Nama : Ferza Reyaldi
NIM : 09021281924060
Kelas : TI REGULER
Mata Kuliah : Data Mining

Data Mining with Weka

1.1 Introduction

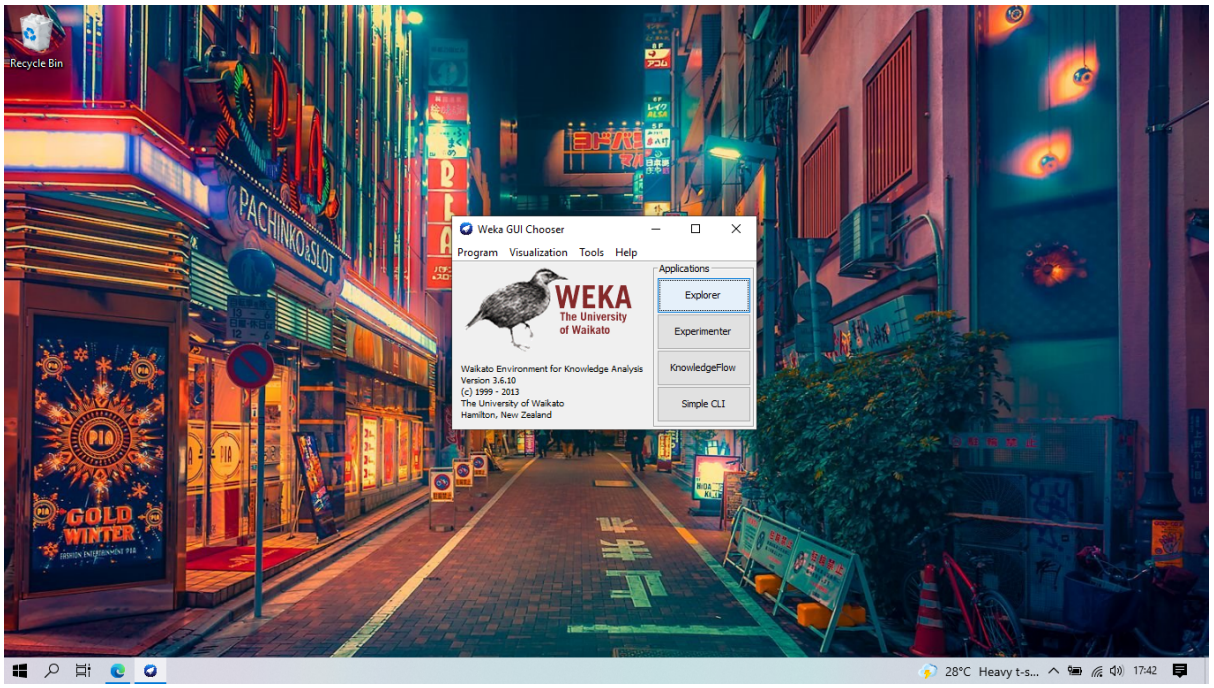
Data Mining berperan mengubah data mentah tersebut menjadi informasi yang lebih berguna dan bisa dipakai untuk memprediksi apa yang akan terjadi berikutnya dan bisa dipakai di dunia nyata. Dan sering dipakai di dunia nyata misal di supermarket untuk membuat harga khusus untuk tiap pembeli.

Jika dibandingkan antara Machine Learning dan Data Mining, Data Mining merupakan aplikasi yang digunakan, sedangkan Machine Learning adalah algoritma yang digunakan untuk melakukan Data Mining.

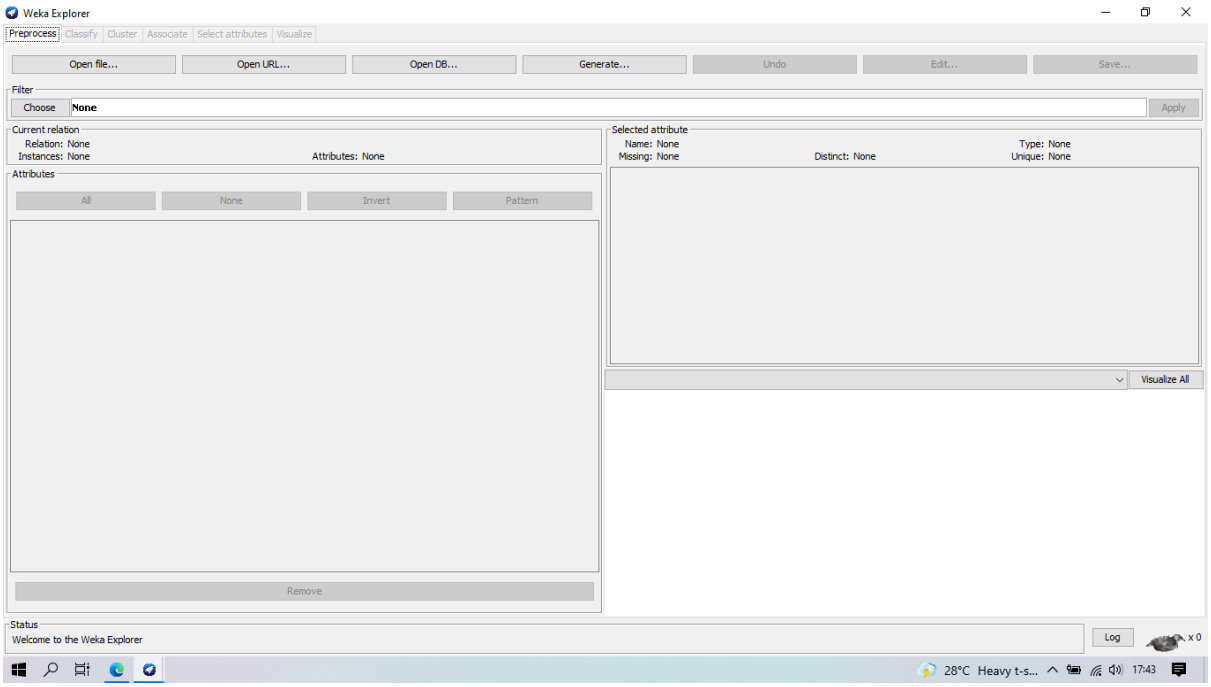
Dalam melakukan data mining, terdapat software yang khusus digunakan untuk mempermudah proses data mining, yaitu WEKA. WEKA (Waikato Environment for Knowledge Analysis) adalah data mining toolkit atau workbench yang bisa dijalankan di Windows, Mac, atau bahkan Linux yang bisa didownload gratis karna bersifat open source software.

1.2 Exploring the Explorer

Buka Aplikasi Weka, kemudian muncul tampilan awal aplikasi Weka.

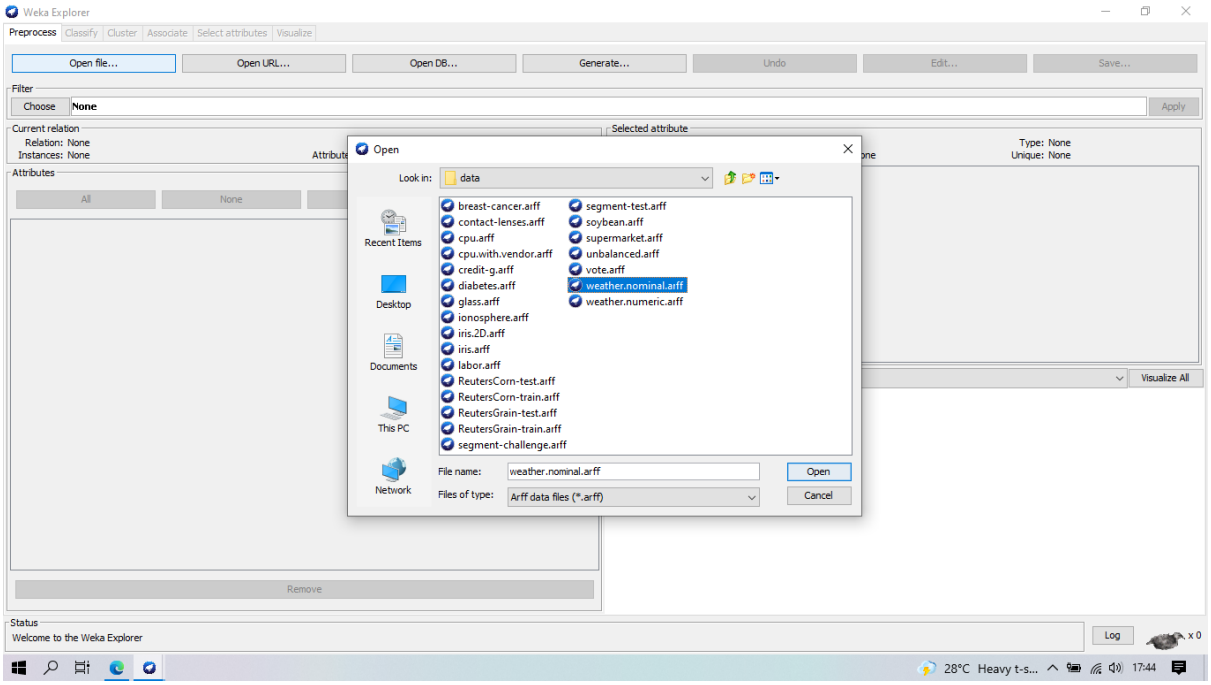


Klik Explorer, sehingga muncul tampilan Explorer.

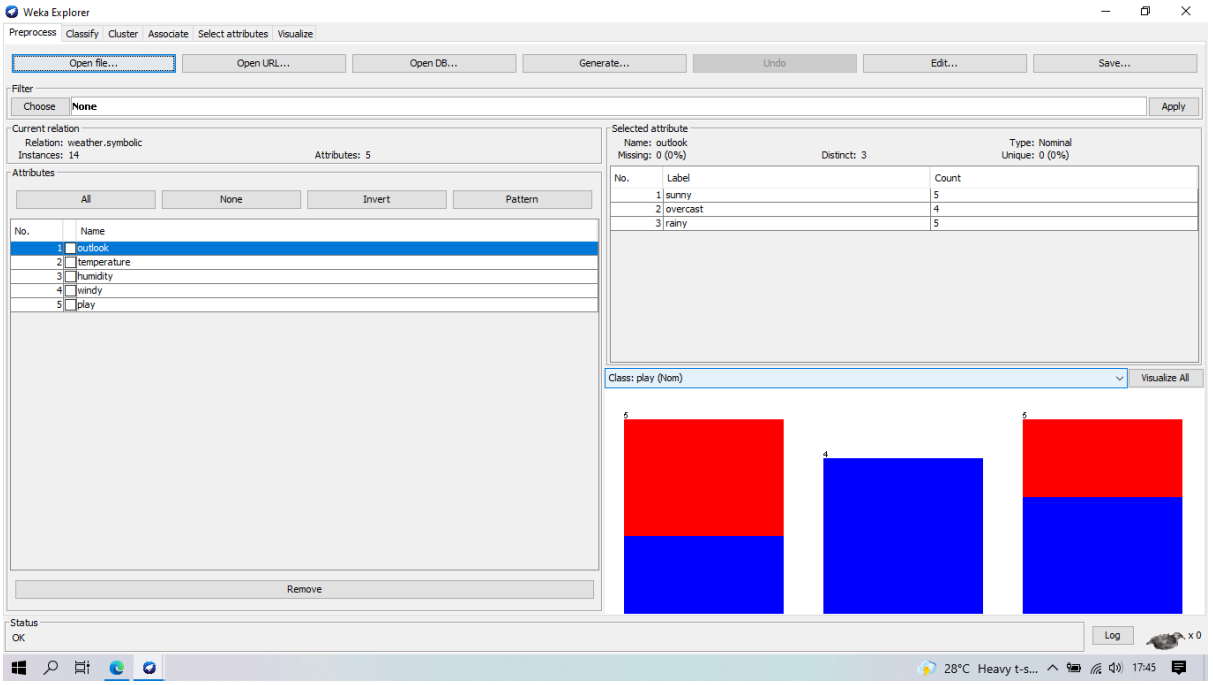


1.3 Exploring Datasets

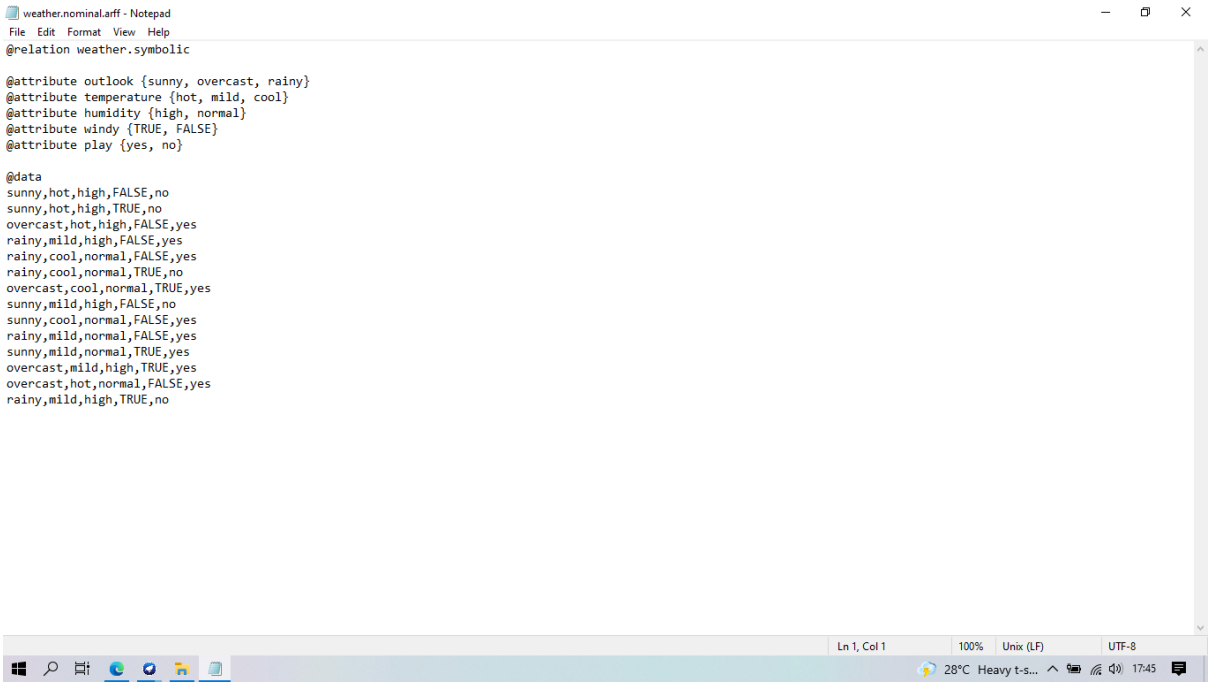
Klik Open File untuk menambah dataset. Sebagai contoh, pilih weather-nominal.arff.



Klik Open, kemudian muncul tampilan berikut berupa atribut, label pada setiap atribut, dan visualisasinya.

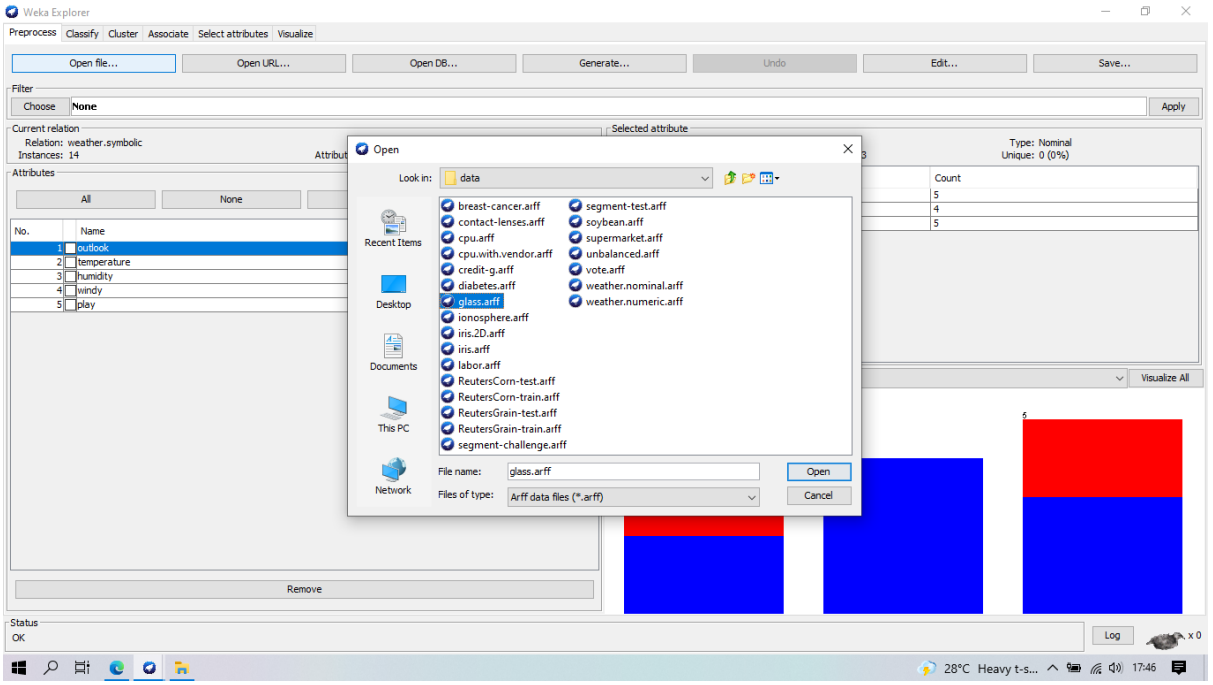


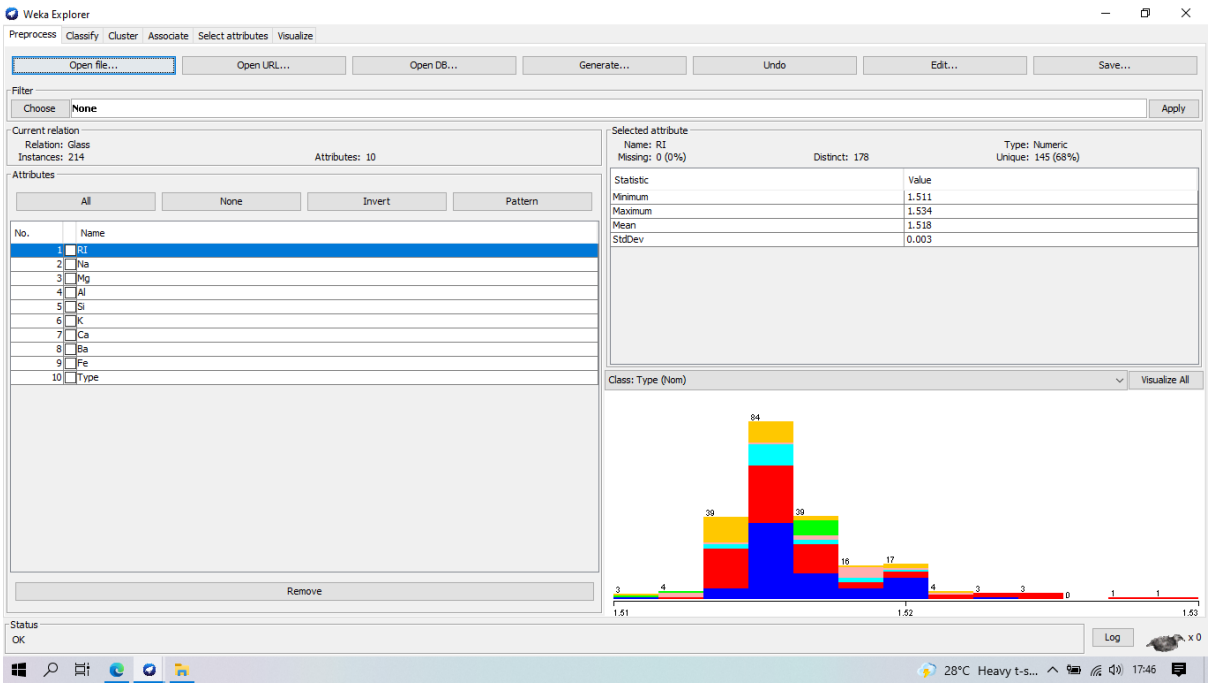
File data set berformat arff, buka file tersebut untuk melihat struktur isi dari file dataset tersebut.



