

Pengembangan Aplikasi Klasifikasi Teks E-mail Keluhan Akademik Pelanggan Studi Kasus: Operasional Kalbis Institute

Daniel Alexander Philip

Informatika, Fakultas Ilmu Komputer dan Desain, Institut Teknologi dan Bisnis Kalbis
Jalan Pulomas Selatan Kav.22, Jakarta 13210
Email: danielsahelangi8@gmail.com

Abstract: *This study aims to develop an application that can classify messages written by Kalbis students. The data used is a text message of student questions or complaints that have been sent to the Academic Operations and Finance through the Gmail service. The message data will be classified into Academic Operation and Finance classes. The data will be weighted based on the frequency of its occurrence in all messages using the Term Frequency – Inverse Document Frequency (TF-IDF) method. Then the weighted data will be classified using the Multinomial Naïve Bayes method. The Incremental Method is a method used to develop applications. The results obtained in this study are applications that are able to classify student messages with a model accuracy value of 86%, with each precision, recall and f1-score : 0.9, 0.83 and 0.84 in iteration one and application interface with objects in it that can be works fine in iteration two.*

Keywords: *Classification text, Gmail, Multinomial Naïve Bayes, TF-IDF*

Abstrak: *Penelitian ini bertujuan mengembangkan aplikasi yang dapat melakukan klasifikasi terhadap pesan yang ditulis oleh mahasiswa Kalbis. Data yang digunakan adalah pesan teks pertanyaan atau keluhan mahasiswa yang pernah dikirim ke Academic Operation dan Finance melalui layanan Gmail. Data pesan tersebut akan diklasifikasi ke dalam kelas Academic Operation dan Finance. Data akan dihitung bobotnya berdasarkan frekuensi kemunculannya di seluruh pesan menggunakan metode Term Frequency – Inverse Document Frequency (TF-IDF). Kemudian data yang sudah dihitung bobotnya akan diklasifikasi menggunakan metode Multinomial Naïve Bayes. Metode Inkremental adalah metode yang digunakan untuk mengembangkan aplikasi. Hasil yang diperoleh dalam penelitian ini adalah aplikasi yang mampu mengklasifikasi pesan mahasiswa dengan nilai akurasi model sebesar 86 %, dengan masing nilai precision, recall dan f1-score : 0.9, 0.83 dan 0.84 di iterasi satu dan antarmuka aplikasi dengan objek yang di dalamnya yang dapat berfungsi dengan baik di iterasi dua.*

Kata kunci: *Klasifikasi teks, Gmail, TF-IDF, Multinomial Naïve Bayes*

I. PENDAHULUAN

Dalam menjalani pendidikannya di perkuliahan, seorang mahasiswa tentu pernah mengalami kendala atau memiliki keperluan seputar perkuliahan. Mahasiswa akan menyampaikan keluhan atau pertanyaan terkait kendala atau keperluan mereka ke pihak kampus.

Di Kalbis Institute ada bagian *Academic Operation* yang melayani persoalan perkuliahan secara umum dari mahasiswa seperti permohonan surat magang, kualitas mengajar dosen, dan permasalahan fasilitas kampus. Adapula bagian *Finance* merupakan bagian yang melayani secara spesifik masalah keuangan mahasiswa seperti pembayaran SKS, BP3, dan lain - lain seperti yang telah disampaikan oleh perwakilan dari *Academic Operation* melalui wawancara mengenai layanan Operasional Akademik Kalbis

Institute Berdasarkan wawancara yang dilakukan dengan perwakilan *Academic Operation* dikatakan bahwa *Academic Operation* menampung semua pesan keluhan atau pertanyaan dari mahasiswa terkadang persoalan yang berkaitan dengan *Finance* melalui layanan Gmail selama masa pandemi ini. Hal ini dikarenakan ada mahasiswa yang tidak tahu kalau permasalahan yang berkaitan dengan keuangan bisa disampaikan langsung ke Gmail bagian *Finance*. Demikian jika *Academic Operation* menerima pesan dari mahasiswa yang berkaitan dengan keuangan, maka *Academic Operation* akan meneruskan pesan tersebut ke Gmail *Finance* untuk ditindaklanjuti.

Oleh karena itu agar pesan yang berkaitan dengan keuangan dapat langsung sampai ke *Finance* tanpa ada yang masuk *Academic Operation* lagi, maka peneliti mengusulkan untuk membuat aplikasi yang dapat mengklasifikasi pesan yang ditulis oleh mahasiswa Kalbis. Pesan akan diklasifikasi ke dalam kategori *Academic Operation* dan *Finance*. Agar aplikasi dapat melakukan hal itu maka digunakan model *machine learning* untuk mengklasifikasi pesan.

