

# **Klasifikasi Tingat Keseriusan Mahasiwa FILKOM dalam mempelajari *Computer Science***

Disusun oleh:

Pengenalan Pola - S  
Kelompok – 1  
Azzam Syawqi Aziz  
155150207111132  
Pengenalan Pola  
Desember/2017



TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
2017

## BAB 1

### I. Latar Belakang

Diawali oleh pengalaman penulis yang seringkali memberikan materi dan diskusi kepada kerabat dan teman dekat di filkom perihal *computer science*, penulis banyak menemukan jurang perbedaan yang luas antara standar kompetensi yang seharusnya dimiliki oleh seorang mahasiswa *computer science* dengan yang ada di sekitar. Hal ini diperkuat oleh maraknya keluhan industri tentang kompetensi pelamar maupun keahlian para mahasiswa di dunia kerja.

Hal inilah yang membuat penulis selalu berusaha memberikan motivasi dan ajakan untuk selalu serius menekuni bidang yang sedang kita alami, apapun itu. Sehingga seseorang tidak lagi menghabiskan banyak waktu untuk hal yang tersier ataupun bahkan hal yang tidak bermanfaat yang jauh sekali dari nuansa akademisi.

Sudah telalu banyak hal yang kita ataupun orang tua kita investasikan dan perjuangkan, baik moril dan materil, hari-hari yang mana seharusnya penuh dengan semangat kebermanfaatan dan karya, dan tidak lagi ada waktu yang terbuang untuk hidup yang hanya sekali ini.

Dengan adanya sistem ini penulis banyak berharap para mahasiswa lebih sadar, bahwa mereka seharusnya memiliki standar kompetensi yang diharapkan.

## BAB 2

### II. Definisi Sistem

Klasifikasi Tingkat Keseriusan Mahasiswa FILKOM dalam mempelajari Computer Science

Sistem yang kami bangun adalah sistem yang melakukan klasifikasi pada tingkat keseriusan mahasiswa FILKOM dalam mempelajari *Computer Science*.

Kami membangun sistem tersebut sebagai pengganti pakar untuk bisa melakukan klasifikasi terhadap seorang mahasiswa dalam menentukan tingkat keseriusan belajar dengan mengajukan beberapa soal, total soal berjumlah 3 soal, setiap soal berasal dari 3 kategori keilmuan di bidang *Computer Science*, kategori pertama adalah dibidang Sistem Operasi, kedua dibidang *Programming*, ketiga dibidang Kecerdasan Buatan, masing-masing soal yang dijawab oleh *user* diberikan nilai berupa benar, sedang atau salah, kemudian sistem melakukan kalkulasi dengan menghitung probabilitas kemunculan yang didapatkan dari data training yang sudah sistem miliki, sistem kemudian memberikan keluaran keputusan hasil akhir, apakah mahasiswa tersebut serius, cukup serius atau tidak serius.

Sistem ini kami buat berbasis *web* dan diimplementasikan menggunakan bahasa *Common Lisp*, sistem dapat menerima inputan nilai satu mahasiswa dalam setiap prosesnya.

## BAB 3

### III. Methode

## I. Dasar Teori Klasifikasi yang digunakan

Kami menggunakan metode *Naive Bayes Classifier* dalam sistem ini, pemilihan *Naive Bayes Classifier* dari pada *Bayes* karena pada data latih yang kami miliki tidak terdapat banyak kombinasi, sehingga sistem tidak bisa mengeluarkan hasil komputasi jika menggunakan metode *Bayes Classifier* dengan jumlah data uji yang sedikit dan tidak beragam. Oleh karena itulah kami menggunakan *Naive Bayes Classifier*.

Langkah yang dilakukan pada proses kalkulasi menggunakan metode *Naive Bayes Classifier* adalah dengan proses sebagai berikut

1. Proses pengumpulan data dari sumber primer, dengan menggunakan 3 fitur berupa 1 pertanyaan pada masing-masing fitur berdasarkan 3 kategori, yaitu kategori *Secutiry*, *Programming*, dan *Operating System*. dan 3 kelas berupa (Serius, Cukup Serius dan Tidak Serius). Sehingga menghasilkan data sebagai berikut

No	Nama	F1	F2	F3	Kelas
1	DWP	S	S	S	TS
2	SAK	S	S	S	TS
3	KJH	B	B	S	SS
4	A	S	S	S	TS
5	HR	S	S	S	TS
6	RI	D	D	S	SS
7	PGS	S	S	S	TS
8	FAZ	S	S	S	TS
9	MS	S	S	S	TS
10	FNA	S	S	S	TS
11	A1	S	S	S	TS
12	MZM	S	S	S	TS
13	RAW	S	S	S	TS
14	MAQ	S	S	S	TS
15	FF	S	S	S	TS
16	MDR	S	S	S	TS
17	AFY	S	S	S	TS
18	DMN	D	D	D	SS
19	R	S	B	B	SS
20	MAR	S	S	S	TS
21	HHO	S	S	S	TS
22	DPW	D	S	S	TS
23	I	S	S	S	TS
24	DNF	B	B	B	S
25	L	S	D	S	TS
26	DD	S	D	S	TS
27	QOB	S	S	S	TS
28	DI	S	D	S	TS
29	WDR	S	S	S	TS
30	FAW	S	S	S	TS
31	MGA	S	D	S	TS
32	SA	S	B	D	SS
33	LF	S	S	S	TS
34	N	S	S	S	TS
35	RF	S	S	D	TS