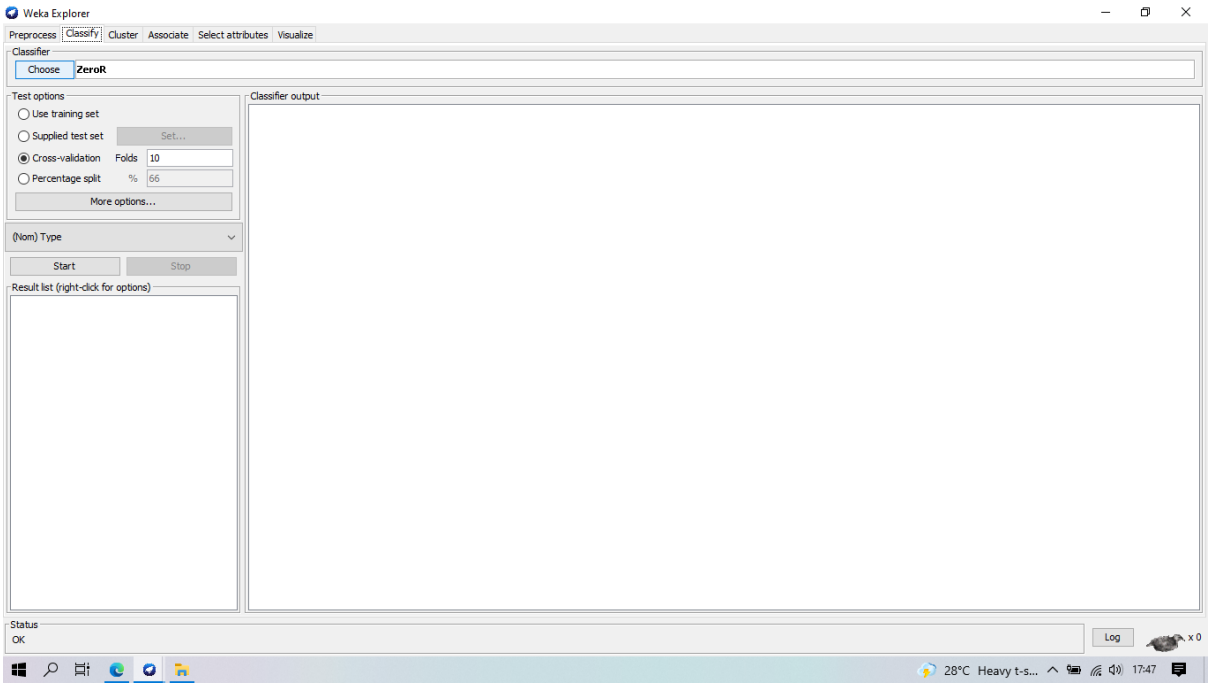
1.4 Building a Classifier

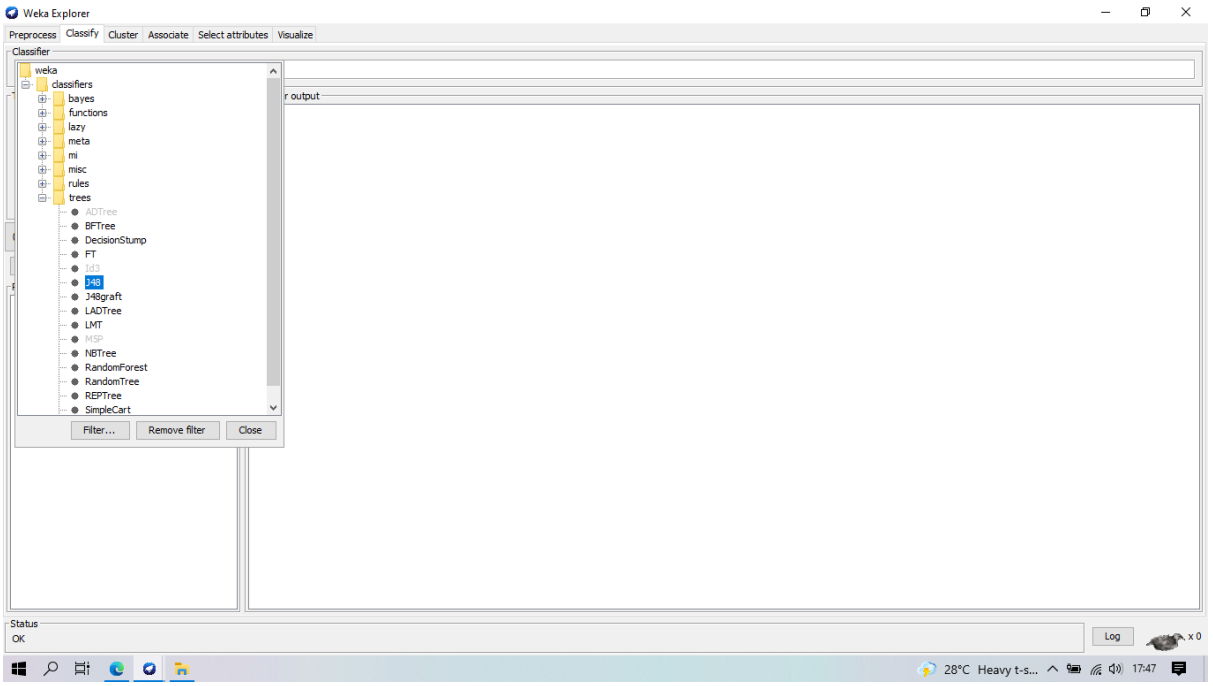
Buka file dataset glass.arff



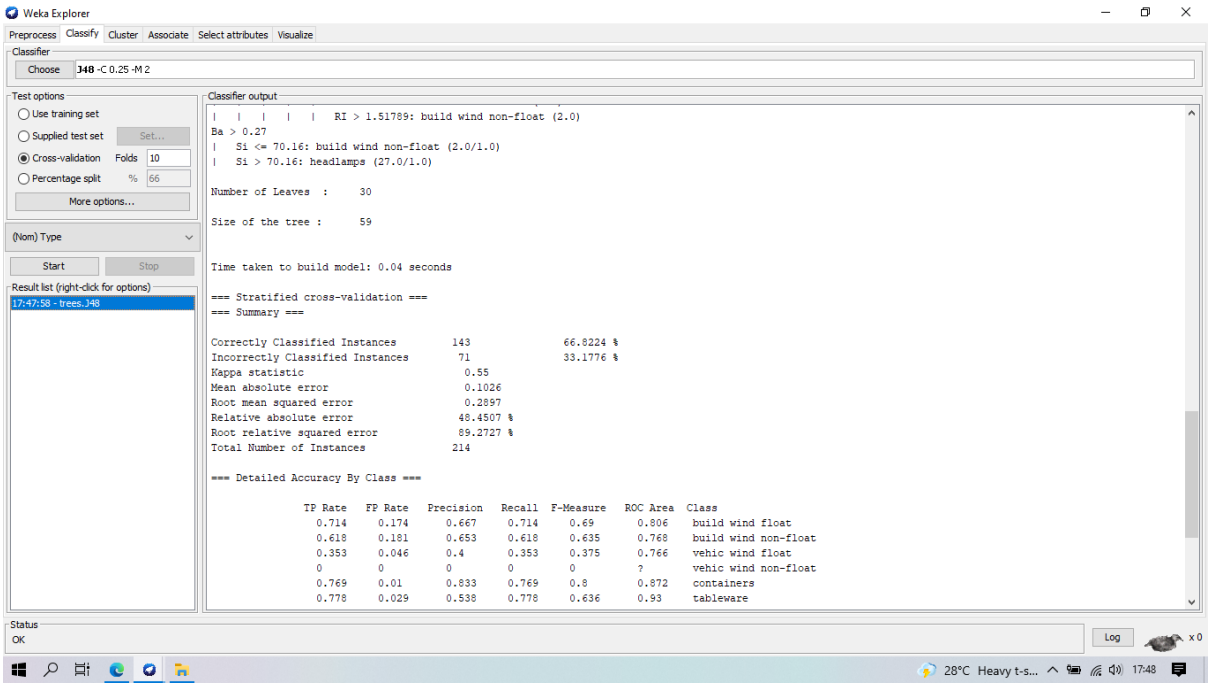


Klik tab Classify, kemudian pilih tipe classifier, yaitu Tree J48.

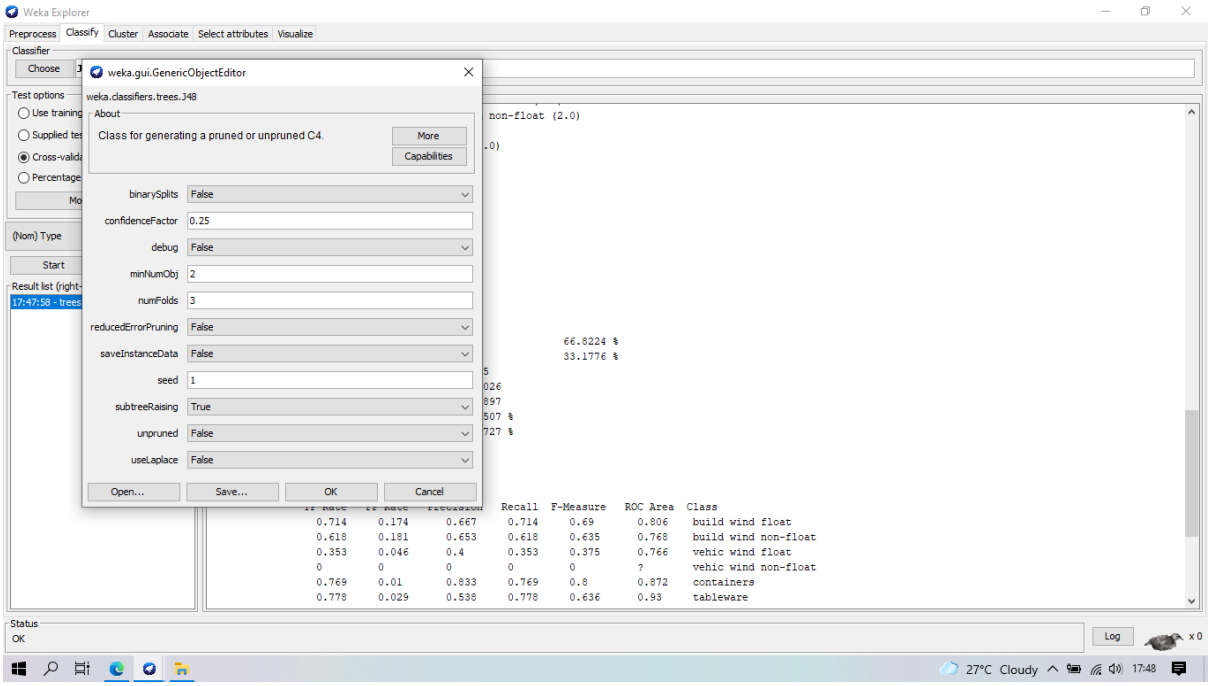




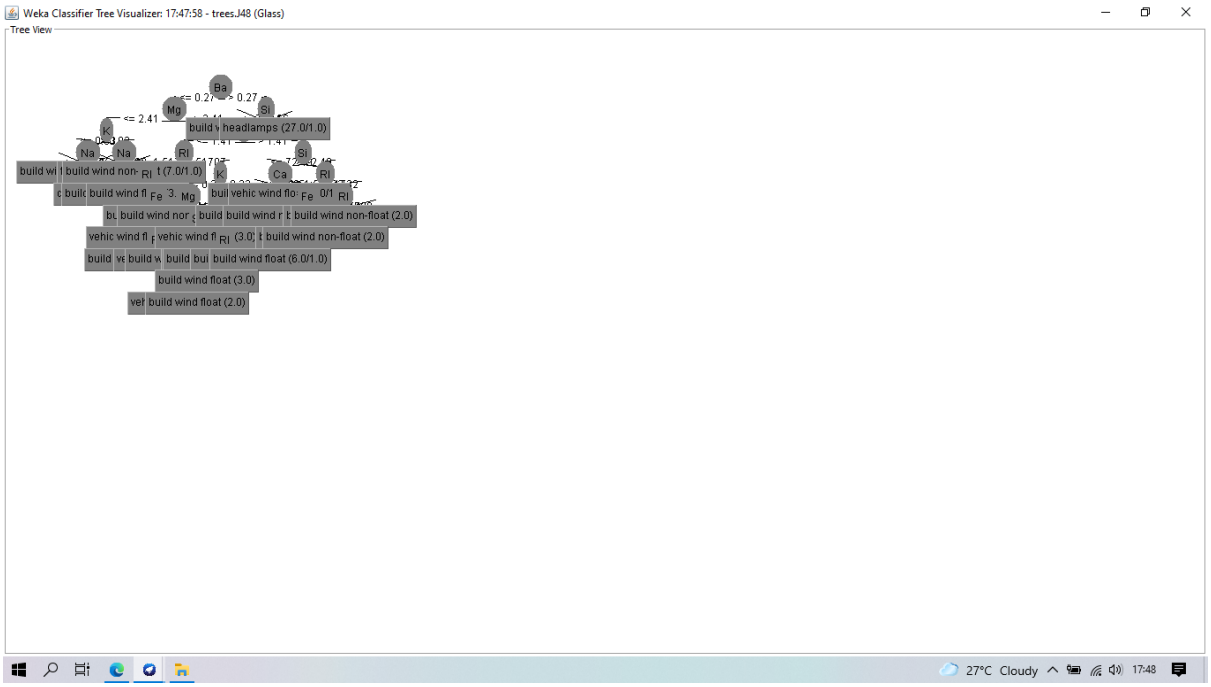
Buka hasil dari klasifikasi, dengan cara klik Start.



Bisa juga dilakukan konfigurasi pada parameter untuk J48.

