**LAPORAN TUGAS KECIL II IF3170 INTELEGENSI BUATAN**

**Eksplorasi Waikato Environment for**

**Knowledge Analysis (WEKA)**



**Disusun oleh:**

**13514039 – Ari Pratama Zhorifiandi**

**13514086 – Azka Hanif Imtiyaz**

**TEKNIK INFORMATIKA**

**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA**

**INSTITUT TEKNOLOGI BANDUNG**

**DAFTAR ISI**

1. **DESKRIPSI PERSOALAN DAN TOOLS**
2. **IMPLEMENTASI WEKA API PADA JAVA**
3. **ANALISIS**
4. **KESIMPULAN**

**BAB I. DESKRIPSI PERSOALAN DAN TOOLS**

1. **DESKRIPSI PERSOALAN**

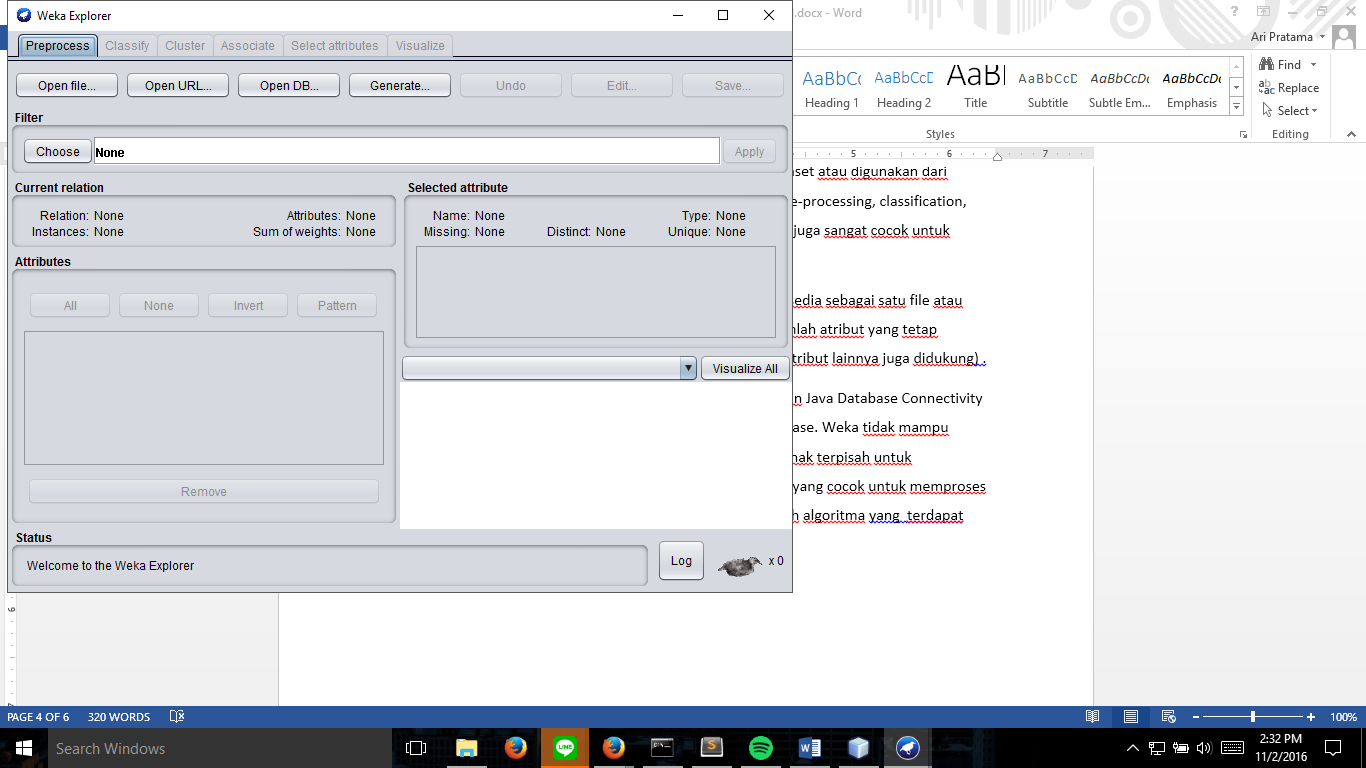
Program yang dibuat pada tugas ini adalah sebuah program yang mengimplementasikan API dari WEKA. Program tersebut:

1. Diimplementasikan dengan Bahasa Java
2. Dapat membaca dataset yang diberikan
3. Dapat mengaplikasikan filter yang mengubah tipe atribut, misalnya Discretize atau NumericToNominal.
4. Dapat melakukan pembelajaran dataset dengan skema 10-fold cross validation
5. Dapat melakukan pembelajaran dataset dengan skema full-training
6. Dapat menyimpan (save) model/hipotesis hasil pembelajaran ke sebuah file eksternal
7. Dapat membaca (read) model/hipotesis dari file eksternal
8. Dapat membuat instance baru sesuai masukan dari pengguna untuk setiap nilai atribut
9. Dapat melakukan klasifikasi dengan memanfaatkan model/hipotesis dan instance sesuai masukan pengguna pada poin 8.
10. Mengimplementasikan kelas dengan menggunakan iris.arff
11. **TOOLS YANG DIGUNAKAN**

Weka adalah aplikasi yang menyimpan koleksi algoritma machine learning untuk pengerjaan data mining. Algoritmanya dapat di terapkan langsung pada dataset atau digunakan dari program Java buatan sendiri. Weka memiliki tools untuk data pre-processing, classification, regression, clustering, association rules, dan visualization. Weka juga sangat cocok untuk membangun skema machine learning yang baru.

Semua teknik Weka didasarkan pada asumsi bahwa data tersedia sebagai satu file atau relasi yang flat, di mana setiap titik data digambarkan oleh sejumlah atribut yang tetap (biasanya, atribut numerik atau nominal, tetapi beberapa jenis atribut lainnya juga didukung) .

Weka juga menyediakan akses ke database SQL menggunakan Java Database Connectivity dan dapat memproses hasil yang dikembalikan oleh query database. Weka tidak mampu melakukan multi-relasional data mining, tetapi ada perangkat lunak terpisah untuk mengkonversi koleksi tabel database dihubungkan ke satu tabel yang cocok untuk memproses menggunakan Weka. Bidang lain yang saat ini tidak tercakup oleh algoritma yang terdapat dalam Weka adalah sequence modeling.