2. Melakukan pengelompokan tiap kelas agar mudah dalam melakukan perhitungan.

[illegible]

3. Kemudian kita menggunakan Rumus *Naive Bayes Classifier* untuk melakukan proses perhitungan, contoh

$$P(Wa|x) = P(Wa) \cdot \prod_{i=1}^k P(x_i|Wa)$$

$P(Wa|x)$  = Kemungkinan di kelas  $Wa$  terhadap data  $x$

$P(Wa)$  = *Prior probability*

$Wa$  = Kelas yang dicari

$x$  = Data yang ditanyakan

$k$  = Fitur

4. Dari rumus diatas, hal pertama yang dilakukan adalah mencari probabilitas di setiap kelas, yaitu diawali dengan menghitung *Prior probability*, yaitu banyak data pada suatu kelas yang dicari dibandingkan dengan seluruh data yang ada.
5. Mencari *Likelihood* dengan cara menghitung jumlah data yang ditanyakan pada  $f_1$  (fitur pertama) pada kelas yang sedang dicari, seterusnya hingga fitur ke 3.
6. melakukan langkah ke 4 dan 5, pada kelas lainnya.
7. Menghitung hasil probabilitas setiap kelas.
8. Membandingkan seluruh hasil akhir setiap kelas.
9. Nilai tertinggi adalah nilai yang menjadi jawaban, sehingga hasil komputasi akhir adalah kelas yang memiliki nilai tertinggi.

## II. Proses Training

Proses training dilakukan oleh seorang *knowledge engineer* dengan bantuan seorang pakar dengan proses dimana seorang *knowledge engineer* pada kasus ini memberikan hasil interview yang didapatkan dari data primer, data interview tersebut kemudian dicek oleh seorang pakar, dan diberikan nilai pada setiap fitur dan diberikan hasil akhir kelas pada salah satu data, proses ini dilakukan terus menerus hingga semua data memiliki kelas akhir dengan mempertimbangkan nilai fitur-fitur yang ada, dalam hal ini seorang pakar berperan penting dari proses penilaian fitur-fitur setiap data hingga memberikan nilai kelas pada setiap data.

- Data yang digunakan untuk proses training berjumlah 35 lima data primer.

Proses training dilakukan dengan cara sebagai berikut.

1. Seorang *knowledge engineer* memberikan data primer hasil wawancara.

```
50 (defvar *class-ts* (list
51   '(f1 ("f" "f" "f" "f" "f" "f" "f" "f" "f" "f"
52         "f" "f" "f" "f" "f" "m" "f" "f" "f" "f"
53         "f" "f" "f" "f" "f" "f" "f" "f"))
54   '(f2 ("f" "f" "f" "f" "f" "f" "f" "f" "f" "f"
55         "f" "f" "f" "t" "f" "f" "f" "f" "m" "f"
56         "f" "m" "m" "f" "f" "f" "f"))
57   '(f3 ("f" "f" "f" "f" "f" "f" "f" "f" "f" "f"
58         "f" "f" "t" "f" "f" "f" "f" "f" "f" "f"
59         "f" "f" "f" "f" "f" "f" "m"))))
60
61 (defvar *class-ss* (list '(f1 ("t" "m" "m" "f"))
62   '(f2 ("t" "m" "m" "t"))
63   '(f3 ("f" "f" "m" "m"))))
64
65 (defvar *class-s* (list '(f1 ("t"))
66   '(f2 ("t"))
67   '(f3 ("t"))))
68
```

Proses perhitungan data pada fase training dilakukan sebagai berikut

1. Pakar melakukan penilaian pada setiap soal

[illegible]

2. Data hasil penilaian pakar kemudian di bentuk datasheet oleh *Knowledge Engineer*.

No	Nama	F1	F2	F3	Kelas
1	DWP	S	S	S	TS
2	SAK	S	S	S	TS
3	KJH	B	B	S	SS
4	A	S	S	S	TS
5	HR	S	S	S	TS
6	RI	D	D	S	SS
7	PSS	S	S	S	TS
8	FAZ	S	S	S	TS
9	MS	S	S	S	TS
10	FNA	S	S	S	TS
11	A1	S	S	S	TS
12	MZM	S	S	S	TS
13	RAW	S	S	S	TS
14	MAA	S	S	S	TS
15	FF	S	S	S	TS
16	MDR	S	S	S	TS
17	AFY	S	S	S	TS
18	DMN	D	D	D	SS
19	R	S	B	B	SS
20	MAR	S	S	S	TS
21	HHO	S	S	S	TS
22	DPW	D	S	S	TS
23	I	S	S	S	TS
24	DNE	B	B	B	S
25	L	S	D	S	TS
26	DD	D	D	S	TS
27	DOB	S	S	S	TS
28	DI	S	D	S	TS
29	WDR	S	S	S	TS
30	RAW	S	S	S	TS
31	MSA	S	D	S	TS
32	SA	S	B	D	SS
33	LF	S	S	S	TS
34	N	S	S	S	TS
35	RF	S	S	D	TS

3. Data yang sudah di tata dalam datasheet, kemudian diinputkan kedalam sistem, dimana kemudian data tersebut digunakan untuk kalkulasi pada data testing nantinya.

