# *Outlier Detection* pada data *California Environmental Data Exchange Network* (CEDEN) menggunakan metode *Supervised Machine Learning*

**Abstrak**

Masalah utama pada pengolahan data untuk *data mining* adalah adanya *outlier*, yaitu data yang anomali. Paper ini mendeteksi keberadaan *outlier* pada data yang berjudul *Water Quality* dimana data berasal dari *California Environmental Data Exchange Network (CEDEN)* dengan menggunakan dua metode *Supervised Machine Learning* yaitu algoritma *Naive Bayes* dan *K-Nearest Neighbor (KNN).* Algoritma *Naive Bayes* adalah algoritma yang memprediksi peluang di masa depan berdasarkan pengalaman di masa sebelumnya. Algoritma *K-Nearest Neighbor (KNN)* memetakan data berdasarkan titik k yang paling dekat. Dari hasil pengujian yang telah dilakukan menggunakan kedua algoritma, algoritma *K-Nearest Neighbor* memiliki hasil yang lebih baik dibandingkan dengan algoritma *Naive Bayes*.

**Kata kunci :** *outlier, CEDEN, Supervised Machine Learning, Naive Bayes, KNN*

---

## 1. Pendahuluan

Penelitian ini dibuat berdasarkan dengan penelitian yang sudah ada sebelumnya yang menggunakan data berisi perhitungan jumlah polusi pada air. Tujuan dari dibuatnya data tersebut adalah untuk mengetahui jumlah polusi pada air yang ada pada danau dan sungai. Polusi ini dapat mempengaruhi lingkungan untuk masyarakat di United States (US) beraktifitas sehari-harinya. Sesuai sejarah pada negara-negara lainnya di seluruh dunia, polusi pada air akan terus meningkat hari demi hari, oleh karena itu data yang dipakai bersifat *realtime.*

Karena data yang digunakan berupa nominal dan memiliki *record* yang banyak, dapat dipastikan data tersebut memiliki nilai *outlier*nya masing-masing pada setiap atribut. Data yang termasuk ke dalam kategori *outlier* akan membuat pengolahan data menjadi tidak rata dalam pembagian kelasnya. Oleh karena itu, perlu dicari data mana saja yang termasuk *outlier* dan dipisahkan dengan data lainnya yang bukan termasuk *outlier.*

Penelitian ini hanya berfokus pada pendeteksian *outlier* tetapi nilai tersebut tidak di *handle*. Tujuannya agar data *outlier* terdeteksi dan kemudian dipisahkan dari data lainnya. Hal ini disebabkan oleh data *outlier* dapat membuat pengolahan data menjadi tidak rata dengan nilai lainnya. Oleh karena itu, pemisahan ini sangat penting bagi sebuah data sebelum data diolah ke *data mining.*

Berbeda dari penelitian sebelumnya yang menggunakan empat metode dan membandingkannya. Penelitian ini

hanya menggunakan dua dari banyaknya algoritma *supervised machine learning* untuk mendeteksi keberadaan *outlier*. Pada penelitian kali ini, algoritma yang akan digunakan adalah algoritma *naive bayes* dan algoritma *K-nearest neighbor* untuk mendeteksi dan memisahkan nilai *outlier* dari yang bukan *outlier.*

## 2. Landasan Teori dan Metode

### 2.1. *Outlier*

*Outlier* adalah data yang nilainya menyimpang jauh dari nilai pada data yang lainnya pada serangkaian data. Keberadaan *outlier* ini akan membuat analisis pada sebuah data menjadi bias atau tidak merata. Nilai *outlier* ini juga sering disebut sebagai nilai ekstrim, baik itu ekstrim besar maupun ekstrem kecil.

Data *outlier* tersebut bisa didapatkan jika sudah menentukan nilai untuk batas atas dan batas bawah pada sebuah rangkaian data. Jika nilai data tersebut bernilai lebih besar atau berada di atas dari batas atas maka nilai tersebut termasuk nilai ekstrim besar. Sedangkan, jika nilai data tersebut bernilai lebih kecil atau berada di bawah dari batas bawah maka nilai tersebut termasuk nilai ekstrim kecil.

Pada penelitian kali ini, yang akan dilakukan hanyalah mengelompokkan nilai ekstrim-ekstrim tersebut ke dalam kelas yang berbeda dengan nilai rata-rata lainnya. Tujuan pengelompokkan tersebut adalah untuk memberi label bahwa data yang dimaksudkan mempunyai label 0 atau data tersebut memiliki nilai yang termasuk *outlier.*

Pemisahan nilai *outlier* pada serangkaian data dianggap penting pada setiap tahap *preprocessing* di semua metode yang akan dipakai. Pengidentifikasian *outlier* juga penting untuk beberapa variabel di serangkaian data kategorikal atau ketika menggunakan metode linear maupun nonlinear.

