**BAB IV**

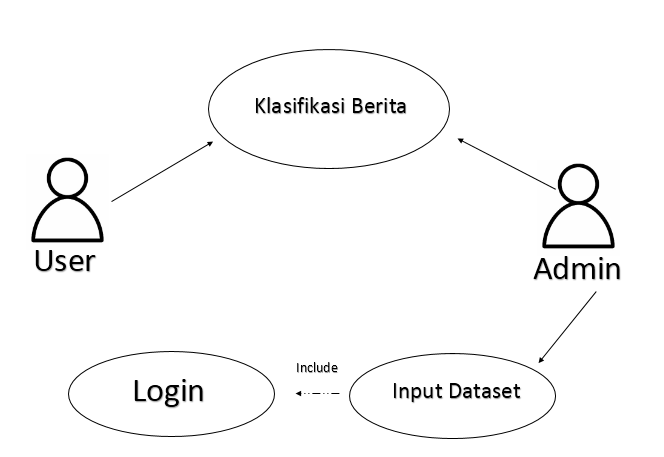
**HASIL**

Pada tahap ini dilakukan analisi secara menyeluruh dari *system* yang telah dibuat. Akan dijelaskan rincian perancangan system dan implementasi dalam sebuah bahasa pemrograman. Selanjutnya akan dilakukan ujicoba/ *testing* guna mencari kekurangan dan kelebihan pada *system* yang dibuat sehingga dapat dijadikan bahan pertimbangan guna melakukan pengembangan penelitian selanjutnya.

**A. Desain Aplikasi**

Pada tahap ini terdapat beberapa desain perancangan *system* meliputi *Usecase Diagram,Diagram Konteks,* Alurdata *user* , Alur data *admin*, Penyimpanan data*,*

**1. Usecase Diagram**

****

**Gambar 4.1 *Usecase Diagram***

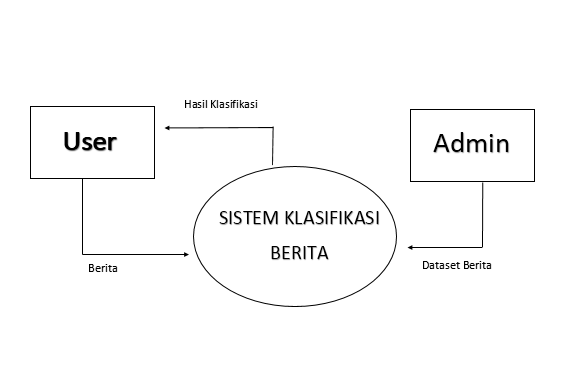
Pada gambar 4.1 di atas menjelaskan tentang alur *system*  yang dibuat. Dalam diagram di atas dapat dilihat bahwa user hanya bias memberikan inputa untuk klasifikasi berita sedangkan admin dapat menambah dataset ke dalam database.

**Tabel 4.1 *Usecase* Deskripsi**

|  |
| --- |
| Nama *Usecase* Alur Sistem |
| Aktor *User, Admin*  *Normal Course* 1. User Memasukkan berita yang ingin diklasifikasi  2. Admin login untuk menginputkan *database*  *Pre-Condition* 1. User memiliki berita yang ingin diklasifikasi  2. Admin terdaftar dalam system  *Post-Condition* 1. User mendapatkan hasil klasifikasi  2. Admin Berhasil mengiputkan data |

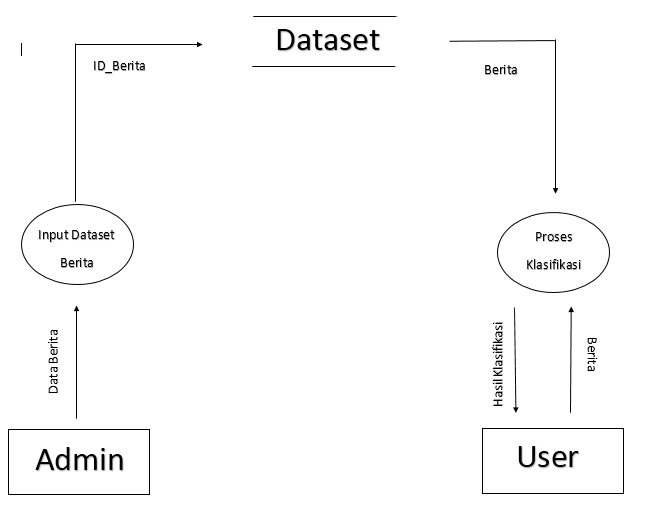
**2. *Diagram Konteks***

Dalam pembuatan system diperlukan *diagram konteks* yang digunakan untuk mempermudah memahami aliran data dari aplikasi yang dibuat Pada Gambar 4.2 diketahui bahwa user memberikan *input* kedalam system klasifikasi berita dan admin memberikan *input* berupa dataset berita sehingga *system* dapat memberikan *output* hasil identifikasi berita kepada user.

