"

Layer (type)	Output Shape	Param #
dense_6 (Dense)	(None, 6)	72
dense_7 (Dense)	(None, 6)	42
dense_8 (Dense)	(None, 1)	7

Total params: 121

Trainable params: 121

Non-trainable params: 0

11. proses training yang dilakukan sampai 100 kali

```
# Meenjalankan ANN
MesinKlasifikasi.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics = ['accuracy'])
# Fitting ANN ke training set
MesinKlasifikasi.fit(X_train, y_train, batch_size = 10, epochs = 100)
800/800 [=====] - 2s 3ms/step - loss: 0.4141 - accuracy: 0.8355
Epoch 8/100
800/800 [=====] - 3s 4ms/step - loss: 0.4127 - accuracy: 0.8335
Epoch 9/100
800/800 [=====] - 2s 3ms/step - loss: 0.4115 - accuracy: 0.8338
Epoch 10/100
800/800 [=====] - 2s 3ms/step - loss: 0.4106 - accuracy: 0.8353
Epoch 11/100
800/800 [=====] - 3s 4ms/step - loss: 0.4098 - accuracy: 0.8354
Epoch 12/100
800/800 [=====] - 4s 5ms/step - loss: 0.4089 - accuracy: 0.8349
Epoch 13/100
800/800 [=====] - 2s 3ms/step - loss: 0.4081 - accuracy: 0.8359
Epoch 14/100
800/800 [=====] - 2s 3ms/step - loss: 0.4077 - accuracy: 0.8353
Epoch 15/100
800/800 [=====] - 2s 3ms/step - loss: 0.4070 - accuracy: 0.8353
Epoch 16/100
800/800 [=====] - 2s 3ms/step - loss: 0.4065 - accuracy: 0.8360
Epoch 17/100
800/800 [=====] - 2s 2ms/step - loss: 0.4063 - accuracy: 0.8354
```

12. Melakukan prediksi dengan data testing

```
y_pred = MesinKlasifikasi.predict(X_test)
y_pred = (y_pred > 0.5)
```

```
y_pred

array([[False],
       [False],
       [False],
       ...,
       [False],
       [False],
       [False]])
```

1. Hasil Akurasi mengatakan bahwa hasil akurasi adalah 84%

```
predictions = MesinKlasifikasi.predict(X_test)
predictions
test_loss, test_acc = MesinKlasifikasi.evaluate(X_test, y_test)
print('test_acc:', test_acc)
```

```
63/63 [=====] - 0s 2ms/step - loss: 0.394
test_acc: 0.8429999947547913
```

13. menghitung confusion matrix

```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
```

```
cm

array([[1544,  51],
       [ 262, 143]])
```

7.7. POST TEST

Jawablah pertanyaan berikut (Total Skor: 100):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-03	CPMK-01	lakukan proses klasifikasi Jaringan Syaraf Tiruan dari dataset apa saja yang kalian dapatkan!	60
2.			buatlah sebuah tabel tuning yaitu tabel proses penentuan beberapa hidden layer yang dipakai dengan hasil akurasi!	40

7.8. HASIL CAPAIAN PRAKTIKUM

Diisi oleh asisten setelah semua assessment dinilai.

No	Bentuk Assessment	CPL	CPMK	Bobot	Skor (0-100)	Nilai Akhir (Bobot x Skor)
1.	Pre-Test	CPL-03	CPMK-01	20%		
2.	Praktik	CPL-03	CPMK-01	30%		

3.	Post-Test	CPL-03	CPMK-01	50%	
					Total Nilai



LEMBAR JAWABAN PRE-TEST DAN POST-TEST PRAKTIKUM

<p>Nama : NIM :</p>	<p>Asisten: Paraf Asisten:</p>	<p>Tanggal: Nilai:</p>
---------------------------------------	--------------------------------------------------	------------------------------------------

PRAKTIKUM 8: KLASIFIKASI CONVOLUTIONAL NEURAL NETWORK (CNN)

Pertemuan ke : 8

Total Alokasi Waktu : 90 menit

- Materi : 15 menit
- Pre-Test : 15 menit
- Praktikum : 45 menit
- Post-Test : 15 menit

Total Bobot Penilaian : 100%

- Pre-Test : 20 %
- Praktik : 30 %
- Post-Test : 50 %

Pemenuhan CPL dan CPMK:

CPL-03	Deskripsi dari CPL tsb...
CPMK-01	Deskripsi dari CPMK tsb...

