**LAPORAN TUGAS BESAR**

**KECERDASAN BUATAN**

****

Disusun oleh :

Ubassy Abdillah 1301148282

M. Zakie Faiz Rahiemy 1301144422

Raja Ryan Pradana 1301140302

Akhmad Fadilah Ramadhan 1301140222

Adrian Gusti Nurcahyo 1301140092

**UNIVERSITAS TELKOM**

**BANDUNG**

**2016**

**DAFTAR ISI**

[1. Deskripsi Masalah 3](#_Toc468271517)

[2. Proses Pengolahan 3](#_Toc468271518)

[2.1 Sintaks keseluruhan dari program : 3](#_Toc468271519)

[2.1. Penjelasan Setiap Fungsi 8](#_Toc468271520)

[3. Metode yang digunakan 9](#_Toc468271521)

[4. Screenshot dari program yang dibuat 9](#_Toc468271522)

[5. Sumber referensi 9](#_Toc468271523)

# **Deskripsi Masalah**

Kondisi awal dari permasalahan di tugas besar Kecerdasa Buatan adalah kami diberikan 90.000 baris data, dimana setiap baris tersebut memiliki sepuluh kolom atribut yang memengaruhi nilai kelas yang ada di kolom kesebelas, atau dengan kata lain kami diberikan 990.000 sel nilai yang harus dianalisis. Yang harus kami lakukan adalah, menentukan aturan dari data yang telah diberikan, apakah suatu baris dengan atribut-atribut tertentu termasuk ke kelas 0 atau justru masuk ke kelas 1.

Untuk mengklasifikasi baris data tersebut, kami diperbolehkan untuk menggunakan berbagai metode namun yang kami gunakan adalah metode Fuzzy Logic, atau istilah lainnya adalah logika samar.

# **Proses Pengolahan**

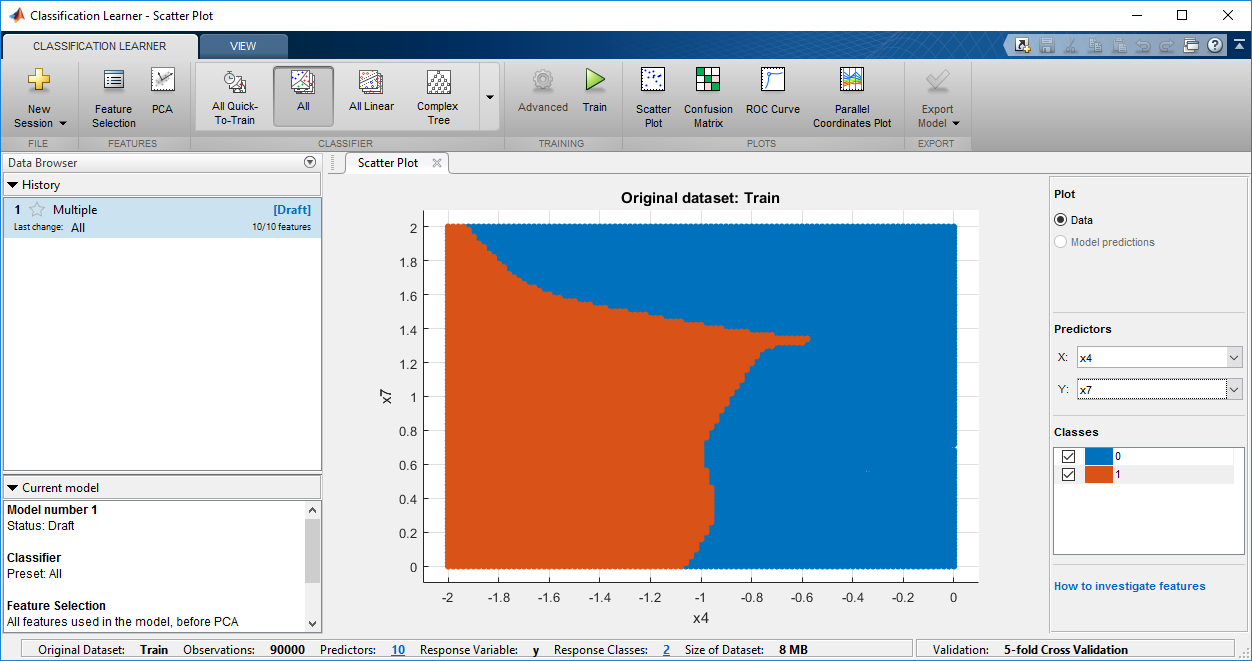
Dalam mengerjakan tugas besar, pertama-tama kami menganalisis data dari file Train.xlsx. Setelah menemukan data yang kami butuhkan, dalam implementasinya kami menggunakan bahasa java dan menggunakan netbeans sebagai compiler.

