**Algoritma K-Nearest Neighbor (KNN) untuk Klasifikasi**

Algoritma KNN atau K-Nearest Neighbor adalah salah satu algoritma yang banyak digunakan di dunia *machine learning* untuk kasus klasifikasi. Disini saya akan membahas tentang cara kerja algoritma KNN.

**Apa itu Algoritma KNN?**

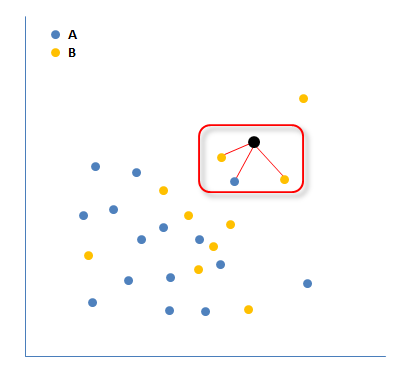
Seperti yang telah disebutkan, algoritma KNN merupakan algoritma klasifikasi yang bekerja dengan mengambil sejumlah K data terdekat (tetangganya) sebagai acuan untuk menentukan kelas dari data baru. Algoritma ini mengklasifikasikan data berdasarkan *similarity* atau kemiripan atau kedekatannya terhadap data lainnya.

Dalam K-Nearest Neighbor, data point yang berada berdekatan disebut “neighbor” atau  “tetangga”.

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.

Perhatikan gambar ilustrasi di bawah ini.



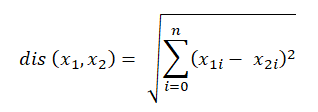
Dari gambar di atas, ada sejumlah data point yang terbagi menjadi dua kelas yaitu A (biru) dan B (kuning). Misalnya ada data baru (hitam) yang akan kita prediksi kelasnya menggunakan algoritma KNN. Dari contoh di atas, nilai K yang digunakan adalah 3. Setelah dihitung jarak antara titik hitam ke masing-masing data point lainnya, didapatkan 3 titik terdekat yang terdiri dari 2 titik kuning dan satu titik biru seperti yang diilustrasikan di dalam kotak merah, maka kelas untuk data baru (titik hitam) adalah B (kuning).

**Menghitung jarak dengan Euclidean Distance**

Untuk menghitung jarak antara dua titik pada algoritma KNN digunakan metode ***Euclidean Distance*** yang dapat digunakan pada *1-dimensional space*, *2-dimensional space*, atau *multi-dimensional space*.

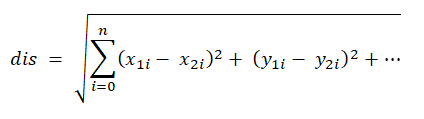
*1-dimensional space* berarti perhitungan jarak hanya menggunakan satu variabel bebas (*independent variable), 2-dimensional-space* berarti ada dua variabel bebas, dan *multi-dimensional space* berarti ada lebih dari dua variabel.

Secara umum, formula Euclidean distance pada *1-dimensional space* adalah sebagai berikut.



Formula di atas dapat digunakan jika jumlah *independent variable* hanya ada satu variabel. Lalu, bagaimana jika ada banyak variabel yang digunakan?

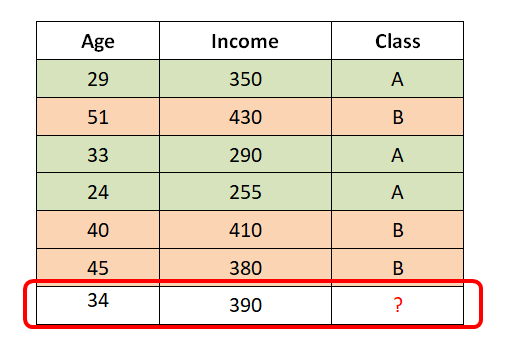
Jika ada lebih dari satu, kita dapat menjumlahkannya seperti di bawah ini.