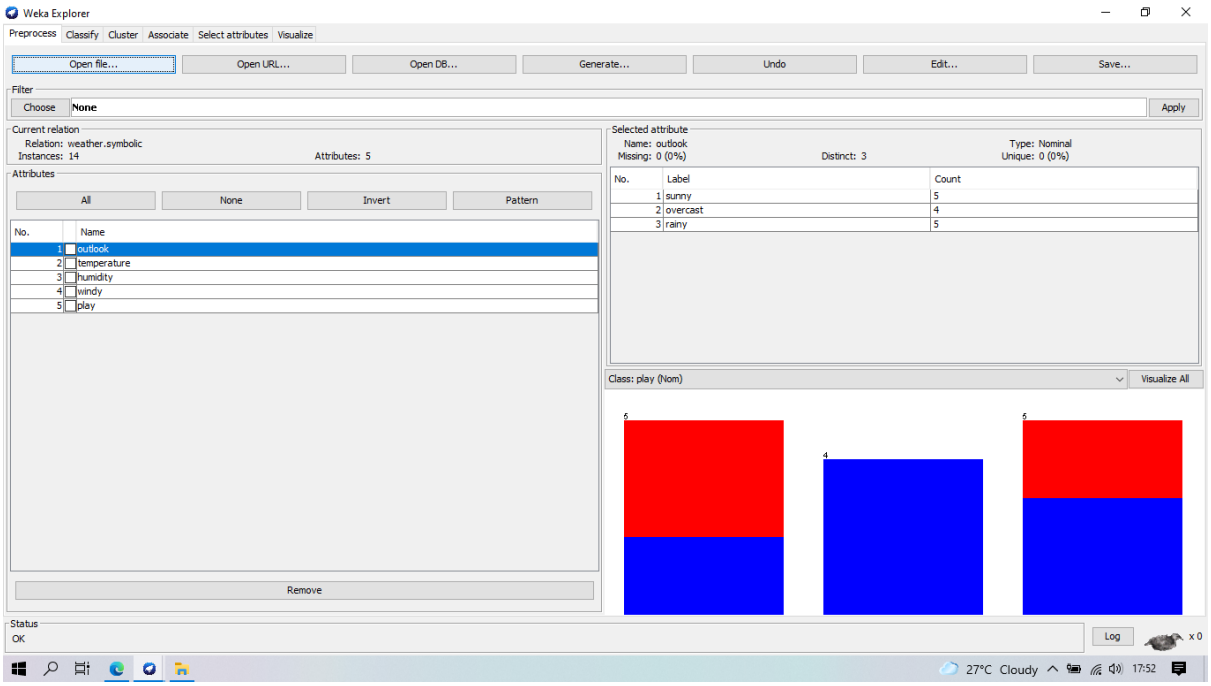


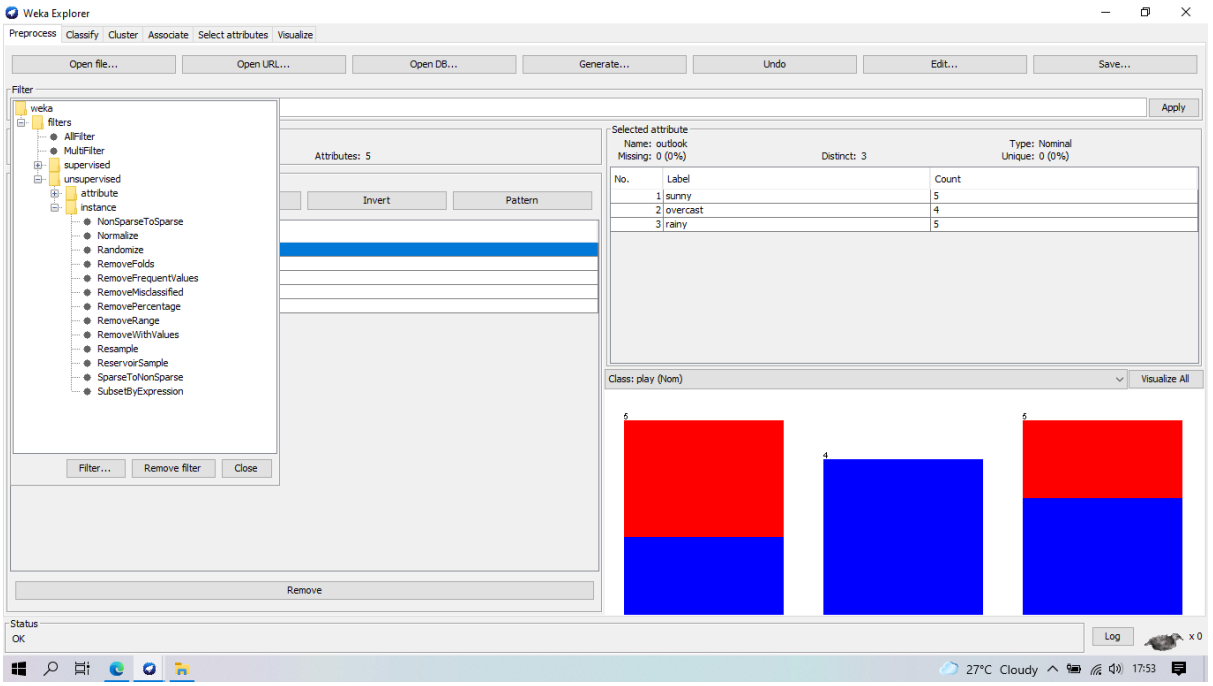
Visualisasi tree j48 untuk dataset glass.arff.



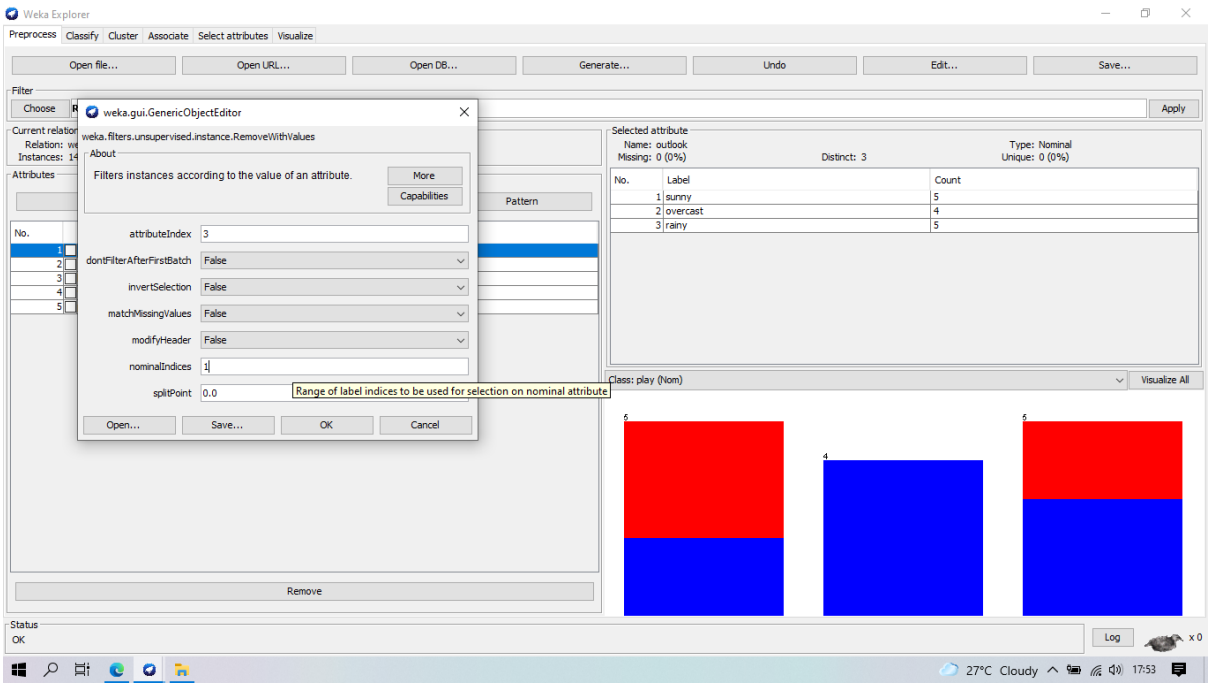
1.5 Using a Filter

Pilih dataset weather-nominal.arff, kemudian cari filter yang ingin digunakan. Untuk contoh, gunakan filter instance di folder unsupervised, yaitu removeWithValues.

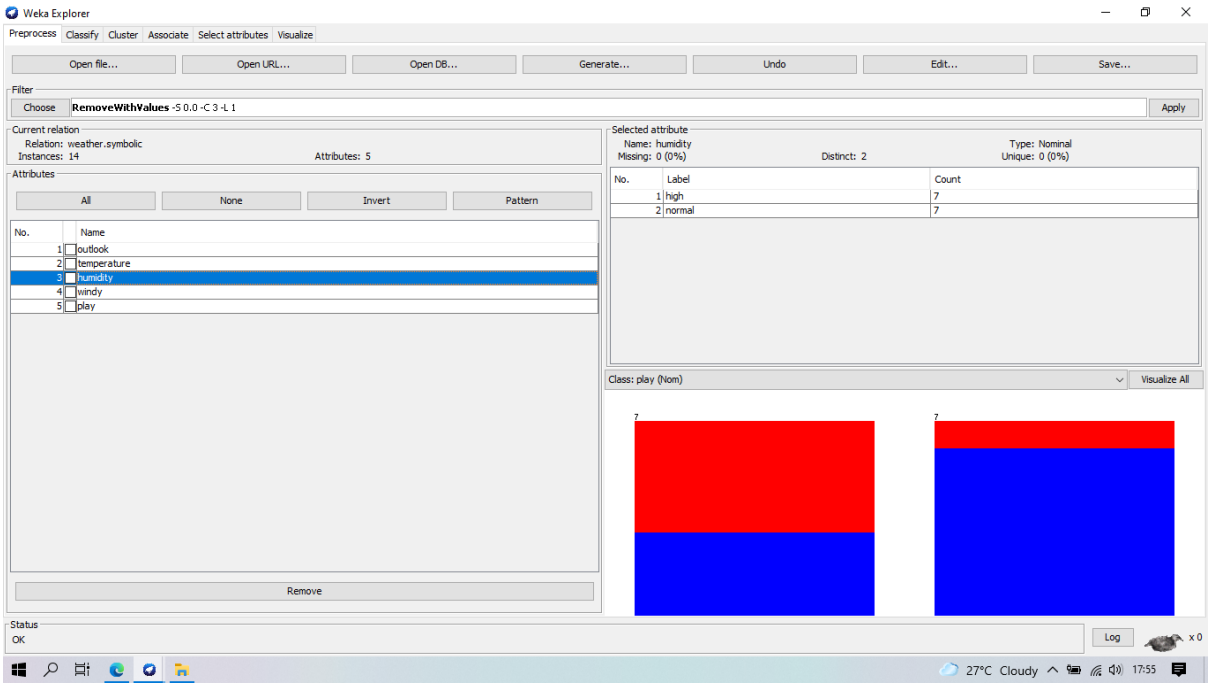




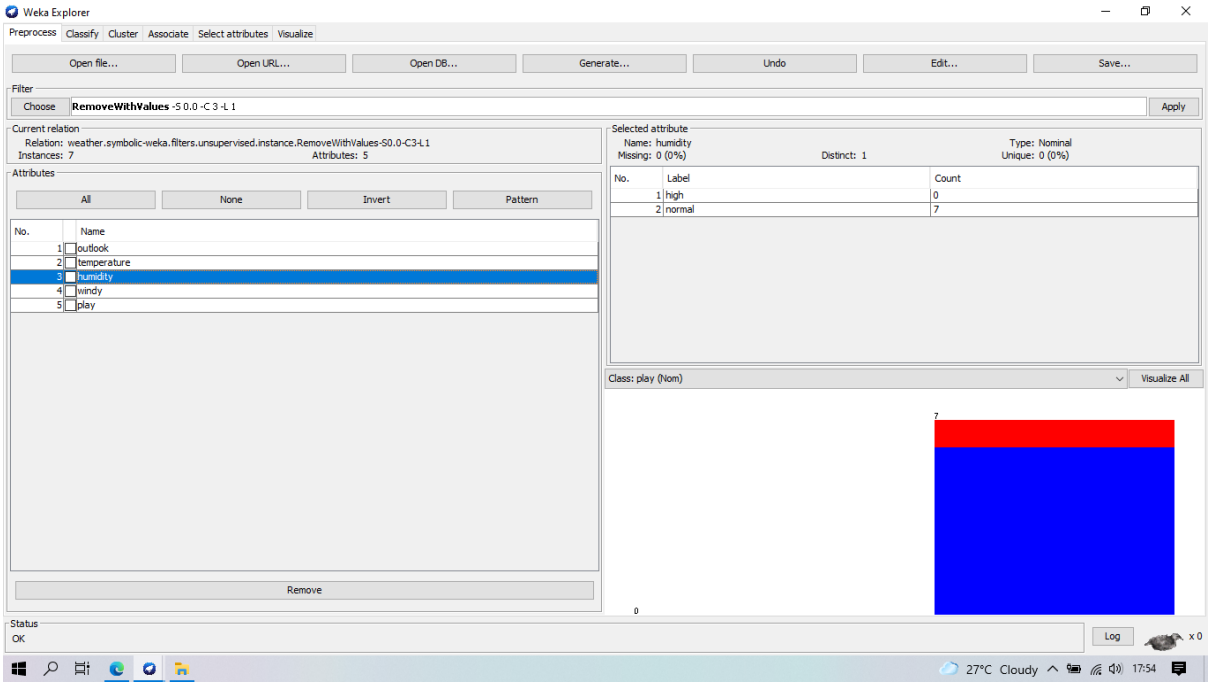
Klik RemoveWithValues agar muncul jendela untuk mengatur parameter sesuai keinginan.



Klik Open, kemudian klik Apply untuk melihat perubahan.



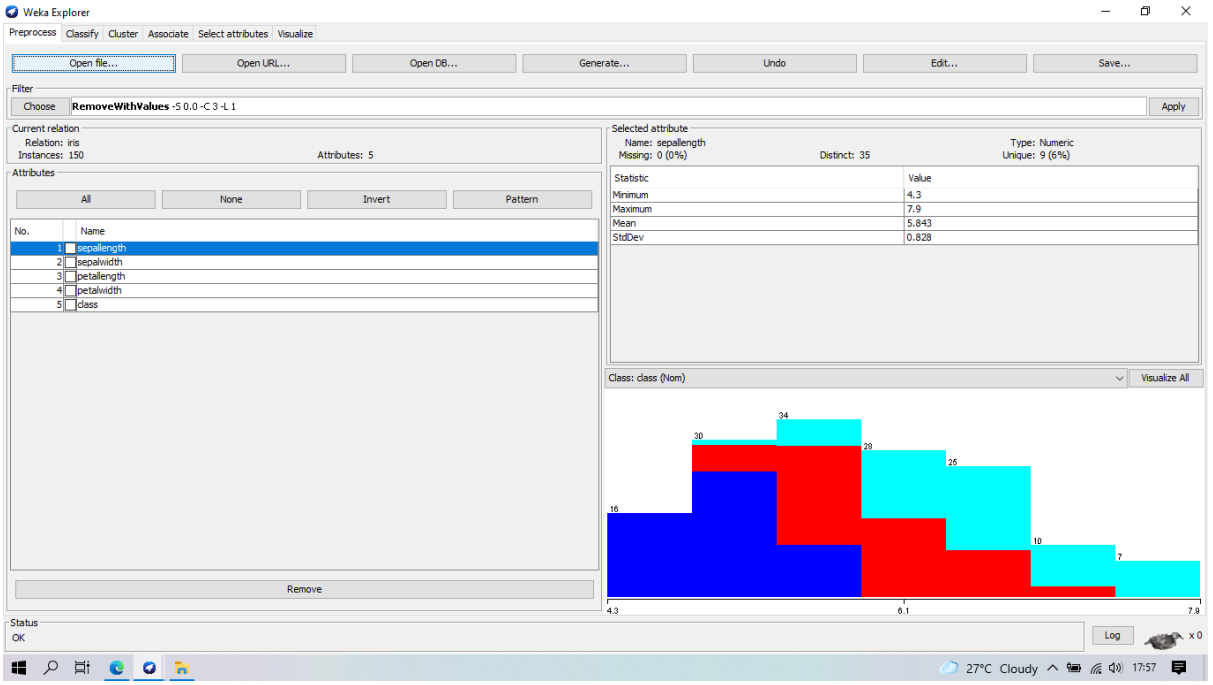
Sebelum diterapkan filter

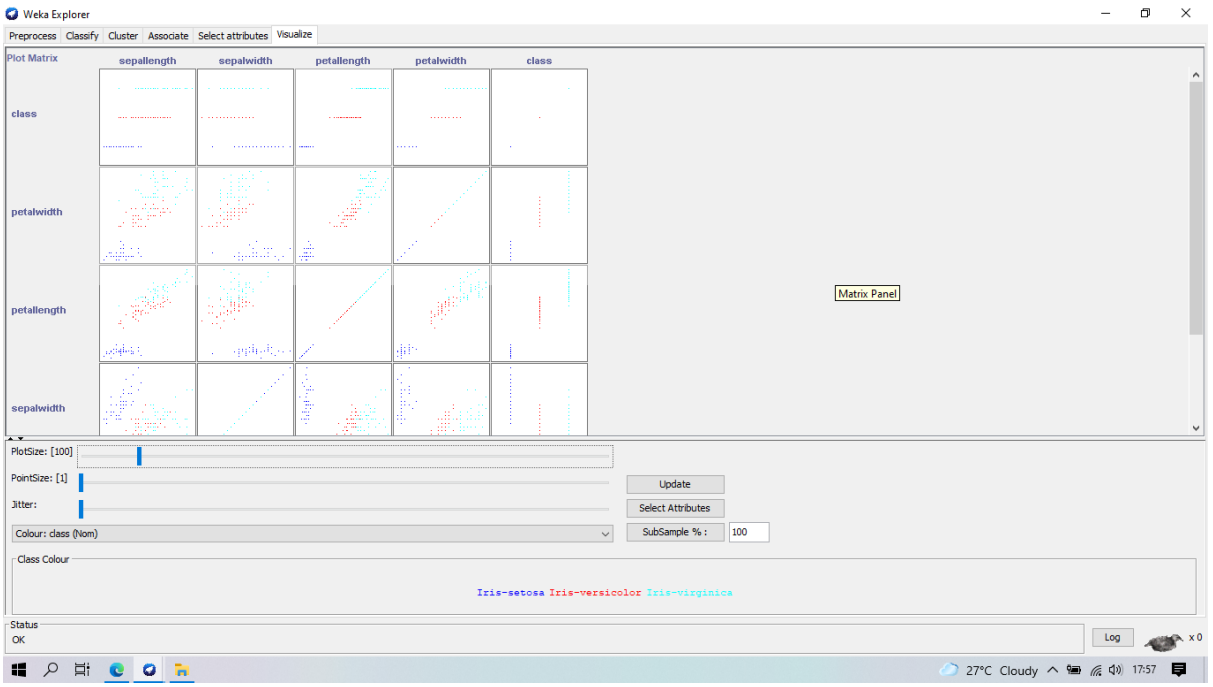


Setelah diterapkan filter

1.6 Visualizing Your Data

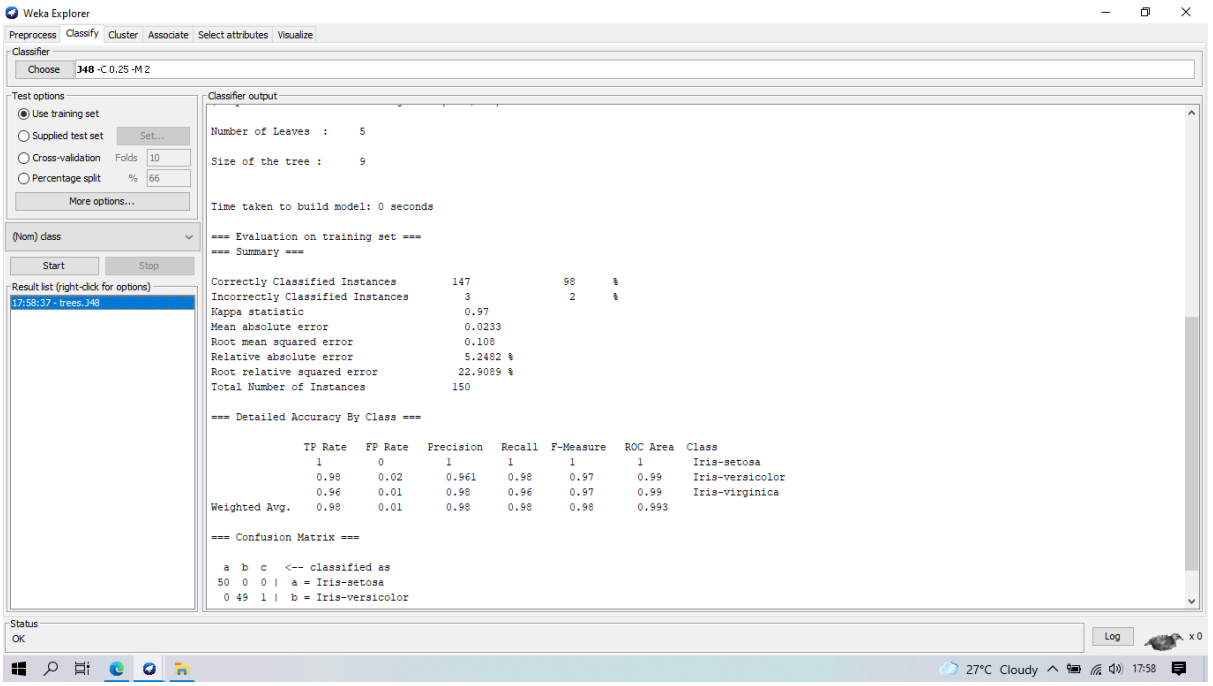
Load dataset iris.arff, Klik tab Visualize untuk melihat visualisasi dari dataset.