*Gambar 1 Aplikasi Desktop Weka pada Windows*

**BAB II. IMPLEMENTASI WEKA API PADA JAVA**

1. **SOURCE CODE**

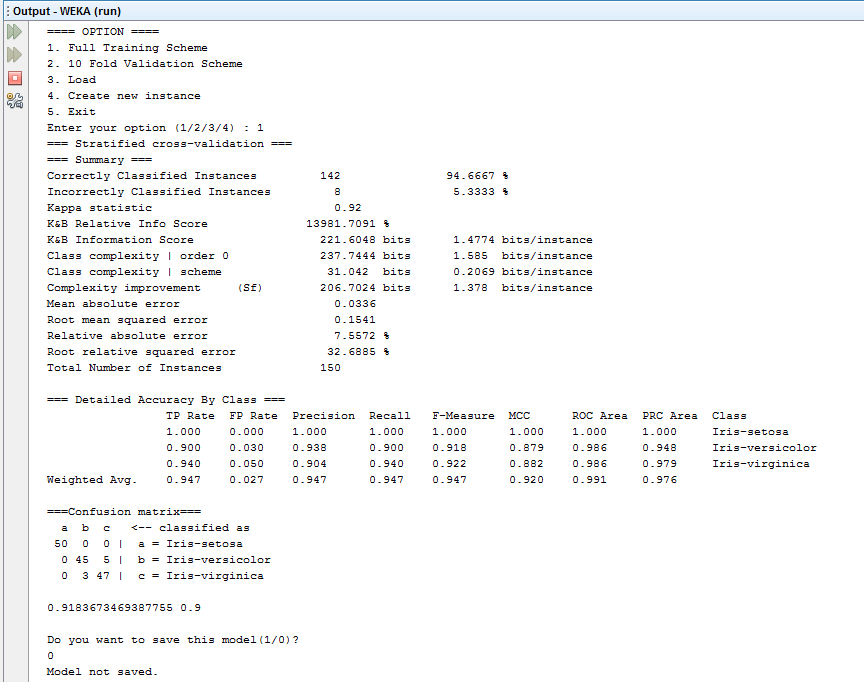
|  |
| --- |
| **/\***  **\* Made by Ari Pratama and Azka Hanif**  **\*/**  **package weka;**  **import java.io.BufferedReader;**  **import java.io.FileOutputStream;**  **import java.io.FileReader;**  **import java.io.ObjectOutputStream;**  **import java.util.Random;**  **import java.util.Scanner;**  **import weka.classifiers.Classifier;**  **import weka.core.Instances;**  **import weka.classifiers.Evaluation;**  **import weka.classifiers.bayes.NaiveBayes;**  **import weka.core.DenseInstance;**  **import weka.filters.Filter;**  **import weka.filters.supervised.attribute.\*;**  **public class WEKA {**    **void saveModel(Classifier C, String namaFile) throws Exception {**  **//SAVE**  **// serialize model**  **ObjectOutputStream oos = new ObjectOutputStream(**  **new FileOutputStream(namaFile));**  **oos.writeObject(C);**  **oos.flush();**  **oos.close();**  **}**  **public static void main(String[] args) throws Exception {**  **// IMPORT file \*.arff**  **WEKA w = new WEKA();**    **//Pilihan SKEMA**  **boolean validasi = false;**  **do {**  **//Read iris.arff**  **BufferedReader breader = null;**  **breader = new BufferedReader(new FileReader("src\\weka\\iris.arff"));**  **Instances inputTrain = new Instances (breader);**  **inputTrain.setClassIndex(inputTrain.numAttributes() -1);**  **breader.close();**  **//FILTER**  **Discretize filter = new Discretize();**  **filter.setInputFormat(inputTrain);**  **Instances outputTrain = Filter.useFilter(inputTrain,filter);**  **Evaluation eval = new Evaluation(outputTrain);**  **//ALGORITMA YANG DIGUNAKAN**  **NaiveBayes nB = new NaiveBayes();**    **//Menu**  **Scanner scan = new Scanner(System.in);**  **System.out.println("\n\n=================\n==== OPTION ====");**  **System.out.println("1. Full Training Scheme");**  **System.out.println("2. 10 Fold Validation Scheme");**  **System.out.println("3. Load");**  **System.out.println("4. Create new instance");**  **System.out.println("5. Exit");**  **System.out.print("Enter your option (1/2/3/4) : ");**  **int pilihan = scan.nextInt();**  **switch (pilihan) {**  **case 1:**  **{**  **nB.buildClassifier(outputTrain);**  **eval.evaluateModel(nB,outputTrain);**  **//OUTPUT**  **System.out.println(eval.toSummaryString("=== Stratified cross-validation ===\n" +"=== Summary ===",true));**  **System.out.println(eval.toClassDetailsString("=== Detailed Accuracy By Class ==="));**  **System.out.println(eval.toMatrixString("===Confusion matrix==="));**  **System.out.println(eval.fMeasure(1)+" "+eval.recall(1));**  **System.out.println("\nDo