

Pengembangan Aplikasi Helpdesk Ticketing System Dengan Algoritma Multinomial Naïve Bayes Classifier

Muhammad Dery Setiawan¹, Eliyani*²

^{1,2} Prodi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana
Jl. Meruya Selatan No.31, RT.4/RW.1, Meruya Sel., Kec. Kembangan, Kota Jakarta Barat, Daerah Khusus
Ibukota Jakarta 11610

Email : 41517120135@student.mercubuana.ac.id¹, eliyani@mercubuana.ac.id²

*) Corresponding Author

(received: 25-06-22, revised: 14-07-22, accepted: 18-08-22)

Abstract

PT FGS Infotama is an Online Trading Application service provider company that does not only focus on the early stages of implementation, but also on several other important matters, such as training, development, maintenance and socialization related to the applied software. Currently, complaints are received via email, telephone or whatsapp, so it is relatively difficult to manage incoming complaints. This research is intended to create a web based helpdesk ticketing system that is equipped with a classification of the level of urgency of complaints, and several other features such as Frequently Asked Questions, News which contains information to clients from the Company, and Notification of incoming tickets and when tickets have been completed. The method used to develop this application is the Rapid Application Development method, while the algorithm used to classify complaints is the Multinomial Naive Bayes Classifier. Using a data split of 80% training data and 20% test data, the accuracy for data from Whatsapp is 73.6% and data from email is 61.9% with a precision level for High and Medium Classification reaching 100%, and Low Classification 71%.

Keyword: multinomial naïve bayes classifier, helpdesk ticketing system, classification

Abstrak

PT FGS Infotama adalah perusahaan penyedia jasa Aplikasi Online Trading yang tidak hanya fokus pada implementasi tahap awal, melainkan beberapa hal penting lainnya, seperti pelatihan, pengembangan, perawatan dan sosialisasi terkait dengan software yang diterapkan. Saat ini keluhan diterima melalui email, telepon maupun whatsapp sehingga relatif sulit dalam mengelola keluhan yang masuk. Tujuan dari penelitian ini adalah membuat helpdesk ticketing system berbasis yang dilengkapi dengan klasifikasi tingkat urgensi keluhan, dan beberapa fitur lain seperti Frequently Asked Question, News yang berisi informasi ke klien dari Perusahaan, dan Notifikasi tiket masuk dan ketika tiket telah diselesaikan. Metode yang digunakan untuk mengembangkan aplikasi ini adalah Rapid Application Development, sementara algoritma yang digunakan untuk mengklasifikasi keluhan adalah Multinomial Naive Bayes Classifier. Algoritma ini diimplementasikan pada form create ticket untuk mengklasifikasikan tingkat urgensi tiket rendah, sedang dan tinggi. Menggunakan split data 80% data latih dan 20% data uji, diperoleh akurasi untuk data dari Whatsapp 73.6% dan data dari email sebesar 61.9% dengan tingkat presisi untuk Klasifikasi High dan Medium mencapai 100%, dan Klasifikasi Low 71%.

Kata Kunci: multinomial naïve bayes classifier, helpdesk ticketing system, klasifikasi

I. Pendahuluan

PT FGS Infotama adalah vendor bidang finansial dan sekuritas yang sudah beroperasi sejak tahun 2007 dan berlokasi di Jakarta. Perusahaan ini menyediakan jasa Aplikasi Online Trading yang tidak hanya fokus pada implementasi tahap awal saja, melainkan beberapa hal penting lainnya, seperti pelatihan, pengembangan, perawatan dan sosialisasi terkait dengan software yang diterapkan.

Untuk meningkatkan kualitas pelayanan kepada pelanggan, Perusahaan membuka jalur komunikasi untuk menampung pertanyaan maupun masukan dari pelanggan, namun masih menggunakan sistem email, whatsapp

dan telepon. Sementara penggunaan media-media tersebut akan kurang efisien dan efektif untuk jumlah keluhan yang semakin banyak karena semakin sulit melakukan kontrol terhadap keluhan-keluhan tersebut (Darmawan dan Senjaya, 2018 dalam [1]).