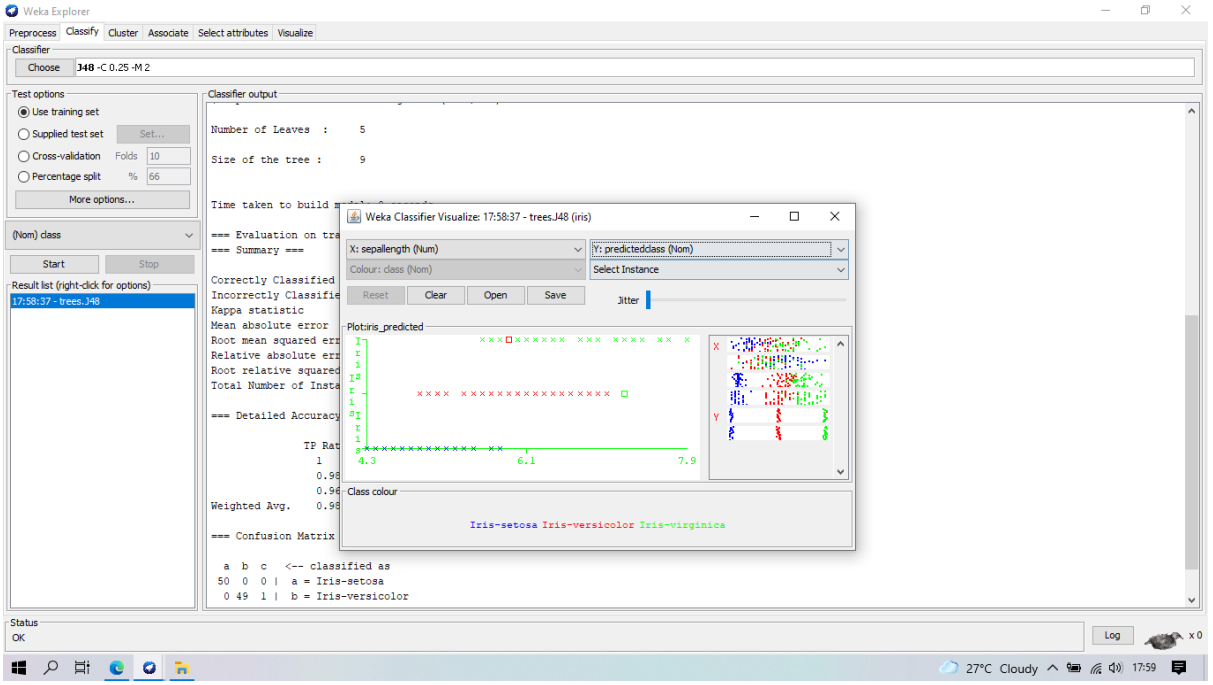


Klik salah satu visualisasinya untuk melihat lebih besar. Juga bisa dimodifikasi atribut yang ingin dijadikan sumbu X dan sumbu Y.

Selain itu, juga bisa dilihat visualisasi untuk hasil klasifikasi. Pertama lakukan klasifikasi pada tab classify. Untuk contoh, digunakan algoritma tree J48.

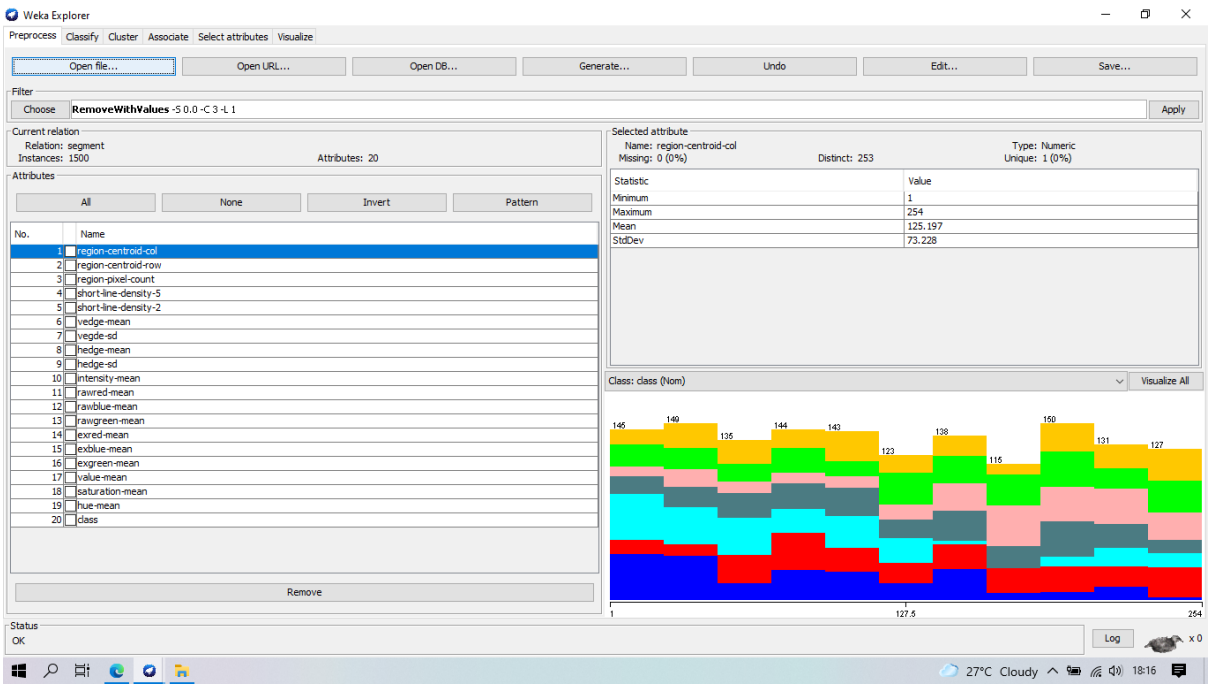


Klik kanan pada result list, kemudian pilih visualize classifier error.

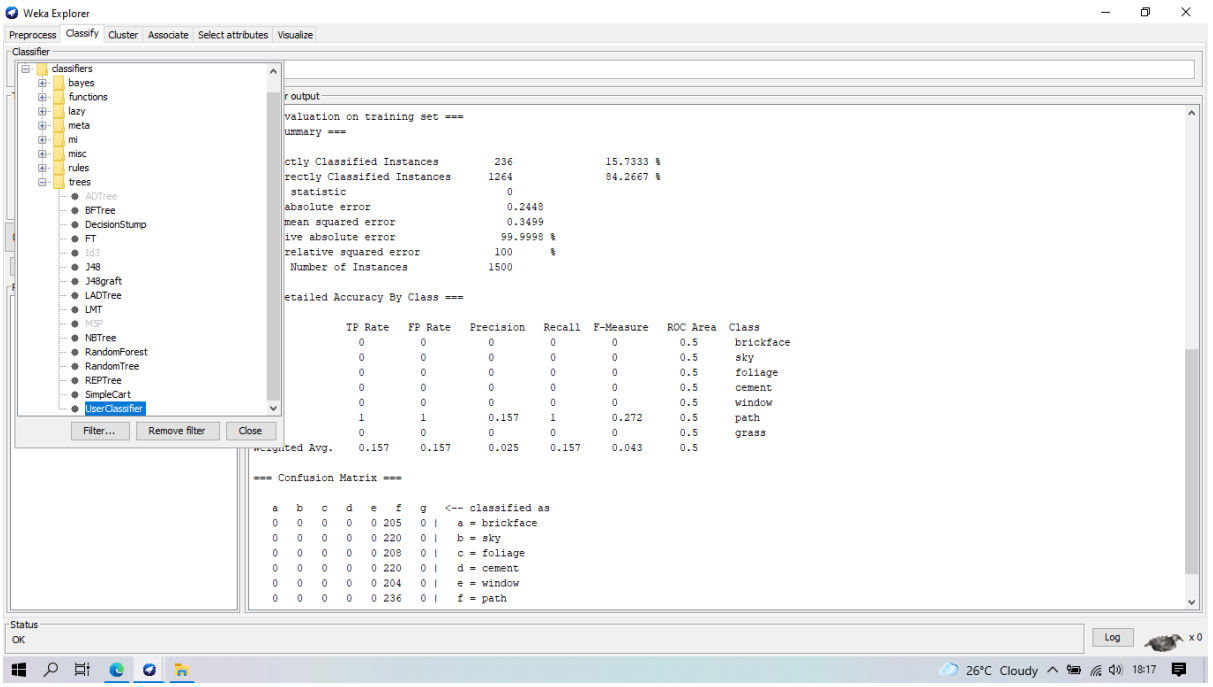


2.1 Be a Classifier!

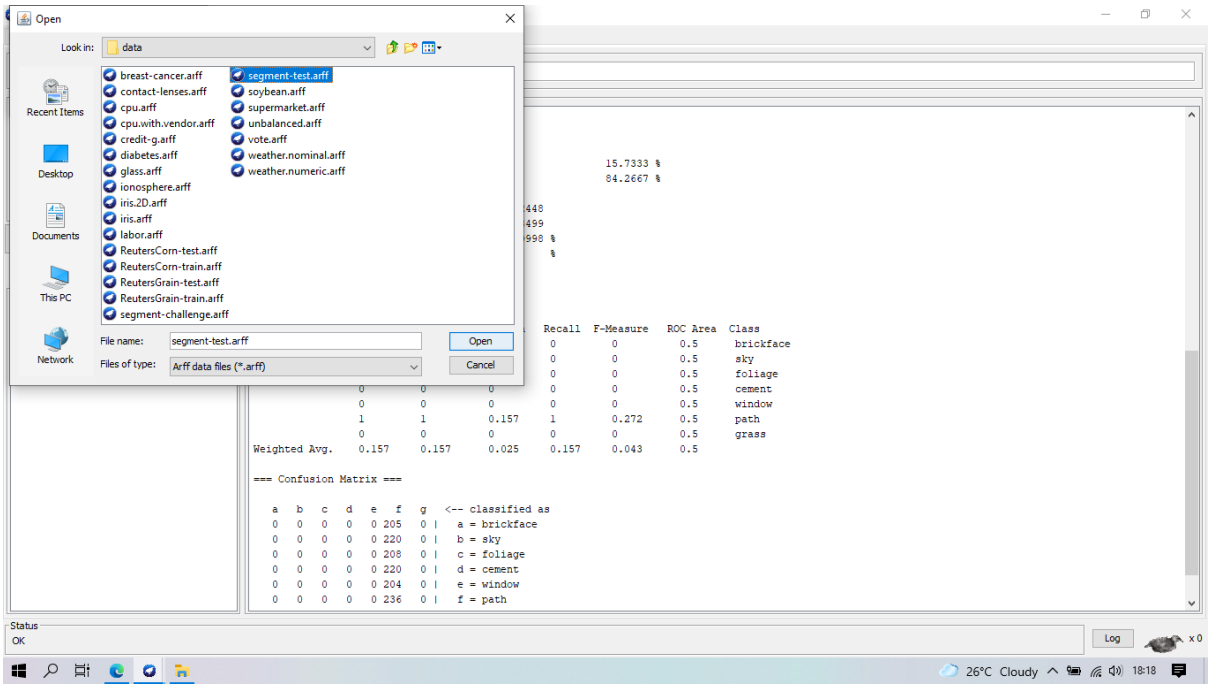
Load segment-challenge dataset.



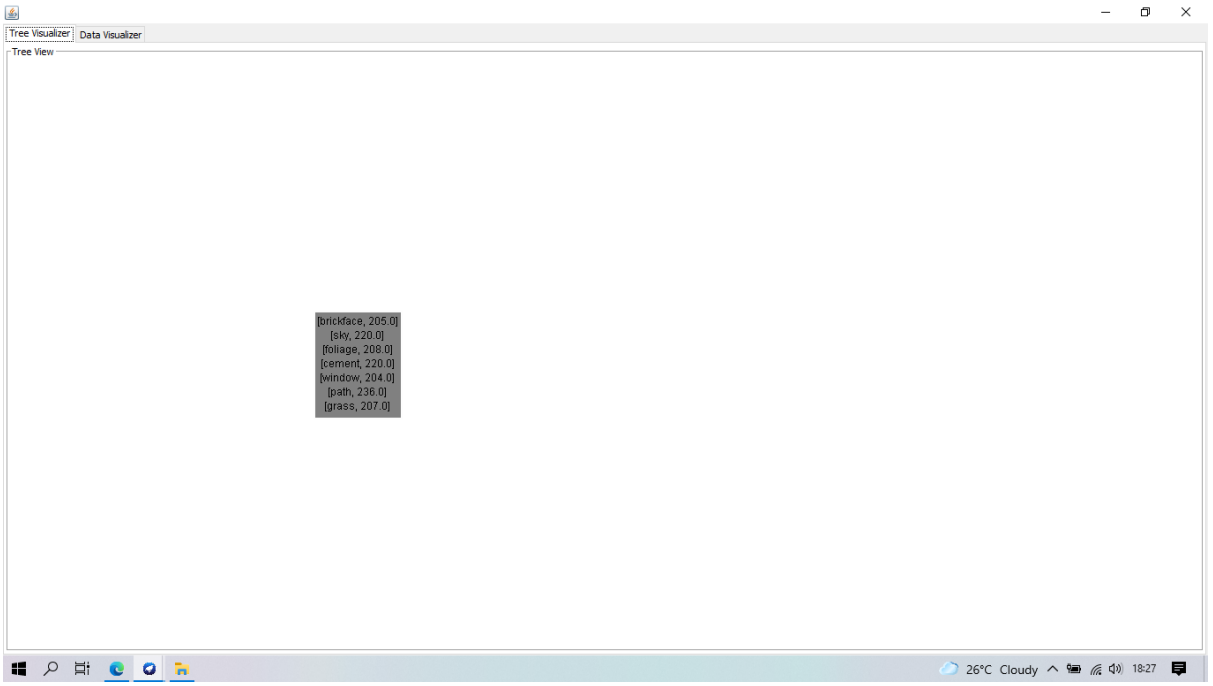
Buka tab classify, pilih userClassifier.



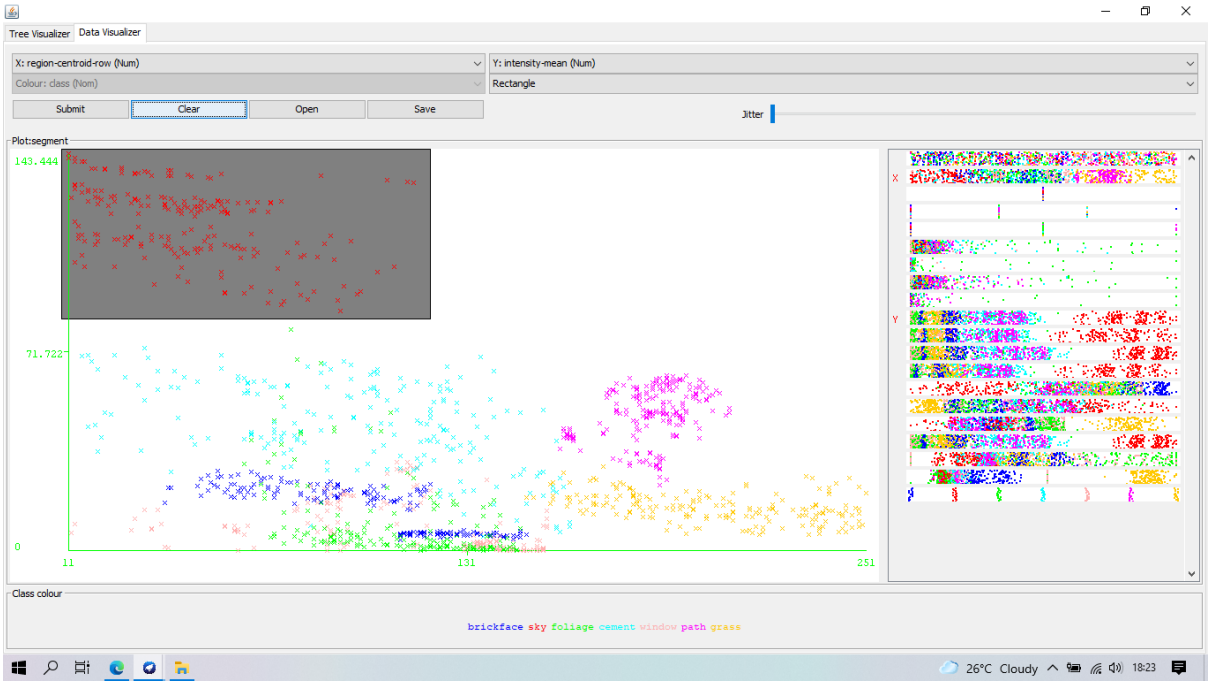
Pada test options, pilih supplied test set, kemudian klik set. Tujuannya untuk memilih dataset yang digunakan sebagai testing untuk classifier yang digunakan. Pilih segment-test.arff sebagai testing dataset.



