

## CHAPTER 5

### IMPLEMENTATION AND TESTING

#### 5.1 Implementation

This project is implemented in Java Language. This sub-chapter, will explain how to use the program and how the program works. In this project, the program has 4 feature, there are message detection, news detection(by input link), trainer message, trainer news.

##### 5.1.1 Message

Message help us to detect news without input the link of news. This feature is very useful when we want to detect news from media social based on chat messenger.

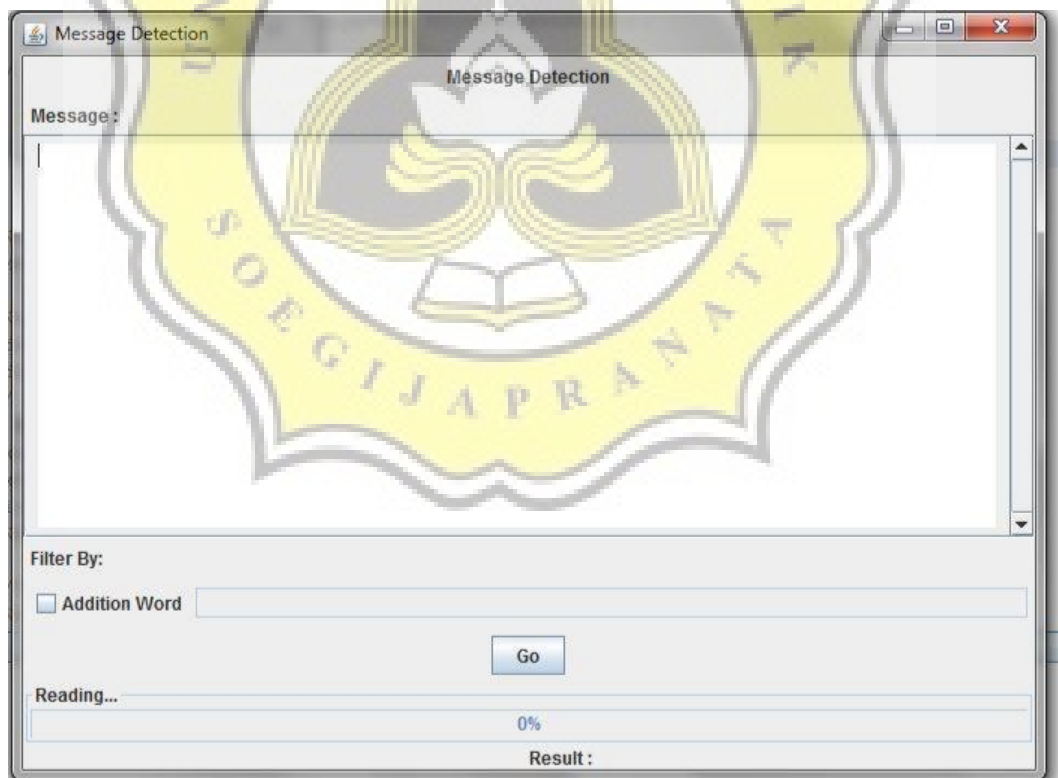


Illustration 5.1: GUI Message Classification

The code of Message page is shows in figure below.

```

1.
2. if (event.getActionCommand()=="Go" && cek() ==true ) {
3.
4.     try {
5.
6.         Message a=new Message(message.getText().toString());
7.         CaseFoldingMessage b=new CaseFoldingMessage();
8.         TokenizeMessage c=new TokenizeMessage();
9.         StemmingMessage d=new StemmingMessage();
10.        FilteringMessage e=new FilteringMessage();
11.        TextMiningMessage f=new TextMiningMessage();
12.        TextMiningMessageTrainer g=new TextMiningMessageTrainer();
13.        NaiveBayesMessage h=new NaiveBayesMessage();
14.        if(!(txtword.getText().toString()).equals("")){
15.            AdditionWordM i=new
16.            resultadd.setText("Result of addition
17.            word:"+i.cek(txtword.getText().toString()));
18.
19.        }
20.
21.        result.setText(h.getResult());
22.    } catch (SQLException ex) {
23.    } catch (IOException ex) {
24.    }
25.    }

```

In code above, shows that there are some process before classify news/message into hoax or real. Firstly program call Message class which is used to get data from user input and save into database. After it program call CaseFolding class which is used to change all content into low case in order to make data more easily to be proceed. Program call TokenizeMessage to parsing all content of word. Stemming class is used for get stem word of content, and Filtering class is used for delete all stop word in order to make analysis data more accurate. TextMiningMessage class is used for calculate TF(Term Frequency) of message.

On line 14-19, program will check whether user input addition word on txtword(text field), if user input text then program will call AdditionWordM. AdditionWordM is used for searching word in news document.

Last, NaiveBayesMessage class is used to calculate probability of hoax or real news.

