# PERTEMUAN KE-1:
# PENGANTAR BAHASA PEMROGRAMAN PYTHON UNTUK DATA MINING

## A. TUJUAN

Setelah menempuh praktikum ini diharapkan mahasiswa mampu

1. memahami dasar pemrograman Python yang mendukung *data mining*.
2. membuat program sederhana yang mendasari *data mining*.

## B. MATERI

Python adalah bahasa pemrograman yang sangat efektif untuk data mining karena menyediakan berbagai alat dan pustaka (libraries) yang mendukung berbagai tahap analisis data, mulai dari pengumpulan data hingga visualisasi dan pembuatan model prediksi. Salah satu pustaka paling populer adalah Pandas, yang memungkinkan pengguna untuk melakukan manipulasi data dengan mudah melalui struktur data seperti DataFrame. NumPy mendukung komputasi numerik dengan array multidimensi, yang sangat berguna untuk operasi matematika yang intensif. Pustaka untuk visualisasi data seperti Matplotlib dan Seaborn memungkinkan pembuatan grafik yang informatif, membantu pengguna dalam memahami pola data secara visual. Scikit-learn menyediakan berbagai algoritma untuk pembelajaran mesin untuk klasifikasi, regresi, dan clustering yang dapat digunakan dalam data mining, sedangkan TensorFlow dan PyTorch mendukung pengembangan model pembelajaran mendalam (deep learning) untuk analisis data yang lebih kompleks. Selain itu, NLTK dan spaCy sangat berguna dalam pemrosesan bahasa alami (NLP), memungkinkan analisis teks dan pengolahan data tidak terstruktur. Untuk ekstraksi data dari web, pustaka seperti BeautifulSoup dan Scrapy memudahkan pengguna dalam melakukan web scraping untuk mengumpulkan data yang dibutuhkan. Semua pustaka ini menjadikan Python sebagai pilihan utama bagi para data scientist dan profesional di bidang data mining untuk mengolah dan menganalisis data dalam berbagai bentuk dan skala. Berikut adalah daftar pustaka utama yang sering digunakan untuk data mining:

### 1. Pandas

Pandas adalah pustaka Python yang menyediakan struktur data seperti DataFrame dan Series, yang memungkinkan manipulasi data tabular yang sangat efisien. Pustaka ini sangat berguna untuk pembersihan data, pengolahan data, dan analisis data statistik. Dengan Pandas, kita bisa melakukan operasi seperti filtering, grouping, aggregating, dan merging data. Pandas berguna untuk ,emproses data yang tidak terstruktur, membersihkan data yang hilang atau duplikat, mengubah format data, dan melakukan analisis statistik dasar.

### 2. NumPy

NumPy adalah pustaka Python yang digunakan untuk komputasi numerik dan menyediakan array multidimensi, yang lebih efisien dibandingkan dengan struktur data Python standar. NumPy juga memiliki berbagai fungsi matematika untuk operasi vektorisasi dan statistik. Komputasi numerik tingkat lanjut, operasi matematika dengan array besar, dan digunakan bersama dengan pustaka lain seperti Pandas dan SciPy untuk analisis data.

### 3. Matplotlib

Matplotlib adalah pustaka visualisasi data yang memungkinkan pengguna membuat berbagai jenis grafik dan plot, seperti grafik garis, batang, histogram, dan scatter plot. Visualisasi ini membantu pengguna untuk lebih memahami pola dan distribusi dalam data. Kegunaan dari Matplotlib adalah untuk membuat visualisasi dasar seperti grafik garis, diagram batang, dan histogram untuk analisis data.

### 4. Seaborn

Seaborn dibangun di atas Matplotlib dan memberikan API yang lebih sederhana untuk visualisasi statistik yang lebih menarik dan informatif. Seaborn mendukung visualisasi distribusi data, hubungan

antar variabel, dan analisis korelasi. Kegunaan Seaborn adalah untuk visualisasi data statistik, termasuk heatmap, box plot, violin plot, dan pair plot, yang lebih mudah digunakan dan lebih menarik secara visual dibandingkan Matplotlib.