8.1. DESKRIPSI CAPAIAN PEMBELAJARAN

Setelah mengikuti praktikum ini mahasiswa diharapkan mampu:

5. Melakukan preprocessing untuk text, gambar dan numerik
6. Menentukan teknik yang tepat untuk melakukan preprocessing

8.2. INDIKATOR KETERCAPAIAN PEMBELAJARAN

Indikator ketercapaian diukur dengan:

CPL-01	CPMK-01	Kemampuan mahasiswa dalam menerapkan preprocessing pada dataset yang dimiliki
--------	---------	-------------------------------------------------------------------------------

8.3 TEORI PENDUKUNG

Convolutional Neural Network (CNN) atau dapat disebut juga dengan *ConvNet* merupakan salah satu model dari *deep learning* yang digunakan sebagai metode analisis dan klasifikasi objek. Metode CNN menggunakan pendekatan arsitektur dari Multi-Layer Perceptron (MLP) (Pai and Giridharan 2019). CNN menggunakan perhitungan matriks piksel dari citra digital sebagai operasi utama pada data input (Anthimopoulos et al. 2016). Data latih yang digunakan pada CNN merupakan *Supervised Learning* yaitu data yang sudah diberi label sebelumnya. Metode CNN memiliki beberapa lapisan seperti lapisan masukan (*input layer*), lapisan ekstraksi fitur (*feature extraction layer*) yang terdiri dari lapisan konvolusi dan lapisan subsampling atau lapisan pooling (*pooling layer*) (Dewa, Fadhilah, and

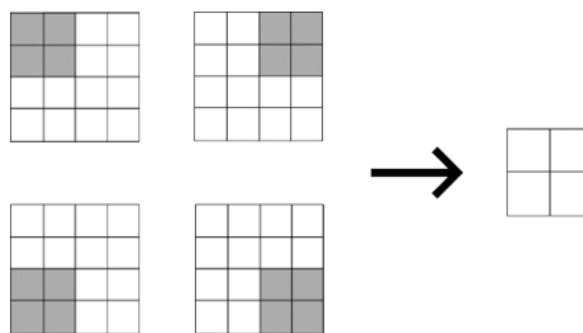
Afiahayati 2018). Lapisan yang terakhir adalah *fully-connected layer* yaitu matriks da terakhir akan diubah menjadi vektor(Gu et al. 2018).

a. Lapisan Konvolusi (*Convolutional Layer*)

Lapisan konvolusi merupakan lapisan pertama yang berfungsi sebagai tempat melakukan ekstraksi fitur (*feature extraction*) dari lapisan masukan (*input layer*) sebelumnya yang berupa piksel. Proses di dalam lapisan konvolusi ini merupakan lapisan atau blok rancangan utama pada CNN, di mana sebagian besar teknik komputasi dikerjakan pada lapisan ini. Dasar dari proses konvolusi ini sebenarnya berupa *dot product* antara filter dengan bidang reseptif kecil yang terdapat pada citra *input* yang memiliki ukuran sama dengan ukuran pada filter (Suyanto, Ramadhani, and Mandala 2019).

b. *Stride*

Stride merupakan sebuah parameter yang digunakan untuk menghitung jumlah pergeseran pada filter. Apabila *stride* bernilai 1 maka filter konvolusi akan bergerak atau bergeser sebanyak 1 piksel secara mendatar atau horizontal kemudian bergerak vertikal. Fungsi dari *stride* adalah mengatur berapa matrix yang akan dilompati. Gambar di bawah mengilustrasikan jumlah *stride* adalah 2



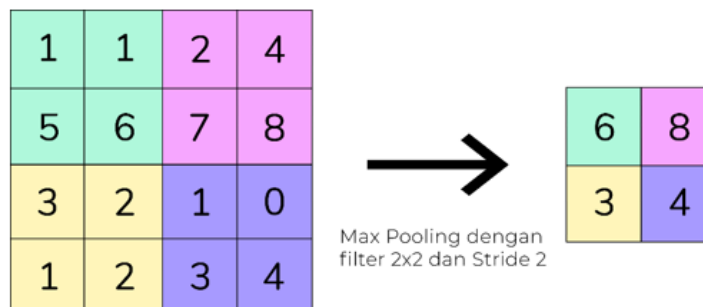
c. *Padding*

Padding atau juga disebut dengan *zero padding* merupakan teknik menambah parameter piksel dengan nilai 0 pada tiap sisi. Salah satu tujuan menambahkan *padding* ini adalah melakukan pengaturan terhadap dimensi keluaran (*output*) dari lapisan konvolusi supaya dimensi dari masukan (*input*) dapat terjaga ukurannya.