## BAB 4

### IV. Hasil dan Pembahasan (result)

#### I. Proses Testing

Proses testing dilakukan dengan cara melakukan uji coba berupa data yang diinputkan kedalam sistem yang kemudian di komputasi hingga mengeluarkan kelas akhir.

Langkah-langkah proses testing tersebut dengan menggunakan methode *Naive Bayes Classifier*, sehingga urutan langkah nya pun sama dengan penjelasan metode, kurang lebih prosesnya seperti berikut

1. Proses testing sama halnya menggunaakn methode *Naive Bayes Classifier* sebagai berikut.

$$P(Wa|x) = P(Wa) \cdot \prod_{i=1}^k P(x|Wa-k)$$

$P(Wa|x)$  = Kemungkinan di kelas  $Wa$  terhadap data  $x$

$P(Wa)$  = *Prior probability*

$Wa$  = Kelas yang dicari

$x$  = Data yang ditanyakan

$k$  = Fitur

Atau dalam sistem nya kita menggunakan interface sebagai berikut

	Interface 1
1	<code>(defroute "/inference" (&amp;key _parsed)</code>
2	<code>  (let ((params (cdr (assoc "items" _parsed :test</code>
3	<code>    #'string=))))</code>
4	<code>    (funcall #'hitung-semua-fitur-di-ts params)</code>
5	<code>    (funcall #'hitung-semua-fitur-di-ss params)</code>
6	<code>    (funcall #'hitung-semua-fitur-di-s params)</code>
7	<code>    (funcall #'hitung-semua-kelas)</code>
8	<code>    (funcall #'yield-hasil))</code>
9	<code>  (render #P"inferensi.html"</code>
10	<code>    `(:hasil-keseriusan ,*hasil-akhir*))</code>

2. Dari rumus diatas, hal pertama yang dilakukan adalah mencari probabilitas di setiap kelas, yaitu diawali dengan menghitung *Prior probability*, yaitu banyak data pada suatu kelas yang dicari dibandingkan dengan seluruh data yang ada.

3. Kemudian dilanjutkan dengan perhitungan berapa banyak muncul nya suatu data dalam fitur tertentu dikelas yang dicari

	Cari jumlah fitur per kelas
1	;;;; cari berapa banyak key di tiap fitur
2	
3	;; dikelas ts
4	
5	(defun get-key-di-f1-ts (key)
6	(let ((isi (alexandria:flatten (cdr (first *class-
7	ts*))))
8	(dolist (x isi)
9	(cond
10	((equal x key) (incf *key-f1-di-ts*))))
11	
12	(defun get-key-di-f2-ts (key)
13	(let ((isi (alexandria:flatten (cdr (second *class-
14	ts*))))
15	(dolist (x isi)
16	(cond
17	((equal x key) (incf *key-f2-di-ts*))))
18	
19	(defun get-key-di-f3-ts (key)
20	(let ((isi (alexandria:flatten (cdr (third *class-
21	ts*))))
22	(dolist (x isi)
23	(cond
24	((equal x key) (incf *key-f3-di-ts*))))
25	
26	;; di kelas ss
27	
28	(defun get-key-di-f1-ss (key)
29	(let ((isi (alexandria:flatten (cdr (first *class-
30	ss*))))
31	(dolist (x isi)
32	(cond
33	((equal x key) (incf *key-f1-di-ss*))))
34	
35	(defun get-key-di-f2-ss (key)
36	(let ((isi (alexandria:flatten (cdr (second *class-
37	ss*))))
38	(dolist (x isi)
39	(cond
40	((equal x key) (incf *key-f2-di-ss*))))
41	
42	(defun get-key-di-f3-ss (key)
43	(let ((isi (alexandria:flatten (cdr (third *class-
44	ss*))))
45	(dolist (x isi)
46	(cond
47	((equal x key) (incf *key-f3-di-ss*))))
48	
49	;; di kelas s
50	
51	(defun get-key-di-f1-s (key)
52	(let ((isi (alexandria:flatten (cdr (first *class-
53	s*))))
54	(dolist (x isi)
55	(cond
56	((equal x key) (incf *key-f1-di-s*))))
57	



```

58 (defun get-key-di-f2-s (key)
59   (let ((isi (alexandria:flatten (cdr (second *class-
60 s*))))))
61     (dolist (x isi)
62       (cond
63         ((equal x key) (incf *key-f2-di-s*))))))
64
65 (defun get-key-di-f3-s (key)
66   (let ((isi (alexandria:flatten (cdr (third *class-
67 s*))))))
68     (dolist (x isi)
69       (cond
70         ((equal x key) (incf *key-f3-di-s*))))))
71