** **Gambar 4.2 Diagram Konteks**

**2.1 Aliran Data**

Pada tahap ini, alur diagram yang sudah dibuat pada diagram konteks akan diuraikan dengan rinci sehingga didapat aliran yang detail dan dapat dilihat dengan jelas.Berikut adalah Aliran data pada *system* klasifikasi berita *hoax :*

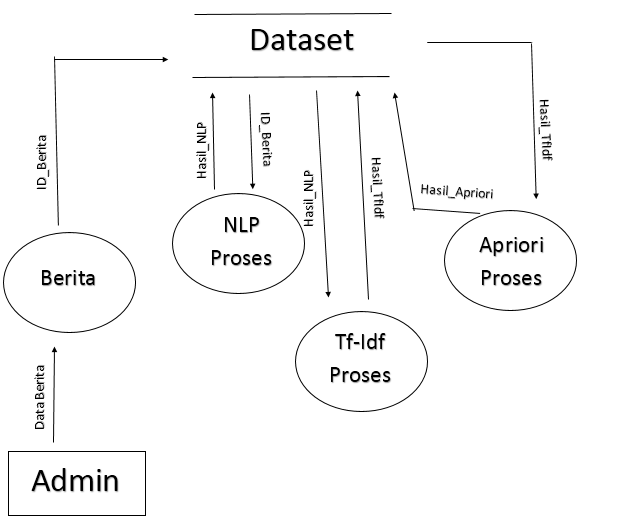
**

**Gambar 4.3 Aliran Data**

Pada gambar 4.3 Admin dapat memberikan *input* berupa *input dataset* berita yang nantinya akan dijadikan *dataset* untuk proses klasifikasi sedangkan user dapat memberikan *input* berupa berita yang nantinya akan diklasifikasi dengan menggunakan dataset input dari admin.

**2.2 Aliran Data *Admin***

Dalam tahap ini, aliran data yang telah dibuat pada aliran data akan dijabarkan lebih rinci sehingga dapat dipahami dengan jelas. Berikut aliran data admin :

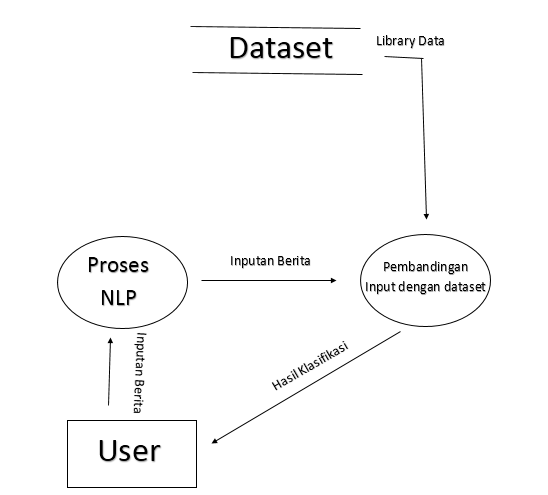


**Gambar 4.4 *Data Flow Diagram Level 1 Input Dataset***

Pada gambar 4.4 dapat diamati bahwa *input* dari admin akan di lanjutkan kedalam *database* lalu di proses dengan NLP proses pencarian kata umum dari sebuah kata sebelum diolah ,Tf-Idf yakni proses penacrian bobot kata dan Apriori *methode* untuk mencari nilai support *.*Sehingga didapat dataset yang bias digunakan untuk proses klasifikasi pada *input* user.

**2.3 Aliran Data *User***

Setelah dilakukan *input dataset,* Selanjutnya adalah proses *input* dari user. Berikut adalah Aliran data *User* pada gambar4.5 *:*

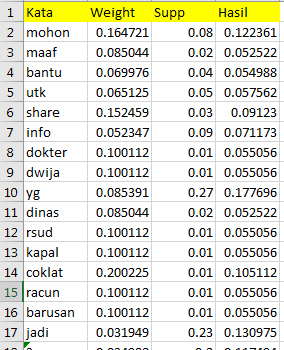


**Gambar 4.5Aliran Data User**

Pada tahap ini dilakukan proses NLP(*Natural Language Processing*) yang digunakan untuk pencarian kata natural.Selanjutnya, dilakukan proses pencarian kata pada dataset yang telah dibuat.Setelah proses pencarian maka akan mucul hasil dari klasifikasi berita.

