PENGOLAHAN DATA MENGGUNAKAN TOOLS DATA MINING

Tujuan Pembelajaran :

Mahasiswa mampu menggunakan tools data mining untuk mengolah data

.

12.1 Pengertian WEKA

Weka merupakan aplikasi yang dibuat dari bahasa pemrograman java yang dapat digunakan untuk membantu pekerjaan data mining (penggalian data). Weka berisi beragam jenis algoritma yang dapat digunakan untuk memproses dataset secara langsung atau bisa juga dipanggil melalui kode bahasa java. Weka berisi peralatan seperti pre-processing, classification, regression, clustering, association rules dan visualization. Weka dapat juga digunakan untuk memproses big data dan dikembangkan guna memenuhi skema machine learning (ML). Weka bersifat open source dibawah lisensi GNU General Public License.

Gambar 1. Simbol Aplikasi Weka

Weka tersedia bagi pengguna Linux, OS X, dan Windows. Untuk mengunduh GNU Weka silahkan mengunjungi halaman Download Weka. Pada praktikum kali ini menggunakan fitur klasifikasi untuk mengolah datanya dengan metode Klasifikasi k-Nearest Neighbor.

12.2 Metode Klasifikasi k-Nearest Neighbor

Metode klasifikasi k-Nearest Neighbor atau biasa disingkat kNN. kNN memiliki perbedaan cara kerja dibandingkan dengan metode-metode klasifikasi lainnya. metode klasifikasi secara umum akan membentuk model, yang merupakan fungsi pemetaan input output dari data latih dan menggunakan model yang telah terbentuk untuk memperkirakan output dari suatu input yang baru. Pada metode decision tree, model berbentuk tree, sedangkan pada Naive Bayes, model yang digunakan berupa fungsi probabilitas. Metode kNN tidak memiliki prosedur pelatihan (training) sehingga tidak terdapat pembentukan model. Metode kNN bekerja bedasarkan asumsi bahwa suatu

data akan memiliki kelas atau kategori yang sama dengan data yang berada disekitarnya. Konsep ini disebut konsep ketetanggaan.

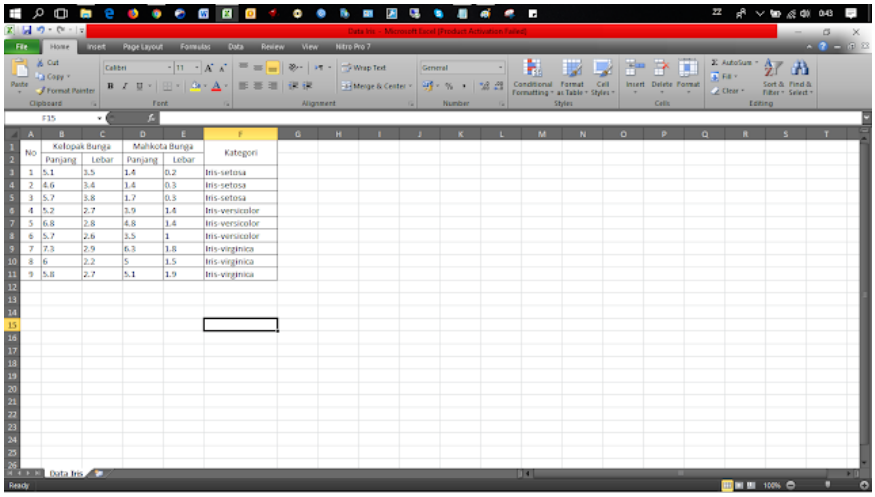
12.3 Algoritma kNN

Algoritma kNN bekerja dengan cara menemukan k tetangga terdekat dari suatu data yang belum diketahui kelasnya (data uji). Penentuan k tetangga terdekat dilakukan dengan menghitung jarak dari data uji ke setiap data latih yang ada. Selanjutnya dipilih sejumlah k data yang memiliki jarak terdekat. Kelas dari data uji ditentukan dari mayoritas kelas dari k data latih terdekat.

12.4 Implementasi kNN pada Weka

Metode kNN diimplementasikan pada Weka dengan nama IBk (instance-based learning with parameter k). Contoh berikut akan menampilkan penggunaan IBk untuk mengelompokan data Iris berdasarkan panjang dan lebar dari kelopak dan mahkota bunga.