### 2.2. Supervised Machine Learning

*Supervised Machine Learning* adalah sebuah pembelajaran yang sudah terdapat data untuk dilatih, yang kemudian terdapat variabel yang menjadi target dari data yang diuji. Tujuan dari pembelajaran ini adalah mengelompokkan data sesuai dengan kelasnya masing-masing. Algoritma-algoritma yang ada pada metode ini ada banyak, tetapi yang akan dipakai pada penelitian ini hanya dua yaitu *Naive Bayes Classifier* dan *K-Nearest Neighbor Classifier.*

#### 2.2.1. Naive Bayes

*Naive Bayes* adalah salah satu metode pembelajaran mesin yang paling efisien dan efektif. Metode *naive bayes* terbukti bekerja dengan baik dalam berbagai jenis data baik data yang *dependent* maupun yang *independent,* kedua keadaan data tersebut tidak mempengaruhi pengklasifikasian pada *naive bayes.*

*Naive Bayes* merupakan pengklasifikasian dengan metode probabilitas dan statistika, yaitu memprediksi peluang di masa depan berdasarkan pengalaman di masa sebelumnya sehingga dikenal dengan Teorema Bayes. Teorema tersebut dikombinasikan dengan Naive dimana diasumsikan kondisi antar atribut saling bebas. Klasifikasi *Naive Bayes* diasumsikan bahwa ada atau tidak ciri

tertentu dari sebuah kelas tidak ada hubungannya dengan ciri dari kelas lainnya.

Rumus untuk algoritma *Naive Bayes* adalah sebagai berikut.

$$P(C|F_1...F_n) = \frac{P(C)P(F_1...F_n|C)}{P(F_1...F_n)}$$

Variabel C merepresentasikan kelas, sementara variabel $F_1...F_n$ merepresentasikan karakteristik petunjuk yang dibutuhkan untuk melakukan klasifikasi. Peluang masuknya sampel karakteristik tertentu dalam kelas C (*Posterior*) adalah peluang munculnya kelas C (sebelum masuknya sampel tersebut, seringkali disebut *prior*), dikali dengan peluang kemunculan karakteristik-karakteristik sampel pada kelas C (disebut juga *likelihood*), dibagi dengan peluang kemunculan karakteristik-karakteristik sampel secara global (disebut juga *evidence*).

### 2.2.2. K-Nearest Neighbor

Algoritma k-Nearest Neighbor adalah algoritma supervised learning dimana hasil dari *instance* yang baru diklasifikasikan berdasarkan mayoritas dari kategori k-tetangga terdekat.

Tujuan dari algoritma ini adalah untuk mengklasifikasikan obyek baru berdasarkan atribut dan sample-sample dari training data.

Ada banyak cara untuk mengukur jarak kedekatan antara data baru dengan data lama (data *training*), diantaranya *euclidean distance* dan *manhattan distance* (*city block distance*), yang paling sering digunakan adalah *euclidean distance*, yaitu:

$$\sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + ... + (a_n - b_n)^2}$$

Dimana a = $a_1, a_2, \ldots, a_n$, dan b = $b_1, b_2, \ldots, b_n$ mewakili n nilai atribut dari dua *record*.

## 3. Data

Dataset yang digunakan telah di*download* pada 20 April 2020 dan data totalnya berisi 60.000 *records.* Data yang berjudul "Measurements of various surface water pollutants in the California Environmental Data Exchange Network (CEDEN)" berasal dari kumpulan data di *California Environmental Data Exchange Network* (CEDEN). Karena pada penelitian ini hanya mendeteksi nilai *oulier*-nya saja, maka tidak semua atribut akan diolah menggunakan algoritma yang telah ditentukan. Atribut yang digunakan pada penelitian ini hanya atribut 'Result' yang nanti akan dicari nilai *outlier*nya dari nilai-nilai atribut tersebut. Atribut kedua yang digunakan adalah atribut 'Unit' yang merupakan satuan dari nilai yang ada di result.

Dapat dilihat dari nilai yang ada di atribut 'Unit', data ini memiliki dua satuan nilai yang berbeda. Oleh karena itu, pendeteksian outlier pun juga dilakukan dua kali. Pendeteksian dilakukan masing-masing sesuai dengan satuannya. Pengklasifikasian - nya pun juga nanti akan berbeda antar keduanya.