**2.3 Penyimpanan *Data***

Untuk penyimpanan dataset , aplikasi yang dibuat memiliki 2 buah tempat penyimpanan yakni berupa excel. Proses pertama yakni penyimpanan data yang di *input* oleh admin kedalam excel. Langkah selanjutnya yakni proses pengolahan data sehingga didapat nilai suatu kata yang disimpan dalam bentuk .csv. Dilakukan penyimpanan dalam bentuk excel dikarenakan data yang disimpan sangat banyak sehingga untuk memaksimalkan kinerja proses maka penyimpanan dilakukan didalam excel.

******

**Gambar 4.6 Penyimpanan Hasil Proses dalam Bentuk Excel**

Gambar 4.6 yakni penyimpanan excel yang digunakan untuk menyimpan data yang telah di proses.

**B. Penulisan Code Program**

Pada bab ini akan dijelaskan secara detail penulisan code program yang digunakan untuk proses klasifikasi berita *hoax* menggunakan apriori.

1. *Preprocessing* Dokumen

Pada tahap ini yakni pengolahan dokumen yang masih mentah menjadi dokumen yang dapat diproses oleh system. Data latih yang digunakan yakni 100 buah data berita yang terindikasi berita *hoax.* Alur proses pada *preprocessing* ini yakni *tokenizing* (pemisahan kalimat) ,*Case Folding* (penyetaraan huruf menjadi huruf kecil), *Stopwords removal* (Penghilangan kata tidak penting) dan Stemming (Perubahan menjadi kata dasar). Berikut adalah source code yang digunakan untuk *preprocessing :*

|  |
| --- |
| #Case Folding /Huruf Kecil  text =text.casefold()  #tokenizin  text=text.split()    #Stopword Removal / Menghilangkan kata tidak penting  factory = StopWordRemoverFactory()  stopword = factory.create\_stop\_word\_remover()  text = stopword.remove(text)  #Steeming kata dasar  factory = StemmerFactory()  stemmer = factory.create\_stemmer()  text=stemmer.stem(text) |

**Gambar 4.7 *Source Code Preprocessing***

1. Proses *Input* Data Latih

Setelah melalui *preprocessing* maka selanjutnya data akan di masukkan ke dalam *database* . *Source code input* data kedalam *database* dapat dilihat pada gambar 4.8*.*

|  |
| --- |
| def database(x,b,c):  import pymysql  # Open database connection  db = pymysql.connect("localhost","xiinlaw","Arcasevenvold04","skripsi" )  # prepare a cursor object using cursor() method  cursor = db.cursor()    sql = """INSERT INTO beritax (Judul\_Berita, #ganti  Berita,JumlahKata)  VALUES (%s,%s,%s)"""  val=(x,b,c)  try:  # Execute the SQL command  cursor.execute(sql,val)  # Commit your changes in the database  db.commit()  except:  # Rollback in case there is any error  db.rollback()  # disconnect from server  db.close() |

**Gambar 4.8 *Source Code Input Data* Latih**

1. Proses hitung Tf-Idf dan Apriori

Setelah melalui proses input db, maka langkah selanjutnya yakni proses perhitungan untuk membuat *library* kata yang nantinya digunakan untuk pembanding data uji. Perhitungan dilakukan dengan menggunakan metode Tf-Idf dan Apriori. S*ource code* dari proses hitung dapat dilihat pada gambar 4.9.

|  |
| --- |
| # idf dan apriori  #jumlah dokumen yang mengandung kata  sqli = "SELECT ID FROM beritax WHERE Berita LIKE %s"  b = berita[r]  a = ('%' + b + '%')  val = (a)  cursor.execute(sqli, val)  a = cursor.fetchall()  rr = len(a)  rr=float(rr)  Kata.append(berita[r])  print('Kata ke : ',r)  print(berita[r],' Jmlh :',rr)  Idf = math.log(100 / rr)  apriori = rr/100  TfIdf=Tf\*Idf |