Adapun penelitian terdahulu dilakukan oleh Haryono, dkk yang menerapkan *Naïve Bayes* untuk mengklasifikasi pesan keluhan dan gangguan ke PLN melalui *call center* 123. Pesan tersebut akan diklasifikasi ke dalam 3 kategori yaitu bidang jaringan, bidang konstruksi dan bidang

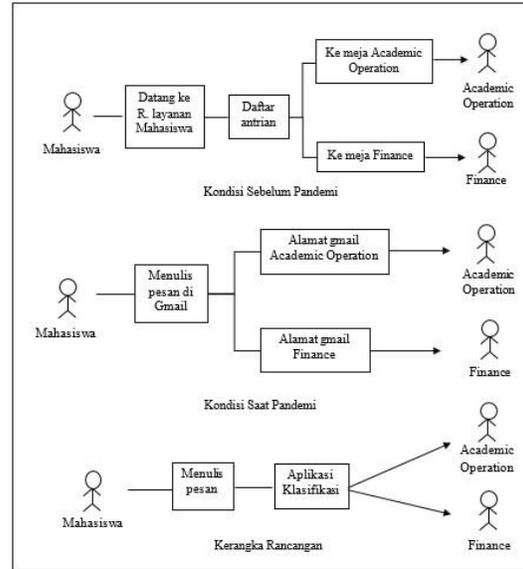
transaksi energi. Menghasilkan aplikasi yang mengklasifikasi pesan keluhan dan gangguan ke dalam 3 kelas tersebut dengan tingkat akurasi sebesar 95% [1, hlm. 10]. Kemudian penelitian juga dilakukan oleh Lifeng, dkk yang menerapkan *Naïve Bayes* untuk mengklasifikasi pesan keluhan kepada departemen kereta api. Pada penelitian ini dilakukan perbandingan 3 metode ekstraksi fitur yaitu *TF-IDF*, *TextRank*, *Word2Vec*. Hasil penelitian mengatakan bahwa klasifikasi terhadap pesan keluhan kepada departemen kereta api menggunakan *TF-IDF* memiliki akurasi tertinggi yaitu sebesar 77,8% [2, hlm. 8]. Lalu ada penelitian yang dilakukan oleh Alan Chavez yang menggunakan *Multinomial Naïve Bayes* dengan ekstraksi fitur *TF-IDF* untuk mengklasifikasi terhadap pesan elektronik ke dalam kategori *spam* dan tidak *spam*. Menghasilkan sistem yang dapat mengklasifikasi pesan elektronik tersebut dengan akurasi sebesar 98% [3, hlm. 12]. Maka berdasarkan penelitian – penelitian tersebut, akan digunakan *Multinomial Naïve Bayes* untuk klasifikasi dan *TF-IDF* untuk ekstraksi fitur.

Hal yang dilakukan terlebih dahulu adalah mengumpulkan data yang berupa kumpulan pertanyaan atau keluhan dari mahasiswa menggunakan kuesioner dan disimpan ke berkas yang berformat CSV. Kemudian data yang telah disimpan diberi label *Academic Operation* dan *Finance* secara manual. Setelah itu, dilakukan tahap – tahap pra-pemrosesan. Lalu, data dibagi menjadi data latih sebanyak 90% dan data uji sebanyak 10%. Sebelum data latih dan data uji digunakan untuk melatih dan menguji model, maka data latih dan data uji akan

melewati tahap pembobotan. Setelah itu dilakukan tahap pelatihan pada model yang akan digunakan yaitu *Multinomial Naïve Bayes*, yang kemudian dilakukan tahap pengujian terhadap model dengan data uji. Lalu digunakan *Confusion Matrix* untuk melihat performa model dalam klasifikasi serta akurasi modelnya. Untuk evaluasi terhadap model klasifikasi dapat dilakukan dengan cara melakukan perhitungan rata – rata *precision*, *recall*, dan *f-measure*. Lalu model diuji kembali dengan menggunakan data kasus yang merupakan kalimat yang jadi masukan ke dalam aplikasi untuk memastikan model dapat mengklasifikasi pesan.