Berdasarkan analisis pada sistem berjalan yang ada pada saat ini di PT FGS Infotama, terdapat beberapa hambatan dalam penanganan keluhan klien, antara lain:

1. Keluhan disampaikan menggunakan sistem *email*, *whatsapp* dan telepon, hal ini membuat tim kesulitan dalam mengontrol keluhan yang masuk, terkadang klien keliru dalam menyampaikan keluhan sehingga keluhan diterima oleh divisi yang tidak memiliki kapabilitas dalam menyelesaikan keluhan tersebut
2. Tim perlu menganalisis keluhan yang perlu diselesaikan terlebih dahulu, proses ini akan menunda waktu penyelesaian keluhan
3. Pelanggan terkadang mengeluhkan masalah yang sama dengan pelanggan lainnya sehingga tim harus menjelaskan hal tersebut berulang-ulang, hal ini memperlambat tim dalam menangani keluhan yang lain karena harus menjawab hal yang sudah di tanyakan secara berulang-ulang
4. Tim harus menghubungi pelanggan satu per satu ketika ada informasi yang ingin disampaikan ke pelanggan.

Untuk meningkatkan efektivitas penanganan keluhan pelanggan, penelitian ini dimaksudkan untuk mengembangkan *smart helpdesk ticketing system*. *Helpdesk* adalah sistem manajemen untuk menanggapi permintaan *user* berupa pertanyaan, layanan, dukungan teknis, atau keluhan tentang layanan pada suatu organisasi, menggunakan sistem penomoran (*request ticket*) untuk melacak tindakan penyelesaian [2]. Beberapa penelitian telah dilakukan untuk pengembangan *Helpdesk Ticketing System* [3], [4], [5], [6], dan [7]. Sistem umumnya dikembangkan berbasis *web* dengan berbagai metode pengembangan, antara lain *Prototype*, *Waterfall*, dan *Rapid Application Development*. Diperoleh hasil, sistem ini selain membuat pengguna lebih mudah dalam melaporkan masalah, juga membuat teknisi *IT* lebih mudah dalam mengelola laporan.

Untuk mempercepat penanganan, keluhan dapat diklasifikasikan berdasarkan kriteria tertentu yang dibutuhkan. *Naïve bayes classifier* digunakan untuk mengklasifikasi keluhan agar keluhan tersebut bisa langsung didistribusikan kepada bidang-bidang yang relevan dalam penanganannya tanpa perlu melalui operator [8]. Pada penelitian ini, keluhan diklasifikasikan berdasarkan tingkat urgensinya menggunakan algoritma *Multinomial Naive Bayes Classifier*.

Multinomial Naïve Bayes Classifier adalah model yang dikembangkan dari algoritma *Bayes* untuk mengklasifikasi teks, di mana klasifikasi ditentukan berdasarkan perhitungan frekuensi setiap kata yang muncul pada dokumen [9]. Algoritma *Multinomial Naive Bayes Classifier* telah diujicoba untuk mengklasifikasi surat keluar sehingga nomor surat dapat ditentukan secara otomatis. Tingkat *accuracy*, *precision*, *recall*, dan *F-measure* dari penerapan algoritma tersebut untuk kepentingan ini berturut-turut sebesar 89,58%, 79,17%, 78,72%, dan 77,05% [10]. Algoritma ini juga digunakan pada penelitian pengelolaan teks guna mengetahui akurasi hasil program pengkategorian otomatis dengan hasil akurasi yang kurang baik pada kategori politik yaitu 40% untuk 100 data latih dan 60% untuk 200 data latih namun akurasi cukup baik mencapai 90% pada kategori ekonomi baik dengan 100 maupun 200 data latih [11]. Klasifikasi artikel berita secara otomatis dengan menggunakan 10.500 *dataset* dan tujuh kategori, kinerja algoritma ini menunjukkan tingkat *accuracy* 96%, *precision* 96%, *recall* 96% dan *f1-score* 96% [12].

II. Metodologi Penelitian

Aplikasi *Helpdesk Ticketing System* pada penelitian ini dikembangkan menggunakan metode *Rapid Application Development (RAD)*. Keunggulan metode ini antara lain waktu pemrosesan menjadi lebih singkat, dan sistem aplikasi dihasilkan secara cepat dan akurat. Tiga tahap dalam metode RAD yaitu: *requirement planning*, *workshop design*, dan implementasi [10].

1. Requirement Planning

Wawancara dan observasi digunakan untuk mengumpulkan data dalam rangka menguraikan sistem atau cara kerja di perusahaan sehingga dapat diketahui proses klasifikasi serta hambatan yang terjadi di perusahaan tersebut guna perencanaan sistem yang efektif dan efisien dalam penanganan keluhan pelanggan.