TextMiningMessageTrainer is used for calculate TF-IDF value from training data . TF-IDF value from TextMiningMessageTrainer is used to calculate like hood in Naive Bayes algorithm.



```

26. public double calculateTFIDF(String i,int j,HashMap
    idfword)
27. {
28.     double temp=0;
29.     Object df=0;
30.     double nilai=0;
31.     int hasil = 0;
32.     for (Map.Entry<Integer,
        List<Stopword_Message_Trainer_Table>> entry :
        messageById.entrySet())
33.     {
34.         Integer key = entry.getKey();
35.         List<Stopword_Message_Trainer_Table> values = entry.getValue();
36.         if (key == j){
37.             for(Stopword_Message_Trainer_Table d : values){
38.                 if(d.getKonten() != null && d.getKonten().equals(i))
39.                 {
40.                     hasil=hasil+1;
41.                 }
42.             }
43.         }
44.         if(idfword.containsKey(i)){
45.             df= idfword.get(i);
46.             nilai=(double)df;
47.
48.             temp=hasil*nilai;
49.         }
50.         return temp;
51.     }

```

In code above, shows that function to calculateTFIDF. This function will return tfidf of each word and id of news. Variable i means the word, j means the id of the word and idfword is a lot of idf value that already being calculated . On line 32 , there is a iteration of word to help program get the right word in order to

iterate line 36. Line 36 is an iteration to calculate document frequency. Lastly on line 48 idf value will be multiples with document frequency of each id.

In NaiveBayesMessage, there are some value that needed, which is probability hoax news and probability of real news. If probability of hoax is bigger than real probability, then the news will be categorize as a hoax news.

```

52. public String getResult(){
53.     String hasilnya="";
54.     double real=0;
55.     double hoax=0;
56.     real=getProbabilityReal();
57.     hoax=getProbabilityHoax();
58.     if(real>hoax){
59. hasilnya="real dengan probabilitas hoax :"+hoax +" dan
    probabilitas real :"+real;
60.     } else hasilnya="hoax dengan probabilitas
    hoax :"+hoax +" dan probabilitas real :"+real;
61.     return hasilnya;
62.     }

```

Each of probability need prior value and like hood value.

```

63. public double getProbabilityReal(){
64.     priorreal=priorRealNews();
65.     prior_content_real=getTotalContentPriorReal();
66.     real_probability=prior_content_real*priorreal;
67.     return real_probability;
68. }

```

Here is a function to return prior probability value. This function need amount of real training document and amount of all training data.

```

69. public double priorRealNews(){
70.     priorreal=(double) (jumlahdokumenreal)/
    (jumlahdokumen);
71.     return priorreal;
72. }

```

Meanwhile, like hood value need some parameter such as total of word on hoax class (wcreal), value of tf idf . In specific word on hoax/real class(angka) and total word(W). After it all of like hood value will be put on like hood real.

```

73.   private void likelihoodContentRealAllTheme() {
74.
75.       getWCREal(fileName3);
76.       for(String string: wreal.keySet()){
77.           double angka=0d;
78.           angka=wreal.get(string);
79.           likelihoodR= ((angka +1)/(wcreal+W));
80.
81.           if(!likelihoodreal.containsKey(string)){
82.               likelihoodreal.put(string, likelihoodR);
83.           }
84.
85.       }
86.
87.   }

```



### 5.1.2 News

News help us to detect news by input the link of news. This feature is very useful when we want to detect news based on specific theme of news. The code of News page is shows in figure below. Basically News and Message menu have same logic on calculate Naive Bayes Algorithm. The difference between them only on data source. On Message all of training data will be calculated, but on NaiveBayesThemeNews only will calculated training data that has a same theme of news with the set data.

```

88.   if(event.getActionCommand()=="Go with Theme" && cek()
      ==true
89.   ) {
90.       try {
91.           News a =new News
92.               (txtlink.getText().toString(),sourcenews,themenews)
          ;
93.           CaseFolding b=new CaseFolding();
94.           TokenizeNews c=new TokenizeNews();
95.           Stemming d=new Stemming();
96.           Filtering e=new Filtering();
97.           TextMining f=new TextMining();
98.           TextMiningTrainer g=new TextMiningTrainer();
99.           NaiveBayesThemeNews h=new NaiveBayesThemeNews();
100.          System.out.println("Naive Bayes done");
101.
102.          if(txtword.getText().toString().equals("")){
103.
104.              AdditionWord i=new AdditionWord(
105.                  txtword.getText().toString());
106.              resultadd.setText
107.                  ("Result of addition word:"
108.                  +i.cek(txtword.getText().toString()));;
109.          }
110.          result.setText(h.getResult());

```

### 5.1.3 Trainer Message

This feature is used for get training data and save into database.

### 5.1.4 Trainer News

This feature is used for get training data and save into database with input link of news.

## 5.2 Testing

### 5.2.1 First Testing

On first testing, researcher try to compare Naive Bayes with specific theme of news to Naive Bayes without limiting the theme.

Testing Data

Source : Tribun News

Title : Ada Tikus dan Kecoak di Rumah Anies-Sandi, Pemprov  
DKI Anggarkan Rp 266 Juta

Theme : News

Evaluation result of all testing data are showed on below.