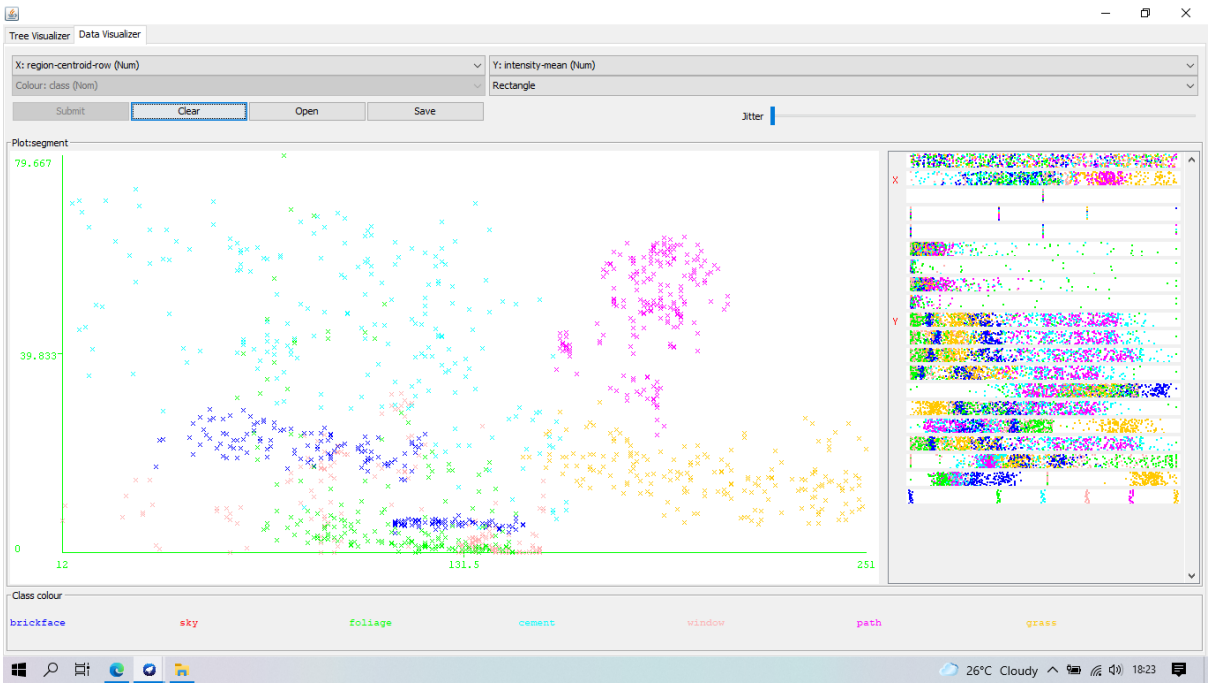
Klik start. Kemudian muncul jendela baru. Klik tab data visualizer, untuk melihat visualisasinya. Pilih sumbu X dan sumbu Y yang kira-kira sesuai sehingga nilai yang berwarna sama terkelompok cukup baik.



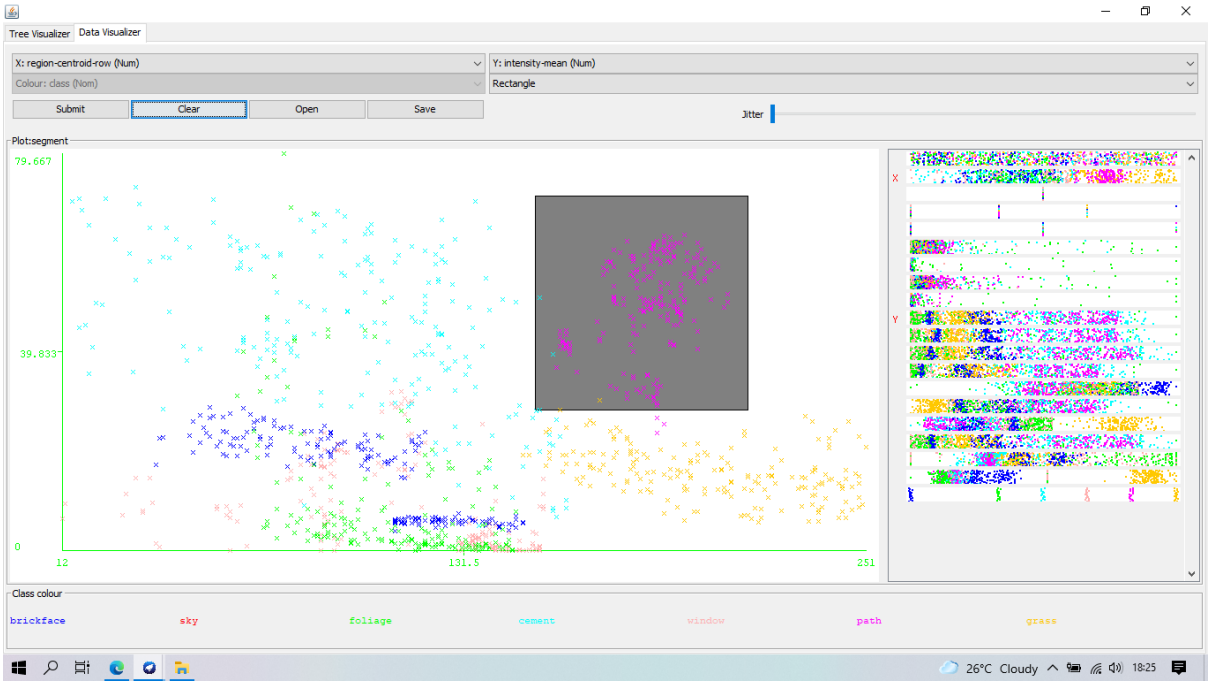
Lakukan data selection pada dropdown select instance, pilih rectangle. Kemudian pilih area dengan koordinat berwarna merah. Kemudian klik submit.

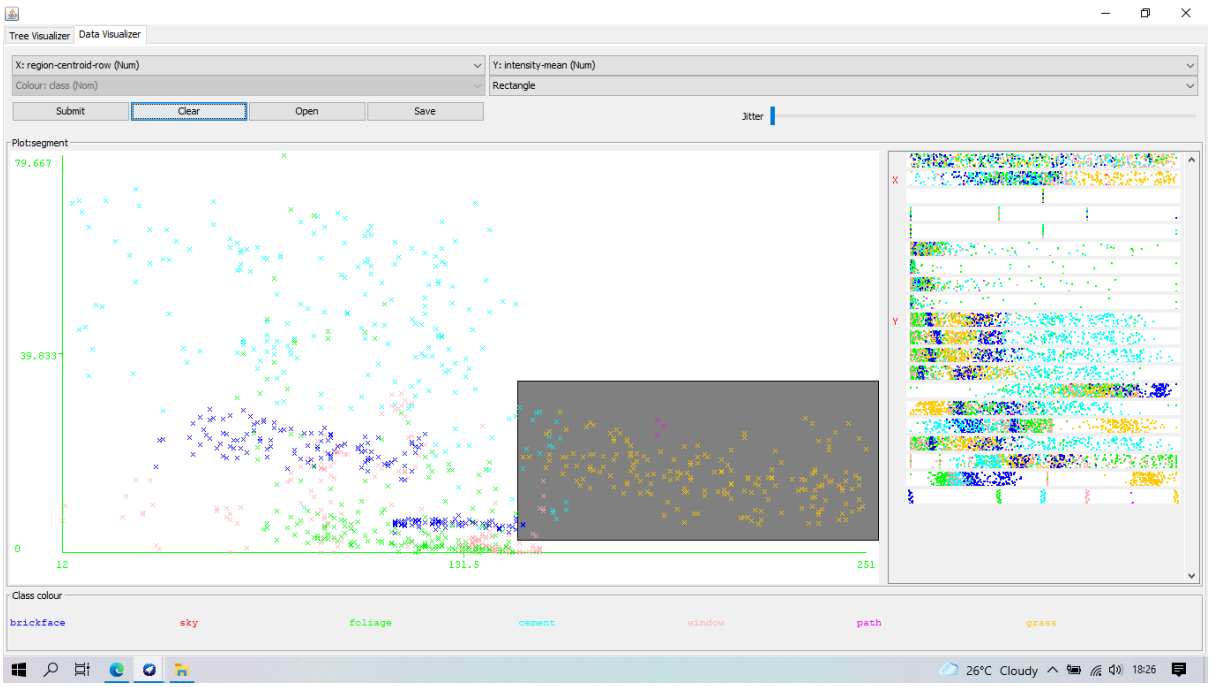


Visualisasinya akan berubah seperti gambar berikut. Hal ini dikarenakan nilai yang telah diseleksi telah diklasifikasikan, sehingga yang terlihat hanya yang belum diklasifikan oleh User.

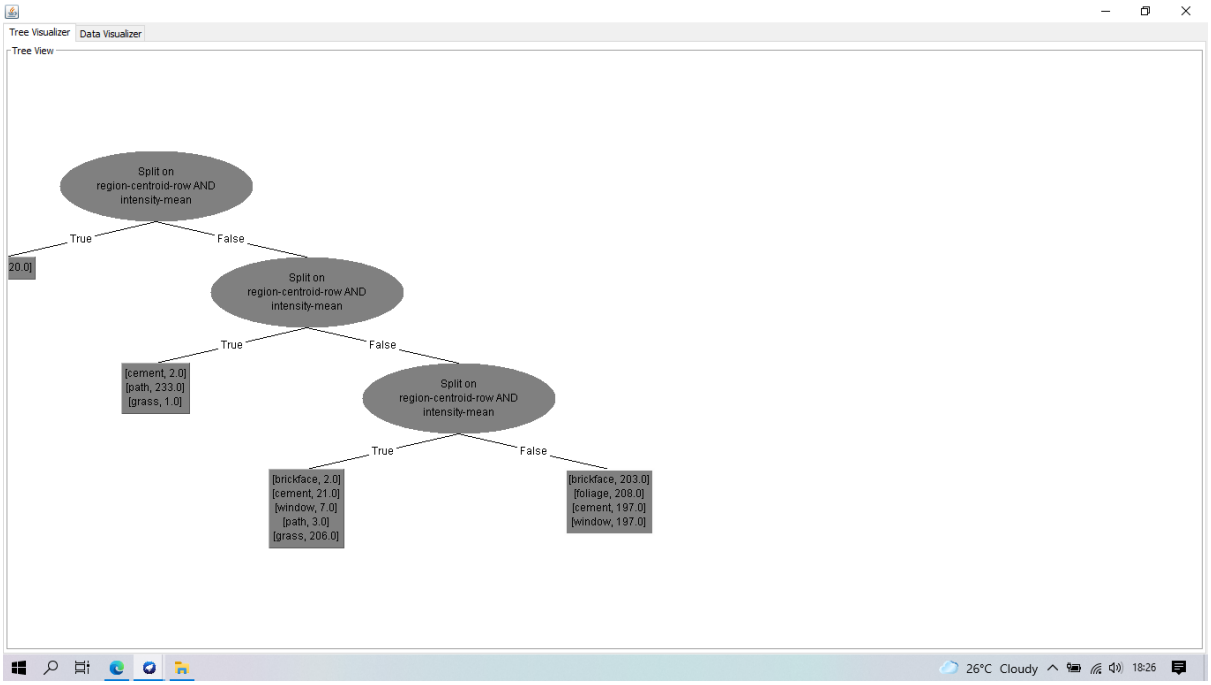


Lanjutkan dengan memilih warna lain, seperti ungu, biru, kuning dan seterusnya, sampai semuanya telah terklasifikasi.



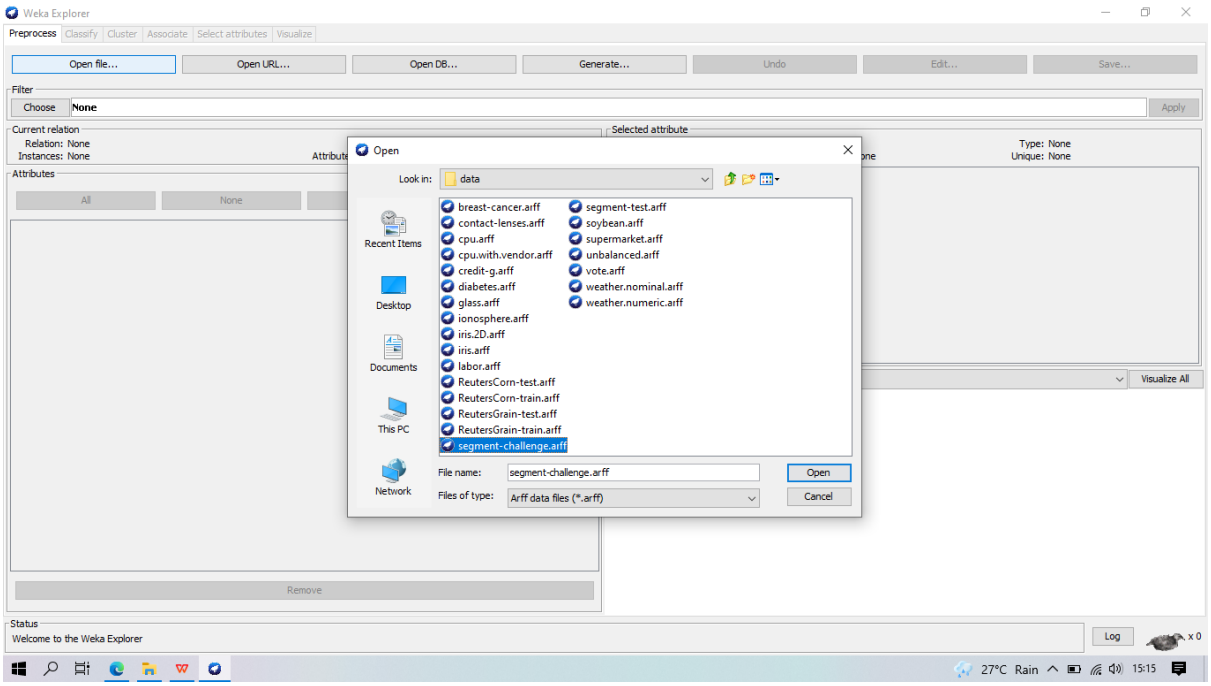


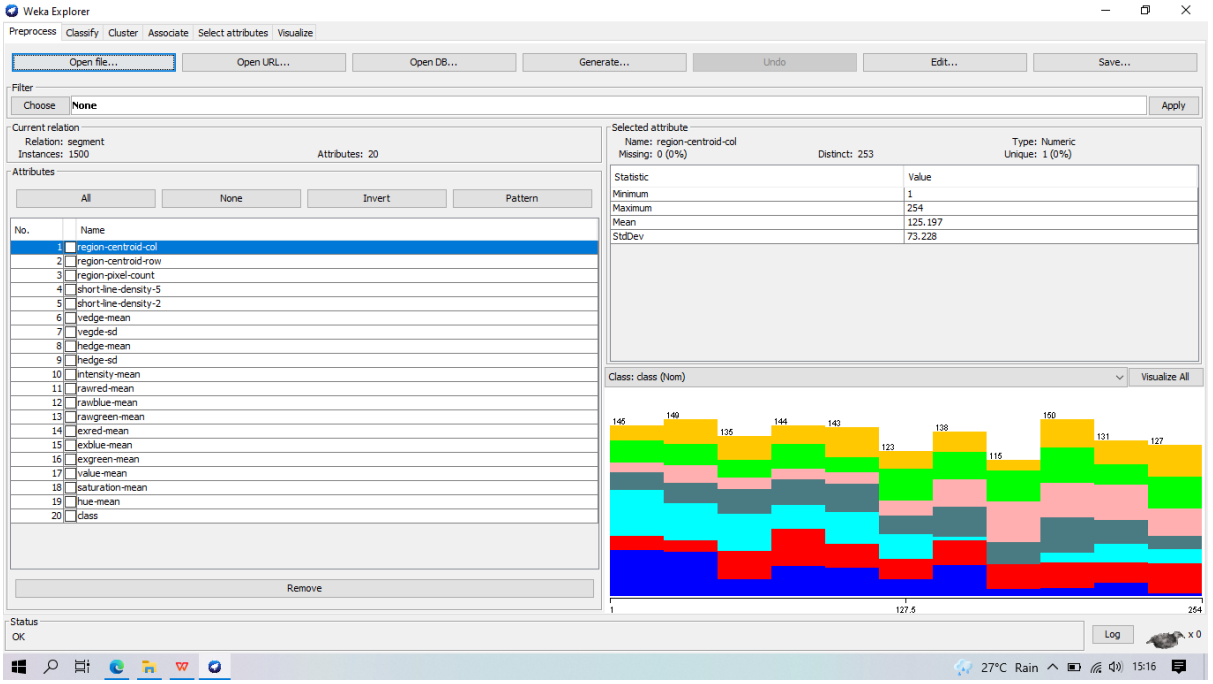
Setelah melakukan klasifikasi, klik tab Tree Visualizer. Terlihat bahwa tree yang ditampilkan berbeda dengan sebelumnya, instance telah diklasifikasikan dengan cukup baik.