2. Workshop Design

Aplikasi dirancang menggunakan *UML (Unified Modeling Language)* yaitu *use case*, diagram aktivitas dan *Entity Relationship Diagram* [13].

3. Implementasi

Web dibangun menggunakan bahasa pemrograman PHP 7.4 dan Laravel 8 dan algoritma *Multinomial Naive Bayes Classifier* untuk mengklasifikasikan tingkat urgensi keluhan dikembangkan menggunakan bahasa pemrograman Python.

III. Hasil dan Pembahasan

A. Sistem Aplikasi

Analisis sistem adalah penguraian suatu sistem informasi yang lengkap ke dalam komponen-komponennya sehingga masalah, peluang, hambatan yang dihadapi, dan kebutuhan yang diharapkan dapat diidentifikasi dan dievaluasi, serta dapat diusulkan perbaikannya [14]. *Work flow* sistem penanganan keluhan klien pada perusahaan yang diamati disajikan pada Gambar 1. Sistem penanganan keluhan ini masih dilakukan secara manual, seperti penerimaan keluhan masih menggunakan media sosial atau email, dan keluhan dan pertanyaan dianalisis dan dijawab satu demi satu.



Gambar 1. *Work flow* penanganan keluhan

Untuk mengatasi kelemahan tersebut, diharapkan penanganan keluhan pelanggan dapat berlangsung lebih efektif dan efisien, dengan fitur-fitur yang mampu mempermudah *user* dalam menyampaikan keluhan hingga mempercepat proses penyelesaian keluhan, seperti :

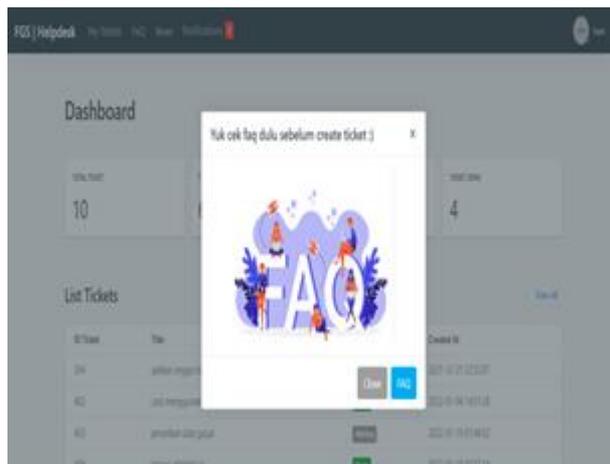
1. Memberikan keluhan secara *online* melalui aplikasi *web*
2. Keluhan dapat langsung tersampaikan ke divisi tujuan
3. Klasifikasi tingkatan urgensi keluhan secara sistematis
4. Menampilkan notifikasi keluhan masuk maupun keluhan yang sudah terselesaikan
5. Menampilkan informasi-informasi yang ingin disampaikan ke pelanggan.
6. Menampilkan informasi berupa pertanyaan-pertanyaan yang sering diajukan

Beberapa fitur yang ada pada aplikasi yang dibangun meliputi: pengelolaan data staff dan pelanggan, pengelolaan data laporan tiket, *Frequently Ask Question* (FAQ) dan *News*. *Workflow* sistem yang dikembangkan ditunjukkan pada Gambar 2.



Gambar 2. *Workflow Helpdesk Ticketing System*

Usecase Diagram disajikan pada Gambar 3 di mana pengguna aplikasi terdiri dari pelanggan, admin, IT Support, dan *Developer*. Pelanggan dapat menginput tiket, melihat status tiket, melihat FAQ dan *News*. Untuk dapat melakukan semua aktivitas, pengguna harus melakukan login terlebih dahulu dengan memasukkan email dan *password* yang terdaftar. Halaman awal aplikasi berupa halaman login disajikan pada Gambar 4. Sebelum



Gambar 5. Pop Up FAQ

Selanjutnya pengguna dapat mengajukan tiket dan melihat *progress* penyelesaian tiket.