you want to save this model(1/0)? ");**  **int c = scan.nextInt();**  **if (c == 1 ){**  **System.out.print("Please enter your file name (\*.model) : ");**  **String infile = scan.next();**  **w.saveModel(nB,infile);**  **}**  **else {**  **System.out.print("Model not saved.");**  **} break;**  **}**  **case 2:**  **{**  **nB.buildClassifier(outputTrain);**  **eval.crossValidateModel(nB, outputTrain, 10, new Random(1));**  **//OUTPUT**  **System.out.println(eval.toSummaryString("=== Stratified cross-validation ===\n" +"=== Summary ===",true));**  **System.out.println(eval.toClassDetailsString("=== Detailed Accuracy By Class ==="));**  **System.out.println(eval.toMatrixString("===Confusion matrix==="));**  **System.out.println(eval.fMeasure(1)+" "+eval.recall(1));**  **System.out.println("\nDo you want to save this model(1/0)? ");**  **int c = scan.nextInt();**  **if (c == 1 ){**  **System.out.print("Please enter your file name (\*.model) : ");**  **String infile = scan.next();**  **w.saveModel(nB,infile);**  **}**  **else {**  **System.out.print("Model not saved.");**  **} break;**  **}**  **case 3:**  **//LOAD**  **// deserialize model**  **System.out.print("Please enter the file name : ");**  **String namaFile = scan.next();**  **Classifier cls = (Classifier) weka.core.SerializationHelper.read(namaFile);**  **eval.crossValidateModel(cls, outputTrain, 10, new Random(1));**  **System.out.println(eval.toSummaryString("=== Stratified cross-validation ===\n" +"=== Summary ===",true));**  **System.out.println(eval.toClassDetailsString("=== Detailed Accuracy By Class ==="));**  **System.out.println(eval.toMatrixString("===Confusion matrix==="));**  **System.out.println(eval.fMeasure(1)+" "+eval.recall(1));**  **break;**  **case 4:**  **System.out.println();**  **//ADD New Instance**  **nB.buildClassifier(inputTrain);**  **//Copy attributes from instances**  **DenseInstance buffer = new DenseInstance(inputTrain.firstInstance());**  **//Initialization**  **buffer.setDataset(inputTrain);**  **buffer.setMissing(inputTrain.classIndex());**  **//Input**  **for (int i = 0; i < inputTrain.classIndex(); i++){**  **System.out.print("Enter the value for " + buffer.attribute(i).name() + ": ");**  **double val = scan.nextDouble();**  **buffer.setValue(i, val);**  **}**  **//Classify**  **double res = nB.classifyInstance(buffer);**  **buffer.setValue(inputTrain.classIndex(), res);**  **inputTrain.add(buffer);**  **System.out.println("Class: " + buffer.stringValue(inputTrain.classIndex()));**  **break;**  **case 5:**  **validasi = true;**  **break;**  **default:**  **System.out.println("Wrong input!");**  **break;**  **}**  **}**  **while (!validasi);**  **}**  **}** |