```

4. Menghitung hasil probabilitas yang dimiliki setiap kelas.

	Hitung probabilitas per kelas
1	(defun hitung-kelas-ts ()
2	(setq *hasil-ts*
3	(*
4	(* (/ *key-f1-di-ts* 35) (/ *key-f2-di-ts* 35) (/
5	*key-f3-di-ts* 35))
6	(/ 29 35))))
7	
8	(defun hitung-kelas-ss ()
9	(setq *hasil-ss*
10	(*
11	(* (/ *key-f1-di-ss* 5) (/ *key-f2-di-ss* 5) (/
12	*key-f3-di-ss* 5))
13	(/ 5 35))))
14	
15	(defun hitung-kelas-s ()
16	(setq *hasil-s*
17	(*
18	(* (/ *key-f1-di-s* 1) (/ *key-f2-di-s* 1) (/
19	*key-f3-di-s* 1))
20	(/ 1 35))))

5. Membandingkan nilai hasil akhir setiap kelas.

	Hitung probabilitas per kelas
1	;; yield-hasil
2	
3	(defun yield-hasil ()
4	(cond
5	((and (> *hasil-ts* *hasil-ss*)
6	(> *hasil-ts* *hasil-s*))
7	(setq *hasil-akhir* "Anda Kurang Serius"))
8	((and (> *hasil-ss* *hasil-ts*)
9	(> *hasil-ss* *hasil-s*))
10	(setq *hasil-akhir* "Anda Cukup Serius"))
11	((and (> *hasil-s* *hasil-ts*)
12	(> *hasil-s* *hasil-ss*))

13	<code>(setq *hasil-akhir* "Selamat, Anda Serius"))))</code>
----	---

6. Nilai tertinggi adalah nilai yang menjadi jawaban, sehingga hasil komputasi akhir adalah kelas yang memiliki nilai tertinggi.

- Data yang digunakan dalam proses testing adalah 15 data.

Contoh perhitungan manual pada suatu data misalnya sebagai berikut

1. Data testing yang di cari adalah **salah-benar-salah**. Atau bisa diatrikan **false-true-false (f-t-f)**.
2. Mengikuti rumus *Naive Bayes Classifier*.

$$P(Wa | x) = P(Wa) \cdot \prod_{i=1}^k P(x_i | Wa - k)$$

3. Mencari probabilitas setiap kelas

1. Mencari *prior probability*

1. Kelas TS

$$\frac{29}{35}$$

2. Kelas SS

$$\frac{5}{35}$$

3. Kelas S

$$\frac{1}{35}$$

2. Mencari *likelihood*

1. Kelas TS

$$\frac{28}{35} \cdot \frac{0}{35} \cdot \frac{28}{35}$$

2. Kelas SS

$$\frac{2}{5} \cdot \frac{2}{5} \cdot \frac{2}{5}$$

3. Kelas S

$$\frac{0}{1} \cdot \frac{1}{1} \cdot \frac{0}{1}$$

#### 4. Menghitung hasil setiap kelas

##### 1. Kelas TS

$$P(TS|FTF) = \frac{29}{35} \cdot \left( \frac{28}{35} \cdot \frac{0}{35} \cdot \frac{28}{35} \right) = 0$$

##### 2. Kelas SS

$$P(SS|FTF) = \frac{5}{35} \cdot \left( \frac{2}{5} \cdot \frac{2}{5} \cdot \frac{2}{5} \right) = 0,05714$$

##### 3. Kelas S

$$P(S|FTF) = \frac{1}{35} \cdot \left( \frac{0}{1} \cdot \frac{1}{1} \cdot \frac{0}{1} \right) = 0$$

#### 5. Membandingkan, hasil setiap kelas dengan mencari yang terbesar, sehingga hasilnya adalah, kelas **SS (Cukup Serius)**

Akurasi yang dimiliki sistem ini adalah

##### 1. Penilaian data latih dari pakar

NO	F1	F2	F3	KELAS
1	S	S	S	TS
2	S	S	S	TS
3	S	S	S	TS
4	S	S	S	TS
5	S	S	S	TS
6	S	S	D	TS
7	S	S	S	TS
8	S	S	S	TS
9	S	S	S	TS
10	S	S	S	TS
11	S	S	B	SS
12	S	S	S	TS
13	S	B	D	SS
14	S	B	S	TS
15	D	B	S	SS

## 2. Hasil dari sistem

<b>NO</b>	<b>F1</b>	<b>F2</b>	<b>F3</b>	<b>KELAS</b>
1	S	S	S	TS
2	S	S	S	TS
3	S	S	S	TS
4	S	S	S	TS
5	S	S	S	TS
<b>6</b>	S	S	D	TS
7	S	S	S	TS
8	S	S	S	TS
9	S	S	S	TS
10	S	S	S	TS
<b>11</b>	S	S	B	TS
12	S	S	S	TS
<b>13</b>	S	B	D	SS
<b>14</b>	S	B	S	TS
<b>15</b>	D	B	S	SS