B. Implementasi Algoritma

Untuk melakukan klasifikasi keluhan klien, data yang digunakan berupa keluhan yang diberikan klien kepada perusahaan melalui aplikasi *Whatsapp* dengan jumlah data sebanyak 624 data dan melalui email sebanyak 104 data. Data-data tersebut dikelompokkan ke dalam 3 kategori yaitu *high*, *medium* dan *low* berdasarkan tingkat urgensinya. Tingkat urgensi tinggi (*high*) seperti *server down*, namun jika keluhan meliputi hal yang harus ditangani tetapi relatif bisa ditunda maka akan dimasukkan ke kategori *medium*, dan untuk keluhan yang berupa pertanyaan terkait fungsionalitas seperti bagaimana cara menggunakan aplikasi akan dimasukkan ke kategori *low*. Beberapa contoh data keluhan dan kategorinya disajikan pada Gambar 6.

1	questions,label			
2	tolong refresh memory dan restart server,high			
3	tolong restart service desktop skrg,high			
4	tolong setup server baru untuk dev,medium			
5	tolong setup server baru untuk prod ,medium			
6	tolong setup server dev,medium			
7	tolong setup server drc,medium			
8	tolong setup server live,medium			

Gambar 6. Sampel data keluhan

Beberapa tahapan dalam pengklasifikasian tingkat urgensi keluhan menggunakan algoritma *multinomial naïve bayes classifier*, di antaranya menghilangkan tanda baca dan angka, menghapus kata-kata yang kurang informatif dari teks (*stopword*), dan melakukan *stemming* untuk menghilangkan infleksi kata ke bentuk dasarnya dengan *script* pemrograman seperti yang ditampilkan pada Gambar 7, Gambar 8 dan Gambar 9. Pelabelan data dilakukan secara manual.

```
[3] def remove_punctuation_and_number(text):  
    '''a function for removing punctuation'''  
    import string  
    import re  
    for sp in string.punctuation:  
        text = text.replace(sp, " ")  
    text = re.sub(r"\d+", "", text)  
    return text.replace('/\s\s+/g', ' ')  
  
df['questions'] = df['questions'].apply(remove_punctuation_and_number)
```

Gambar 7. Menghilangkan Tanda Baca dan Angka

StopWords

```
[ ] sw = ["aja", "agan", "barang", "bell", "baru", "ada", "adalah", "ajaa", "akan", "aku", "atas", "boot", "boss", "agak", "agar", "apa", "bisa", "bisa", "alhamdulillah", "alhamdulillah", "akhir", "allah", "ane", "arn", "atau", "audah", "bang", "aaaa", "about", "adik", "ah", "admin", "an"]  
def stopwords(text):  
    '''a function for removing the stopwords'''  
    import re  
    text = [word.lower() for word in text.split()]  
    for word in text:  
        for stop in sw:  
            if word==stop:  
                text.remove(word)  
    text = " ".join(text)  
    text = re.sub(r'(\.+)1+', r'\1', text)  
    return text  
  
df['questions'] = df['questions'].apply(stopwords)  
df.head()
```

	questions	label
0	alert disk usaga sudah dari mesin fgs	high
1	berapa order pas open market	medium
2	kesalahan input saham tolong bantu di wd	medium
3	mook derivative statusnya masih sendng dari p...	medium
4	saham masuk ke user sudah bisa di transaksikan	medium

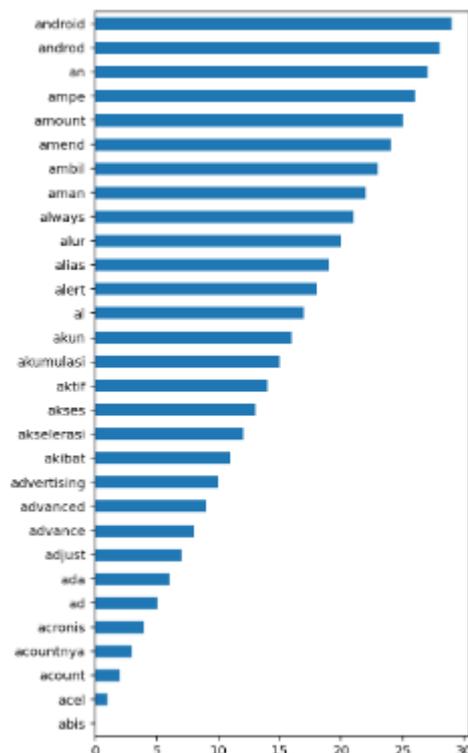
Gambar 8. Stopword

```
! #import Sastrawi package  
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory  
# create stemmer  
factory = StemmerFactory()  
stemmer = factory.create_stemmer()  
def stemming(text):  
    text = [stemmer.stem(word) for word in text.split()]  
    return " ".join(text)  
df['questions'] = df['questions'].apply(stemming)
```