II. METODE PENELITIAN

Di kondisi sebelum pandemi pada Gambar 1, jika mahasiswa ingin menyampaikan pertanyaan atau keluhannya, mereka harus mengunjungi ruang layanan mahasiswa yang ada di lantai dasar gedung Kalbis Institute. mereka harus mendaftar antrian untuk memilih apakah mereka akan ke bagian *Academic Operation* atau *Finance*. Di kondisi saat pandemi, mahasiswa menyampaikan pertanyaan atau keluhannya melalui layanan Gmail dengan menuliskan alamat Gmail dari kedua bagian tersebut. Di kerangka rancangan, mahasiswa pertanyaan atau keluhannya melalui aplikasi klasifikasi pesan yang dapat mengklasifikasi pesan yang ditulis oleh mahasiswa ke dalam dua label yang sudah didefinisikan sebelumnya yaitu *Academic Operation* dan *Finance*.



Gambar 1 Kerangka Pemikiran

A. Proses Pembuatan Dataset

Pada awal penelitian, dibuat sebuah kuesioner yang akan diisi oleh mahasiswa Kalbis dengan pesan yang pernah mahasiswa kirim ke bagian *Academic Operation* dan *Finance* melalui layanan Gmail. Kuesioner disebar dari tanggal 16 sampai 17 Februari 2020 secara *online*. Saat itu diperoleh data pesan sebanyak 65 pesan.

1	teks	kategori
2	waktu tambahan untuk membayar sks/bp3	Finance
3	permintaan tanda tangan mengenai magang	Academic Operation
4	Saya mau tanya, sertifikat Kalbispheration saya apakah sudah jadi atau belum? Ke Academic Operation	Academic Operation
5	Terkait pembayaran bp3, apakah saya telah melakukan pembayaran untuk tagih Finance	Finance
6	Tolong lunasi uang kuliah saya	Finance
7	Permohonan surat praktik kerja	Academic Operation
8	Ac kalbis kurang dingin	Academic Operation
9	Perihal Surat Permohonan Untuk Magang	Academic Operation
10	Bukti Pembayaran	Finance
11	Subject lamaran magang Nama : ***** Nim : ***** Dear Academic Operat Academic Operation	Academic Operation

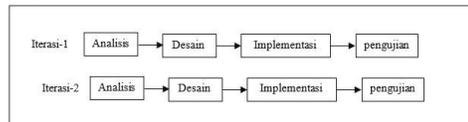
Gambar 2 Dataset Pesan Mahasiswa

Gambar 2 adalah *dataset* pesan mahasiswa dalam berkas CSV. Hal ini dilakukan dengan cara menyalin data pesan yang terdapat dalam kuesioner, lalu menempelkannya ke dalam Microsoft Excel dan memberikan label kepada tiap pesan secara manual. Setelah itu pesan disimpan dengan ekstensi CSV.

B. Pengembangan Perangkat Lunak

Pada penelitian ini, digunakan model Inkremental sebagai metode pengembangan perangkat lunak. Pengembangan perangkat lunak yang dilakukan secara bertahap dikarenakan kebutuhan perangkat lunak yang dipecah menjadi beberapa bagian membuat model ini dipilih dalam penelitian ini. Pada penelitian ini, model inkremental dibagi menjadi dua tahapan yaitu : iterasi satu dan dua, agar hasil memenuhi kebutuhan penelitian.

Digambar ulang dari sumber [4, hlm. 417]



Gambar 3 Tahapan Model Inkremental

Seperti yang terdapat pada Gambar 3, berikut adalah tahapan yang terjadi di tiap iterasi [5, hlm. 3]:

1. Analisis : tahap melakukan menganalisis akan hal – hal yang dibutuhkan untuk membangun perangkat lunak.
2. Desain : tahap melakukan perancangan arsitektur dari perangkat lunak agar menghasilkan landasan untuk tahap implementasi.
3. Implementasi : tahap pembuatan kode dari perangkat lunak dengan menyesuaikan landasan yang dihasilkan di tahap desain
4. Pengujian : tahap melakukan pengujian terhadap perangkat lunak baik pengujian *error* dan fitur – fitur perangkat lunak.