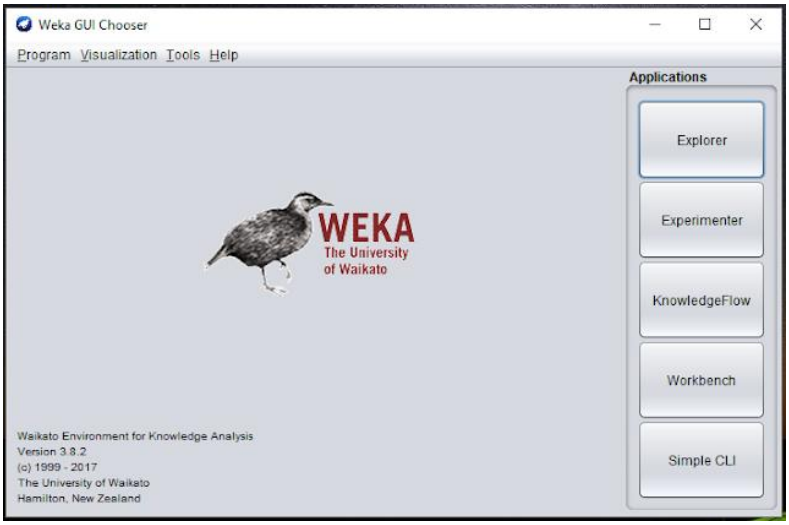
Langkah pertama yang dilakukan adalah Data Iris.arff yang sebelumnya dibuat mengguakan microsoft excel dengan format CSV (Comma delimited).



No	Kelopak Bunga		Mahkota Bunga		Kategori
	Panjang	Lebar	Panjang	Lebar	
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.6	3.4	1.4	0.3	Iris-setosa
3	5.7	3.8	1.7	0.3	Iris-setosa
4	5.2	2.7	1.9	1.4	Iris-versicolor
5	6.8	2.8	4.8	1.4	Iris-versicolor
6	5.7	2.6	3.5	1	Iris-versicolor
7	7.3	2.9	6.3	1.8	Iris-virginica
8	6	2.2	5	1.5	Iris-virginica
9	5.8	2.7	5.1	1.9	Iris-virginica