***full source code: https://github.com/zhorifiandi/WEKAstarter***

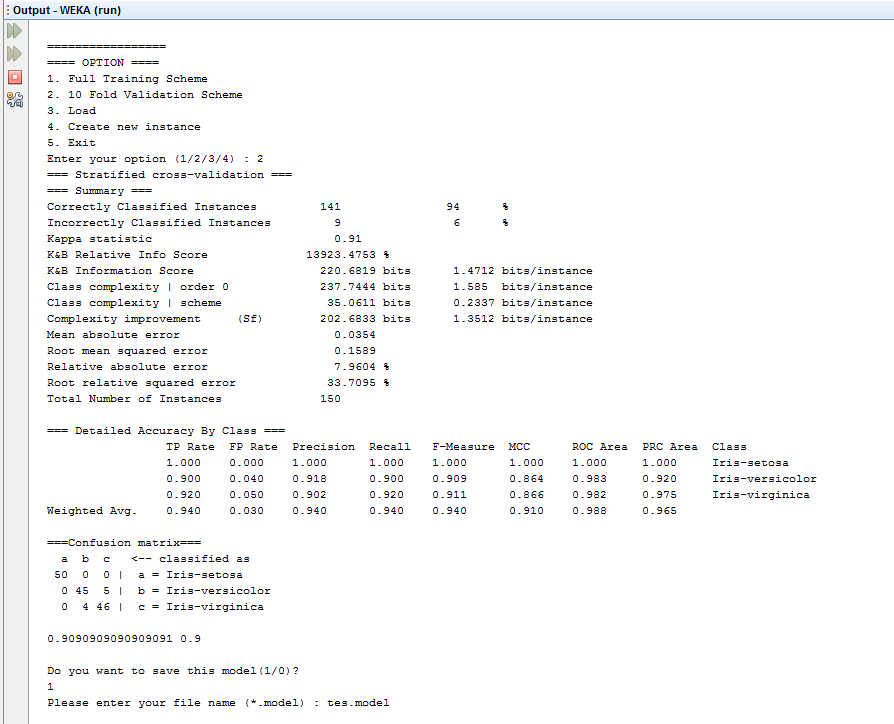
1. **HASIL IMPLEMENTASI**

Untuk pengujian hasil implementasi, dataset yang digunakan adalah iris.arff

**Pembelajaran dengan Skema Full Training**

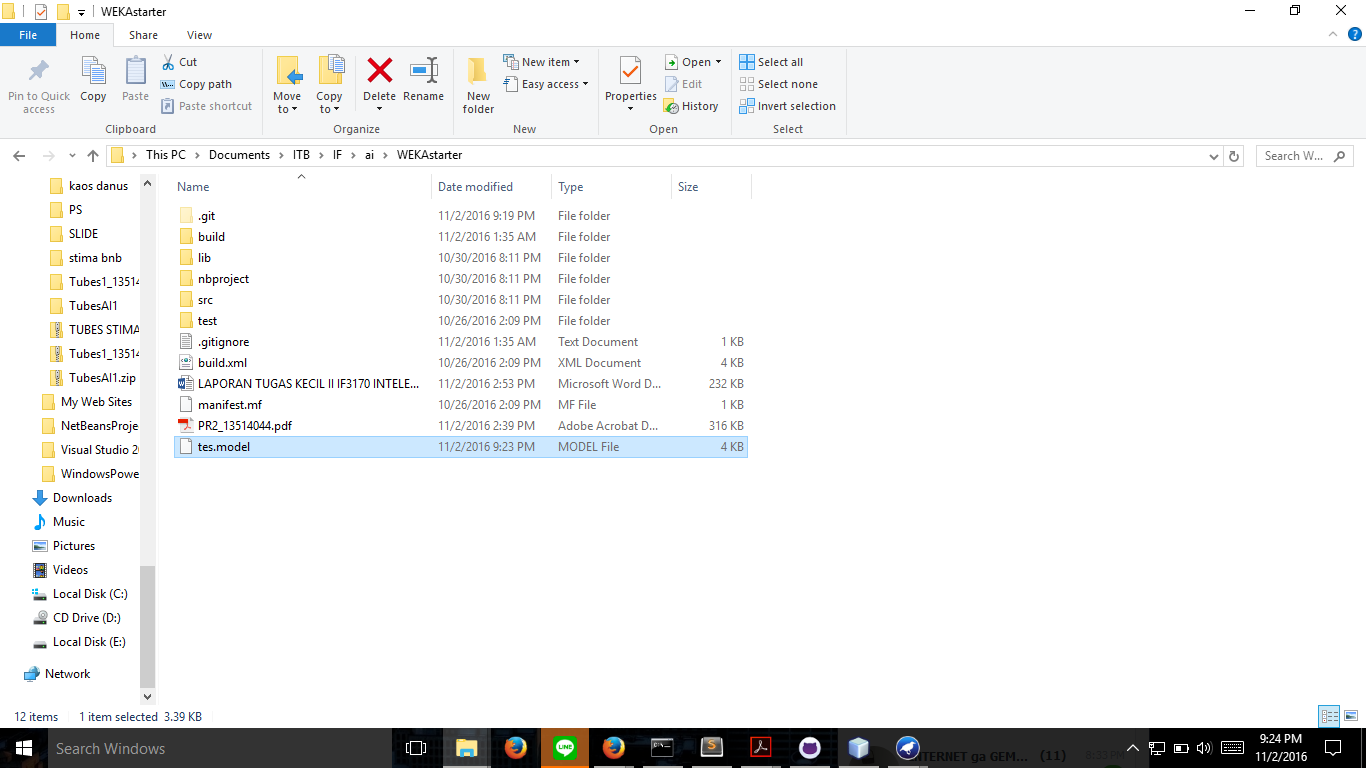