5. **Scikit-learn**

Scikit-learn adalah pustaka Python untuk pembelajaran mesin (machine learning) yang menyediakan banyak algoritma untuk klasifikasi, regresi, clustering, dan reduksi dimensi. Selain itu, pustaka ini juga mencakup alat untuk pemodelan dan evaluasi model. Kegunaan Scikit-learn adalah untuk membangun dan menguji model pembelajaran mesin seperti K-Nearest Neighbors (KNN), Support Vector Machines (SVM), regresi linear, dan Random Forests. Pustaka ini juga memiliki alat untuk pemilihan fitur dan evaluasi kinerja model.

6. **TensorFlow**

TensorFlow adalah pustaka open-source untuk pembelajaran mesin dan deep learning yang dikembangkan oleh Google. TensorFlow memungkinkan pengembangan dan pelatihan model pembelajaran mendalam (deep learning) yang sangat kompleks menggunakan arsitektur jaringan saraf tiruan. Kegunaan TensorFlow adalah untuk membangun dan melatih model pembelajaran mendalam untuk klasifikasi gambar, analisis teks, pemrosesan bahasa alami, dan banyak lagi. TensorFlow juga mendukung penggunaan GPU untuk mempercepat pelatihan model.

7. **PyTorch**

PyTorch adalah pustaka pembelajaran mendalam yang dikembangkan oleh Facebook, dikenal karena API-nya yang dinamis dan mudah digunakan. PyTorch mendukung pembelajaran mendalam serta komputasi numerik secara umum, dengan keunggulan dalam fleksibilitas dan debugging. Kegunaan PyTorch adalah untuk membangun dan melatih model deep learning dengan jaringan saraf konvolusional (CNN), jaringan saraf berulang (RNN), dan model generatif seperti GANs.

8. **NLTK (Natural Language Toolkit)**

NLTK adalah pustaka yang menyediakan alat untuk pemrosesan bahasa alami (NLP), termasuk tokenisasi, stemming, lemmatization, dan analisis teks. NLTK banyak digunakan dalam proyek analisis teks dan pembelajaran mesin berbasis teks. Kegunaan NLTK adalah untuk menangani analisis dan pemrosesan teks, termasuk tokenisasi kata, analisis sentimen, dan pembuatan model berbasis kata (seperti TF-IDF dan Word2Vec).

9. **spaCy**

spaCy adalah pustaka NLP yang cepat dan efisien, dirancang untuk aplikasi pemrosesan teks komersial dan industri. SpaCy memiliki model untuk analisis sintaksis, entitas pengenalan (NER), dan pengelompokan kata. Kegunaan SpaCy adalah untuk ekstraksi informasi, analisis sintaksis, pengenalan entitas bernama (NER), dan pemrosesan teks dalam jumlah besar.

10. **BeautifulSoup**

BeautifulSoup adalah pustaka Python yang digunakan untuk parsing dan ekstraksi data dari HTML dan XML. Ini sangat berguna untuk web scraping, memungkinkan pengguna untuk menavigasi dan mengekstrak informasi dari halaman web. Kegunaan BeautifulSoup adalah untuk ekstraksi data dari halaman web dengan HTML parsing, memungkinkan pengguna untuk menemukan dan mengumpulkan data dari elemen web seperti tabel atau daftar.

11. **Scrapy**

Scrapy adalah framework open-source untuk web scraping yang memungkinkan pengguna untuk mengekstrak data dari situs web dengan cara yang efisien dan terstruktur. Scrapy memungkinkan pengguna untuk melakukan crawling, ekstraksi data, dan menyimpannya dalam berbagai format seperti JSON atau CSV. Kegunaan Scrapy adalah untuk Web scraping dalam skala besar, memungkinkan pengguna untuk membangun "spider" untuk mengumpulkan data dari banyak halaman web secara otomatis.

12. **OpenCV**

OpenCV adalah pustaka sumber terbuka untuk pengolahan gambar dan visi komputer, yang memungkinkan pengguna untuk memanipulasi gambar dan video serta mendeteksi objek. OpenCV dapat digunakan dalam berbagai aplikasi mulai dari pemrosesan citra medis hingga pengawasan video.

Kegunaan OpenCV adalah untuk pengolahan gambar, deteksi objek, pengenalan wajah, dan aplikasi visi komputer lainnya.

**13.Statsmodels**

Statsmodels adalah pustaka untuk analisis statistik yang menyediakan berbagai model untuk regresi, analisis deret waktu, dan berbagai uji statistik. Pustaka ini juga mendukung perhitungan interval kepercayaan dan uji hipotesis. Kegunaan Statsmodels adalah untuk analisis regresi, uji statistik, model time series, dan model statistik lainnya.