## 3. Akurasi

<b>Pakar</b>	<b>Sistem</b>
TS	TS
TS	TS
TS	TS
TS	TS
TS	TS
TS	TS
TS	TS
TS	TS
TS	TS
TS	TS
<b>SS</b>	<b>TS</b>
TS	TS
SS	SS
TS	TS
SS	SS

Dari 15 kali data testing yang di inputkan, sistem memiliki satu perbedaan dengan pakar, sehingga nilai akurasinya.

$$P_{15}(\text{akurat}) = \frac{14}{15} \cdot 100\% = 94\%$$

$$P_{15}(\text{tidak akurat}) = \frac{1}{15} \cdot 100\% = 6\%$$

## II. Pembahasan

Dapat kita simpulkan bahwa pada proses testing atau bisa kita sebut proses inferensi, sistem melakukan perhitungan dimana data latih diolah agar menemukan hasil yang di inputkan pada masa testing.

Data latih yang kurang beragam akan menghasilkan hasil 0 jika menggunakan *Bayes Classifier*, karena cara kerjanya adalah membandingkan berapa jumlah kemungkinan suatu nilai data testing muncul dengan yang ada pada data latih, sehingga ketika data testing berupa (benar-salah-benar) misalnya dan data latih tidak memiliki data serupa semua hasil perhitungan akan 0.

Hal ini yang menuntun kami untuk menggunakan metode *Naive Bayes Classifier*, dimana memiliki cara kerja yang berbeda dengan *Bayes Classifier*, *Naive Bayes* menghitung kemungkinan munculnya setiap fitur secara independent atau terpisah. Proses inferensinya yaitu dengan cara menghitung banyaknya nilai yang muncul pada setiap fitur yang terpisah di setiap kelas, kemudian melakukan kalkulasi akhir dengan membandingkan nilai tertinggi yang pada tiap tiap kelas. Dengan proses *Naive Bayes* yang melakukan perhitungan setiap fitur secara independent sistem memiliki kemungkinan yang lebih luas untuk menghasilkan data keluaran dengan kondisi data yang sedikit dan tidak beragam.

Proses testing memiliki hasil akurasi 94% dan memiliki kesalahan karena minimnya data latih yang dimiliki oleh sistem, sehingga tentu keputusan sistem memiliki kekurangan jika dibandingkan dengan hasil pakar, input testing yang semakin beragam juga akan mempengaruhi hasil akurasi, tinggi rendah nya akurasi disebabkan oleh banyak dan beragam tidaknya data latih yang sistem kita miliki.

## V. Lampiran

## I. Source code

```

Hitung probabilitas per kelas
1 ;; inference
2 ;;
3
4 (defparameter *key-f1-di-ts* 0)
5 (defparameter *key-f2-di-ts* 0)
6 (defparameter *key-f3-di-ts* 0)
7
8 (defparameter *key-f1-di-ss* 0)
9 (defparameter *key-f2-di-ss* 0)
10 (defparameter *key-f3-di-ss* 0)
11
12 (defparameter *key-f1-di-s* 0)
13 (defparameter *key-f2-di-s* 0)
14 (defparameter *key-f3-di-s* 0)
15
16 (defparameter *hasil-ts* 0)
17 (defparameter *hasil-ss* 0)
18 (defparameter *hasil-s* 0)
19
20 (defvar *hasil-akhir* "")
21
22 ;; datasheet
23
24
25 (defvar *class-ts* (list
26   '(f1 ("f" "f" "f" "f" "f" "f" "f" "f" "f" "f" "f"
27         "f" "f" "f" "f" "f" "f" "f" "m" "f" "f" "f"
28         "f" "f" "f" "f" "f" "f" "f" "f" "f"))
29   '(f2 ("f" "f" "f" "f" "f" "f" "f" "f" "f" "f" "f" "f"
30         "f" "f" "f" "f" "t" "f" "f" "f" "f" "m" "m"
31         "f" "m" "f" "f" "m" "f" "f" "f" "f"))
32   '(f3 ("f" "f" "f" "f" "f" "f" "f" "f" "f" "f" "f" "f"
33         "f" "f" "f" "f" "t" "f" "f" "f" "f" "f" "f" "f"
34         "f" "f" "f" "f" "f" "f" "f" "m"))))
35
36 (defvar *class-ss* (list '(f1 ("t" "m" "m" "f"))
37   '(f2 ("t" "m" "m" "t"))
38   '(f3 ("f" "f" "m" "m"))))
39
40 (defvar *class-s* (list '(f1 ("t"))
41   '(f2 ("t"))
42   '(f3 ("t"))))
43
44
45 ;;;; cari berapa banyak key di tiap fitur
46
47 ;; dikelas ts
48
49 (defun get-key-di-f1-ts (key)
50   (let ((isi (alexandria:flatten (cdr (first *class-ts*)))))
51     (dolist (x isi)
52       (cond
53         ((equal x key) (incf *key-f1-di-ts*)))))
54
55 (defun get-key-di-f2-ts (key)
56   (let ((isi (alexandria:flatten (cdr (second *class-ts*)))))