0	0	0	0	0	0	0	0
0							0
0							0
0							0
0							0
0							0
0							0
0	0	0	0	0	0	0	0

d. Pooling Layers

Pada arsitektur CNN lapisan *pooling* ini biasanya dilakukan setelah melewati lapisan konvolusi (*convolutional layer*) sebelumnya. Lapisan ini memiliki pekerjaan untuk menjaga ukuran dari data saat proses konvolusi dilakukan yaitu dengan cara mereduksi (*downsampling*). Selain itu, dengan lapisan *pooling* ini juga dapat memudahkan pengaturan *overfitting* dan melakukan pengelolaan data menjadi lebih kecil sehingga mudah untuk diatur. Secara umum proses ini dapat dilakukan menggunakan *average pooling* atau *max-pooling* dengan cara menentukan nilai rata-rata atau nilai maksimum dalam suatu wilayah tertentu (Suyanto dkk., 2019).



e. Fully-Connected Layers

Lapisan yang terhubung secara keseluruhan (*fully-connected layers*) ini merupakan bentuk dari jaringan syaraf tiruan yaitu *Multi-Layer Perceptrons* (MLP). Pada lapisan ini setiap *neurons* akan terhubung penuh ke semua aktivasi pada lapisan sebelumnya (Suyanto 2018). Lapisan konvolusi memiliki perbedaan dengan lapisan ini, karena pada lapisan konvolusi *neuron* hanya terhubung pada jaringan tertentu pada *input*, sedangkan pada lapisan *fully-connected layers* ini semua *neurons* akan terhubung satu dengan yang lainnya. Agar *feature map* dapat dijadikan sebagai masukan (*input*) oleh lapisan *fully-connected layers*, maka harus melakukan proses *reshape* dari *feature map* agar menjadi vektor karena *feature map* yang dihasilkan oleh ekstraksi fitur (*feature extraction*) pada lapisan sebelumnya masih dalam bentuk array *multidimensional*.

f. Fungsi Aktivasi Softmax

Fungsi aktivasi secara sederhana bertujuan untuk menentukan suatu *neuron* apakah aktif atau tidak aktif berdasarkan jumlah bobot (*weighted sum*) dari masukan sebelumnya. Fungsi aktivasi *softmax* merupakan salah satu fungsi aktivasi yang dapat melakukan perhitungan secara baik dalam menangani permasalahan klasifikasi dengan keluaran (*output*) yang memiliki dua atau lebih kelas (*multi-class*). Fungsi aktivasi ini menghitung komputasi kemungkinan atau probabilitas untuk tiap kelas target atas keseluruhan kelas target yang dapat membantu penentuan kelas target kepada *input* yang diberikan.

8.4. HARDWARE DAN SOFTWARE

Hardware dan software yang digunakan dalam praktikum ini yaitu:

19. Komputer.
20. IDE Python (Google collab, Jupyter Notebook)
21. Python 3.x

8.5. PRE-TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.		CPMK-01	Jelaskan perbedaan metode CNN dengan metode deep learning lainnya	40
2.		CPMK-02	Jelaskan kenapa metode CNN menggunakan pendekatan arsitektur Multi-Layer Perceptron (MLP)	60

8.6. LANGKAH PRAKTIKUM

Aturan Penilaian (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Dokumen Pendukung	Skor
1.	CPL-03	CPMK-01	Selesaikan langkah praktikum 1 – 9	Hasil praktikum langkah 1 – 9	
2.	dst				

Langkah-Langkah Praktikum:

1. Import library

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense
```

2. Menginisialisasi model cnn

```
# Inisialisasi CNN
MesinKlasifikasi = Sequential()
```

3. Langkah pertama yaitu Convolution dimana layer proses utama yang mendasari sebuah CNN

```
# Langkah 1 - Convolution
MesinKlasifikasi.add(Conv2D(filters = 32, kernel_size=(3, 3),
                             input_shape = (128, 128, 3), activation = 'relu'))
```

4. Langkah kedua yaitu Pooling layer untuk mereduksi input secara spasial (mengurangi jumlah parameter) dengan operasi down-sampling. Umumnya, metode pooling yang digunakan adalah max pooling atau mengambil nilai terbesar dari bagian tersebut.

```
# Langkah 2 - Pooling
MesinKlasifikasi.add(MaxPooling2D(pool_size = (2, 2)))
```

5. Menambah Convolution layer

```
# Menambah convolutional layer
MesinKlasifikasi.add(Conv2D(32, (3, 3), activation = 'relu'))
MesinKlasifikasi.add(MaxPooling2D(pool_size = (2, 2)))
```

6. Langkah ketiga yaitu Flatten layer merupakan operasi yang mengubah matriks menjadi vektor satu dimensi. Proses flattening mengubah feature map yang telah diperoleh dari layer sebelumnya menjadi vektor satu dimensi agar feature map tersebut dapat diklasifikasikan dengan fully-connected layer dan softmax.