**14.Gensim**

Gensim adalah pustaka yang digunakan untuk pemodelan topik dan analisis semantik dalam teks, seperti Latent Dirichlet Allocation (LDA). Gensim juga mendukung pembelajaran mesin berbasis kata dengan menggunakan model seperti Word2Vec. Kegunaan Gensim adalah untuk pemodelan topik, analisis semantik, dan pembuatan model representasi kata.

**15.XGBoost**

XGBoost adalah pustaka pembelajaran mesin untuk boosting yang sangat efisien dan efektif, sering digunakan dalam kompetisi data science. XGBoost adalah implementasi dari algoritma Gradient Boosting yang memfokuskan pada kecepatan dan kinerja. Kegunaan Pustaka XGBoost Membuat model prediksi yang kuat dengan menggunakan teknik boosting, sering digunakan dalam klasifikasi dan regresi untuk dataset besar.

**Dasar Bahasa Pemrogram Python**

Python mendukung bilangan bulat, angka float dan bilangan kompleks. Variabel adalah wadah untuk menyimpan nilai data. Tidak seperti bahasa pemrograman lain, Python tidak memiliki perintah untuk mendeklarasikan variabel. Variabel dapat menyimpan data dari tipe yang berbeda, dan tipe yang berbeda dapat melakukan hal yang berbeda. Python memiliki tipe data berikut bawaan secara bawaan, dalam kategori ini: str, int, float, complex, list, tuple, range, dict, set, frozenset, bool, bytes, bytearray, memoryview. Selain itu, Python juga menyediakan berbagai operator. Operator digunakan untuk melakukan operasi pada variabel dan nilai. Python membagi operator dalam kelompok-kelompok berikut: Operator aritmatika, Operator penugasan, Operator pembanding, Operator logika, Operator identitas, Operator keanggotaan, dan Operator bitwise. Beberapa operator yang sering digunakan akan kita bahas dalam praktikum ini. Python menyediakan Operator Aritmatika. Operator aritmatika digunakan dengan nilai numerik untuk melakukan operasi matematika umum sebagaimana ada pada Tabel 1.1.

**Tabel 1.1.** Operator Aritmatika

| Operator | Nama | Contoh |
|---|---|---|
| + | Penjumlahan | x + y |
| - | Pengurangan | x - y |
| * | Perkalian | x * y |
| / | Pembagian | x / y |
| % | Modulus | x % y |
| ** | pangkat | x ** y |
| // | *Floor division* | x // y |

Python juga menyediakan Operator Penugasan seperti bahasa pemrograman lainnya. Operator penugasan digunakan untuk menetapkan nilai ke variabel. Operator penugasan yang disediakan Python dapat dilihat pada Tabel 1.2.

**Tabel 1.2.** Operator Penugasan

| Operator | Contoh | Kesaamaan Arti |
|---|---|---|
| = | x = 5 | x = 5 |
| += | x += 3 | x = x + 3 |
| -= | x -= 3 | x = x – 3 |

| | | |
|---|---|---|
| *= | x *= 3 | x = x * 3 |
| /= | x /= 3 | x = x / 3 |
| %= | x %= 3 | x = x % 3 |
| //= | x //= 3 | x = x // 3 |
| **= | x **= 3 | x = x ** 3 |
| &= | x &= 3 | x = x & 3 |
| \|= | x \|= 3 | x = x \| 3 |
| ^= | x ^= 3 | x = x ^ 3 |
| >>= | x >>= 3 | x = x >> 3 |
| <<= | x <<= 3 | x = x << 3 |

Operator Perbandingan pada Python digunakan untuk membandingkan dua nilai. Operotor Operator Perbandingan pada Python dapat dilihat pada Tabel 1.3.

**Tabel 1.3.** Operator Perbandingan

| Operator | Nama | Contoh |
|---|---|---|
| == | Sama dengan | x == y |
| != | Tidak sama dengan | x != y |
| > | Lebih dari | x > y |
| < | Lebih kecil dari | x < y |
| >= | Lebih besar dari atau sama dengan | x >= y |
| <= | Lebih kecil dari atau sama dengan | x <= y |

Operator logika pada Python digunakan untuk menggabungkan pernyataan bersyarat. Operasi yang ada pada Python dapat dilihat pada Tabel 1.4.