## **2.1 Sintaks keseluruhan dari program :**

|  |
| --- |
| package ai; |
|  |
| import java.io.File; |
| import java.io.FileInputStream; |
| import java.io.FileNotFoundException; |
| import java.io.IOException; |
| import org.apache.poi.ss.usermodel.Cell; |
| import org.apache.poi.ss.usermodel.FormulaEvaluator; |
| import org.apache.poi.ss.usermodel.Row; |
| import org.apache.poi.xssf.usermodel.XSSFSheet; |
| import org.apache.poi.xssf.usermodel.XSSFWorkbook; |
|  |
| /\*\* |
| \* |
| \* @author Novak |
| \*/ |
| public class Ai { |
|  |
| double x4[] = new double[90001]; |
| String x4s[] = new String[90001]; |
| double x7[] = new double[90001]; |
| double hasil[] = new double[90001]; |
| String x7s[] = new String[90001]; |
| int total = 0; |
| double yee[] = new double[90001]; |
| /\*\* |
| \* @param args the command line arguments |
| \*/ |
|  |
| public void convert() { |
| for (int i = 1; i < total; i++) { |
| x4[i] = (100/2)\*(x4[i]-0)+100; |
| x7[i] = (100/2)\*(x7[i]-2)+100; |
| } |
| } |
|  |
| public void comparison() { |
| double bener=0; |
| for (int i = 1; i < total; i++) { |
| if (hasil[i]==yee[i]) { |
| bener++; |
| } |
| } |
| System.out.println(bener); |
| double akurasi = (bener/90000)\*100; |
| System.out.println("Akurasi : "+akurasi+" %"); |
| } |
|  |
| public void inferensi(){ |
| for (int i = 1; i < total; i++) { |
| if (x4s[i].equals("LOWLOW")) { |
| yee[i]=1; |
| } else if (x4s[i].equals("LOW") && x7s[i].equals("MEGAHIGH")) { |
| yee[i]=0; |
| } else if (x4s[i].equals("LOW")) { |
| yee[i]=1; |
| } else if ((x4s[i].equals("MID") && x7s[i].equals("MEGAHIGH")) || (x4s[i].equals("MID") && x7s[i].equals("HIGHHIGH"))) { |
| yee[i]=0; |
| } else if (x4s[i].equals("MID")) { |
| yee[i]=1; |
| } else if (x4s[i].equals("HIGH") && x7s[i].equals("HIGH")) { |
| yee[i]=1; |
| } else if (x4s[i].equals("HIGH")) { |
| yee[i]=0; |
| } else if (x4s[i].equals("HIGHHIGH")) { |
| yee[i]=0; |
| } else if (x4s[i].equals("MEGAHIGH")) { |
| yee[i]=0; |
| } |
| } |
| } |
|  |
| public void inferensi2(){ |
| for (int i = 1; i < total; i++) { |
| if (x4s[i].equals("LOWLOW")) { |
| yee[i]=1; |
| } else if (x4s[i].equals("LOW") && x7s[i].equals("MEGAHIGH")) { |
| yee[i]=0; |
| } else if (x4s[i].equals("LOW")) { |
| yee[i]=1; |
| } else if ((x4s[i].equals("MID") && x7s[i].equals("HIGHHIGH")) || (x4s[i].equals("MID") && x7s[i].equals("HIGHHIGH"))) { |
| yee[i]=0; |
| } else if (x4s[i].equals("MID") && x7s[i].equals("MEGAHIGH")) { |
| yee[i]=0; |
| } else if (x4s[i].equals("MID")) { |
| yee[i]=1; |
| } else if (x4s[i].equals("HIGH")) { |
| yee[i]=0; |
| } else if (x4s[i].equals("HIGHHIGH")) { |
| yee[i]=0; |
| } else if (x4s[i].equals("MEGAHIGH")) { |
| yee[i]=0; |
| } |
| } |
| } |
|  |
| public void inferensi3(){ |
| for (int i = 1; i < total; i++) { |
| if (x4s[i].equals("LOWLOW")) { |
| yee[i]=1; |
| } else if (x4s[i].equals("LOW") && x7s[i].equals("MEGAHIGH")) { |
| yee[i]=0; |
| } else if (x4s[i].equals("LOW")) { |
| yee[i]=1; |
| } else if (x4s[i].equals("MID") && x7s[i].equals("LOWLOW")) { |
| yee[i]=1; |
| } else if ((x4s[i].equals("MID") && x7s[i].equals("MEGAHIGH")) || (x4s[i].equals("MID") && x7s[i].equals("HIGHHIGH"))) { |
| yee[i]=0; |
| } else if (x4s[i].equals("MID")) { |
| yee[i]=1; |
| } else if (x4s[i].equals("HIGH")) { |
| yee[i]=0; |
| } else if (x4s[i].equals("HIGHHIGH")) { |
| yee[i]=0;  } else if (x4s[i].equals("MEGAHIGH")) {  yee[i]=0;  }  }  }    public void fuzzy() {  double lowlow = 10;  double low = 31;  double mid = 52;  double high = 73;  double highhigh = 79;  double megahigh = 100;  for (int i = 1; i < total; i++) {  if (x4[i]<=lowlow) {  x4s[i] = "LOWLOW";  } else if (x4[i]<=low) {  x4s[i] = "LOW";  } else if (x4[i]<=mid) {  x4s[i] = "MID";  } else if (x4[i]<=high) {  x4s[i] = "HIGH";  } else if (x4[i]<=highhigh) {  x4s[i] = "HIGHHIGH";  } else {  x4s[i] = "MEGAHIGH";  }  if (x7[i]<=lowlow) {  x7s[i] = "LOWLOW";  } else if (x7[i]<=low) {  x7s[i] = "LOW";  } else if (x7[i]<=mid) {  x7s[i] = "MID";  } else if (x7[i]<=high) {  x7s[i] = "HIGH";  } else if (x7[i]<=highhigh) {  x7s[i] = "HIGHHIGH";  } else {  x7s[i] = "MEGAHIGH";  }  }  }    public void isi() throws FileNotFoundException, IOException {  File f = new File("Train.xlsx");  FileInputStream file = new FileInputStream(f);  XSSFWorkbook wb = new XSSFWorkbook(file);  XSSFSheet sheet = wb.getSheetAt(0);  FormulaEvaluator forlula = wb.getCreationHelper().createFormulaEvaluator();  for (Row row : sheet) {  for (Cell cell : row) {  if (cell.getColumnIndex() == 4) {  switch(forlula.evaluateInCell(cell).getCellType()){  case (Cell.CELL\_TYPE\_NUMERIC):  x4[total] = cell.getNumericCellValue();  break;  }  } else if (cell.getColumnIndex() == 7) {  switch(forlula.evaluateInCell(cell).getCellType()){  case (Cell.CELL\_TYPE\_NUMERIC):  x7[total] = cell.getNumericCellValue();  break;  }  } else if (cell.getColumnIndex() == 11) {  switch(forlula.evaluateInCell(cell).getCellType()){  case (Cell.CELL\_TYPE\_NUMERIC):  hasil[total] = cell.getNumericCellValue();  break;  }  }  }  total++;  }  }    public static void main(String[] args) throws FileNotFoundException, IOException {  // TODO code application logic her  Ai a = new Ai();  a.isi();  a.convert();  a.fuzzy();  a.inferensi3();  for (int i = 1; i < a.total; i++) {  System.out.print(i+" ");  System.out.println(a.x4s[i]+"\t\t"+a.x7s[i]+"\t\t\t"+a.hasil[i]+"\t"+a.yee[i]);  }  a.comparison();  }  } |