Naive Bayes News with all theme

Table 5.1: First Testing (A)

No	All Data	Training Hoax Data	Training Real Data	Data	Run Time	Naive Bayes (H/R)	
1	25	14	11	1	39sc	H ( $1.56 \times 10^{-101}$ )	( $1.34 \times 10^{-103}$ )
2	54	27	27	1	44sc	H( $1.68 \times 10^{-115}$ )	( $2.36 \times 10^{-115}$ )
3	75	37	38	1	45sc	H( $9.95 \times 10^{-114}$ )	( $1.35 \times 10^{-118}$ )

From the results of the testing process can be concluded the more training data owned, the longer the time required to perform the process of classification. This is caused by the process of information retrieval / text mining that serves to find the similarity of each training data so that more training data will make the process of retrieval information is longer. Naive Bayes News with specific theme

Table 5.2: First Testing (B)

No	All Data	Training Hoax Data	Training Real Data	Data	Run Time	Naive Bayes (H/R)	
1	25	14	11	1	41sc	H( $2.62 \times 10^{-95}$ )	( $3.19 \times 10^{-95}$ )
2	54	27	27	1	45sc	H( $1.68 \times 10^{-109}$ )	( $2.36 \times 10^{-10}$ )
3	75	37	38	1	45sc	H( $4.80 \times 10^{-111}$ )	( $7.917 \times 10^{-113}$ )
4	100	50	50	1	52sc	H( $4.80 \times 10^{-114}$ )	( $1.20 \times 10^{-114}$ )

The result is Naive Bayes with specific theme of news has a proximity value between hoax probability and real probability rather than Naive Bayes without limiting the theme.

### 5.2.2 Second Testing

On second testing, researcher try to calculate accuracy and error rate of Naive Bayes. On this testing, researcher try to make two different source data toward 2 testing data. Source data are divided into random topic data and specific topic data. Random source data means training data 's topic are retrieved randomly, specific topic data means researcher already make sure set data has a similarity with training data. Test Data are divided into random test data and specific test data.

1. Training data : Random

Test Data : Random

Theme : All

Table 5.3: Second Testing Result (A)

No	TrainD	T_Hoax	T_Real	TestD	Precision	Recall	Accuracy	Error Rate
1	76	38	38	8	0.5	0.67	62.5%	37.5%



2. Training data : Random

Test Data : Random

Theme : Specific to their origin theme

Table 5.4: Second Testing Result (B)

No	TrainD	T_Hoax	T_Real	TestD	Precision	Recall	Accuracy	Error Rate
1	76	38	38	8	0.5	0.4	37.5%	62.5%

3. Training data : Random

Test Data : Random

Theme : (other theme except their origin theme)

Table 5.5: Second Testing Result (C)

No	TrainD	T_Hoax	T_Real	TestD	Precision	Recall	Accuracy	Error Rate
1	76	38	38	8	0.5	0.5	50%	50%

4. Test Data : Specific

Training Data : Specific

Theme : All

Table 5.6: Second Testing Result (D)

No	TrainD	T_Hoax	T_Real	TestD	Precision	Recall	Accuracy	Error Rate
1	76	38	38	8	1	1	100%	0%

## 5. Test Data Topic : Specific

Training Data Topic : Specific

Theme : Specific to their origin theme / Other theme except their origin theme.

Table 5.7: Second Testing Result (E)

No	TrainD	T_Hoax	T_Real	TestD	Precision	Recall	Accuracy	Error Rate
1	76	38	38	8	0.75	0.75	75%	25%

## 6. Training Data Topic :

Random Test Data Topic

: Specific Theme : All

Table 5.8: Second Testing Result (F)

No	TrainD	T_Hoax	T_Real	TestD	Precision	Recall	Accuracy	Error Rate
1	76	38	38	8	0.5	0.67	62.5%	37.5%

## 7. Training Data Topic : Random

Test Data Topic : Specific

Theme : Specific to their origin theme / not

Table 5.9: Second Testing Result (G)

No	TrainD	T_Hoax	T_Real	TestD	Precision	Recall	Accuracy	Error Rate
1	76	38	38	8	0.75	1	87.5%	12.5%

8. Training Data Topic :  
 Specific Test Data Topic  
 :Random Theme : All

Table 5.10: Second Testing Result (H)

No	TrainD	T_Hoax	T_Real	TestD	Precision	Recall	Accuracy	Error Rate
1	76	38	38	8	1	0.8	87.5%	12.5%

9. Test Data Topic :Random  
 Training Data Topic :  
 Specific Theme : Specific

Table 5.11: Second Testing Result (I)

No	TrainD	T_Hoax	T_Real	TestD	Precision	Recall	Accuracy	Error Rate
1	76	38	38	8	0.75	0.42	37.5%	62.5%

From the results of the testing process in the first experiment has a small tendency of accuracy. This is due to the similarity of topics between training data and test data that tend to be low. High accuracy can be achieved in Experiment 4 and Experiment 5 because of the similarity of topics between training data and test data. Because of the above experiments it can be concluded that the more specific the topics of training data and the higher the similarity of topics between training data and performance test data Multi-nominal Naive Bayes will be better.