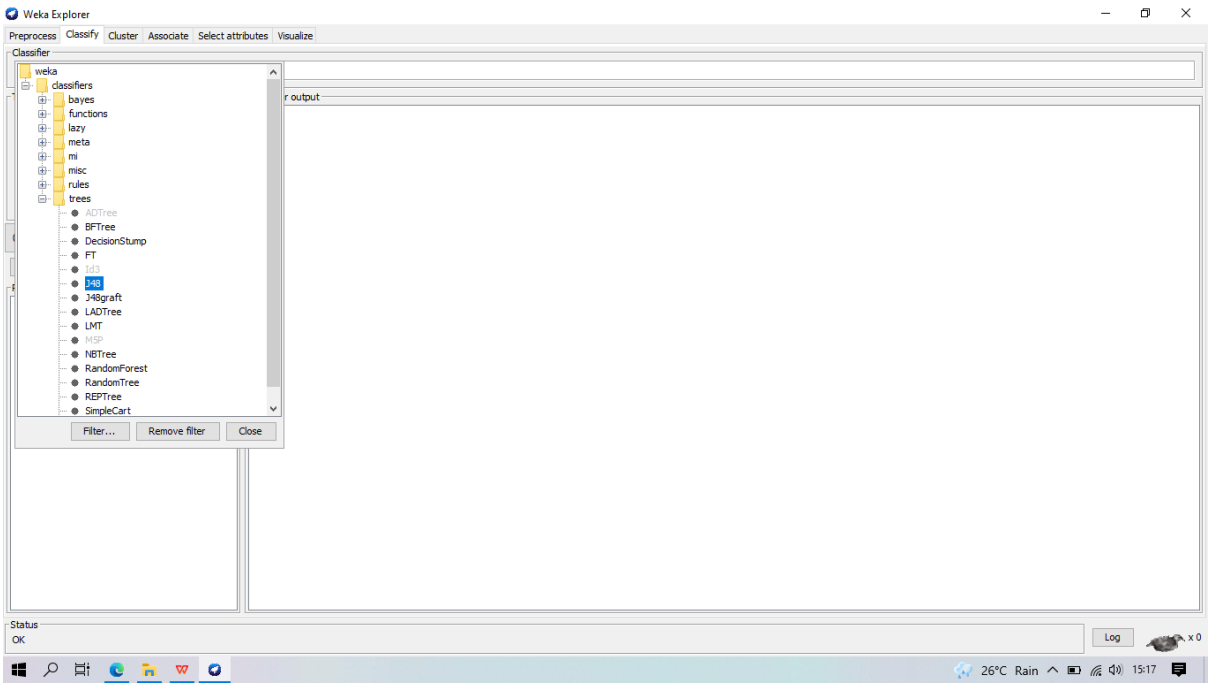
2.2 Training and Testing

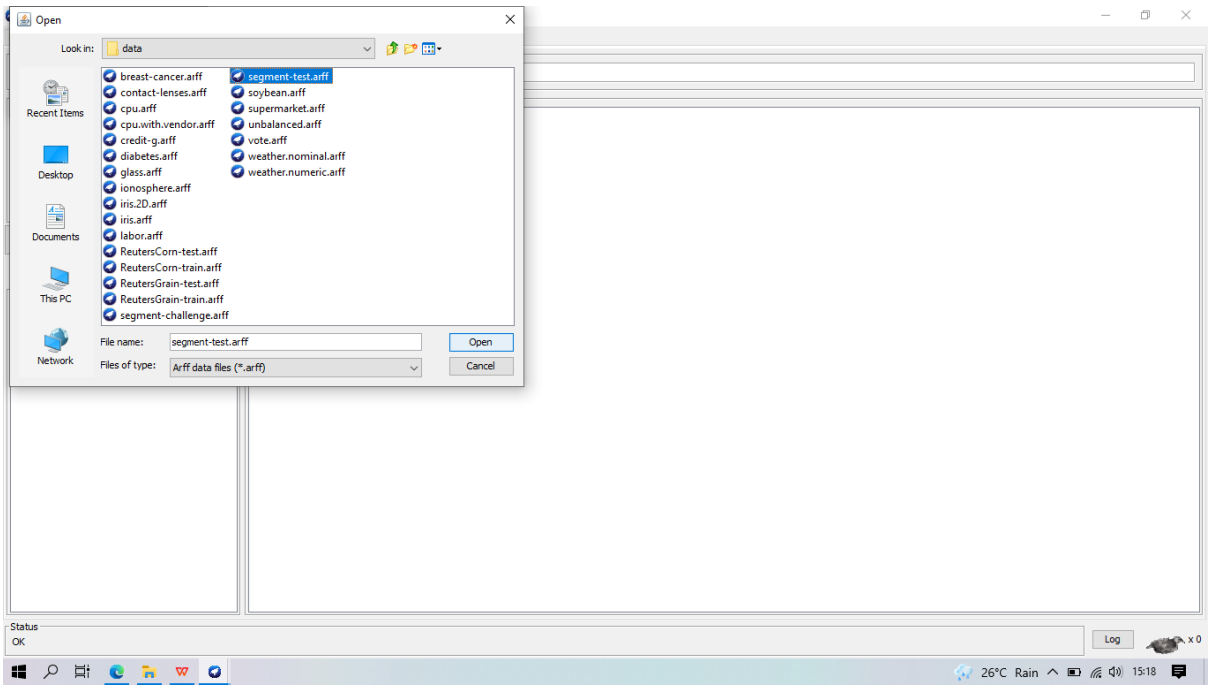
Load dataset segment-challenge.arff



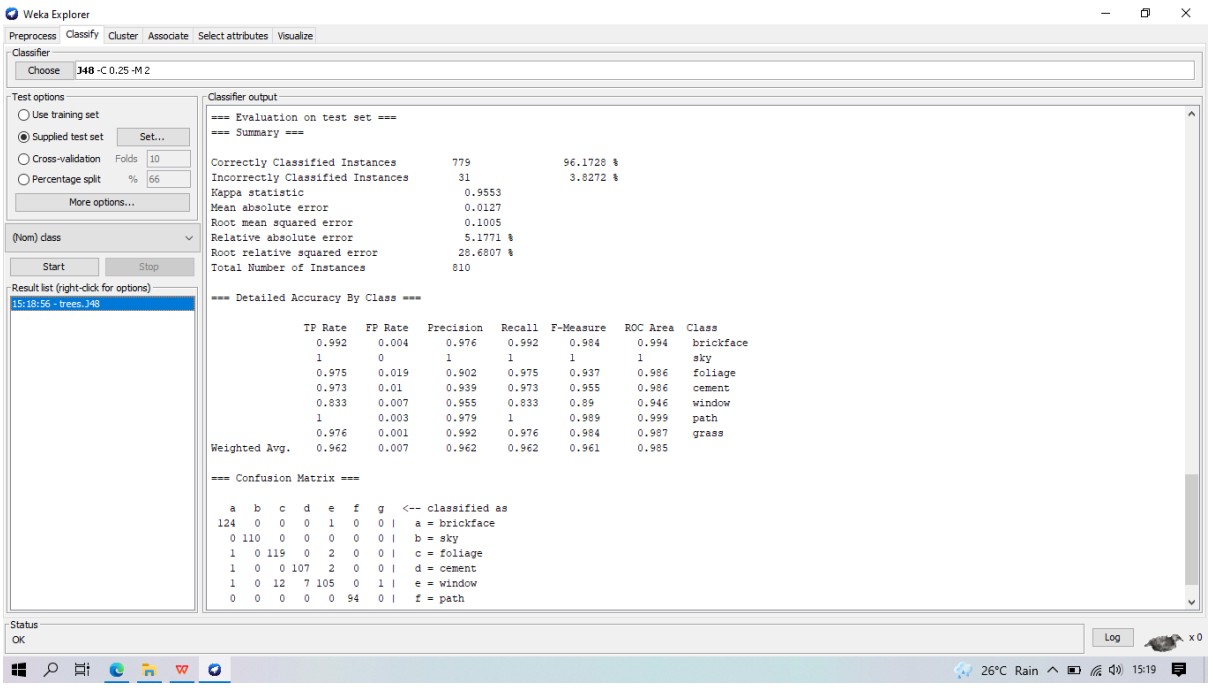


Pada tab classify, pilih classifier J48 dan supplied test set segment-test.arff.

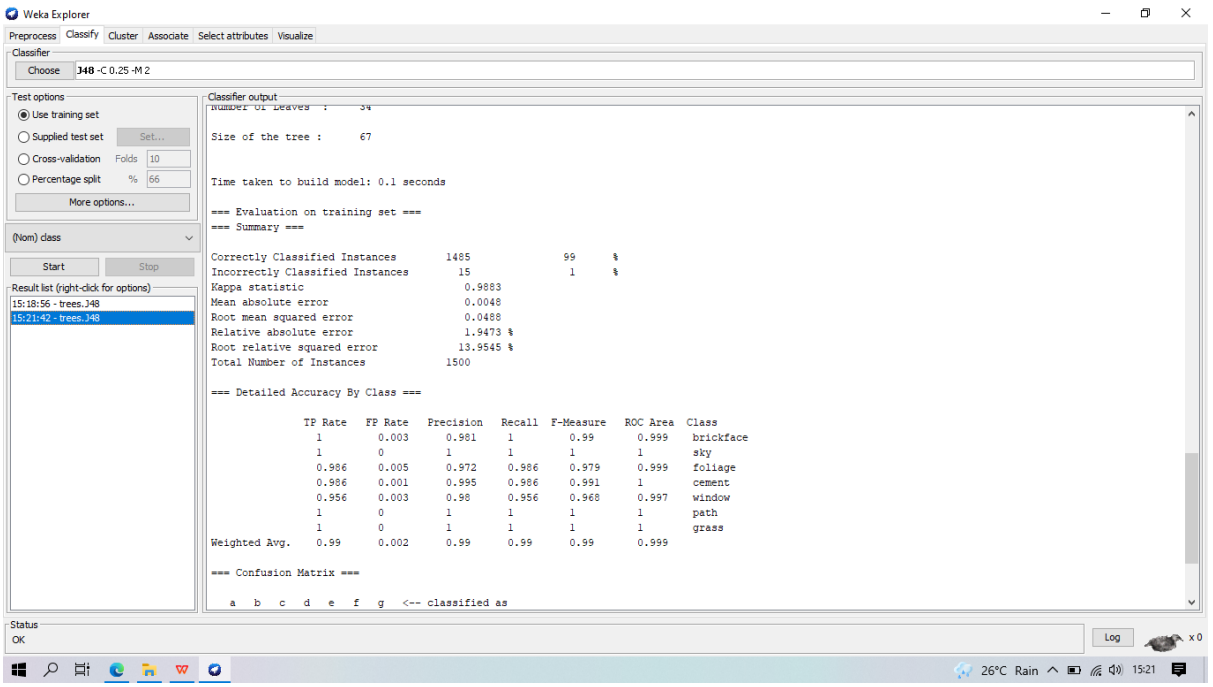




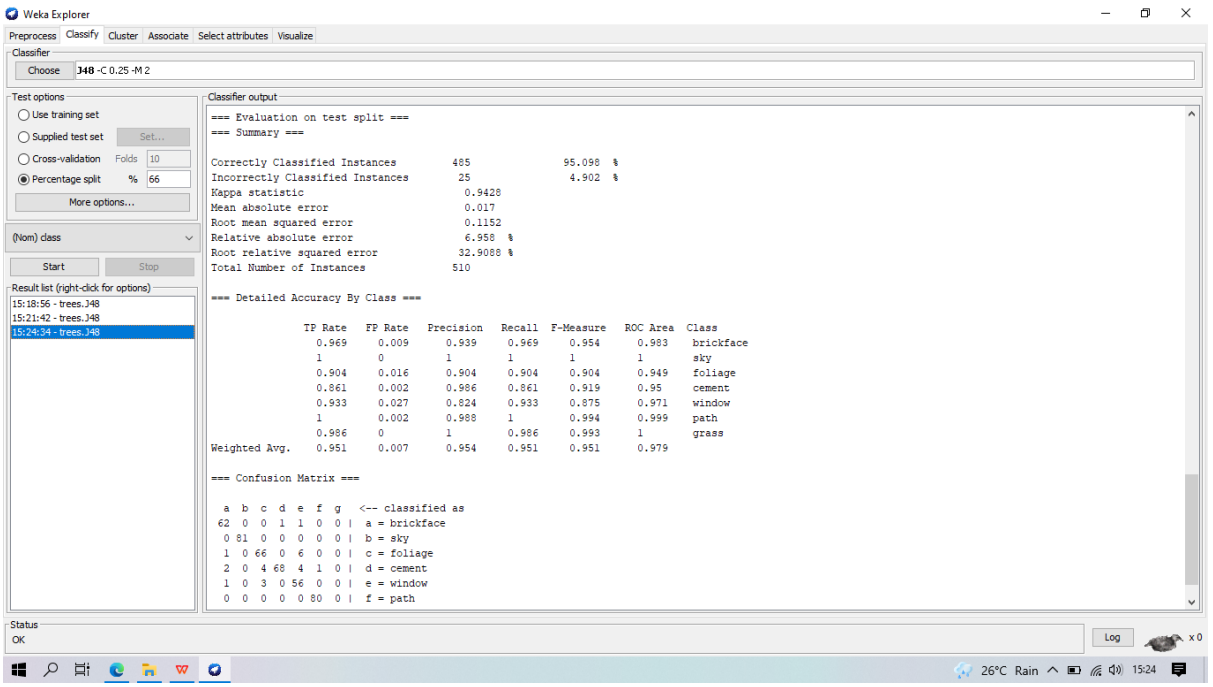
Klik Start. Akurasi hasil testing adalah 96%.



Ubah test option menjadi use training set. Akurasi berupa menjadi 99% (hal ini tidak diseharusnya dilakukan dalam melakukan testing).

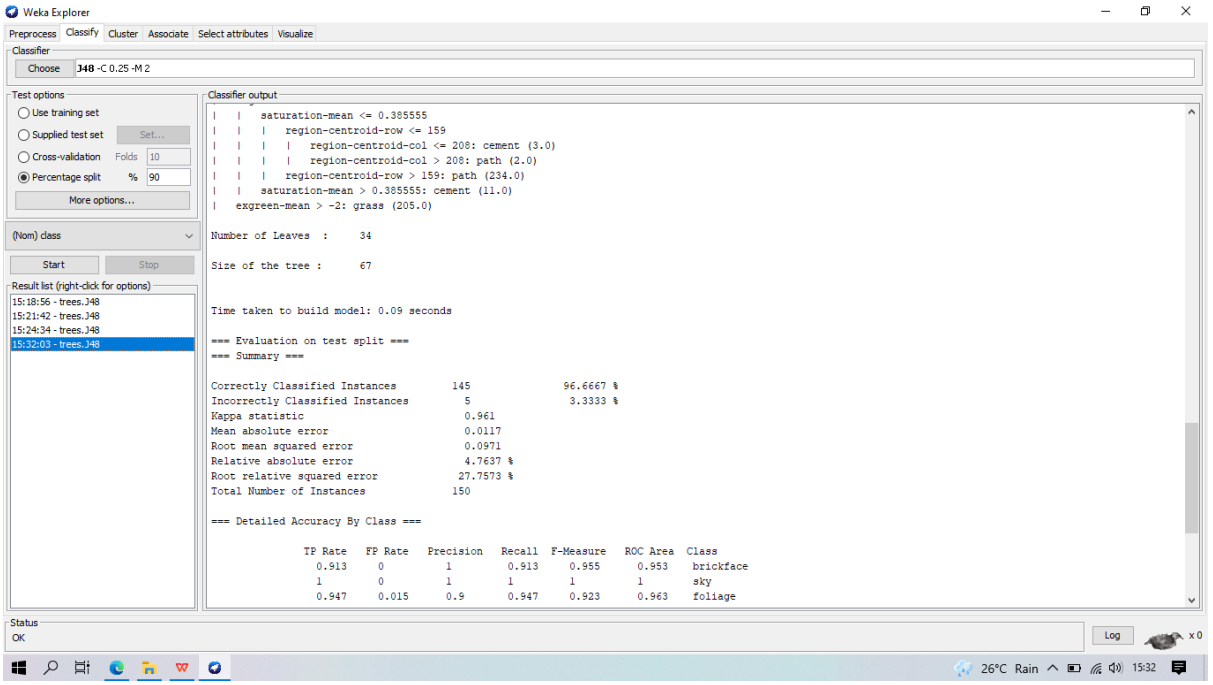


Jika ingin menggunakan satu dataset saja, pilih percentage split pada test option, kemudian diatur persentase training set. Sebagai contoh, persentase diatur menjadi 66%. Akurasi yang didapatkan 95%.