```
# Langkah 3 - Flattening
MesinKlasifikasi.add(Flatten())
```

7. Langkah keempat yaitu membuat dua hidden layer

```
# Langkah 4 - Full connection
MesinKlasifikasi.add(Dense(units = 128, activation = 'relu'))
MesinKlasifikasi.add(Dense(units = 1, activation = 'sigmoid'))
```

8. Menjalankan model CNN yang sudah dibuat

```
# Menjalankan CNN
MesinKlasifikasi.compile(optimizer = 'adam',
                        loss = 'binary_crossentropy',
                        metrics = ['accuracy'])
```

9. menjalankan metode CNN dengan data testing dan data training

```
# Menjalankan CNN ke training dan test set
from tensorflow.keras.preprocessing.image import ImageDataGenerator

train_datagen = ImageDataGenerator(rescale = 1./255,
                                   shear_range = 0.2,
                                   zoom_range = 0.2,
                                   horizontal_flip = True)

test_datagen = ImageDataGenerator(rescale = 1./255)

training_set = train_datagen.flow_from_directory('dataset/training_set',
                                                target_size = (128, 128),
                                                batch_size = 32,
                                                class_mode = 'binary')

test_set = test_datagen.flow_from_directory('dataset/test_set',
                                           target_size = (128, 128),
                                           batch_size = 32,
                                           class_mode = 'binary')