Gambar 9. Stemming Menggunakan Sastrawi

Pembobotan kata dan *word vectorization* dengan mengubah kumpulan teks menjadi *numerical feature vectors* dilakukan dengan metode *tf-idf* (*Term Frequency — Inverse Document Frequency*). Contoh hasil pembobotan kata disajikan pada Gambar 10 dan *script* program *word vectorization* disajikan pada Gambar 11. Pada tahap *word vectorization* digunakan *max_features=5000* yang artinya akan ada maksimal 5000 kata yang unik. Selanjutnya

dilakukan *vectorized Train_X* dan *Test_X* sehingga nantinya setiap baris akan berisikan *integer number* yang unik dan kepentingan suatu kata diasosiasikan dengan kalkulasi yang dilakukan oleh *tf-idf*. *Split* data dilakukan dengan membagi data menjadi 80% data *training* dan 20 % data *testing* dengan *script* seperti yang ditunjukkan pada Gambar 12.



Gambar 10. Contoh hasil kata teratas dengan pembobotan *tf-idf*

```
# Word Vectorization
Tfidf_vect = TfidfVectorizer(max_features=5000)
Tfidf_vect.fit(df['questions'])
Train_X_Tfidf = Tfidf_vect.transform(Train_X)
Test_X_Tfidf = Tfidf_vect.transform(Test_X)
```

Gambar 11. *Word Vectorization*

```
Train_X, Test_X, Train_Y, Test_Y = model_selection.train_test_split( df['questions'], df['label'], test_size=0.2 )
```

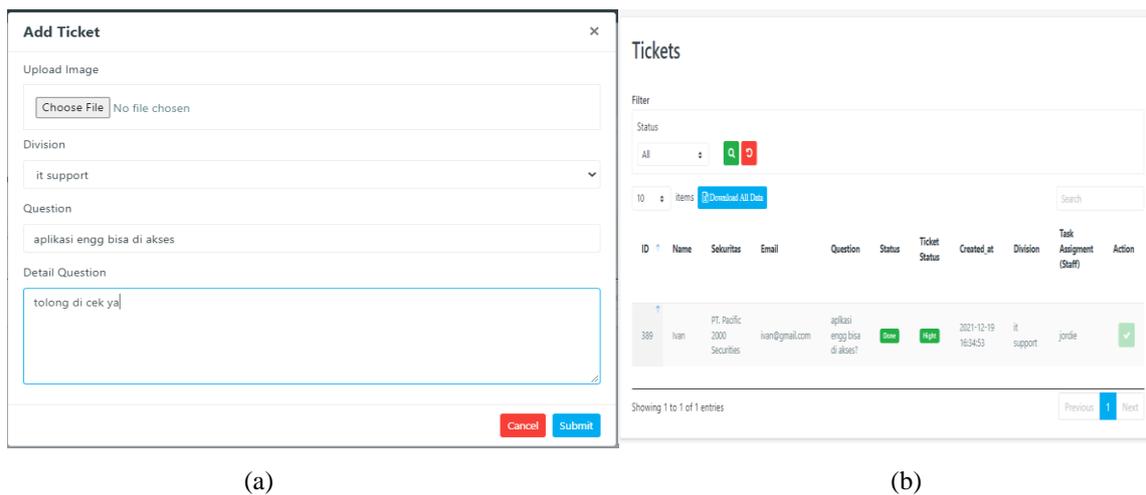
Gambar 12. *Split Data Training dan Data Testing*

Hasil pengujian yang telah dilakukan menunjukkan bahwa akurasi data yang berasal dari email sebesar 61.90% dan data yang berasal dari *Whatsapp* sebesar 73.6%. Lebih kecilnya tingkat akurasi data dari *email* bisa disebabkan karena jumlah data yang lebih sedikit dibandingkan jumlah data dari *Whatsapp*. Laporan klasifikasi diperoleh dari *confusion matrix* yang memberikan informasi lebih rinci tentang kinerja model [15]. Tingkat presisi, *recall*, *F1-score*, dan *support* hasil klasifikasi disajikan pada Tabel 1. Terlihat pada Tabel 1., *tingkat presisi untuk Klasifikasi High dan Medium mencapai 100%, dan Klasifikasi Low 71%*.