Agar dapat lebih memahami cara kerja algoritma KNN menggunakan Euclidean distance, mari kita coba terapkan pada contoh kasus.

**Contoh cara kerja algoritma KNN**

Sebagai contoh, misalnya kita memiliki data *customer* seperti di bawah ini.



Dari tabel di atas, ada 6 data yang sudah berlabel dan 1 satu data baru yang harus kita tentukan kelasnya dengan detail informasi sebagai berikut:

* Ada 2 kelas yaitu A dan B
* **Age** dan **Income** adalah *independent variables* atau variabel yang nilainya tidak dipengaruhi oleh variabel lain dan akan digunakan untuk menghitung jarak
* **Class** merupakan *dependent variable*, variabel yang nilainya dipengaruhi oleh variabel lain (Age dan Income)

**Tentukan nilai K**

Pertama, kita harus menentukan nilai K terlebih dahulu. Penentuan nilai K ini tidak ada rumus pastinya. Namun satu tips yang dapat dipertimbangkan, yakni jika kelas berjumlah genap maka sebaiknya nilai K-nya ganjil, sebaliknya jika kelas berjumlah ganjil maka sebaiknya nilai K-nya genap.

Mengapa demikian?

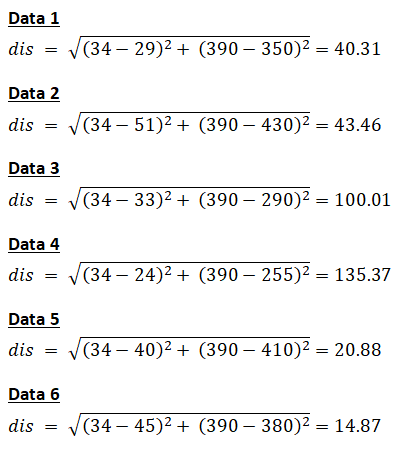
Logikanya begini, misalnya seperti contoh di atas ada dua kelas (genap) A dan B, jika kita ambil nilai K genap juga misalnya 4, maka akan ada kemungkinan hasil 4 tetangga terdekat terdiri dari 2 kelas A dan 2 kelas B sehingga sama hasilnya, bisa A bisa B. Namun jika ambil nilai K ganjil, misal 3 atau 5, maka otomatis akan ada yang lebih banyak jumlahnya.

Dalam prakteknya di Python, Anda dapat menghitung menggunakan kode program untuk mencari nilai K terbaik dari berbagai opsi nilai (misalnya dari K=2 sampai K=10).

Karena disini saya akan memberikan contoh cara kerja algoritma KNN, jadi saya akan langsung tentukan nilai K yang digunakan yaitu K=3.

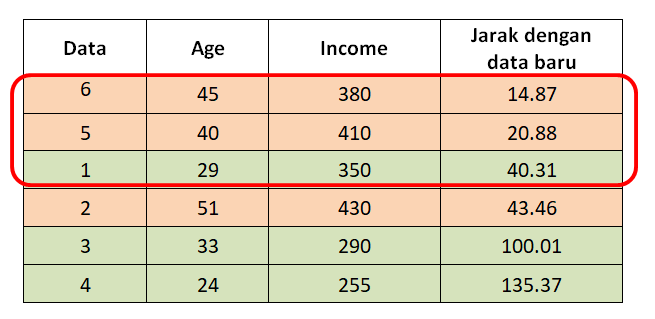
**Hitung jarak antara data baru dan masing-masing data lainnya**

Kita akan menghitung jarak menggunakan metode Euclidean distance di atas. Ada 6 data yang akan kita hitung.



**Ambil tiga data dengan jarak terdekat**

Dari perhitungan Euclidean distance di atas, jika kita rangkum dari jarak terdekat, hasilnya seperti di bawah ini.



Dari hasil di atas, dapat ditarik kesimpulan bahwa data baru tersebut masuk ke dalam kelas B karena dari tiga tetangga terdekat, ada dua yang masuk kelas B, sementara hanya satu yang masuk kelas A.