MesinKlasifikasi.fit_generator(training_set,
                               steps_per_epoch = 8000/32,
                               epochs = 50,
                               validation_data = test_set,
                               validation_steps = 2000/32)
```

8.7. POST TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-03	CPMK-01	Tambahkan prediksi pada langkah praktikum tersebut	100
2.	dst			

8.8. HASIL CAPAIAN PRAKTIKUM

Diisi oleh asisten setelah semua assessment dinilai.

No	Bentuk Assessment	CPL	CPMK	Bobot	Skor (0-100)	Nilai Akhir (Bobot x Skor)
1.	Pre-Test	CPL-03	CPMK-01	20%		
2.	Praktik	CPL-03	CPMK-01	30%		
3.	Post-Test	CPL-03	CPMK-01	50%		
Total Nilai						

LEMBAR JAWABAN PRE-TEST DAN POST-TEST PRAKTIKUM



Nama : NIM :	Asisten: Paraf Asisten:	Tanggal: Nilai:
-------------------------------	------------------------------------------	----------------------------------

--



- Pertemuan ke** : 9
- Total Alokasi Waktu** : 90 menit
- Materi : 15 menit
 - Pre-Test : 15 menit
 - Praktikum : 45 menit
 - Post-Test : 15 menit

- Total Bobot Penilaian** : 100%
- Pre-Test : 20 %
 - Praktik : 30 %
 - Post-Test : 50 %

Pemenuhan CPL dan CPMK:

CPL-03	Deskripsi dari CPL tsb...
CPMK-01	Deskripsi dari CPMK tsb...

9.1. DESKRIPSI CAPAIAN PEMBELAJARAN

Setelah mengikuti praktikum ini mahasiswa diharapkan mampu:

7. Melakukan preprocessing untuk text, gambar dan numerik
8. Menentukan teknik yang tepat untuk melakukan preprocessing

9.2. INDIKATOR KETERCAPAIAN PEMBELAJARAN

Indikator ketercapaian diukur dengan:

CPL-01	CPMK-01	Kemampuan mahasiswa dalam menerapkan preprocessing pada dataset yang dimiliki
--------	---------	-------------------------------------------------------------------------------

9.3. TEORI PENDUKUNG

K-means merupakan algoritma *clustering*. **K-means Clustering** adalah salah satu “*unsupervised machine learning algorithms*” yang paling sederhana dan populer. *K-Means Clustering* adalah suatu metode penganalisaan data atau metode *Data Mining* yang melakukan proses pemodelan tanpa supervisi (*unsupervised*) dan merupakan salah satu metode yang melakukan pengelompokan data dengan sistem partisi.

K-means clustering merupakan salah satu metode *cluster analysis* non hirarki yang berusaha untuk mempartisi objek yang ada kedalam satu atau lebih *cluster* atau kelompok objek berdasarkan karakteristiknya, sehingga objek yang mempunyai karakteristik yang sama dikelompokkan dalam satu cluster yang sama dan objek yang mempunyai karakteristik yang berbeda dikelompokkan kedalam *cluster* yang lain. Metode *K-Means Clustering* berusaha mengelompokkan data yang ada ke dalam beberapa kelompok, dimana data dalam satu kelompok mempunyai karakteristik yang sama satu

sama lainnya dan mempunyai karakteristik yang berbeda dengan data yang ada di dalam kelompok yang lain.

Data clustering menggunakan metode *K-Means Clustering* ini secara umum dilakukan dengan algoritma dasar sebagai berikut:

1. Tentukan jumlah *cluster*
2. Alokasikan data ke dalam *cluster* secara *random*
3. Hitung *centroid*/rata-rata dari data yang ada di masing-masing *cluster*
4. Alokasikan masing-masing data ke *centroid*/rata-rata terdekat
5. Kembali ke Step 3, apabila masih ada data yang berpindah cluster atau apabila perubahan nilai *centroid*, ada yang di atas nilai *threshold* yang ditentukan atau apabila perubahan nilai pada *objective function* yang digunakan di atas nilai *threshold* yang ditentukan

Beberapa Permasalahan yang Terkait Dengan *K-Means Clustering*

Beberapa permasalahan yang sering muncul pada saat menggunakan metode *K-Means* untuk melakukan pengelompokan data adalah:

1. Ditemukannya beberapa model clustering yang berbeda
2. Pemilihan jumlah cluster yang paling tepat
3. Kegagalan untuk *converge*
4. *Outliers*
5. Bentuk *cluster*

Contoh aplikasi K-means

Contoh penggunaan K-means clustering antara lain:

- Segmentasi pasar (market segmentation)
- Segmentasi citra
- Kompresi gambar
- Klasifikasi citra penginderaan jauh

Seperti metode clustering lainnya, K-means digunakan dalam penelitian eksploratori, confirmatori dan eksplanatori.

Untuk tujuan-tujuan eksploratori, k-means dapat dimanfaatkan untuk melengkapi proses Exploratory Data Analysis (EDA), selain menggunakan analisis statistik deskriptif dan visualisasi data.

Sedangkan dalam proses confirmatori dan eksplanatori, K-means clustering dapat digunakan untuk melakukan konfirmasi terhadap teori-teori yang sudah ada.

Selain itu, algoritma ini juga digunakan untuk melakukan identifikasi jika tiba-tiba terjadi perubahan cluster setelah data baru masuk. Pada tahap terakhir, hasil cluster dapat digunakan sebagai dasar melakukan strategi atau kebijakan terhadap suatu masalah tertentu yang dikaji.

9.4. HARDWARE DAN SOFTWARE

Hardware dan software yang digunakan dalam praktikum ini yaitu:

22. Komputer.
23. IDE Python (Google collab, Jupyter Notebook)
24. Python 3.x

9.5. PRE-TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.			Apa yang dimaksud dengan K-means clustering ?	30
2.			Sebutkan contoh aplikasi yang menggunakan metode K-means clustering !	35
3.			Sebutkan kelebihan dan kelemahan dari metode K-means clustering !	35

9.6. LANGKAH PRAKTIKUM

Aturan Penilaian (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Dokumen Pendukung	Skor
1.	CPL-03	CPMK-01	Selesaikan langkah praktikum 1 – 9	Hasil praktikum langkah 1 – 9	
2.	dst				

Langkah-Langkah Praktikum:

1. Mengimport library

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

2. Mengimpor dataset pengunjung mall dengan variabel X menampung nilai kolom ke 3 dan 4 yaitu pendapatan dan juga pengeluaran.

```
# Mengimpor dataset
dataset = pd.read_csv('Pengunjung_mall.csv')
X = dataset.iloc[:, [3, 4]].values