**Pembelajaran dengan Skema 10 Fold Validation**

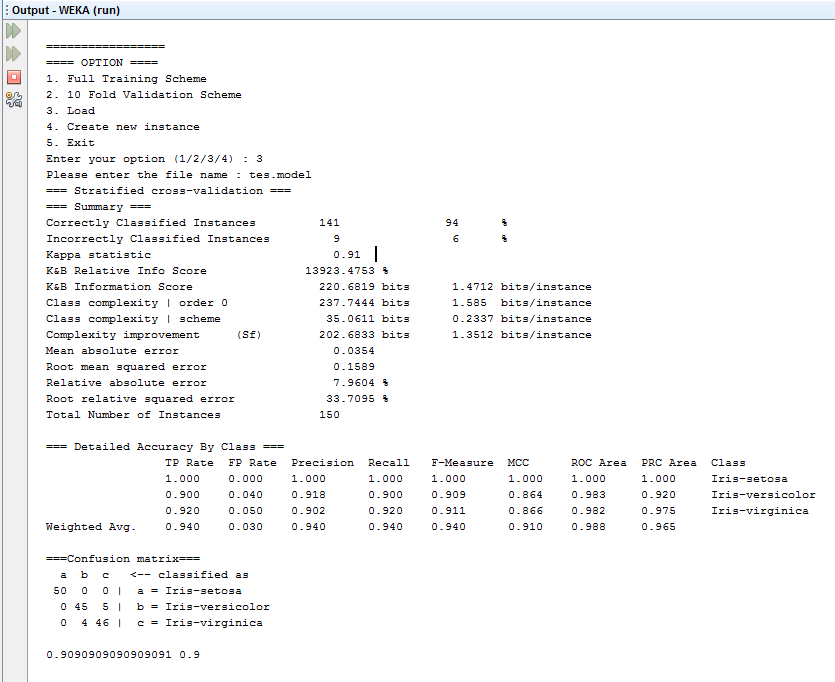


**Menyimpan Model pada File Eksternal**

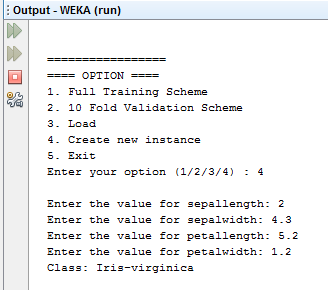




**Membaca Model dari File Eksternal**



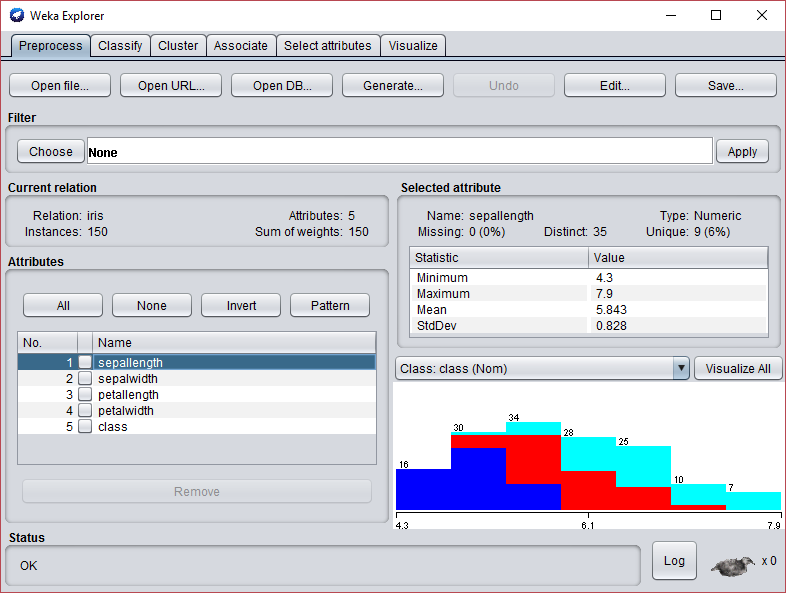
**Membuat Instance Baru sesuai Masukan dari Pengguna**