**Tabel 1.4.** Operator Logika

| Operator | Penjelasan | Contoh |
|---|---|---|
| and | Hasilnya True if kedua pernyataan bernilai true | x < 5 and x < 10 |
| or | Hasilnya True jika satu dari pernyataaan bernilai true | x < 5 or x < 4 |
| not | kebalikan hasil, bernilai False jika hasil bernilai benar true | not(x < 5 and x < 10) |

**Pernyataan if**

Python mendukung kondisi logis yang biasa dari matematika: **Sama dengan**: a == b, **Tidak Sama dengan**: a! = B, **Kurang dari**: a <b, **Kurang dari atau sama dengan**: a <= b, **Lebih besar dari**: a> b**, Lebih besar dari atau sama dengan**: a> = b. Sebuah "pernyataan if" ditulis dengan kata kunci if. Python bergantung pada indentasi (spasi putih di awal baris) untuk mendefinisikan lingkup dalam kode. Bahasa pemrograman lain sering menggunakan kurung kurawal untuk tujuan ini.

**Contoh 1.1**

**Kode Program 1.1.** Contoh program yang menggunakan if.

| No | Kode Program |
|---|---|
| 1 | `a = 200` |
| 2 | `b = 33` |
| 3 | `if b > a:` |
| 4 | `  print("b is greater than a")` |
| 5 | `elif a == b:` |
| 6 | `  print("a and b are equal")` |
| 7 | `else:` |
| 8 | `  print("a is greater than b")` |

Dalam contoh 1.1. a lebih besar dari b, jadi kondisi pertama tidak benar dan kondisi elif juga tidak benar, maka program mencetak ke layar bahwa " `a is greater than b` ".

**Pernyataan Perulangan (while dan for)**

Python memiliki dua perintah loop primitif yaitu while loops dan for loops. Dengan loop while, maka dapat menjalankan serangkaian pernyataan selama suatu kondisi benar. Dengan pernyataan break maka dapat menghentikan loop bahkan jika kondisi while benar.

**Contoh 1.2**

**Kode Program 1.2.** Contoh program yang menggunakan while dan break

| No | Kode Program |
|---|---|
| 1 | `i = 1` |
| 2 | `while i < 6:` |
| 3 | `  print(i)` |
| 4 | `  if i==4:` |
| 5 | `  break` |
| 6 | `  i += 1` |

Sebuah for loop digunakan untuk mengulangi urutan (baik list, tupel, dict, set, atau string). Contoh 1.3 memperlihatkan perintah for untuk untuk mengambil elemen dari suatu list.

**Contoh 1.3**

**Kode Program 1.3.** Contoh program yang menggunakan for dan break

| No | Kode Program |
|---|---|
| 1 | `fruits = ["apple", "banana", "cherry"]` |
| 2 | `for x in fruits:` |
| 3 | `  print(x)` |
| 4 | `  if x == "banana":` |
| 5 | `  break` |

**Contoh 1.4**

Buatlah program untuk menguji apakah dua kelompok memiliki rata-rata yang berbeda secara signifikan dengan menggunakan Pustaka statsmodels.

**Jawaban**

**Kode Program 1.4.** Program untuk menguji apakah dua kelompok memiliki rata-rata yang berbeda secara signifikan.

| No | Kode Program |
|---|---|
| 1 | `from statsmodels.stats.weightstats import ttest_ind` |
| 2 | `import numpy as np` |
| 3 | `# Contoh data dua kelompok` |
| 4 | `group1 = np.random.normal(50, 10, 30)` |
| 5 | `group2 = np.random.normal(55, 10, 30)` |
| 6 | `# Melakukan uji t` |
| 7 | `t_stat, p_value, df = ttest_ind(group1, group2)` |
| 8 | `print(f"T-statistik: {t_stat:.3f}, P-value: {p_value:.3f}")` |

T-statistik: -1.404, P-value: 0.166

Karena P-value (0.166) lebih besar dari 0.05, kita tidak dapat menolak hipotesis nol. Ini berarti tidak ada cukup bukti statistik untuk mengatakan bahwa dua kelompok memiliki rata-rata yang berbeda secara signifikan..

**Contoh 1.5**

Buatlah program untuk uji normalitas suatu data.