**Gambar 4.9 *Source Code* proses hitung**

1. Proses *Create Library Excel*

Dalam tahap ini yakni tahap penyimpanan kata yang sudah diproses dan dihitung dengan menggunakan apriori dan tf-idf. Hasil perhitungan tersebut akan disimpan dalam bentuk .xlsx. S*ource code* yang digunakan untuk proses *create library excel* dapat dilihat pada gambar 4.10*.*

|  |
| --- |
| # input kata on excel  kata = sh.cell(column=1, row=roww)  kata.value = berita[r]  # Weight  weight = sh.cell(column=2, row=roww)  weight.value = TfIdf  # Support  supp = sh.cell(column=3, row=roww)  supp.value = apriori  # Hasil  hasil = sh.cell(column=4, row=roww)  hasil.value = (apriori+TfIdf)/2  print('Jumlah Kata Yang tersimpan : ', roww-1)  r += 1  roww += 1  wk.save('uji.xlsx') #ganti |

**Gambar 4.10 *Source Code Input Library Into Excel***

1. Proses Input data Uji dan Pembandingan dengan Data Latih

Pada tahap ini user akan memasukkan data berita yang nantinya data akan diproses oleh system untuk dibandingkan dengan data latih sehingga didapat hasil *output*  yang diinginkan oleh user. S*ource code* yang digunakan dapat dilihat pada gambar 4.11.

|  |
| --- |
| def hitung(text):  #pembandingan dengan librari  nlpcount=0  text=text.split()  jmlkata=len(text)  sigmakatasama=0  #print(jmlkata)  for g in text:  wk = openpyxl.load\_workbook("kata.xlsx") #ganti  sh = wk.active  rows = sh.max\_row  for i in range(1, rows + 1):  c = sh.cell(i,1)  h=sh.cell(i,4)  #print(c.value)  if(text[nlpcount]==c.value):  sigmakatasama+=h.value  # print(c.value,h.value,Hasilnlp[nlpcount])  nlpcount+=1  print(sigmakatasama,jmlkata)  ratahasil=sigmakatasama/jmlkata  return ratahasil |

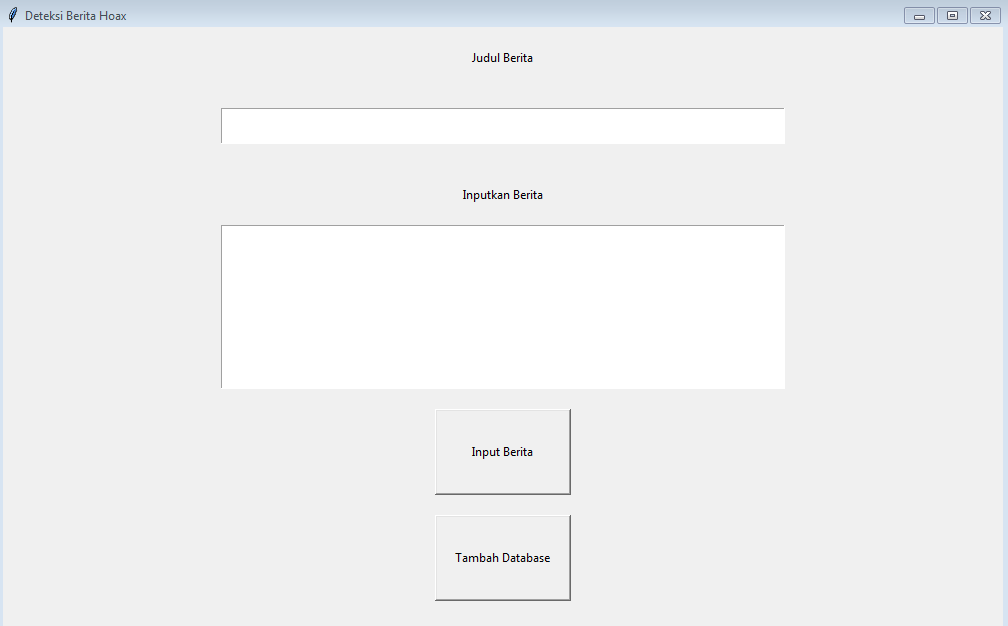
**Gambar 4.11 Proses Pencarian Hasil *Input* Data Uji**

**C. Antar Muka Aplikasi**

Pada tahap ini yakni menjelaskan antar muka *system* yang telah dibuat guna mempermudah pengguna untuk mengoprasikan *system* yang telah dibuat. Terdapat 2 buah antar muka yakni halaman utama (halaman *input* data uji) dan halaman input dataset.