## **Penjelasan Setiap Fungsi**

* + 1. **Method isi**

Method isi bertugas mengambil file train.xlsx yang berisi data-data yang ingin di olah kedalam bentuk array. Data yang diambil hanya data x4, x7, dan y karena dari hasil pesebaran data hanya x4 dan x7 yang tidak beririsan.

* + 1. **Method convert**

Fungsi ini mengubah seluruh nilai x4 dan x7 menjadi skala 0-100.

* + 1. **Method fuzzy**

Fungsi ini digunakan untuk mendefinisikan kelas x4 dan x7.

* + 1. **Method inferensi3**

Fungsi ini digunakan untuk menentukan class mana yang masuk ke dalam kelompok 0 dan class mana yang masuk ke dalam kelompok 1. Hasil tersebut ditampung kedalam array baru.

* + 1. **Method comparison**

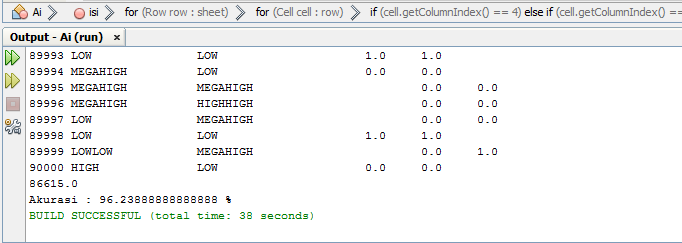
Fungsi ini digunakan untuk menbandingkan hasil yang berada dalam train.xlsx dengan hasil yang telah dibuat di inferensi3. Hasil perbandingan ditampilkan dalam bentuk persen (%).

# **Metode yang digunakan**

Dengan menggunakan metode logika samar, pertama-tama kami harus menentukan kolom mana saja yang mempengaruhi nilai kelas dari suatu baris data beserta batas-batasnya. Setelah melakukan analisis, kami menghasilkan kesimpulan bahwa kolom yang berpengaruh adalah kolom x4 dan x7.

Setelah menemukan batas-batas tersebut, kami langsung melakukan implementasi pada bahasa pemrograman Java.

# **Screenshot dari program yang dibuat**



# **Sumber referensi**

Suyanto, 2014. *Artificial Intelligence : Searching, Reasoning, Planning, dan Learning.* Revisi kedua . Bandung: Informatika.