```

```

57      (dolist (x isi)
58        (cond
59          ((equal x key) (incf *key-f2-di-ts*))))))
60
61 (defun get-key-di-f3-ts (key)
62   (let ((isi (alexandria:flatten (cdr (third *class-ts*)))))
63     (dolist (x isi)
64       (cond
65         ((equal x key) (incf *key-f3-di-ts*))))))
66
67 ;; di kelas ss
68
69 (defun get-key-di-f1-ss (key)
70   (let ((isi (alexandria:flatten (cdr (first *class-ss*)))))
71     (dolist (x isi)
72       (cond
73         ((equal x key) (incf *key-f1-di-ss*))))))
74
75 (defun get-key-di-f2-ss (key)
76   (let ((isi (alexandria:flatten (cdr (second *class-ss*)))))
77     (dolist (x isi)
78       (cond
79         ((equal x key) (incf *key-f2-di-ss*))))))
80
81 (defun get-key-di-f3-ss (key)
82   (let ((isi (alexandria:flatten (cdr (third *class-ss*)))))
83     (dolist (x isi)
84       (cond
85         ((equal x key) (incf *key-f3-di-ss*))))))
86
87 ;; di kelas s
88
89 (defun get-key-di-f1-s (key)
90   (let ((isi (alexandria:flatten (cdr (first *class-s*)))))
91     (dolist (x isi)
92       (cond
93         ((equal x key) (incf *key-f1-di-s*))))))
94
95 (defun get-key-di-f2-s (key)
96   (let ((isi (alexandria:flatten (cdr (second *class-s*)))))
97     (dolist (x isi)
98       (cond
99         ((equal x key) (incf *key-f2-di-s*))))))
100
101 (defun get-key-di-f3-s (key)
102   (let ((isi (alexandria:flatten (cdr (third *class-s*)))))
103     (dolist (x isi)
104       (cond
105         ((equal x key) (incf *key-f3-di-s*))))))
106
107 ;; hitung perkelas
108 (defun hitung-kelas-ts ()
109   (setq *hasil-ts*
110     (*
111       (* (/ *key-f1-di-ts* 35) (/ *key-f2-di-ts* 35) (/ *key-
112 f3-di-ts* 35))
113       (/ 29 35))))
114 (defun hitung-kelas-ss ()
115   (setq *hasil-ss*
116     (*
117       (* (/ *key-f1-di-ss* 5) (/ *key-f2-di-ss* 5) (/ *key-
118 f3-di-ss* 5))
119       (/ 5 35))))
120

```

```

121
122 (defun hitung-kelas-s ()
123   (setq *hasil-s*
124     (*
125       (* (/ *key-f1-di-s* 1) (/ *key-f2-di-s* 1) (/ *key-f3-
126 di-s* 1))
127       (/ 1 35))))
128
129 ;; yield-hasil
130
131 (defun yield-hasil ()
132   (cond
133     ((and (> *hasil-ts* *hasil-ss*)
134           (> *hasil-ts* *hasil-s*))
135      (setq *hasil-akhir* "Anda Kurang Serius"))
136     ((and (> *hasil-ss* *hasil-ts*)
137           (> *hasil-ss* *hasil-s*))
138      (setq *hasil-akhir* "Anda Cukup Serius"))
139     ((and (> *hasil-s* *hasil-ts*)
140           (> *hasil-s* *hasil-ss*))
141      (setq *hasil-akhir* "Selamat, Anda Serius"))))
142
143 ;; abstraction 1
144
145 (defun hitung-semua-kelas ()
146   (funcall #'hitung-kelas-ts)
147   (funcall #'hitung-kelas-ss)
148   (funcall #'hitung-kelas-s))
149
150 (defun hitung-semua-fitur-di-ts (key)
151   (funcall #'get-key-di-f1-ts (first key))
152   (funcall #'get-key-di-f2-ts (second key))
153   (funcall #'get-key-di-f3-ts (third key)))
154
155 (defun hitung-semua-fitur-di-ss (key)
156   (funcall #'get-key-di-f1-ss (first key))
157   (funcall #'get-key-di-f2-ss (second key))
158   (funcall #'get-key-di-f3-ss (third key)))
159
160 (defun hitung-semua-fitur-di-s (key)
161   (funcall #'get-key-di-f1-s (first key))
162   (funcall #'get-key-di-f2-s (second key))
163   (funcall #'get-key-di-f3-s (third key)))
164
165 ;; refresh
166
167 (defun perbaharui ()
168   (defparameter *key-f1-di-ts* 0)
169   (defparameter *key-f2-di-ts* 0)
170   (defparameter *key-f3-di-ts* 0)
171
172   (defparameter *key-f1-di-ss* 0)
173   (defparameter *key-f2-di-ss* 0)
174   (defparameter *key-f3-di-ss* 0)
175
176   (defparameter *key-f1-di-s* 0)
177   (defparameter *key-f2-di-s* 0)
178   (defparameter *key-f3-di-s* 0)
179
180   (defparameter *hasil-ts* 0)
181   (defparameter *hasil-ss* 0)
182   (defparameter *hasil-s* 0))
183