dataset
```

dataset

	IDPelanggan	Kelamin	Usia	Pendapatan (juta Rp)	Rating_pengeluaran (1-100)
0	1	Laki	19	15	39
1	2	Laki	21	15	81
2	3	Perempuan	20	16	6
3	4	Perempuan	23	16	77
4	5	Perempuan	31	17	40
...
195	196	Perempuan	35	120	79
196	197	Perempuan	45	126	28
197	198	Laki	32	126	74
198	199	Laki	32	137	18
199	200	Laki	30	137	83

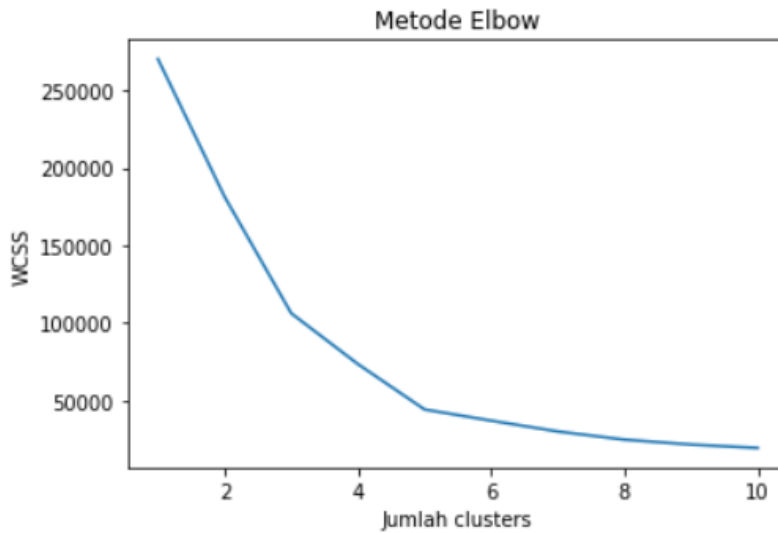
200 rows × 5 columns

3. Nilai X yang berisi pendapatan dan pengeluaran dari pengunjung mall

```
x
array([[ 15,  39],
       [ 15,  81],
       [ 16,   6],
       [ 16,  77],
       [ 17,  40],
       [ 17,  76],
       [ 18,   6],
       [ 18,  94],
       [ 19,   3],
       [ 19,  72],
       [ 19,  14],
       [ 19,  99],
       [ 20, 151],
```

4. Implementasi metode K-Means Clustering menggunakan metode elbow dalam menentukan angka clusternya yang tepat.

```
# Menggunakan metode elbow untuk menentukan angka cluster yang tepat
from sklearn.cluster import KMeans
wcss = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters = i, init = 'k-means++', random_state = 42)
    kmeans.fit(X)
    wcss.append(kmeans.inertia_)
plt.plot(range(1, 11), wcss)
plt.title('Metode Elbow')
plt.xlabel('Jumlah clusters')
plt.ylabel('WCSS')
plt.show()
```

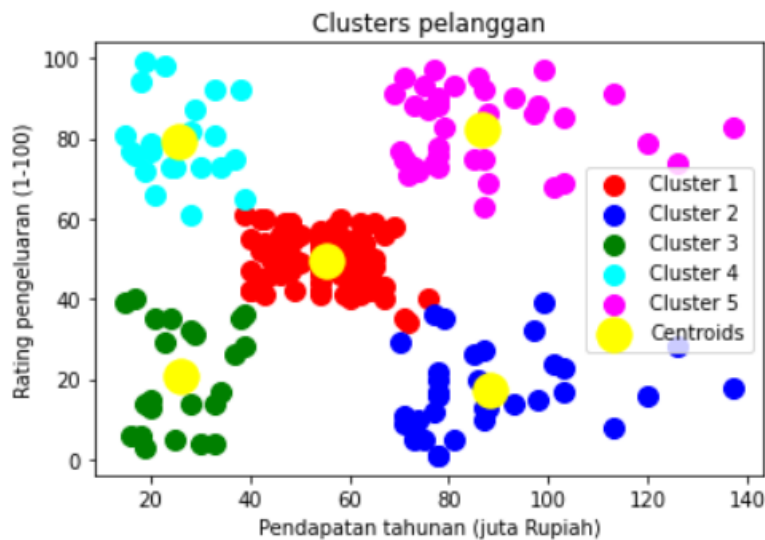


5. Menjalankan metode K-Means Clustering ke dataset

```
kmeans = KMeans(n_clusters = 5, init = 'k-means++', random_state = 42)
y_kmeans = kmeans.fit_predict(X)
```

6. Visualisasi hasil Clusters