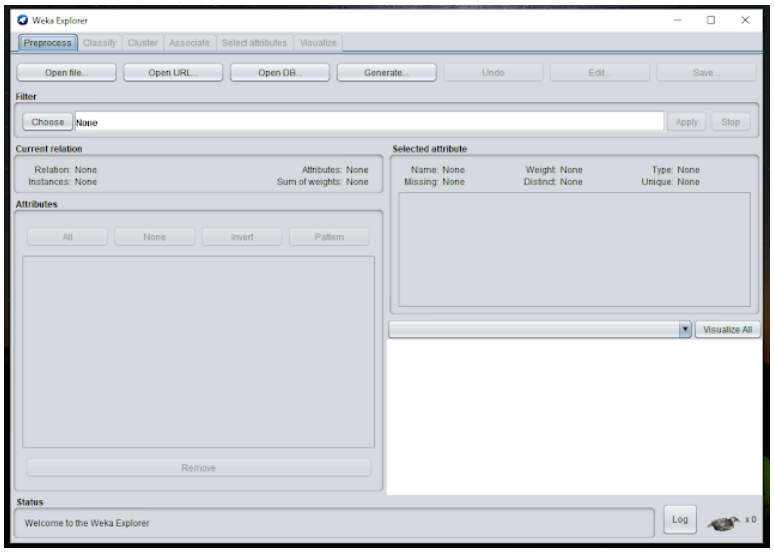
Gambar 2. data iris.csv

Kemudian buka aplikasi Weka, versi Weka yang saya pakai versi 3.8.2



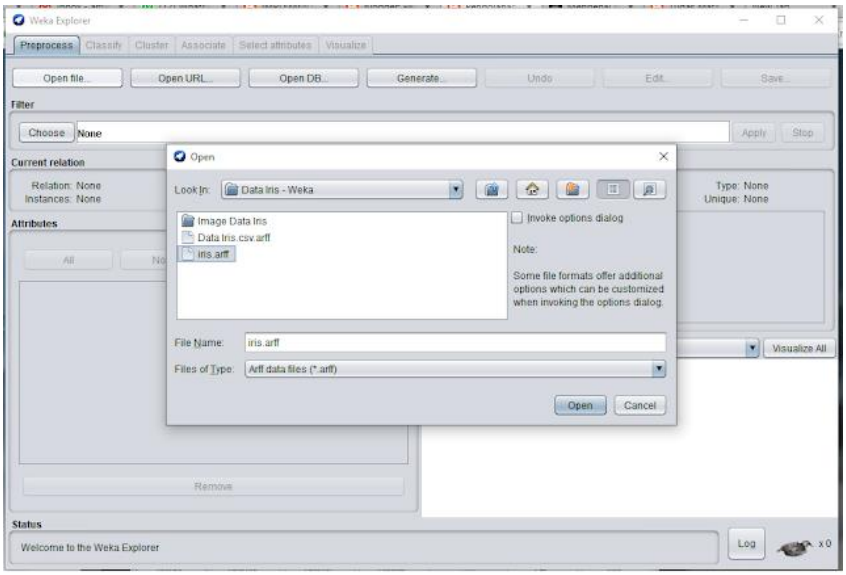
Gambar 3. tampilan Weka 3.8.2

Setelah itu pilih menu Explorer, kemudian akan muncul tampilan seperti pada gambar berikut :