**Klasifikasi Instance Baru**

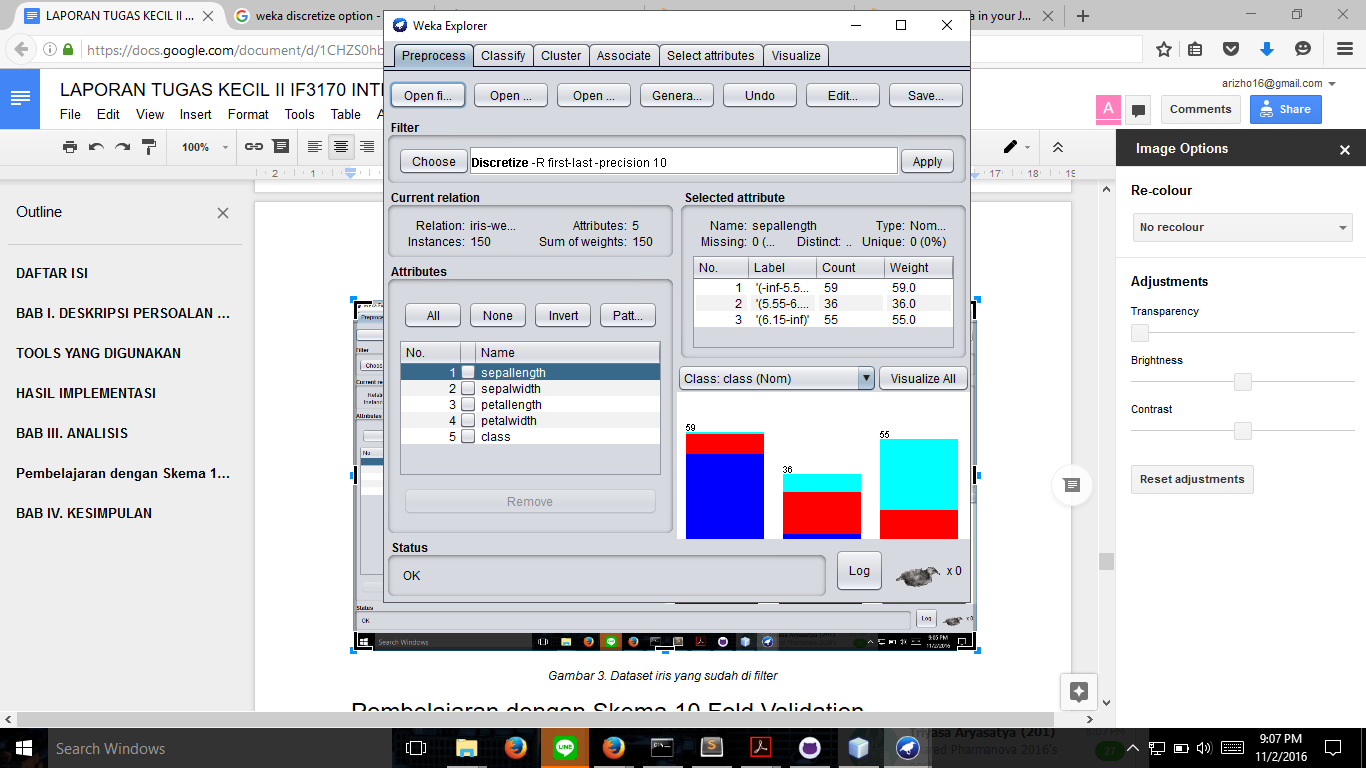


**BAB III. ANALISIS**



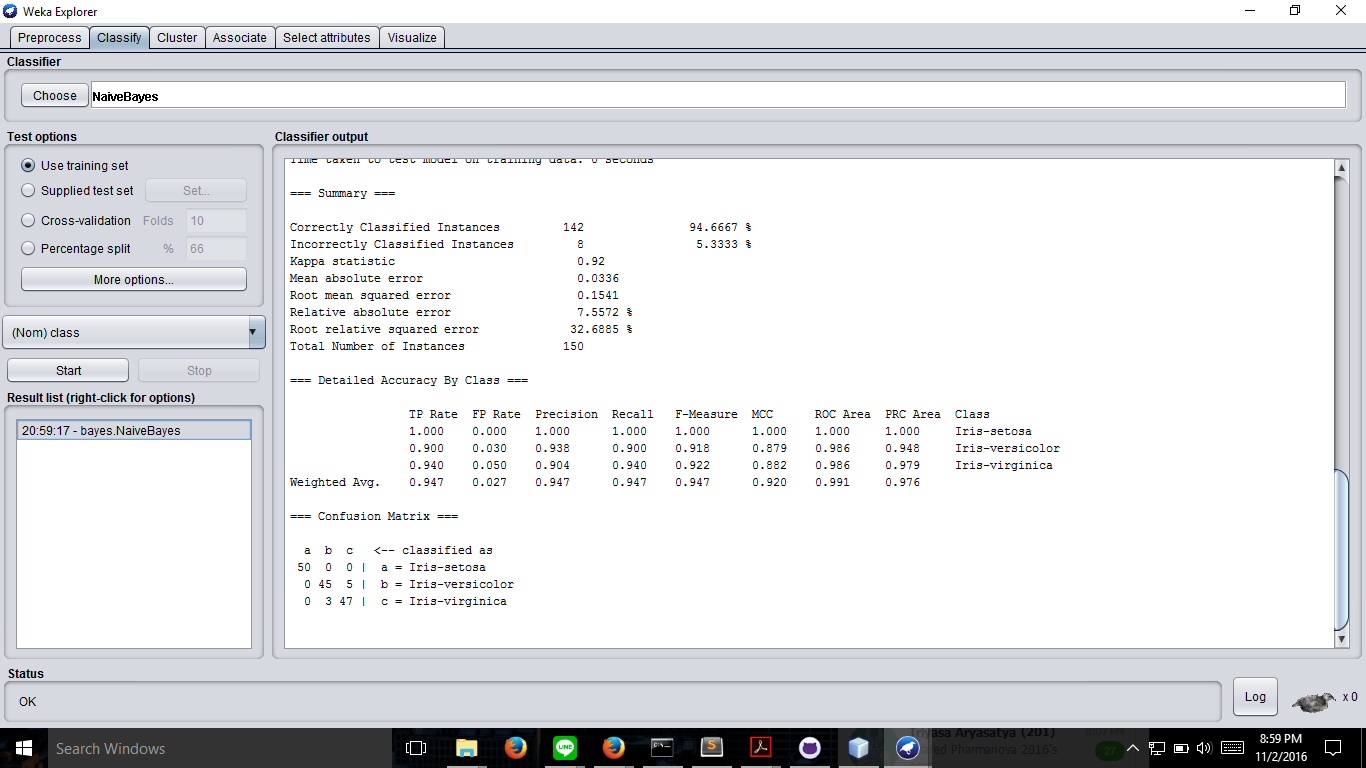
*Gambar 2. iris.arff yang baru dibuka pada GUI Weka*

Untuk perbandingan, kita akan memberikan filter discretize pada dataset iris. Algoritma classifier yang akan digunakan adalah Naive Bayes, sama seperti implementasi program yang telah kami buat.