```
plt.scatter(X[y_kmeans == 0, 0], X[y_kmeans == 0, 1], s = 100, c = 'red', label = 'Cluster 1')
plt.scatter(X[y_kmeans == 1, 0], X[y_kmeans == 1, 1], s = 100, c = 'blue', label = 'Cluster 2')
plt.scatter(X[y_kmeans == 2, 0], X[y_kmeans == 2, 1], s = 100, c = 'green', label = 'Cluster 3')
plt.scatter(X[y_kmeans == 3, 0], X[y_kmeans == 3, 1], s = 100, c = 'cyan', label = 'Cluster 4')
plt.scatter(X[y_kmeans == 4, 0], X[y_kmeans == 4, 1], s = 100, c = 'magenta', label = 'Cluster 5')
plt.scatter(kmeans.cluster_centers[:, 0], kmeans.cluster_centers[:, 1],
            s = 300, c = 'yellow', label = 'Centroids')
plt.title('Clusters pelanggan')
plt.xlabel('Pendapatan tahunan (juta Rupiah)')
plt.ylabel('Rating pengeluaran (1-100)')
plt.legend()
plt.show()
```



9.7. POST TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-03	CPMK-01	lakukan klasifikasi K-Means Clustering dengan dataset apa saja yang kalian dapat	100

9.8. HASIL CAPAIAN PRAKTIKUM

Diisi oleh asisten setelah semua assessment dinilai.

No	Bentuk Assessment	CPL	CPMK	Bobot	Skor (0-100)	Nilai Akhir (Bobot x Skor)
1.	Pre-Test	CPL-03	CPMK-01	20%		
2.	Praktik	CPL-03	CPMK-01	30%		
3.	Post-Test	CPL-03	CPMK-01	50%		
Total Nilai						

LEMBAR JAWABAN PRE-TEST DAN POST-TEST PRAKTIKUM

Nama : NIM :	Asisten: Paraf Asisten:	Tanggal: Nilai:
-------------------------------	------------------------------------------	----------------------------------

PRAKTIKUM 10: CONFUSION MATRIX

Pertemuan ke : 10

Total Alokasi Waktu : 90 menit

- Materi : 15 menit
- Pre-Test : 15 menit
- Praktikum : 45 menit
- Post-Test : 15 menit

Total Bobot Penilaian : 100%

- Pre-Test : 20 %
- Praktik : 30 %
- Post-Test : 50 %

Pemenuhan CPL dan CPMK:

CPL-03	Deskripsi dari CPL tsb...
CPMK-01	Deskripsi dari CPMK tsb...

10.1. DESKRIPSI CAPAIAN PEMBELAJARAN

Setelah mengikuti praktikum ini mahasiswa diharapkan mampu:

9. Melakukan preprocessing untuk text, gambar dan numerik
10. Menentukan teknik yang tepat untuk melakukan preprocessing

10.2. INDIKATOR KETERCAPAIAN PEMBELAJARAN

Indikator ketercapaian diukur dengan:

CPL-01	CPMK-01	Kemampuan mahasiswa dalam menerapkan preprocessing pada dataset yang dimiliki
--------	---------	-------------------------------------------------------------------------------

10.3 TEORI PENDUKUNG

Confusion matrix adalah sebuah tabel yang sering digunakan untuk mengukur kinerja dari model klasifikasi di *machine learning*. Tabel ini menggambarkan lebih detail tentang jumlah data yang diklasifikasikan dengan benar maupun salah.

Confusion matrix adalah salah satu *tools* analitik prediktif yang menampilkan dan membandingkan nilai aktual atau nilai sebenarnya dengan nilai hasil prediksi model yang dapat digunakan untuk menghasilkan metrik evaluasi seperti *Accuracy* (akurasi), *Precision*, *Recall*, dan *F1-Score* atau *F-Measure*.

Confusion matrix adalah metode untuk menguji performa dalam suatu sistem. Pada pengukuran kinerja menggunakan confusion matrix, terdapat empat istilah sebagai representasi hasil proses klasifikasi. Keempat istilah tersebut adalah True Positive (TP) merupakan data positif yang terdeteksi benar, True Negative (TN) merupakan jumlah data negatif yang terdeteksi benar, False Positive (FP) yaitu data negatif namun terdeteksi sebagai data positif, dan False Negative (FN) yaitu kebalikan dari True Positive. Ilustrasi tabel *confusion matrix* dapat dilihat pada gambar berikut.

		Nilai Aktual	
		Positive	Negative
Nilai Prediksi	Positive	TP	FP
	Negative	FN	TN

- True Positive (TP)** : Jumlah data yang bernilai Positif dan diprediksi benar sebagai Positif.
- False Positive (FP)** : Jumlah data yang bernilai Negatif tetapi diprediksi sebagai Positif.
- False Negative (FN)** : Jumlah data yang bernilai Positif tetapi diprediksi sebagai Negatif.
- True Negative (TN)**: Jumlah data yang bernilai Negatif dan diprediksi benar sebagai Negatif.

1. Akurasi

Akurasi merupakan persentase dari total data yang diidentifikasi dan dinilai yang menggambarkan seberapa akurat sistem dapat mengklasifikasikan data secara benar. Dapat dilihat dari persamaan di bawah ini.

$$\text{Akurasi} = \frac{TP}{\text{jumlah semua data}} \times 100\%$$

2. Presisi

Presisi adalah nilai yang menggambarkan jumlah kategori positif yang diklasifikasikan secara benar dengan dibagi total data yang diklasifikasi positif. Dapat dilihat dari persamaan di bawah ini.

$$\text{Presisi} = \frac{TP}{TP+FP} \times 100\%$$

3. Recall

Recall yaitu nilai yang menunjukkan berapa persen data kategori positif yang terklasifikasi dengan benar oleh sistem. Dapat dilihat persamaan di bawah ini.

$$\text{Recall} = \frac{TP}{TP+FN} \times 100\%$$

10.4. HARDWARE DAN SOFTWARE

Hardware dan software yang digunakan dalam praktikum ini yaitu:

25. Komputer.
26. IDE Python (Google collab, Jupyter Notebook)
27. Python 3.x

10.5. PRE-TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.			Jelaskan pengertian confusion matrix	25
2.			Apa yang disebut dengan akurasi, presisi dan recall?	25
3.			Sebutkan metode apa saja yang digunakan untuk mengukur kinerja sistem	25
			Jelaskan perbedaan metode confusion matrix dengan metode yang lain	25

10.6. LANGKAH PRAKTIKUM

Aturan Penilaian (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Dokumen Pendukung	Skor
1.	CPL-03	CPMK-01	Selesaikan langkah praktikum 1 – 7	Hasil praktikum langkah 1 – 7	
2.	dst				

Langkah-Langkah Praktikum:

1. Import library

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

2. Mengimpor dataset dan mengolah data sebelum dilakukan proses perhitungan Confusion Matrix