**Jawaban**

**Kode Program 1.5.** Program untuk untuk uji normalitas suatu data

| No | Kode Program |
|---|---|
| 1 | `from statsmodels.stats.diagnostic import normal_ad` |
| 2 | `# Contoh data` |
| 3 | `data = np.random.normal(0, 1, 100)` |
| 4 | `# Uji normalitas` |
| 5 | `stat, p_value = normal_ad(data)` |
| 6 | `print(f"Statistic: {stat:.3f}, P-value: {p_value:.3f}")` |

Statistic: 0.242, P-value: 0.763

Uji normalitas (misalnya, uji Anderson-Darling, Shapiro-Wilk, atau Kolmogorov-Smirnov) digunakan untuk menentukan apakah data berdistribusi normal. P-value = 0.763 berarti ada 76.3% kemungkinan bahwa data ini berasal dari distribusi normal. Biasanya, kita menggunakan $\alpha = 0.05$ sebagai batas signifikansi: Jika P-value ≤ 0.05, kita menolak hipotesis nol (data tidak berdistribusi normal) dan Jika P-value > 0.05, kita gagal menolak hipotesis nol (tidak ada bukti kuat bahwa data tidak normal). Karena P-value (0.763) jauh lebih besar dari 0.05, maka tidak menolak hipotesis nol, yang berarti data dianggap berdistribusi normal.

**Kode Program 1.6.** Program untuk menghitung MSE dengan pustaka Scikit-learn

| No | Kode Program |
|---|---|
| 1 | `from sklearn.metrics import mean_squared_error` |
| 2 | `y_true = [3, -0.5, 2, 7]` |
| 3 | `y_pred = [2.5, 0.0, 2, 8]` |
| 4 | `mse = mean_squared_error(y_true, y_pred)` |
| 5 | `print("MSE:", mse)` |

## C. TUGAS/LATIHAN SOAL

1. Data tinggi badan (X) dan berat badan (Y) dari 50 orang:
   X = [172, 181, 178, 160, 185, 176, 182, 169, 180, 175, 177, 183, 174, 168, 179, 184, 186, 170, 171, 173, 161, 187, 165, 162, 188, 166, 189, 163, 164, 167, 190, 159, 158, 157, 156, 155, 154, 153, 152, 151, 150, 149, 148, 147, 146, 145, 144, 143, 142, 141]
   y = [58, 72, 68, 55, 75, 66, 70, 60, 71, 65, 67, 73, 64, 59, 69, 74, 76, 61, 62, 63, 54, 77, 57, 55, 78, 58, 79, 56, 57, 58, 80, 53, 52, 51, 50, 49, 48, 47, 46, 45, 44, 43, 42, 41, 40, 39, 38, 37, 36, 35]
   Gunakan Statsmodels untuk membuat model regresi linear dan tampilkan hasilnya.
2. Data jumlah jam belajar (X1), jumlah jam tidur (X2), dan status kelulusan (y) dari 20 siswa:
   X1 = [2, 4, 6, 8, 3, 7, 5, 9, 1, 10, 3, 6, 2, 8, 4, 7, 5, 9, 1, 10]
   X2 = [6, 7, 5, 6, 8, 5, 9, 4, 7, 6, 5, 8, 6, 7, 4, 5, 8, 6, 7, 5]
   y = [0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1]
   Gunakan Statsmodels untuk membuat model regresi logistik dan tampilkan ringkasan hasil regresinya.
3. Data skor ujian dari dua kelas (A dan B):
   kelas_A = [70, 75, 80, 85, 90, 72, 78, 82, 88, 76, 79, 81, 84, 77, 74, 86, 83, 87, 89, 71, 73, 78, 80, 82, 85, 88, 90, 76, 79, 84]
   kelas_B = [74, 77, 79, 85, 83, 81, 86, 82, 89, 75, 76, 78, 80, 84, 88, 90, 92, 77, 79, 85, 87, 89, 91, 73, 75, 79, 82, 84, 86, 88]
   Gunakan Statsmodels untuk melakukan uji t dua sampel (independent t-test) dan interpretasikan hasilnya.
4. Data jumlah pelanggan harian toko selama 20 hari:
   data = [120, 130, 125, 140, 135, 150, 145, 155, 160, 158, 162, 170, 165, 175, 180, 178, 185, 190, 195, 200]. Gunakan Statsmodels untuk membangun model ARIMA dengan parameter (p=2, d=1, q=2) dan tampilkan ringkasan hasil model.