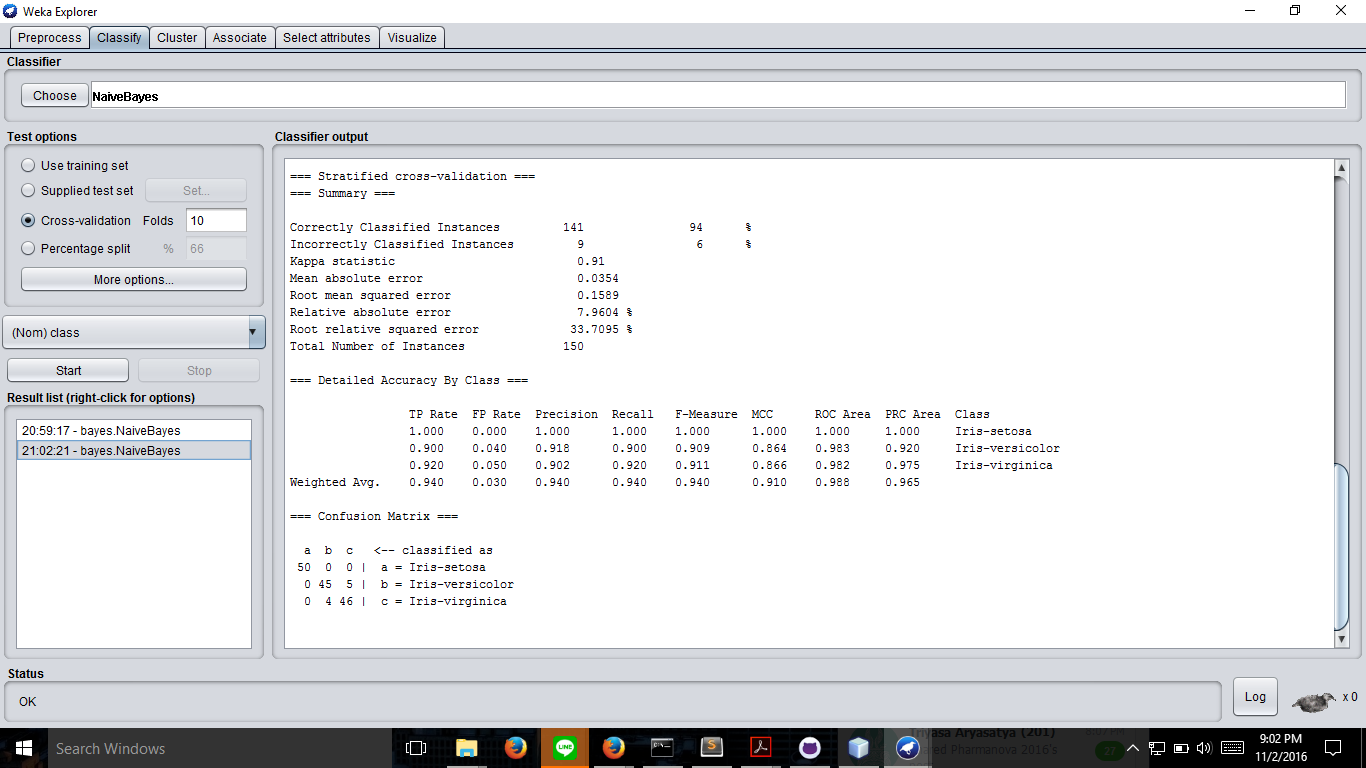


*Gambar 3. Dataset iris yang sudah di filter*

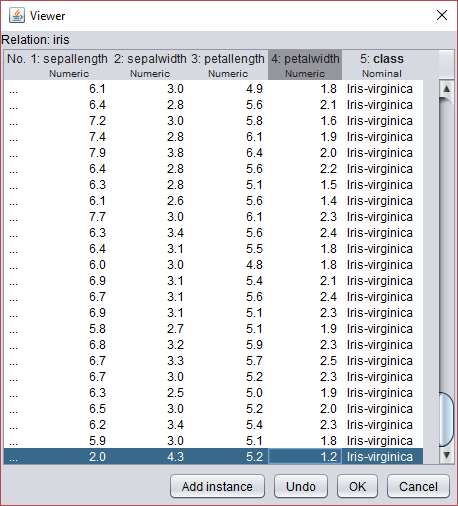
**Pembelajaran dengan Skema Full-Training**



**Pembelajaran dengan Skema 10 Fold Validation**



**Penambahan Instance Baru tanpa Kelas**



**BAB IV. KESIMPULAN**

* Seperti yang dapat dilihat pada hasil implementasi dan analisis pada GUI, hasil implementasi Program Java kami dengan Program Weka GUI adalah sama.
* Implementasi API ini dapat digunakan untuk pembelajaran dataset pada aplikasi yang dibuat secara langsung tanpa perlu melakukan pembelajaran manual menggunakan GUI, sehingga mempermudah programmer nantinya.