1. Halaman Utama

Pada halaman utama ini terdapat satu buah text kolom dan 2 buah button yang digunakan untuk memasukkan data dan untuk menuju ke halaman input dataset. User dapat memasukkan berita yang ingin diidentifikasi ke dalam text kolom. Selanjutnya user dapat menekan tombol input untuk melihat hasil. Hasil akan di tampilkan berbetuk pop up window yang berisi Nilai hoax suatu berita dan terindikasi hoax atau tidak.



**Gambar 4.12 Halaman Utama**

****

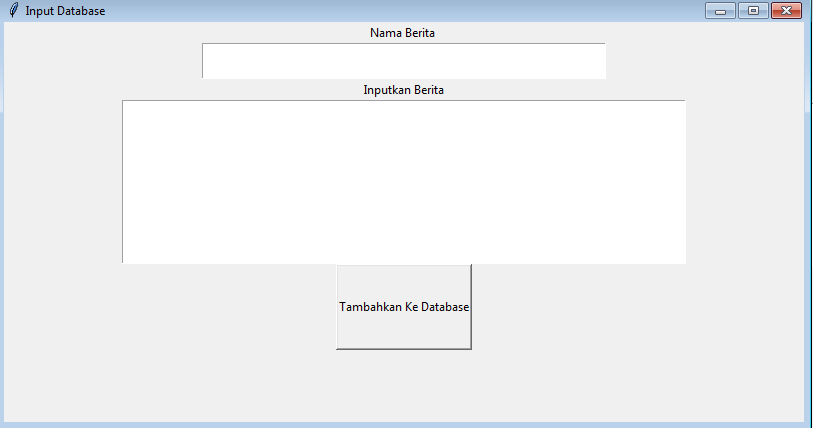
**Gambar 4.13 Hasil Berita Tidak *Hoax***

****

**Gambar 4.14 Hasil Berita Terindikasi *Hoax***

1. Halaman Input Dataset

Pada halaman ini terdapat 2 buah text kolom meliputi judul berita dan isi berita. Setelah admin menginputkan data maka secara otomatis data yang di inputkan akan masuk kedalam database.

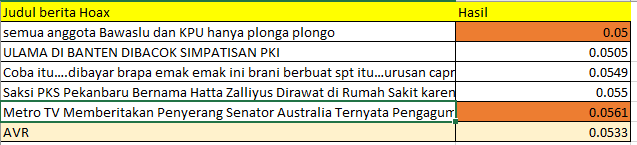


**Gambar 4.15 Input Dataset Berita**

**D. Hasil dan Pengujian *System***

Pada tahap ini adalah tahap pengujian *system* dengan menggunakan data uji yang telah di siapkan berupa 16 data terindikasi berita *hoax* dan 14 buah data berita berupa berita real. Untuk menentukan nilai batas dilakukan uji coba menggunakan 5 buah data terindikasi *hoax*  yang nantinya dicari rataan nilai sehingga didapat hasil nilai batas. Selanjutnya, *system* akandiuji menggunakan batas yang telah ditentukan dan dicari hasil dari *precision and recall*  serta tingkat akurasi menggunakan rumus yang telah dijabarkan pada bab II.

1. Pencarian Nilai Batas

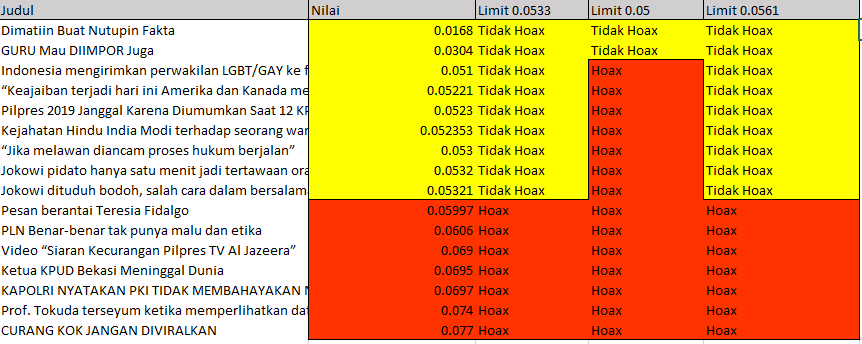


**Gambar 4.16 Data Uji Nilai Batas**

Untuk mencari nilai batas dilakukan uji coba dengan menggunakan 5 buah data uji berita terindikasi *hoax*.Hasil rata-rata, batas atas dan batas bawah dari nilai data uji akan digunakan untuk nilai batas dari pengujian data uji selanjutnya.