Tabel 1. Hasil Klasifikasi

Klasifikasi	Precision	Recall	F1-Score	Support
High	100%	19%	32%	21
Medium	100%	41%	58%	27
Low	71%	100%	83%	79

Pengujian penerapan algoritma pada aplikasi *Helpdesk Ticketing System* yang dibangun dimaksudkan untuk memastikan luaran dari aplikasi apakah sesuai dengan data yang telah diketahui kategorinya. Misalnya diberikan data keluhan yang termasuk kategori *high*, apakah sistem dapat mengklasifikasikan keluhan tersebut sebagai *high*. Gambar 13(a) menunjukkan *form input* data keluhan, dan Gambar 13(b) menunjukkan hasil pengkategorian keluhan, di mana hasil uji aplikasi menunjukkan luaran tingkat urgensi sama seperti yang diharapkan.



Gambar 13. (a) *Form Input* Keluhan dan (b) Hasil klasifikasi Keluhan pada Aplikasi

Pengujian juga dilakukan menggunakan perhitungan secara manual menggunakan rumus dari algoritma *multinomial naive bayes classifier*. Model *multinomial* memperhitungkan frekuensi setiap kata yang muncul pada dokumen. Misal terdapat dokumen *d* dan himpunan kelas *c*. Untuk memperhitungkan kelas dari dokumen *d*, maka dapat dihitung dengan rumus [16] :

Rumus :

$$P(c|term \text{ dokumen } d) = \frac{P(c) \times P(t_1|c) \times P(t_2|c) \times P(t_3|c) \times \dots \times P(t_n|c)}{P(t_n|c)} \quad (2.1)$$

- Keterangan :
- $P(c)$ = Probabilitas *prior* dari kelas *c*
 - t_n = Kata dokumen *d* ke-*n*
 - $P(c|term \text{ dokumen } d)$ = Probabilitas suatu dokumen termasuk kelas *c*
 - $P(t_n|c)$ = Probabilitas kata ke-*n* dengan kelas *c*

Probabilitas *prior* kelas *c* ditentukan dengan rumus:

$$P(c) = \frac{Nc}{N} \quad (2.2)$$

Keterangan :

N_c = Jumlah kelas c pada seluruh dokumen

N = Jumlah seluruh dokumen

Probabilitas kata ke- n ditentukan dengan menggunakan teknik *laplacian smoothing*:

$$(tn|c)^n = \frac{count(tn,c)+1}{count(c)+|v|} \quad (2.3)$$

Keterangan :

$count(t_n,c)$ = Jumlah *term* tn yang ditemukan di seluruh data pelatihan kategori c

$count(c)$ = Jumlah *term* di seluruh data pelatihan dengan kategori c

V = Jumlah seluruh *term* pada data pelatihan

Data hasil pengujian manual disajikan pada Tabel 2.

Tabel 2. Data Pengujian Manual

	<i>Doc</i>	<i>Questions</i>	<i>Label</i>
<i>Training</i>	1	kenapa aplikasi <i>unable to launch</i> ?	<i>high</i>
	2	kenapa aplikasi <i>web</i> tidak dapat di akses?	<i>high</i>
	3	kenapa <i>auto reboot</i> tidak jalan?	<i>high</i>
	4	kenapa <i>cash</i> masih belum masuk juga?	<i>medium</i>
	5	kenapa <i>cash</i> nasabah berbeda jauh dengan yang di bo?	<i>medium</i>
	6	kenapa <i>cash</i> nasabah masih belum masuk?	<i>medium</i>
	7	<i>service</i> apa saja yang perlu di jalankan ketika <i>mock</i> ?	<i>low</i>
	8	<i>service</i> apa saja yang tidak boleh di jalankan pada saat <i>mock</i> ?	<i>low</i>
<i>Testing</i>	9	kenapa aplikasi tidak bisa di akses?	?

