LAPORAN RESMI
PEKAN 7
MACHINE LEARNING

DISUSUN OLEH :
FABYAN KINDARYA
2110191024

BAYESIAN NUMBER

MAIN.JAVA

```java
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.util.List;
import java.util.ArrayList;
import java.util.Arrays;

public class Main {
    public static void main(String[] args) {
        List<List<String>> data_train = readData("src\\number-training.csv");
        List<List<String>> data_test = readData("src\\number-testing.csv");

        List<String> results = bayesian_classification(data_train, data_test);

        System.out.println("Perbandingan Label :");
        System.out.println("Tes\t|\tHasil");
        int data_test_label_index = data_test.get(0).size() - 1;
        for (int i = 0; i < data_test.size(); i++) {
            System.out.println(data_test.get(i).get(data_test_label_index) + "\t|\t" + results.get(i));
        }

        double error = error_results(data_test, results, true);
        System.out.println("\nPersentase Error : " + error + "%");
    }

    public static List<List<String>> readData(String url) {
        final String COMMA_DELIMITER = ",";
        List<List<String>> records = new ArrayList<>();

        try (BufferedReader br = new BufferedReader(new FileReader(url))) {
            String line;

            while ((line = br.readLine()) != null) {
                String[] values = line.split(COMMA_DELIMITER);
                records.add(Arrays.asList(values));
            }
        } catch (IOException e) {
            e.printStackTrace();
        }

        return records;
    }
```

MAIN.JAVA

```
public static List<String> bayesian_classification(List<List<String>> data_train, List<List<String>> data_test) {
       List<String> labels = get_labels(data_train);
       List<String> result = new ArrayList<>();

       for (int i = 0; i < data_test.size(); i++) {
           List<Double> probabilities = new ArrayList<>();

           for (int j = 0; j < labels.size(); j++) {
               double probability = get_probability(data_train, labels.get(j), data_test.get(i));
               probabilities.add(probability);
           }
           int resultIndex = get_biggest_probability_index(probabilities);
           result.add(probabilities.get(resultIndex) != 0 ? labels.get(resultIndex) : "Tidak ada hasil");
       }
       return result;
   }

   public static List<String> get_labels(List<List<String>> data_train) {
       List<String> labels = new ArrayList<>();
       int label_index = data_train.get(0).size() - 1;

       for (List<String> line : data_train) {
           String trainLabel = line.get(label_index);

           if (labels.isEmpty()) {
               labels.add(trainLabel);
           } else {
               boolean labelExist = false;

               for (String label : labels) {
                   if (label.equals(trainLabel)) {
                       labelExist = true;
                       break;
                   }
               }
               if (!labelExist) {
                   labels.add(trainLabel);
               }
           }
       }
       return labels;
   }
```

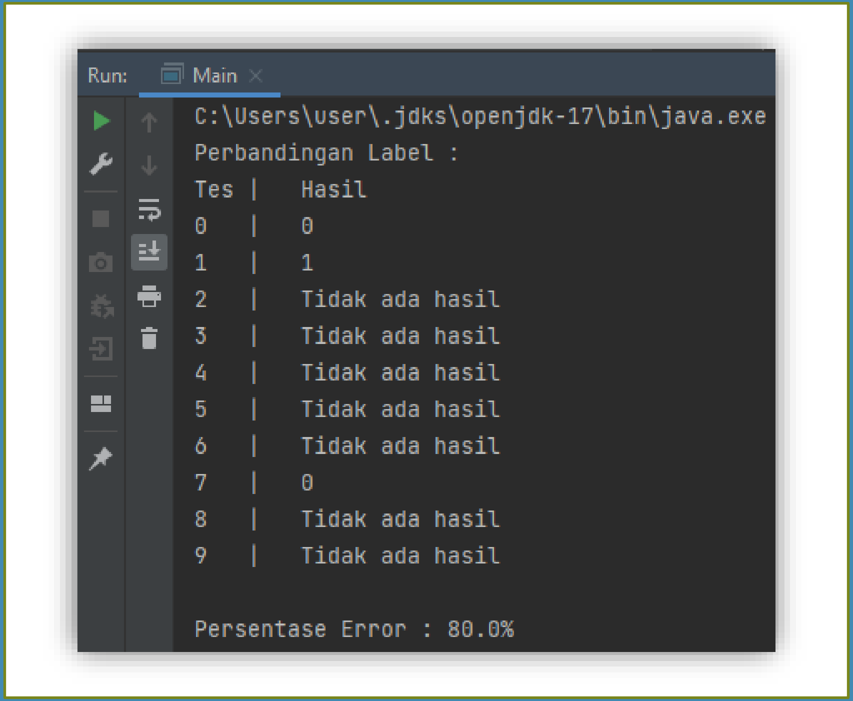
MAIN.JAVA

```
public static double get_probability(List<List<String>> data_train, String label, List<String> params) {
       if (params == null) { // Mencari persentase banyaknya label per keseluruhan data
           int count = 0;
           int label_index = data_train.get(0).size() - 1;
           for (List<String> line : data_train) {
               if (line.get(label_index).equals(label)) {
                   count++;
               }
           }

           return (double) count / (double) data_train.size();
       } else { // Mencari persentase banyaknya data dengan param tertentu terhadap sebuah label
           // per banyaknya label
           double label_probability = get_probability(data_train, label, null);
           int label_index = data_train.get(0).size() - 1;

           double param_probability = 1;
           for (int i = 0; i < params.size() - 1; i++) {
               int param_found = 0;
               int label_found = 0;

               for (List<String> train_row : data_train) {
                   if (train_row.get(label_index).equals(label)) {
                       if (train_row.get(i).equals(params.get(i)) && train_row.get(label_index).equals(label)) {
                           param_found++;
                       }
                       label_found++;
                   }
               }

               param_probability *= ((double) param_found / (double) label_found);
           }

           return param_probability * label_probability;
       }
   }
```

MAIN.JAVA

```java
 public static int get_biggest_probability_index(List<Double> probabilities) {
        int index = 0;

        for (int i = 0; i < probabilities.size(); i++) {
            if (probabilities.get(index) < probabilities.get(i)) {
                index = i;
            }
        }

        return index;
    }

    public static double error_results(List<List<String>> data_test, List<String> results, boolean percentage_return) {
        double error = 0;
        int data_test_label_index = data_test.get(0).size() - 1;

        for (int i = 0; i < data_test.size(); i++) {
            if (!data_test.get(i).get(data_test_label_index).equals(results.get(i))) {
                error++;
            }
        }

        return percentage_return ? error * 100 / results.size() : error / results.size();
    }
}
```

OUTPUT



Run: Main

```
C:\Users\user\.jdks\openjdk-17\bin\java.exe
Perbandingan Label :
Tes |   Hasil
0   |   0
1   |   1
2   |   Tidak ada hasil
3   |   Tidak ada hasil
4   |   Tidak ada hasil
5   |   Tidak ada hasil
6   |   Tidak ada hasil
7   |   0
8   |   Tidak ada hasil
9   |   Tidak ada hasil

Persentase Error : 80.0%
```