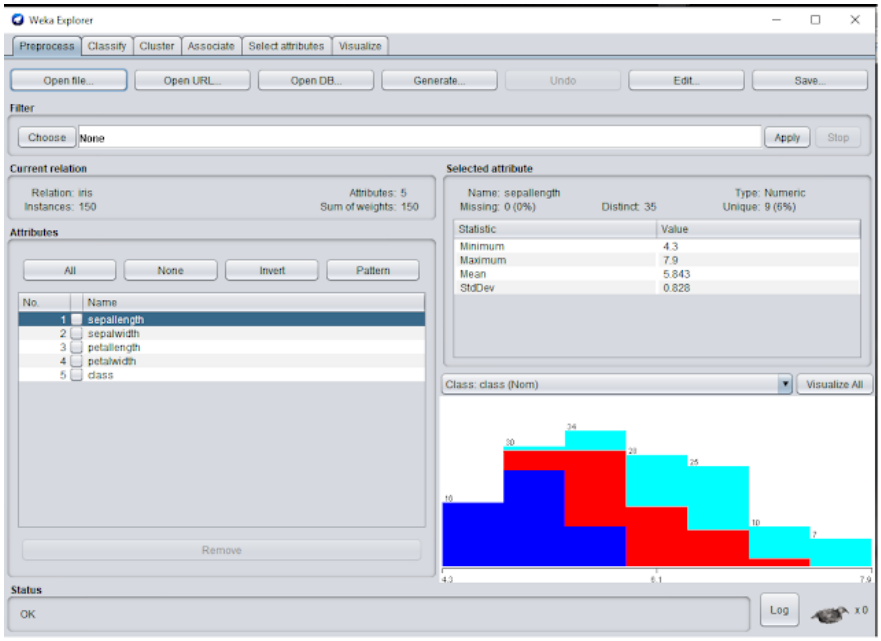
Gambar 4. tampilan menu explorer

Kemudian pilih menu Open file



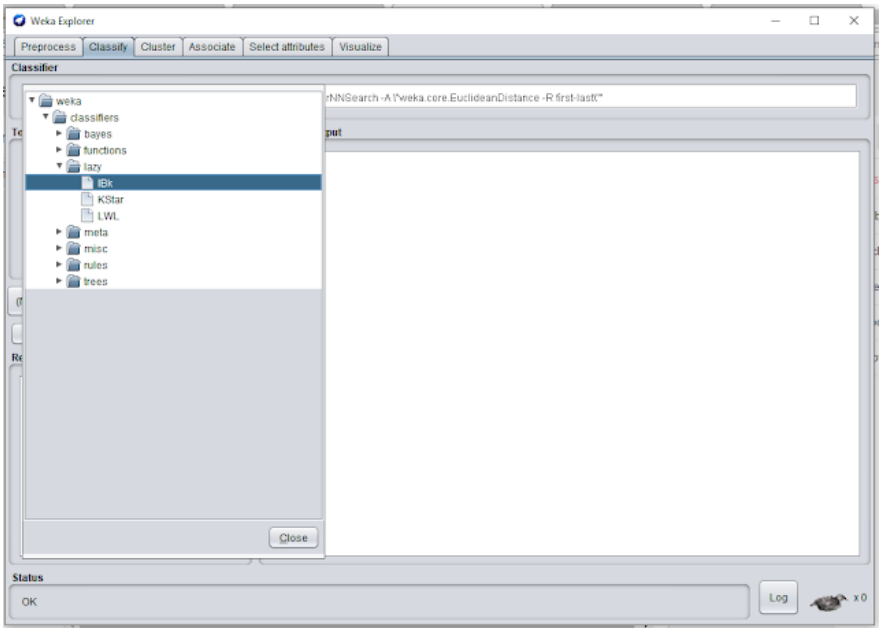
Gambar 5. pengambilan data iris.arff

Berikut tampilan dari data yang sudah di open file



Gambar 6. tampilan open file data iris.arff

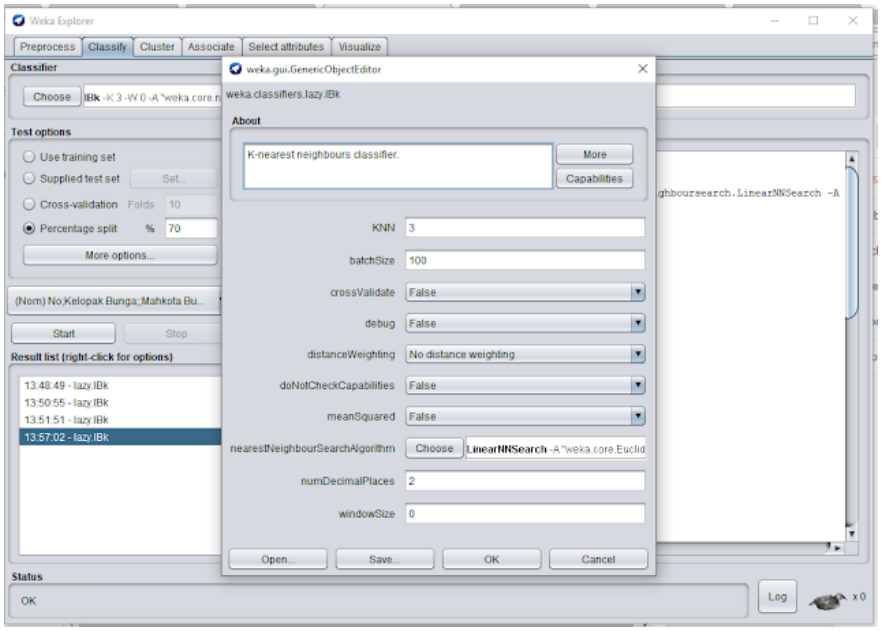
Proses klasifikasi dapat dimulai dengan memilih tab Classify. Tekan tombol Choose untuk memilih metode klasifikasi yang akan digunakan. Algoritma IBk dapat diakses melalui folder Weka > Classifier > lazy > IBk.



Gambar 7. tampilan pemilihan algoritma IBk

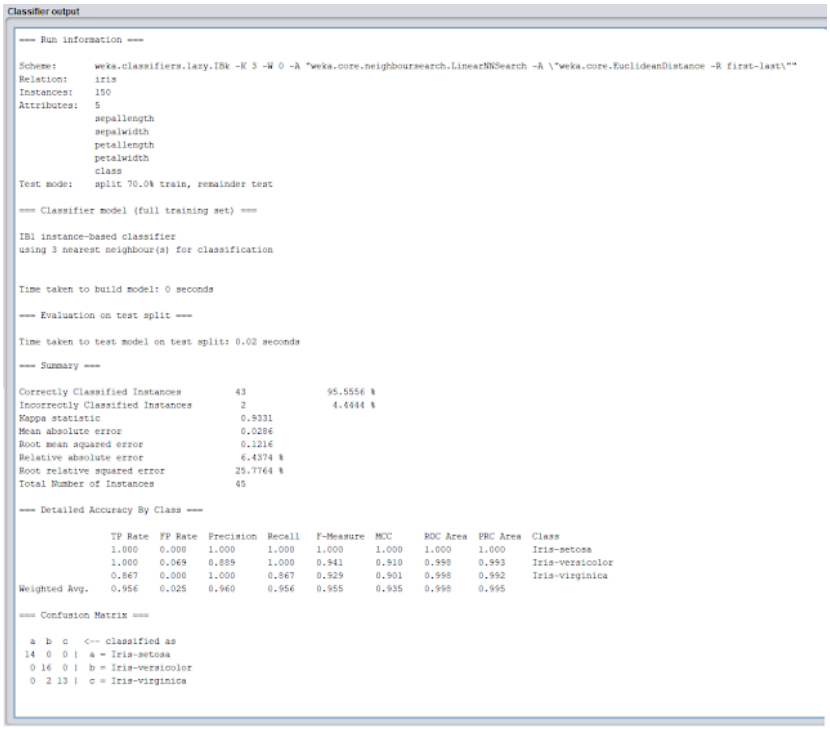
Metode IBk memiliki beberapa parameter yang dapat diatur sesuai dengan kebutuhan. Pengaturan parameter dapat dilakukan dengan klik kanan pada field IBk disamping tombol Choose dan pilih opsi Show properties... Beberapa Opsi yang dapat diatur :