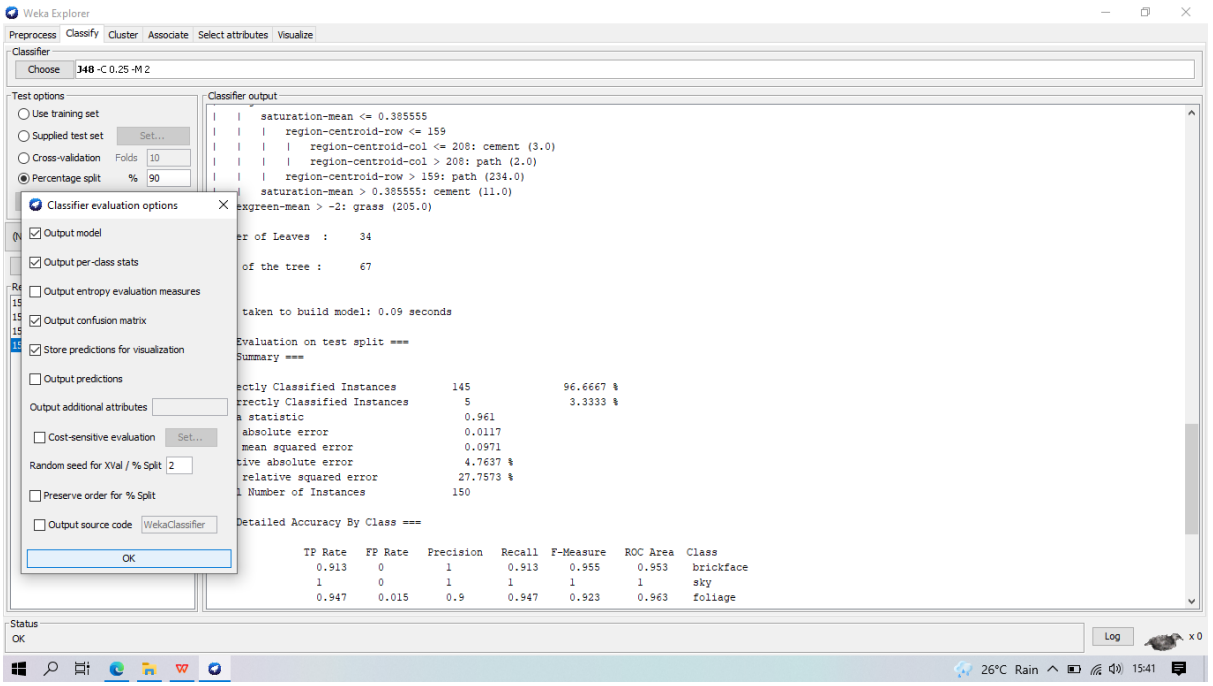


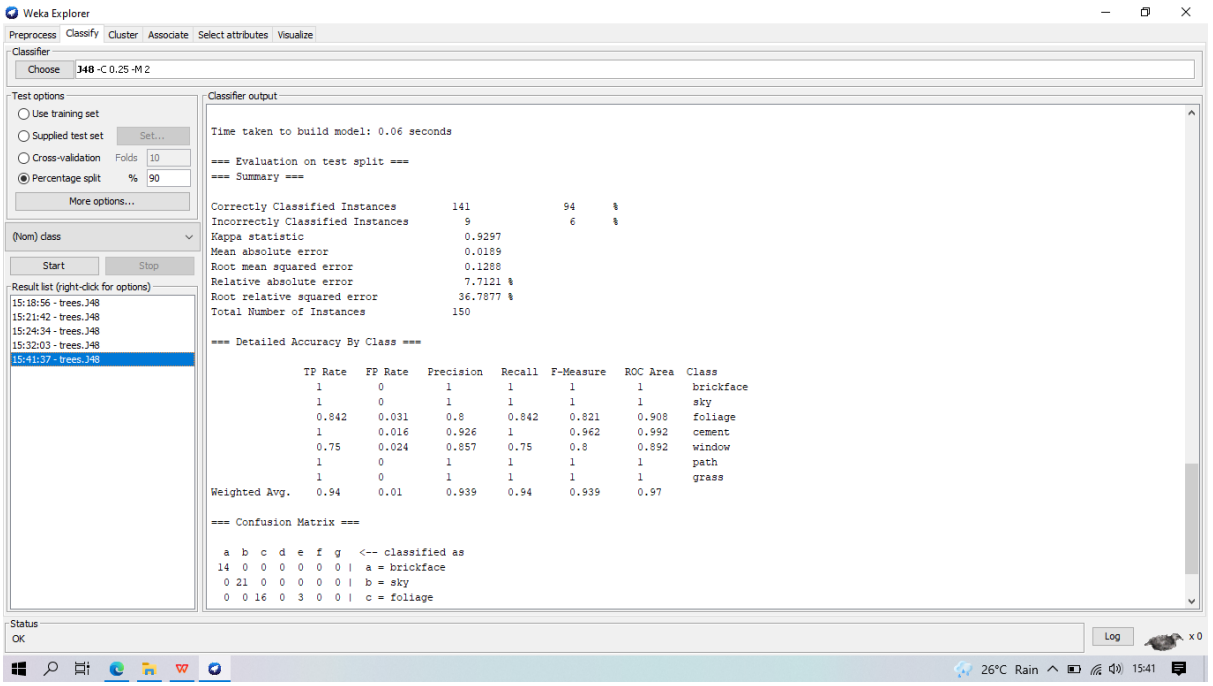
2.3 Repeated Training and Testing

Pada tahap ini tetap menggunakan dataset yang sama dengan tahap sebelumnya yaitu segment-challenge.arff dan classifier yang sama, J48. Karena dataset terdiri 1500 instances, percentage split diatur menjadi 90% (90% untuk training data, 10% untuk testing data). Akurasi yang didapatkan adalah 96,67%



Untuk lebih lanjut, klik more options. Atur random seed for cross-validation menjadi angka yang diinginkan, sebagai contoh, random seed diatur menjadi 2 (Defaultnya bernilai 1). Kemudian klik Start, Akurasi yang didapatkan adalah 94%.





Kemudian diubah lagi nilai random seed for cross-validation sesuai keinginan dan lihat perbedaan akurasi yang didapatkan. Setelah dicoba diubah nilai random seed dari 1-10. Didapatkan hasil sebagai berikut.

Sample mean $\bar{x} = \frac{\sum x_i}{n}$

Variance $\sigma^2 = \frac{\sum (x_i - \bar{x})^2}{n-1}$

Standard deviation σ

0.967
0.940
0.940
0.967
0.953
0.967
0.920
0.947
0.933
0.947

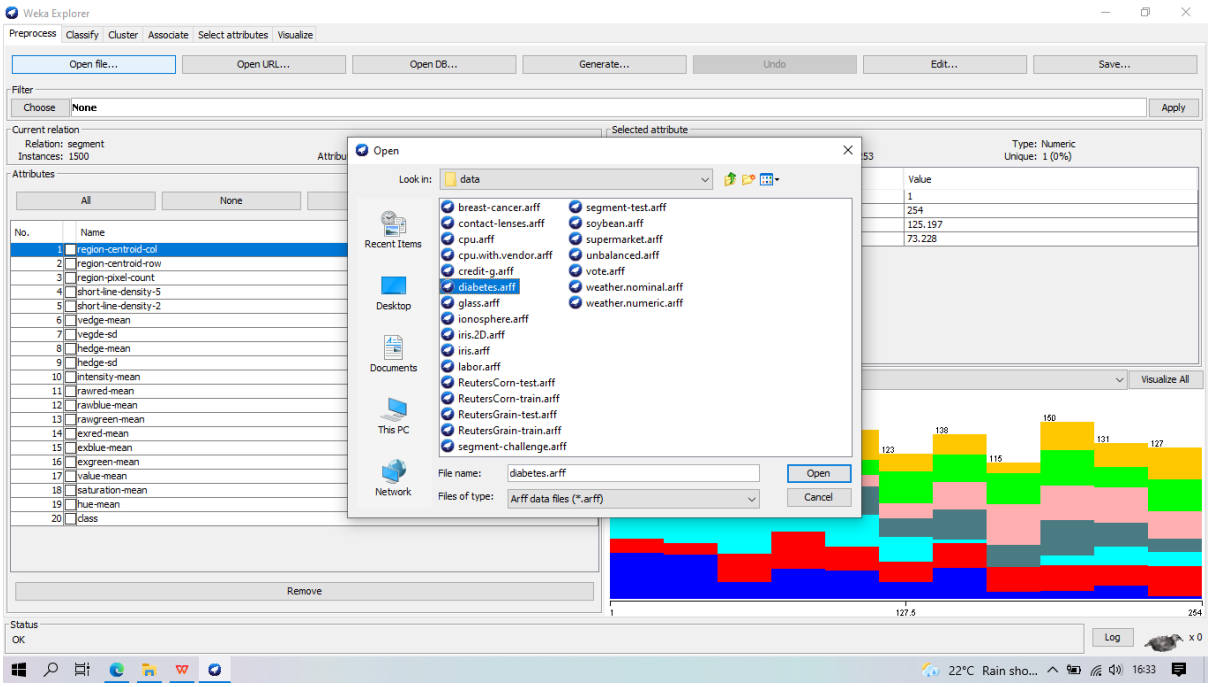
$\bar{x} = 0.949,\ \sigma = 0.018$

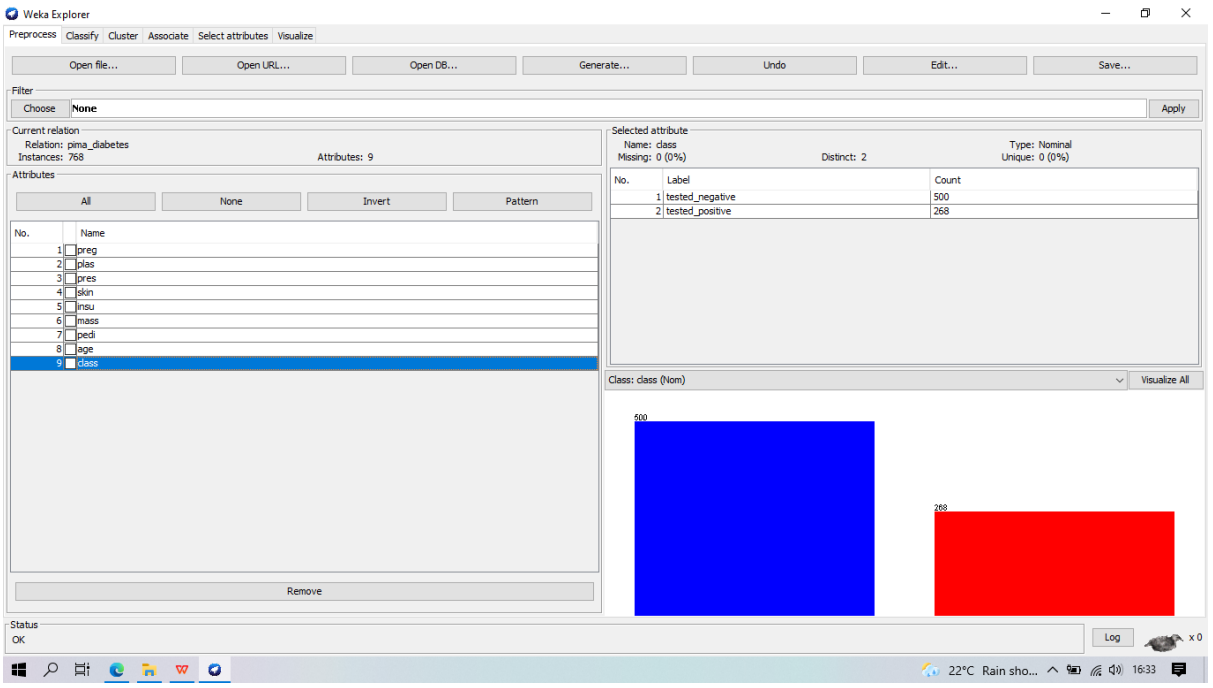
Berdasarkan sampel data tersebut dapat dicari nilai rata-rata sampel dan standar deviasinya. Rata-rata akurasinya adalah 94,9% dan standar deviasinya adalah 1,8%. Sehingga dapat disimpulkan bahwa nilai akurasi berada diantara 93-97%.

2.4 Baseline Accuracy

Dalam membuat model pada suatu dataset, jangan menggunakan classifier sembarang, perlu diperhatikan dataset yang digunakan dan selalu lakukan percobaan *baseline accuracy* (menggunakan classifier *ZeroR*). Dimana classifier yang akan digunakan nantinya harus memiliki akurasi lebih besar dari akurasi classifier *ZeroR*.

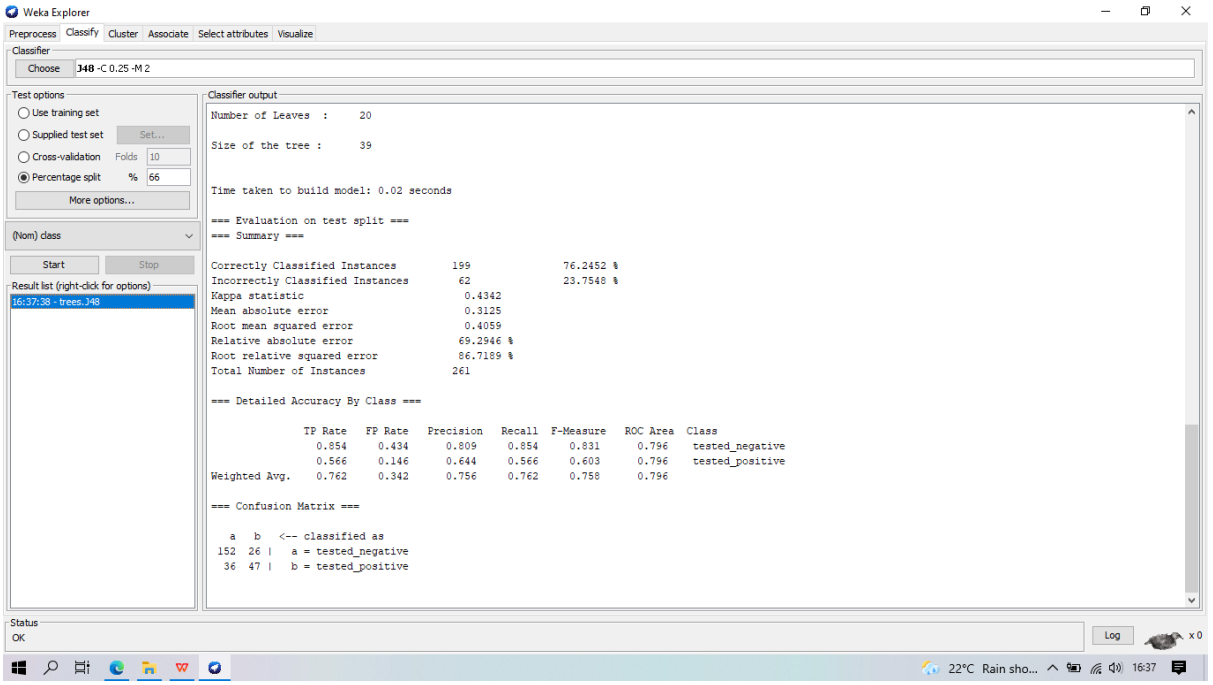
Load dataset diabetes.arff



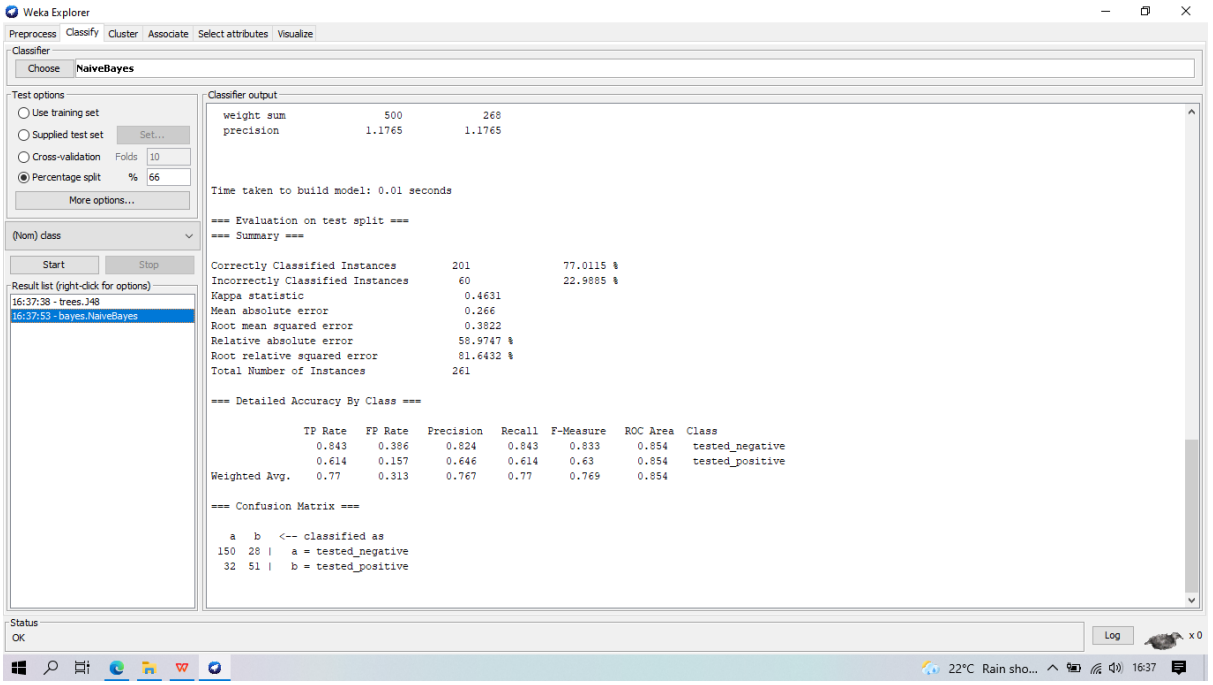


Gunakan berbagai macam classfier untuk membuat model dari dataset diabetes. Sebagai contoh, gunakan classifier trees - J48, bayes - NaiveBayes, lazy - IBk, dan rules - PART. serta perlu juga dilakukan pengecekan *baseline accuracy* menggunakan ZeroR. (percentage split sebesar 66% dan random seed bernilai 1).