Priors :

$$P(h) = \frac{3}{8}, P(m) = \frac{3}{8}, P(l) = \frac{3}{8}$$

Conditional Probabilities :

$$P(\text{kenapa}|h) = (3+1)/(17+35) = 4/52 = 2/26$$

$$P(\text{aplikasi}|h) = (2+1)/(17+35) = 3/52$$

$$P(\text{tidak}|h) = (2+1)/(17+35) = 3/52$$

$$P(\text{bisa}|h) = (0+1)/(17+35) = 1/52$$

$$P(\text{di}|h) = (1+1)/(17+35) = 2/52$$

$$P(\text{akses}|h) = (1+1)/(17+35) = 2/52$$

$$P(\text{kenapa}|m) = (3+1)/(21+35) = 4/56 = 2/28$$

$$P(\text{aplikasi}|m) = (0+1)/(21+35) = 1/56$$

$$P(\text{tidak}|m) = (0+1)/(21+35) = 1/56$$

$$P(\text{bisa}|m) = (0+1)/(21+35) = 1/56$$

$$P(\text{di}|m) = (1+1)/(21+35) = 2/56$$

$$P(\text{akses}|m) = (0+1)/(21+35) = 1/56$$

$$P(\text{kenapa}|l) = (0+1)/(20+35) = 1/55$$

$$P(\text{aplikasi}|l) = (0+1)/(20+35) = 1/55$$

$$P(\text{tidak}|l) = (1+1)/(20+35) = 2/55$$

$$P(\text{bisa}|l) = (0+1)/(20+35) = 1/55$$

$$P(\text{di}|l) = (2+1)/(20+35) = 3/55$$

$$P(\text{akses}|l) = (0+1)/(20+35) = 1/55$$

Choosing a Label :

$$P(h|d9) = 3/8 * 2/26 * 3/52 * 3/52 * 1/52 * 2/52 * 2/52 = 2.7313$$

$$P(m|d9) = 2/28 * 1/56 * 1/56 * 1/56 * 2/56 * 1/56 = 2.5939$$

$$P(l/d9) = 1/55 * 1/55 * 2/55 * 1/55 * 3/55 * 1/55 = 2.1675$$

Berdasarkan perhitungan tersebut jumlah terbesar berada pada prior $P(h|d9)$, maka untuk questions “kenapa aplikasi tidak bisa di akses?” memiliki label “high”. Hasil tersebut telah sesuai dengan hasil klasifikasi pada sistem.

IV. Kesimpulan

Aplikasi *Helpdesk Ticketing System* dibangun menggunakan metode *Rapid Application Development (RAD)* serta Algoritma *Multinomial Naive Bayes Classifier* untuk klasifikasi tingkat urgensi keluhan. Selain dilengkapi dengan fitur pengajuan tiket dan status tiket, aplikasi juga dilengkapi dengan fitur FAQ dan *News*. Semua fitur-fitur ini berjalan seperti yang diharapkan. Data yang digunakan untuk klasifikasi tingkat urgensi keluhan berdasarkan data keluhan oleh pelanggan yang disampaikan melalui *Whatsapp* dan e-mail perusahaan. Menggunakan split data 80% data latih dan 20% data uji, diperoleh akurasi untuk data dari *Whatsapp* 73.6% dan data dari email sebesar 61.9% dengan tingkat presisi untuk Klasifikasi *High* dan *Medium* mencapai 100%, dan Klasifikasi *Low* 71%.