- **kNN** : Jumlah tetangga terdekat (nilai k). Pada percobaan ini pilihlah nilai kNN sebesar 3.
- **distanceWeighting** : Metode pembobotan untuk mengurangi efek dominasi kelas yang memiliki banyak data.
- **CrossValidate** : Jika bernilai True, Weka akan mencari nilai k terbaik, yang nilai nya berada diantara 1 dan k pada parameter kNN.



Gambar 8. parameter pada algoritma IBk

Pengujian dilakukan dengan memecah data Iris menjadi data latih dan data uji dengan proporsi 70% : 30%. Pilihlah opsi Percentage split pada Test options dan isilah dengan angka 70. Pilih fitur (Nom) class sebagai kategori pada data. Klik tombol Start untuk memulai proses klasifikasi. Berikut adalah cuplikan hasil klasifikasi :



Classifier output

```
=== Run information ===

Scheme:       weka.classifiers.lazy.IBk -K 3 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""
Relation:     iris
Instances:    150
Attributes:   5
              sepallength
              sepalwidth
              petallength
              petalwidth
              class
Test mode:    split 70.0% train, remainder test

=== Classifier model (full training set) ===

IB1 instance-based classifier
using 3 nearest neighbour(s) for classification


Time taken to build model: 0 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 0.02 seconds

=== Summary ===

Correctly Classified Instances          43               95.5556 %
Incorrectly Classified Instances         2                4.4444 %
Kappa statistic                          0.9331
Mean absolute error                      0.0286
Root mean squared error                  0.1216
Relative absolute error                  6.4374 %
Root relative squared error             25.7764 %
Total Number of Instances               45

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     Iris-setosa
                 1.000    0.069    0.889      1.000    0.941      0.910    0.998     0.993     Iris-versicolor
                 0.867    0.000    1.000      0.867    0.929      0.901    0.998     0.992     Iris-virginica
Weighted Avg.    0.956    0.025    0.960      0.956    0.955      0.935    0.998     0.995

=== Confusion Matrix ===

  a  b  c   <-- classified as
 14  0  0 |  a = Iris-setosa
  0 16  0 |  b = Iris-versicolor
  0  2 13 |  c = Iris-virginica
```

Gambar 9. Hasil validasi klasifikasi menggunakan IBk

Proses klasifikasi dilakukan dilakukan dengan menggunakan lima fitur dimana satu diantaranya merupakan penanda kelas/kategori. Pengujian dilakukan dengan membagi 150 data menjadi 105 data latih (70%) dan 45 data uji (30%). Dari 45 data uji, terdapat 2 data uji yang diklasifikasikan secara salah dan 43 data uji lainnya diklasifikasikan secara benar. Oleh karena itu, tingkat akurasi yang diperoleh adalah 95,56%.

Berdasarkan informasi yang ditampilkan pada Confusion matrix, kesalahan disebabkan oleh 2 data dengan kelas Iris-virginica yang diklasifikasikan sebagai Iris-versicolor oleh IBk. Selain akurasi, terdapat pula beberapa ukuran validasi lainnya seperti Kappa statistics, RMSE, RAE, dan RRSE.

12.5 Latihan

1. Silahkan dicoba ulang dari setiap langkah praktikum tersebut
2. Carilah salah satu dataset sederhana di kaggle.com dan jelaskan dataset tersebut tentang apa ?
3. Mengolah dataset tsb dengan salah satu tools data mining , silahkan pilih salah satu (WEKA, Rapid Miner, Orange dll)
4. Mengolah dataset tsb dengan membandingkan dua algoritma (misal decision tree dan KNN) lalu jelaskan hasil visualisasinya
5. Buat Laporannya