1. Proses Uji Hasil Data Uji

Pada tahap ini dilakukan 2 buah pengujian yakni pengujian data berupa berita hoax dan pengujian data berupa data real.Proses pengujian dilakukan dengan menggunakan 3 buah nilai batas yang telah ditentukan yakni 0.0533, 0.0561 dan 0.05.



**Gambar 4.17 Data Uji Berita Hoax**



**Gambar 4.18 Data Uji Berita Real**

**Tabel 4.1 Keterangan Tabel**

|  |  |
| --- | --- |
| **Keterangan** | **Posisi** |
| *True Positive* (Berita *Hoax* Terklasifikasi hoax) | Gambar 4.17 bagian bawah |
| *False Negative* (Berita *Hoax* Terklasifikasi Real) | Gambar 4.17 bagian atas |
| *True Negative* (Berita Real Terklasifikasi Real) | Gambar 4.18 bagian atas |
| *False Positive* (Berita Real Terklasifikasi *Hoax*) | Gambar 4.18 bagian bawah |

2.1 Skenario 1

Skenario 1 proses uji dengan menggunakan nilai batas rata-rata dari pencarian nilai limit yakni 0,0533.

**Tabel 4.2 Nilai dari Komponen Perhitungan dengan Batas 0.0533**

|  |  |
| --- | --- |
| Berita | Jumlah |
| *True Positive* (tp)  *False Positive* (fp)  *False Negative* (fn)  *True Negative* (tn) | 7  3  9  11 |

Berdasarkan nilai dari setiap komponen, maka dapat ditentukan nilai *Precision, Recall* dan *Accuracy* sebagai berikut :

*Precision :* tp/(tp+fp) *Recall :* tp/(tp+fn)

= 7/10 = 7/16

= 0,7 = 0,437

*Accuracy = x100*

*=* (7 + 11) / 30 x100

= 60 %

2.2 Skenario 2

Skenario 2 proses uji dengan menggunakan nilai batas bawah dari pencarian nilai limit yakni 0,05.

**Tabel 4.2 Nilai dari Komponen Perhitungan dengan Batas 0.05**

|  |  |
| --- | --- |
| Berita | Jumlah |
| *True Positive* (tp)  *False Positive* (fp)  *False Negative* (fn)  *True Negative* (tn) | 14  3  2  11 |

Berdasarkan nilai dari setiap komponen, maka dapat ditentukan nilai *Precision, Recall* dan *Accuracy* sebagai berikut :

*Precision :* tp/(tp+fp) *Recall :* tp/(tp+fn)

= 14/17 = 14/16

= 0,82 = 0,875

*Accuracy = x100*

*=* (14 + 11 ) / 30 x100

= 83,33 %

2.3 Skenario 3

Skenario 3 proses uji dengan menggunakan nilai batas atas dari pencarian nilai limit yakni 0,0561.

**Tabel 4.2 Nilai dari Komponen Perhitungan dengan Batas 0.0561**

|  |  |
| --- | --- |
| Berita | Jumlah |
| *True Positive* (tp)  *False Positive* (fp)  *False Negative* (fn)  *True Negative* (tn) | 7  2  9  12 |

Berdasarkan nilai dari setiap komponen, maka dapat ditentukan nilai *Precision, Recall* dan *Accuracy* sebagai berikut :

*Precision :* tp/(tp+fp) *Recall :* tp/(tp+fn)

= 7/9 = 7/16

= 0,77 = 0,437

*Accuracy = x100*

*=* (7 + 12) / 30 x100

= 63,33 %

**E. Pembahasan**

Berdasarkan hasil pengujian *system* nilai batas 0.0533 menghasilkan akurasi sebesar 60% , untuk nilai batas 0,561 mendapatkan akurasi sebesar 63,33 % dan untuk nilai batas 0.5 mendapatkan nilai akurasi sebesar 83,33%.Dengan data dari ke 3 buah skenario dapat disimpulkan bahwa nilai batas yang paling efektif digunakan untuk proses klasifikasi berita *hoax* pada system yakni 0.5 dengan tingkat akurasi 83,33 % dengan nilai *precission and recall* 82% dan 87% . Dapat dijelaskan bahwa pengubahan nilai limit sangatlah berpengaruh pada tingkat akurasi system.Dikarenakan nilai dari data uji yang berbeda-beda sehingga untuk dapat mencari kefektifan sebuah system digunakan ujicoba beberapa skenario.