

# **Laporan Tugas 3**

## **Teks dan Web Mining**

Diajukan untuk melengkapi tugas Mata Kuliah Teks dan Web Mining

Oleh:

**MUAMMAR ZIKRI AKSANA**

**1608107010045**



**JURUSAN INFORMATIKA**  
**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**  
**UNIVERSITAS SYIAH KUALA**  
**DARUSSALAM, BANDA ACEH**  
**APRIL, 2019**

## I. Tahapan Penyelesaian

Untuk menyelesaikan tugas ini dilakukan proses :

1. Didapatkan file dengan jumlah minimal sebanyak 8000 untuk kedua kategori dengan cara di-crawling kembali dan file tersebut diekstrak
2. Dilanjutkan dengan proses membangun fitur
3. Setiap fitur yang didapat langsung dituliskan ke dalam file dengan format arff dan svm

## II. Pembacaan dan Stemming kata dari setiap file

Stemming dilakukan per-satu file dan setelah n-grams didapatkan (kamus dibangkitkan juga dengan proses stemming ).

1. ls direktori
2. text dari file di-stemming untuk setiap n-grams yang didapat (per n-grams)
3. Setelah data dasar didapatkan untuk setiap n-grams baru kata tersebut digunakan untuk dibangun fitur.

Program stemming masih belum akurat namun untuk banyak kasus didapatkan output yang diharapkan.

```
ubuntu@  
File Edit View Search Terminal Help  
melukis => lukis  
memasak => masak  
menyanyi => nyanyi  
mempertemu => temu  
memperlihat => lihat  
memberi => beri  
penerbang => terbang  
perantau => rantau  
pemukulan => pukul  
----  
dilukis => lukis  
dimasak => masak  
penyanyi => nyanyi  
penari => tari  
----
```

```
----  
lukisan => lukis  
masakan => masak  
nyanyian => nyanyi  
tarian => tari  
----  
dilukiskan => lukis  
dimasakan => masak  
dinyanyikan => nyanyi  
berlarian => lari  
bernyanyi => nyanyi  
----  
memperlakukan => laku  
dipermasalahan => masalah  
diberikan => berik
```

### III. Membangun Fitur

Fitur untuk setiap file dibangun dengan atribut fitur sebanyak 30 fitur yang ditentukan berdasarkan :

- Jumlah kategori, tugas terdiri dari 2 kategori
- Bagian setiap file terdiri dari 5 bagian, yaitu title, content(atas, tengah, bawah)
- Setiap bagian dihitung dengan memperhatikan jumlah kata (ngrams) yang terdiri dari 1-grams, 2-grams, 3-grams (3 bagian)

maka didapat : jumlah fitur =  $2 \times 5 \times 3 = 30$

Tahapan dalam membangun fitur (dalam proses pembacaan setiap file) :

- Setiap kata yang didapat setelah tahap stemming dihash kepada kamus data diberikan ke fungsi **generateFitur** untuk dihitung skornya.
- **generateFitur(kata) : svm|arff**

**generateFitur(kata) : svm|arff**

- Setiap kata yang diterima telah distemming sebelumnya
- Dihitung total kata yang terdapat di-kamus dari sebuah bagian dengan cara di-hashing lalu dibagi dengan total kata, setiap skor yang didapat dituliskan kedalam file.
- Penulisan kedalam file dilakukan untuk kedua format file (arff dan svm \*pada file terpisah) sehingga perhitungan tidak dilakukan berulang kali.

fitur dalam format arff, header dibuat manual secara terpisah, dan data di-generate dengan program

```
@relation berita
|
@attribute title_a1 NUMERIC
@attribute title_a2 NUMERIC
@attribute title_a3 NUMERIC
@attribute bagian_a1a NUMERIC
@attribute bagian_a1b NUMERIC
@attribute bagian_a1c NUMERIC
@attribute bagian_a2a NUMERIC
@attribute bagian_a2b NUMERIC
@attribute bagian_a2c NUMERIC
@attribute bagian_a3a NUMERIC
@attribute bagian_a3b NUMERIC
@attribute bagian_a3c NUMERIC
@attribute bagian_a4a NUMERIC
@attribute bagian_a4b NUMERIC
@attribute bagian_a4c NUMERIC
```

```
@attribute bagian_a2c NUMERIC
@attribute bagian_a3a NUMERIC
@attribute bagian_a3b NUMERIC
@attribute bagian_a3c NUMERIC
@attribute bagian_a4a NUMERIC
@attribute bagian_a4b NUMERIC
@attribute bagian_a4c NUMERIC
@attribute title_b1 NUMERIC
@attribute title_b2 NUMERIC
@attribute title_b3 NUMERIC
@attribute bagian_b1a NUMERIC
@attribute bagian_b1b NUMERIC
@attribute bagian_b1c NUMERIC
@attribute bagian_b2a NUMERIC
@attribute bagian_b2b NUMERIC
@attribute bagian_b2c NUMERIC
@attribute bagian_b3a NUMERIC
@attribute bagian_b3b NUMERIC
@attribute bagian_b3c NUMERIC
@attribute bagian_b4a NUMERIC
@attribute bagian_b4b NUMERIC
@attribute bagian_b4c NUMERIC
@attribute class {basket,bola}
```

```
[mza@localhost final-fitur-twm]$ cat
basket-bola-arff-40  basket-bola-svm-40  header
basket-bola-arff-50  basket-bola-svm-50
[mza@localhost final-fitur-twm]$ cat basket-bola-arff-40 | head -4
@DATA
0.125,0.0625,0.0208333333333333,0.0208333333333333,0.0208333333333333
33,0.4375,0.21875,0.0729166666666667,0.0729166666666667,0.072916666
6666667,0.5,0.25,0.0833333333333333,0.0833333333333333,0.0833333333
333333,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,basket
0,0,0,0,0,0.375,0.1875,0.0625,0.0625,0.0625,0.5,0.25,0.083333333333
3333,0.0833333333333333,0.0833333333333333,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,basket
0.25,0.125,0.0416666666666667,0.0416666666666667,0.0416666666666667
,0.5,0.25,0.0833333333333333,0.0833333333333333,0.0833333333333333,
0.5,0.25,0.0833333333333333,0.0833333333333333,0.0833333333333333,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,basket
[mza@localhost final-fitur-twm]$
```