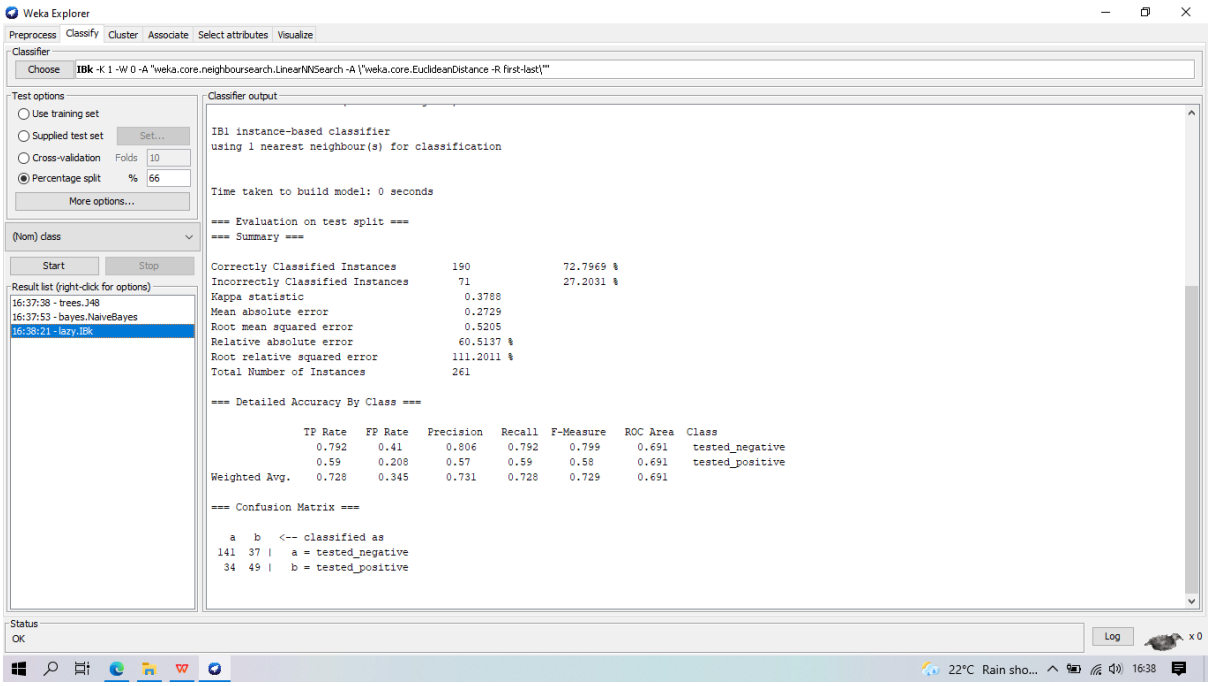
Berikut hasil yang didapatkan.



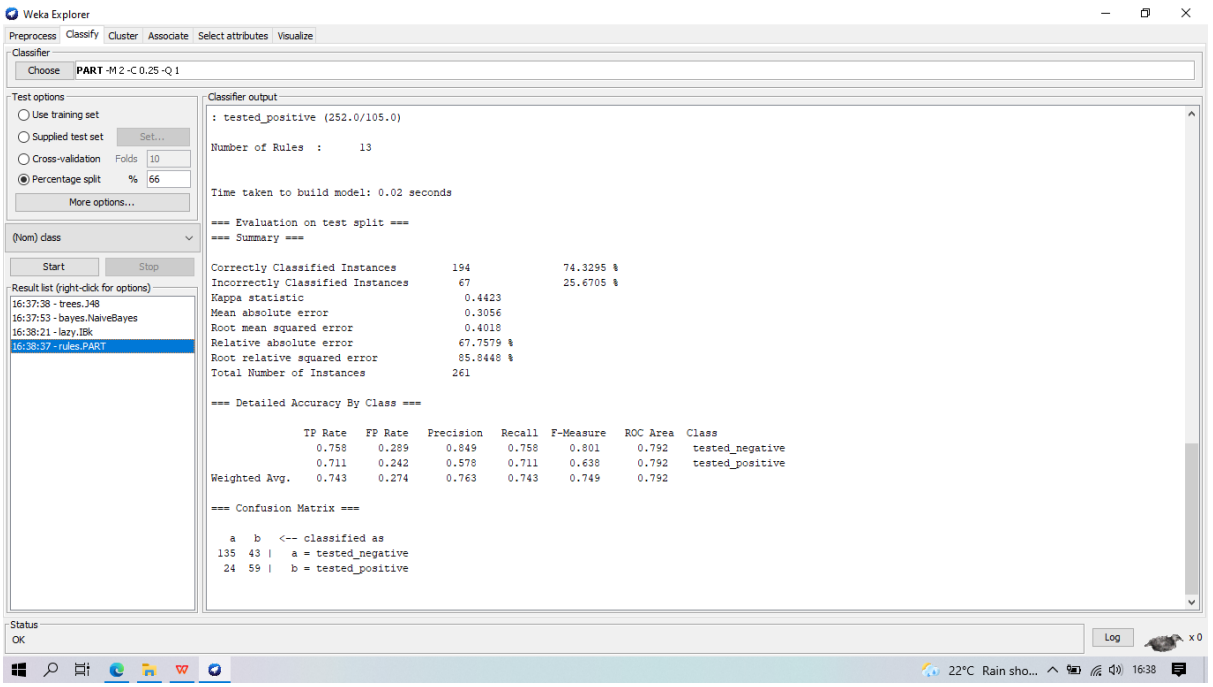
Menggunakan classifier J48 (Akurasi 76%)



Menggunakan classifier NaiveBayes (Akurasi 77%)



Menggunakan classifier IBk (Akurasi 73%)

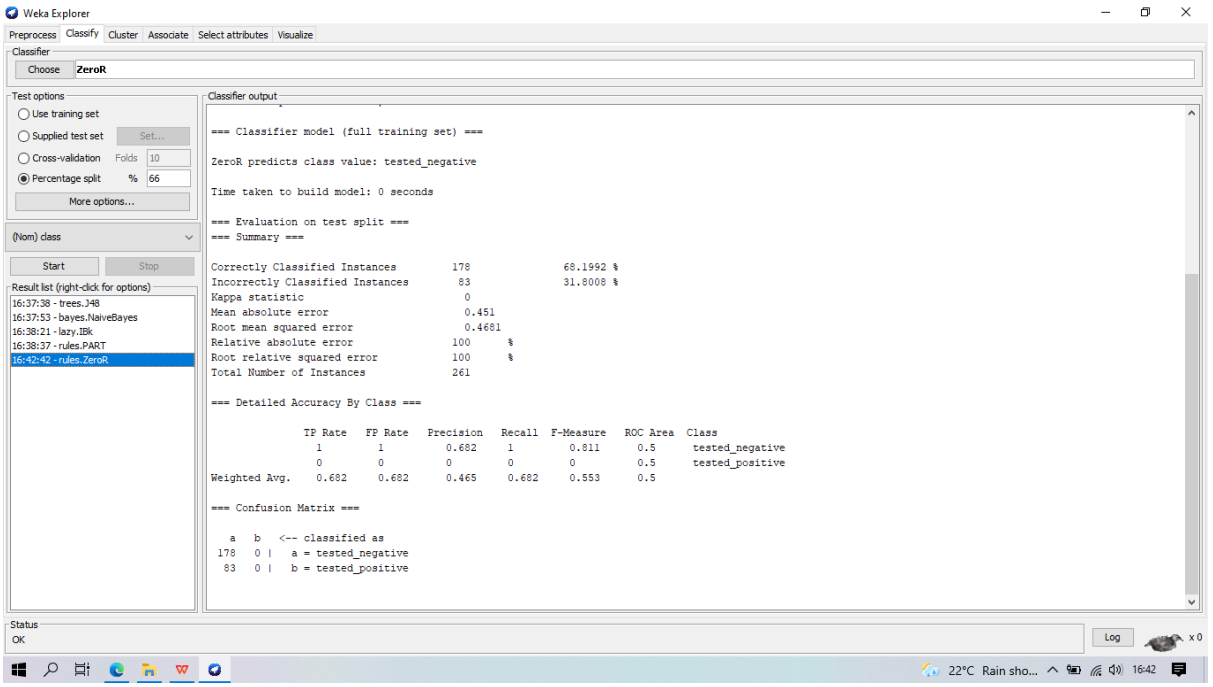


Menggunakan classifier PART (akurasi 74%)

Setelah dicoba setiap classifier, berikut akurasi yang dihasilkan

- ***trees > J48*** **76%**
- ***bayes > NaiveBayes*** **77%**
- ***lazy > IBk*** **73%**
- ***rules > PART*** **74%**

Kemudian lakukan pengecekan pada ZeroR.



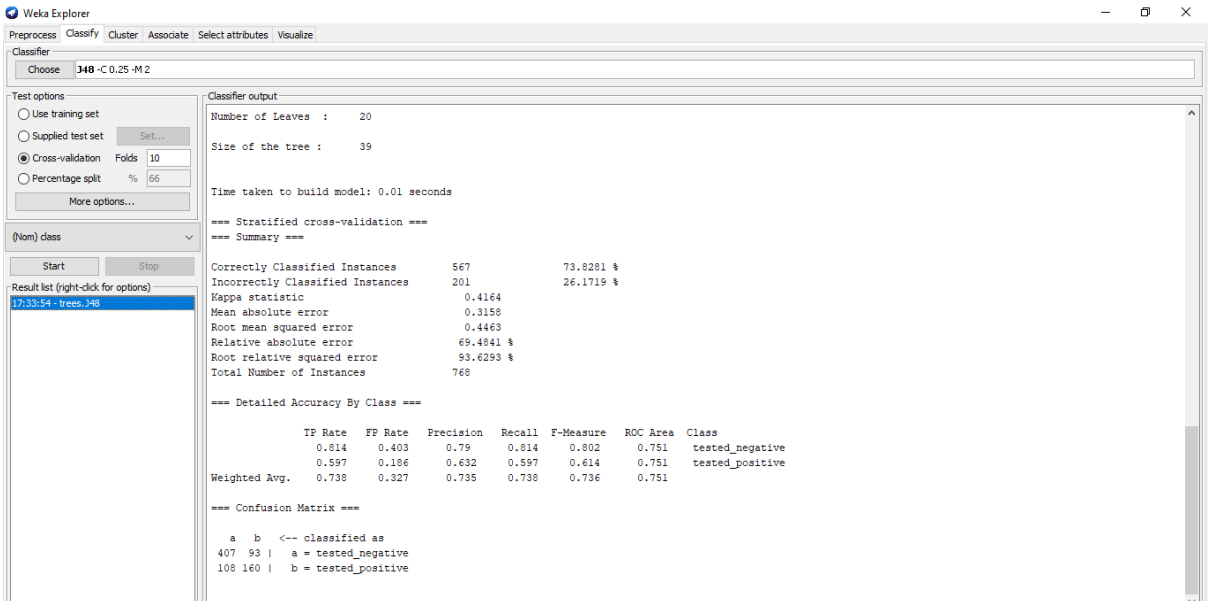
Akurasi yang didapatkan adalah 68%. Jika akurasi yang didapatkan menggunakan suatu classifier tertentu diatas akurasi ketika menggunakan classifier ZeroR, maka classifier tersebut bisa digunakan.

2.5 Cross-validation

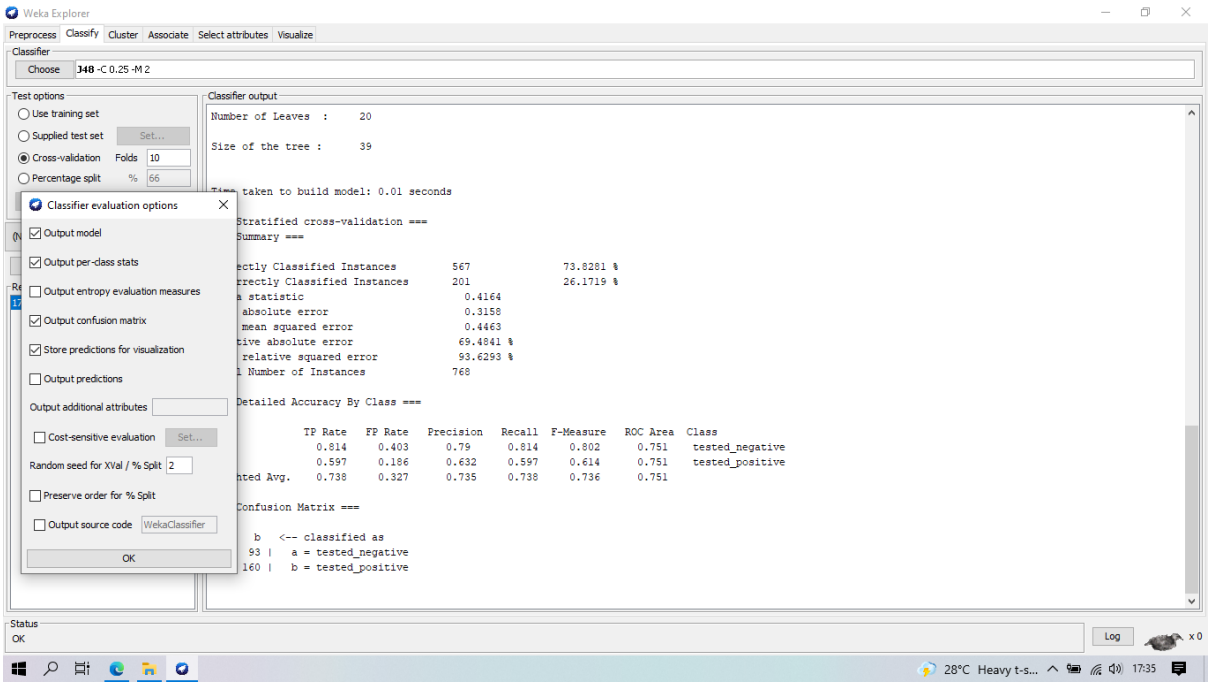
Cross-validation (CV) adalah metode statistik yang dapat digunakan untuk mengevaluasi kinerja model atau algoritma dimana data dipisahkan menjadi dua subset. Cross-validation lebih baik dari repeated holdout. Hal tersebut akan dibuktikan pada praktik 2.6.

2.6 Cross-validation Results

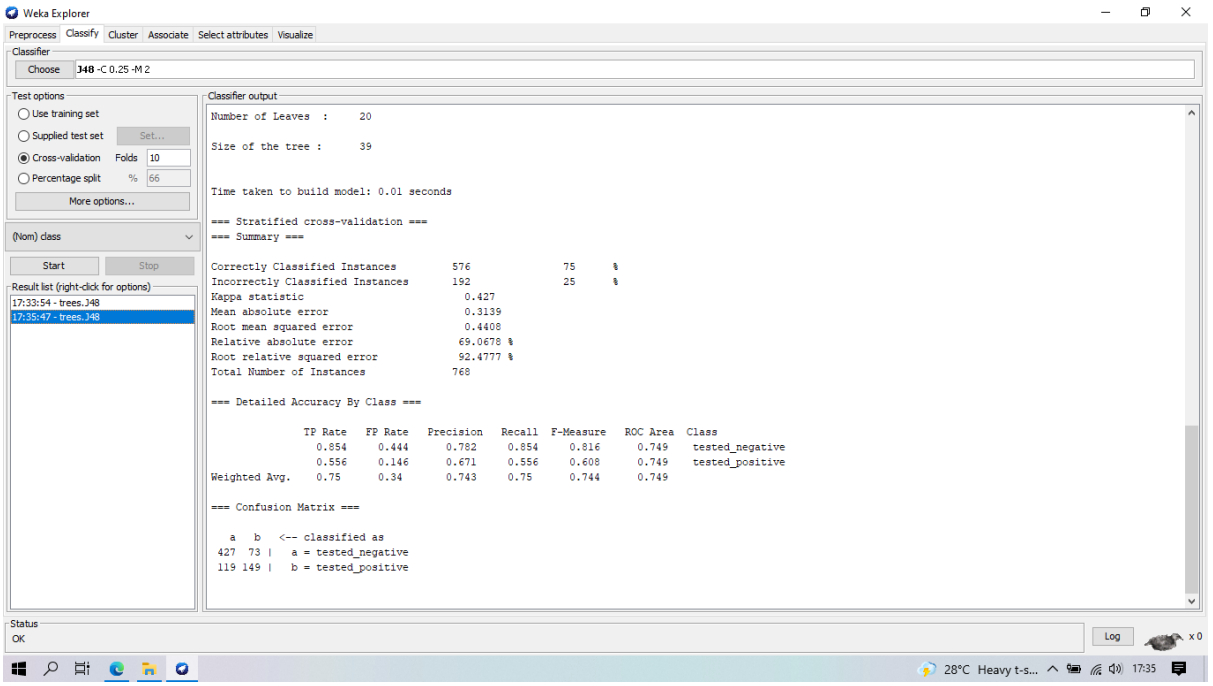
Load dataset diabetes.arff, kemudian classifier J48 untuk membuat model. Pada test option, pilih opsi cross-validation dengan nilai 10. Akurasi yang didapatkan adalah 73,8%



Tetap menggunakan classifier dan test option yang sama, lakukan perubahan pada random seed nya menjadi 2, 3, 4 dan seterusnya. Sebagai contoh, random cross-validation diubah menjadi 2.



Hasilnya mendapatkan akurasi sebesar 75%.



Setelah dicoba dengan random seed berbeda, dari 1 sampai 10, didapatkan data sebagai berikut beserta pembandingnya dengan metode repeated holdout.

Sample mean $\bar{x} = \frac{\sum x_i}{n}$

Variance $\sigma^2 = \frac{\sum (x_i - \bar{x})^2}{n-1}$

Standard deviation σ

	holdout (10%)	cross-validation (10-fold)
	75.3	73.8
	77.9	75.0
	80.5	75.5
	74.0	75.5
	71.4	74.4
	70.1	75.6
	79.2	73.6
	71.4	74.0
	80.5	74.5
	67.5	73.0
	$\bar{x} = 74.8$	$\bar{x} = 74.5$
	$\sigma = 4.6$	$\sigma = 0.9$

Menggunakan cross-validation lebih baik daripada repeated holdout. Cross-validation memperkecil standar deviasi dari akurasi, dalam artian rentang akurasinya semakin kecil.