```



```

184 ;;
185 ;; Routing rules
186
187 (defroute "/" ()
188   (render #P"index.html"))
189
190 (defroute "/inference" (&key _parsed)
191   (let ((params (cdr (assoc "items" _parsed :test #'string=))))
192     (funcall #'hitung-semua-fitur-di-ts params)
193     (funcall #'hitung-semua-fitur-di-ss params)
194     (funcall #'hitung-semua-fitur-di-s params)
195     (funcall #'hitung-semua-kelas)
196     (funcall #'yield-hasil))
197   (render #P"inferensi.html"
198     `(:hasil-keseriusan ,*hasil-akhir*)))
199
200

```

## II. Sceenshot

### I. Benar – benar – benar = (Selamat, Anda Serious)

Silahkan Masukkan Nilai anda

Fitur 1 : Jelaskan bagaimana proses penyimpanan dan pengecekan kata sandi pada Operating System.

Fitur 2 : Jelaskan yang anda ketahui tentang High order Function

Fitur 3 : Kapankah kita menggunakan Bayes Classifier dan Naive Bayes Classifier

© 2017 Amiserious

---

Hasil Keseriusan Anda

Selamat, Anda Serious

### II. Salah – salah – salah = (Anda Kurang Serious)

Silahkan Masukkan Nilai anda

Fitur 1 : Jelaskan bagaimana proses penyimpanan dan pengecekan kata sandi pada Operating System.

Fitur 2 : Jelaskan yang anda ketahui tentang High order Function

Fitur 3 : Kapankah kita menggunakan Bayes Classifier dan Naive Bayes Classifier

Hasil Kesiurutan Anda

Anda Kurang Serius

Terimakasih

### III. Benar – salah – sedang = (Anda Kurang Serius)

Silahkan Masukkan Nilai anda

Fitur 1 : Jelaskan bagaimana proses penyimpanan dan pengecekan kata sandi pada Operating System.

Benar

Fitur 2 : Jelaskan yang anda ketahui tentang High order Function

Salah

Fitur 3 : Kapankah kita menggunakan Bayes Classifier dan Naive Bayes Classifier

Sedang

Inferensikan

Hasil Kesiurutan Anda

Anda Kurang Serius

Terimakasih

### IV. Benar-benar-sedang = (Anda Cukup serius)

Silahkan Masukkan Nilai anda

Fitur 1 : Jelaskan bagaimana proses penyimpanan dan pengecekan kata sandi pada Operating System.

Benar

Fitur 2 : Jelaskan yang anda ketahui tentang High order Function

Benar

Fitur 3 : Kapankah kita menggunakan Bayes Classifier dan Naive Bayes Classifier

Sedang

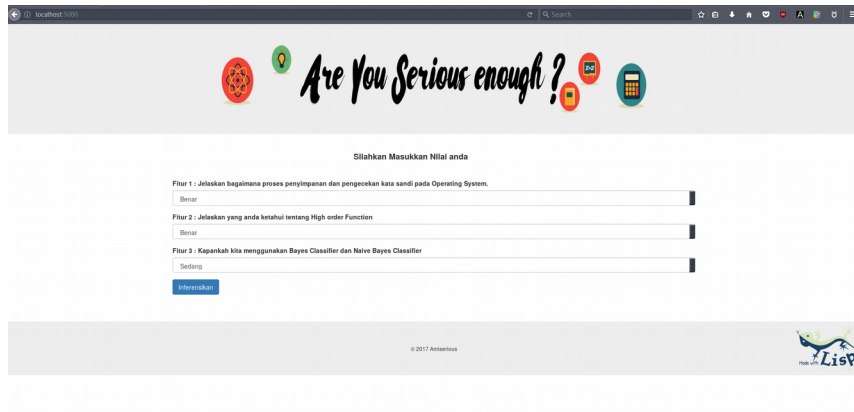
Inferensikan

Hasil Kesiurutan Anda

Anda Cukup Serius

Terimakasih

## V. Full Screenshot



localhost:3000

Are You Serious enough?