Data yang sudah dipisahkan dari data aslinya nantinya akan dihilangkan terlebih dahulu *missing values* nya agar lebih meringkas pengerjaannya. Setelah *missing values* dihapus data yang tadinya berisi 60.000 *records* menjadi 56.035 *records* yang kemudian jika variabel bertipe string didrop data menjadi berisi 55.927 *records.*

## 4. Hasil

Paper ini dilakukan dengan beberapa tahap, yaitu menghilangkan *missing value* pada data, *split data* berdasarkan satuan unit dikarenakan nilai setiap satuan akan berbeda range-nya. Lalu cek *outlier* setiap satuan unit berdasarkan nilai interkuartil, setiap data yang termasuk outlier akan ditambahkan nilai '1' pada atribut 'Class', dan data yang tidak termasuk *outlier* akan ditambahkan nilai '0'. Kemudian split data menjadi data train sebesar 80%, dan data test sebesar 20%. Setelah itu, tahap pendeteksian menggunakan algoritma Naive Bayes dan k-Nearest Neighbor, dan evaluasi menggunakan confusion matrix, F1-Score, Accuracy, Precision, dan Recall.

Hasil yang didapatkan dari algoritma Naive Bayes adalah nilai F1-Score sebesar 57.865%.

| Predicted | 0 | 1 |
|---|---|---|
| Actual | | |
| **0** | 9748 | 315 |
| **1** | 947 | 176 |

```
NAIVE BAYES

F1-SCORE  0.5786479299481497
ACCURACY  0.8871804040765242
PRECISION  0.6349530444685022
RECALL 0.5627101354078389
```

Hasil yang didapatkan dari algoritma k-Nearest Neighbor adalah nilai F1-Score sebesar 96.452%.

```
K NEAREST NEIGHBOR

F1-SCORE  0.9645243400659658
ACCURACY  0.9876631503665296
PRECISION  0.9835996621746069
RECALL 0.9472595135065827
```

| Predicted | 0 | 1 |
|---|---|---|
| Actual | | |
| **0** | 10041 | 22 |
| **1** | 116 | 1007 |

## 5. Kesimpulan

Kesimpulan dari hasil perbandingan dua algoritma pada pendeteksian outlier adalah algoritma k-Nearest Neighbor lebih bagus berdasarkan nilai F1-Score. Dikarenakan pada algoritma k-Nearest Neighbor pengklasifikasian menggunakan nilai K dan dicari titik terdekat dari titik-titik K tersebut. Jika nilai tersebut termasuk ke dalam *outlier* maka semua nilai yang dekat dengan titik K tersebut juga merupakan *outlier.* Oleh karena itu, mendeteksi *outlier* menggunakan algoritma k-Nearest Neighbor memiliki hasil yang lebih baik.

## DAFTAR PUSTAKA


Hadi, A. S., Imon, A. R., & Werner, M. (2009). Detection of outlier. *Wiley Interdisciplinary Reviews: Computational Statistics*, *1*(1), 57-70.

Chowdhury, K. P. (2019). Supervised Machine Learning and Heuristic Algorithms for Outlier Detection in Irregular Spatiotemporal Datasets. *J. Environ. Inform*, *33*.

Konsultanstatistik.com. (2010) Data outlier. Diakses pada 26 Maret 2020, dari https://www.konsultanstatistik.com /2010/05/data-outlier.html

Medium.com. (1 November, 2019) Perbedaan Antara Supervised dan

Unsupervised Learning. Diakses pada 26 Maret 2020, dari https://medium .com/machine-learning-kelompok-2/perbedaan-antara-supervised-dan-unsupervised-learning-fcb18f90e89f

Zhang, H. (2004). The optimality of naive Bayes. *AA*, *1*(2), 3.

Medium.com. (17 Agustus, 2018) Cara Kerja Algoritma k-Nearest Neighbor (k-NN). Diakses pada 26 Maret 2020, dari https://medium.com/bee-solution-partners/cara-kerja-algoritma-k-nearest-neighbor-k-nn-389297de543e

Bustami, B. (2013). Penerapan Algoritma Naïve Bayes Untuk Mengklasifikasi Data Nasabah Asuransi. TECHSI-Jurnal Teknik Informatika, 5(2).

Leidiana, H. (2013). Penerapan algoritma k-nearest neighbor untuk penentuan resiko kredit kepemilikan kendaraan bemotor. PIKSEL: Penelitian Ilmu Komputer Sistem Embedded and Logic, 1(1), 65-76.