Daftar Pustaka

- [1] H. F. R. Wahyudi, “Perancangan Aplikasi *Helpdesk Ticketing* Dengan Penerapan Algoritma *Forward Chaining* (Studi Kasus: PT Idemas Solusindo Sentosa),” *J. Ilmu Tek. dan Komput.*, vol. 4, no. 1, p. 88, 2020, doi: 10.22441/jitkom.2020.v4.i1.010.
- [2] S. Syofian and A. Winandar, “Aplikasi *Helpdesk* Mendukung Sistem *Ticketing*,” *J. Teknol. Inf.*, vol. 4, no. 1, pp. 1–7, 2017, [Online]. Available: <http://ejournal.urindo.ac.id/index.php/JTI/article/view/264/239>.
- [3] R. M. Bahrudin, M. Ridwan, and H. S. Darmojo, “Penerapan *Helpdesk Ticketing System* Dalam Penanganan Keluhan Penggunaan Sistem Informasi Berbasis *Web*,” *Jutis*, vol. 7, no. 1, pp. 71–82, 2019.
- [4] L. Mazia, L. A. Utami, F. K. Bintang, S. Informasi, C. Melayu, and J. Timur, “Rancang Bangun Sistem Informasi *Helpdesk Ticketing* Berbasis *Web* pada PT . Mitra Tiga Berlian Bekasi,” vol. 6, no. 1, pp. 85–89, 2021.
- [5] R. N. Wardhani, M. C. Utami, and I. Y. Saputra, “SISTEM INFORMASI *HELPDESK TICKETING* PADA PT. BANK MEGA Tbk,” *J. Ilm. Matrik*, vol. 22, no. 2, pp. 201–207, 2020, doi: 10.33557/jurnalatrik.v22i2.868.
- [6] A. Alfian, Y. N. Dewi, F. W. Fibriany, H. Rianto, and A. M. Sari, “Rancang Bangun Sistem Informasi *Ticketing Helpdesk* pada *DPMPPTS Pemprov DKI Jakarta*,” *JURIKOM (Jurnal Ris. Komputer)*, vol. 7, no. 2, p. 334, 2020, doi: 10.30865/jurikom.v7i2.2114.
- [7] S. I. Adam, J. H. Moedjahedy, and O. Lengkong, “Pengembangan *IT Helpdesk Ticketing* Sistem Berbasis *Web* di Universitas Klabat,” *CogITO Smart J.*, vol. 6, no. 2, p. 217, 2020, doi: 10.31154/cogito.v6i2.273.217-228.
- [8] W. D. Suryono, Saptono, Ristu, and W. Wiranto, “Implementasi Pengembangan *Smart Helpdesk* di *UPT TIK UNS* Menggunakan Algoritma *Naive Bayes Classifier*,” *Semin. Nas. Apl. Teknol. Inf.*, pp. 39–43, 2017.
- [9] L. M. Siniwi, A. Prahutama, A. R. Hakim, D. Statistika, and U. Diponegoro, “*QUERY EXPANSION RANKING* PADA ANALISIS SENTIMEN MENGGUNAKAN KLASIFIKASI *MULTINOMIAL NAÏVE BAYES* (Studi Kasus : Ulasan Aplikasi Shopee pada Hari Belanja Online Nasional 2020) 1, 2, 3,” vol. 10, pp. 377–387, 2021.
- [10] A. H. Setianingrum, D. H. Kalokasari, and I. M. Shofi, “Implementasi Algoritma *Multinomial Naive Bayes Classifier*,” *J. Tek. Inform.*, vol. 10, no. 2, pp. 109–118, 2018, doi: 10.15408/jti.v10i2.6822.
- [11] Y. Rachmanto, “Uji Akurasi Pengkategorian Otomatis Berita Berbahasa Indonesia Dengan Algoritma *Multinomial Naive Bayes Classifier (NBC)*,” pp. 32–38, 2018.
- [12] F. K. S. Dewi, “Klasifikasi Berita Menggunakan Metode *Multinomial Naive Bayes*,” *SCAN - J. Teknol. Inf. dan Komun.*, vol. 16, no. 3, pp. 1–8, 2021, doi: 10.33005/scan.v16i3.2870.
- [13] S. Susilowati and M. T. Negara, “Implementasi Model *Rapid Application Development (RAD)* Dalam Perancangan Aplikasi *E-Marketplace*,” *J. TECHNO Nusa Mandiri*, vol. 15, no. 1, pp. 25–30, 2018.
- [14] F. Muhammad and S. L. Putri, “Jurnal Teknologi Informasi dan Komunikasi *STMIK* Subang, Oktober 2017 ISSN: 2252-4517,” *Sist. Inf. Pengolah. Data Pegawai Berbas. Web (Studi Kasus Di Pt Perkeb. Nusant. Viii Tambaksari)*, no. April, pp. 1–23, 2017.
- [15] L. A. Andika, P. A. N. Azizah, and R. Respatiwan, “Analisis Sentimen Masyarakat terhadap Hasil *Quick Count* Pemilihan Presiden Indonesia 2019 pada Media Sosial Twitter Menggunakan Metode *Naive Bayes Classifier*,” *Indones. J. Appl. Stat.*, vol. 2, no. 1, p. 34, 2019, doi: 10.13057/ijas.v2i1.29998.
- [16] A. Rahman and A. Doewes, “*Online News Classification Using Multinomial Naive Bayes*,” *ITSMART J. Ilm. Teknol. dan Inf.*, vol. 6, no. 1, pp. 32–38, 2017.