Silahkan Masukkan Nilai anda

Fitur 1 : Jelaskan bagaimana proses penyimpanan dan pengecekan kata sandi pada Operating System.  
Benar

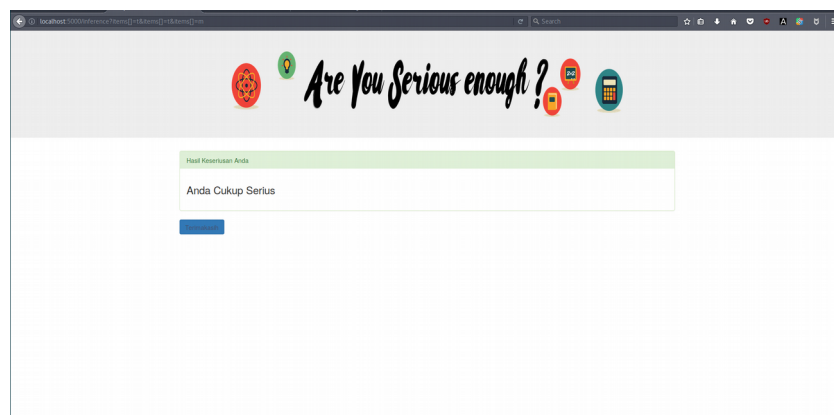
Fitur 2 : Jelaskan yang anda ketahui tentang High order Function  
Benar

Fitur 3 : Kapankah kita menggunakan Bayes Classifier dan Naive Bayes Classifier  
Sedang

Informasikan

© 2017 Anantarius

Logo of LisP



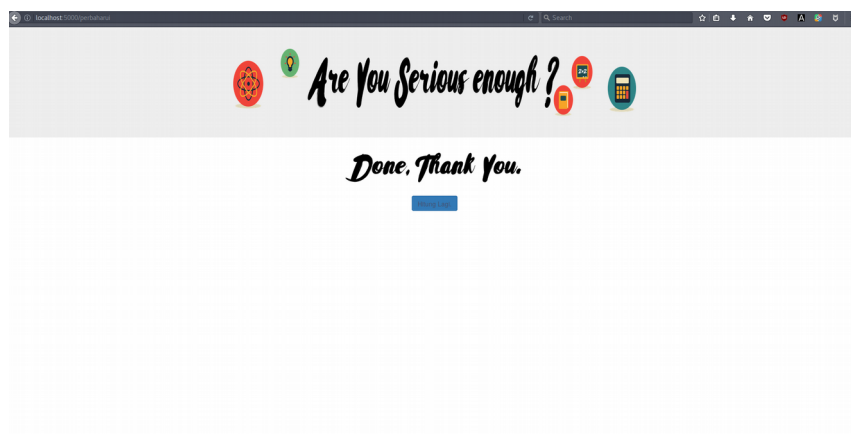
localhost:3000/anonymous/home?name[1]=Anantarius

Are You Serious enough?

Hasil Keresutan Anda

Anda Cukup Serius

Informasikan



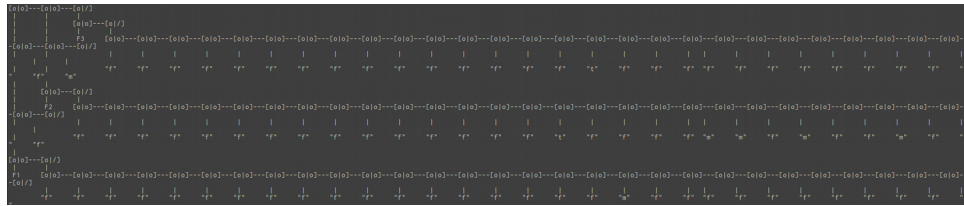
localhost:3000/anonymous

Are You Serious enough?

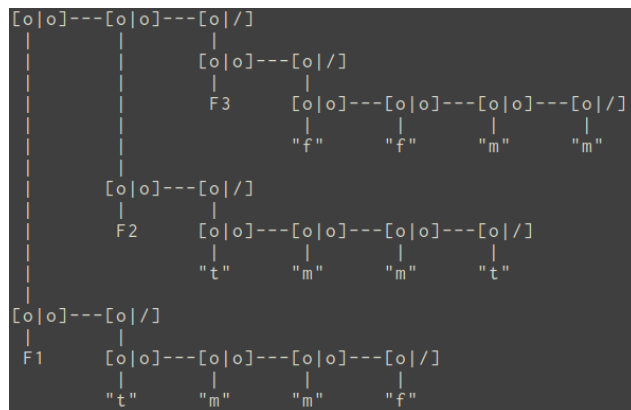
Done, Thank You.

### III. Struktur Data

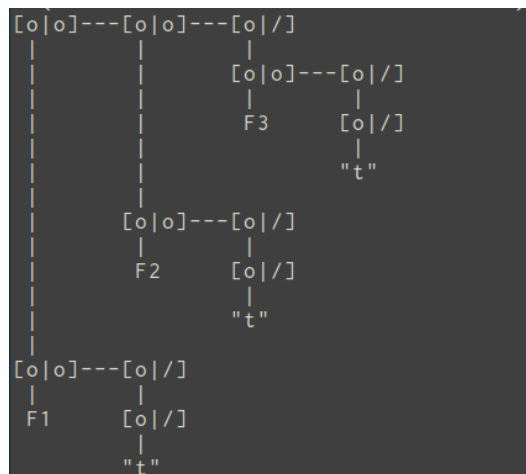
## I. Kelas TS



## II. Kelas SS



### III. Kelas S



## IV. Video

[amiserious] - Seriousness Checker using *Naive Bayes Classifier*  
<https://www.youtube.com/watch?v=tLWQQVkgzA4>