fitur dalam format svm,

```
[mza@localhost final-fitur-twm]$ cat
basket-bola-arff-40  basket-bola-svm-40  header
basket-bola-arff-50  basket-bola-svm-50
[mza@localhost final-fitur-twm]$ cat basket-bola-svm-40 | head -4
basket_bola: 1:0.125 2:0.0625 3:0.0208333333333333 4:0
.0208333333333333 5:0.0208333333333333 6:0.4375 7:0
.21875 8:0.0729166666666667 9:0.0729166666666667 10:0.072916
6666666667 11:0.5 12:0.25 13:0.0833333333333333 14:0.083333
3333333333 15:0.0833333333333333
16:0 17:0 18:0 19:0 20:0 21:0 22:0 23:0 24:
0 25:0 26:0 27:0 28:0 29:0 30:0
basket_bola: 1:0 2:0 3:0 4:0 5:0 6:0.375 7:0
.1875 8:0.0625 9:0.0625 10:0.0625 11:0.5 12:
0.25 13:0.0833333333333333 14:0.0833333333333333 15:0.083333
3333333333
16:0 17:0 18:0 19:0 20:0 21:0 22:0 23:0 24:
0 25:0 26:0 27:0 28:0 29:0 30:0
[mza@localhost final-fitur-twm]$
```

Setelah proses bangun fitur selesai, data yang akan digunakan untuk membangun model di-shuf terlebih dahulu agar data tersebar.

#### IV. Hasil Akhir

Dari hasil yang telah didapatkan dilakukan klasifikasi dengan 2 metode Naive Bayes classifier dan Support Vector Machine

klasifikasi data menggunakan metode svm

The screenshot shows the Weka software interface with the 'Classifier output' window open. The window displays the results of a Support Vector Machine (SVM) classification. The command line at the top is: `SMO -C 1.0 -L 0.001 -P 1.0E-12 -N 0 -V -1 -W 1 -K "weka.classifiers.functions.supportVector.PolyKernel -E 1.0 -C 250007" -calibrator "weka.classifiers.`

The output shows the following metrics:

- Number of kernel evaluations: 1192928 (71.307% cached)
- Time taken to build model: 0.53 seconds
- Stratified cross-validation summary:
  - Correctly Classified Instances: 14743 (98.9995 %)
  - Incorrectly Classified Instances: 149 (1.0005 %)
  - Kappa statistic: 0.974
  - Mean absolute error: 0.01
  - Root mean squared error: 0.1
  - Relative absolute error: 2.6105 %
  - Root relative squared error: 22.8501 %
  - Total Number of Instances: 14892
- Detailed Accuracy By Class:

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
Weighted Avg.	0.990	0.010	0.996	0.990	0.993	0.974	0.990	0.994	bola
- Confusion Matrix:

a	b	<-- classified as
3808	40	a = basket
109	10935	b = bola

didapat output confusion matrix , dimana terdapat 3808 kelas basket dan diklasifikasi benar sebagai kelas basket dan 40 kelas basket yang diklasifikasi sebagai bola ,dan terdapat 10935 kelas bola yang diklasifikasi benar sebagai bola dan terdapat 109 kelas bola yang diklasifikasi sebagai basket, pada percobaan ini didapat hasil akurasi sebesar 98.99%.

## klasifikasi data menggunakan naive bayes

The screenshot shows the 'NaiveBayes' classifier output in a software interface. The left sidebar contains a list of models, with '- bayes.NaiveBayes' selected. The main panel displays the 'Classifier output' for the NaiveBayes model.

**Classifier output**

Time taken to build model: 0.07 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0.2 seconds

=== Summary ===

Correctly Classified Instances	14616	98.1467 %
Incorrectly Classified Instances	276	1.8533 %
Kappa statistic	0.9513	
Mean absolute error	0.0184	
Root mean squared error	0.1353	
Relative absolute error	4.8005 %	
Root relative squared error	30.9127 %	
Total Number of Instances	14892	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
Weighted Avg.	0.955	0.009	0.973	0.955	0.964	0.951	0.996	0.980	basket
	0.991	0.045	0.984	0.991	0.988	0.951	0.980	0.987	bola

=== Confusion Matrix ===

a	b	<-- classified as
3673	175	a = basket
101	10943	b = bola

pada percobaan menggunakan metode klasifikasi naive bayesian didapat confusion matrix dimana terdapat 3673 data yang diklasifikasi sebagai basket dan benar sebagai basket serta 175 data dari kelas basket diklasifikasi sebagai kelas bola, terdapat 10943 data kelas bola diklasifikasi benar sebagai kelas bola dan 101 data kelas bola diklasifikasi sebagai kelas basket dan dari hasil tersebut didapat akurasi klasifikasi sebesar 98.14%.

Dari 2 hasil output sebelumnya didapat metode svm menghasilkan hasil akurasi yang lebih bagus.