```
# membaca dataset
data = pd.read_csv("breast-cancer-wisconsin-data.csv")

# menghapus kolom yang tidak digunakan
data.drop(["Unnamed: 32", "id"], axis=1, inplace=True)

# merubah label M(ganas) = 1 dan B(jinak) = 0
data.diagnosis = [1 if each == "M" else 0 for each in data.diagnosis]
data.head(3) # menampilkan sample data
```

3. Menentukan nilai X dan Y

```
kolom = data.columns.to_list()
x = data[kolom[1:30]]
y = data[kolom[0]]
```


4. Membagi data menjadi dua yaitu data testing dan data training

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y,
                                                    test_size=0.25,
                                                    random_state=42)
```

5. Melakukan prediksi dengan menggunakan metode decision tree

```
from sklearn.tree import DecisionTreeClassifier
model = DecisionTreeClassifier()
model.fit(x_train, y_train)
y_pred = model.predict(x_test) # prediksi
```

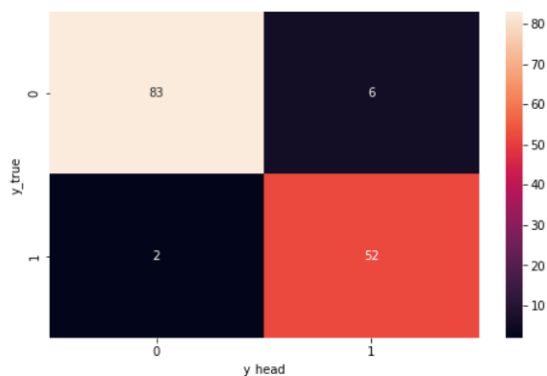
6. Melakukan perhitungan confusion matrix pada hasil prediksi

```
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test, y_pred)
```

```
array([[83,  6],
       [ 2, 52]])
```

7. visualisasi hasil confusion matrix

```
import seaborn as sns
import matplotlib.pyplot as plt
f, ax = plt.subplots(figsize=(8,5))
sns.heatmap(confusion_matrix(y_test, y_pred), annot=True, fmt=".0f", ax=ax)
plt.xlabel("y_head")
plt.ylabel("y_true")
plt.show()
```



10.7. POST TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-03	CPMK-01	hitunglah akurasi, presisi dan recall dari hasil matrix yang dibuat pada langkah praktikum	100
2.	dst			

10.8. HASIL CAPAIAN PRAKTIKUM

Diisi oleh asisten setelah semua assessment dinilai.

No	Bentuk Assessment	CPL	CPMK	Bobot	Skor (0-100)	Nilai Akhir (Bobot x Skor)
1.	Pre-Test	CPL-03	CPMK-01	20%		
2.	Praktik	CPL-03	CPMK-01	30%		
3.	Post-Test	CPL-03	CPMK-01	50%		
Total Nilai						

LEMBAR JAWABAN PRE-TEST DAN POST-TEST PRAKTIKUM

<p>Nama : NIM :</p>	<p>Asisten: Paraf Asisten:</p>	<p>Tanggal: Nilai:</p>
---------------------------------------	--------------------------------------------------	------------------------------------------