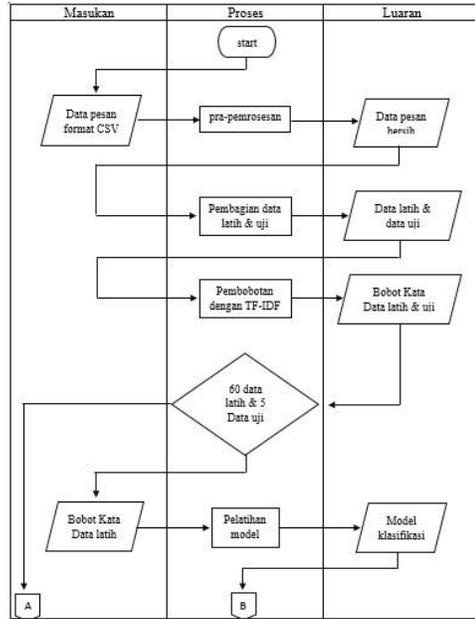
C. Iterasi Satu

Pada tahap ini akan dilakukan analisis terhadap metode – metode apa saja yang dilakukan dalam membangun logika pemrograman dari aplikasi serta pelatihan dan pengujian dari model *machine learning* yang dipilih. Data pesan mahasiswa yang terdapat dalam *dataset* masih dalam keadaan asli saat pesan itu pertama kali dikirim ke bagian *Academic Operation* dan *Finance*. Pesan teks yang ditulis oleh mahasiswa masih terdapat pemakaian huruf besar dan kecil secara bergantian, penggunaan tanda baca, adanya imbuhan dalam suatu kata, serta munculnya kata – kata sambung yang tidak merepresentasikan dan tidak memiliki kaitan terhadap suatu teks.

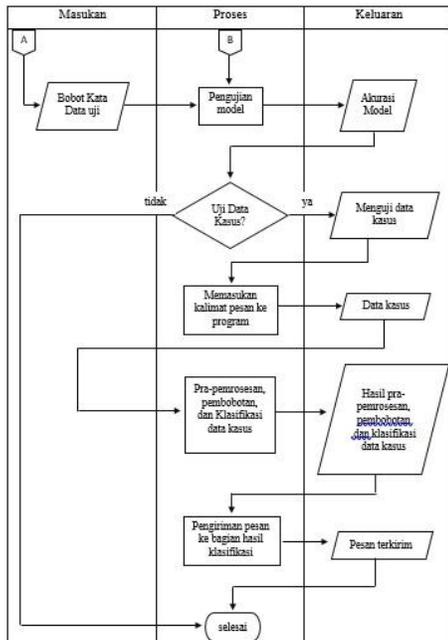
Maka dari itu diperlukan tahap pra-pemrosesan agar data pesan siap untuk diolah. Adapun tahap – tahap pra-pemrosesan yang dilakukan antara lain adalah *case folding*, *remove punctuation*, *stemming* dan *stopword removal*. Setelah dilakukannya pra-pemrosesan, pembobotan terhadap data pesan dilakukan dengan menggunakan TF-IDF agar menurut frekuensinya pada suatu teks setiap kata memiliki bobot. Lalu kemudian dilakukan pelatihan dan pengujian dari model *machine learning* dengan data pesan yang sudah melewati tahap pembobotan. Model yang digunakan untuk mengklasifikasi data pesan adalah *Multinomial Naïve Bayes* dikarenakan model ini memperoleh nilai dari suatu teks melalui frekuensi kemunculan suatu kata pada suatu teks. Lalu model akan diberi data kasus yang merupakan kalimat masukan ke aplikasi untuk melihat hasil klasifikasi yang dilakukan oleh model. Setelah itu data kasus akan dikirim ke tujuan yang sesuai dengan hasil klasifikasi.

Kemudian pada tahap desain dibuatlah desain alur logika pemrograman aplikasi serta pelatihan dan pengujian terhadap model *machine learning* yang akan digunakan

sebagai landasan pada tahap implementasi. Adapun alur dari logika pemrograman terdapat pada Gambar 4 dan Gambar 5.



Gambar 4 Logika pemrograman bagian pertama



Gambar 5 Logika Pemrograman Bagian Kedua

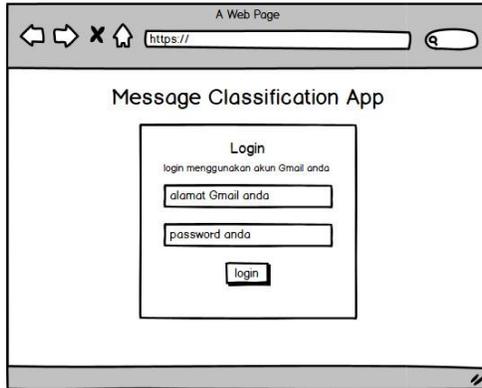
Gambar 4 dan 5 adalah alur dari logika pemrograman yang merupakan inti dari aplikasi. Dari alur yang sudah dibuat, kemudian diimplementasikan ke dalam kode – kode bahasa pemrograman Python. Adapun Python memiliki *library* yang digunakan dalam membuat logika pemrograman antara lain : Pandas, Scikit Learn, Regex, Sastrawi, Matplotlib dan smtplib. Setelah itu di tahap pengujian akan diuji fungsi –fungsi yang terdapat dalam logika pemrograman dengan metode *White – Box*.

D. Iterasi Dua

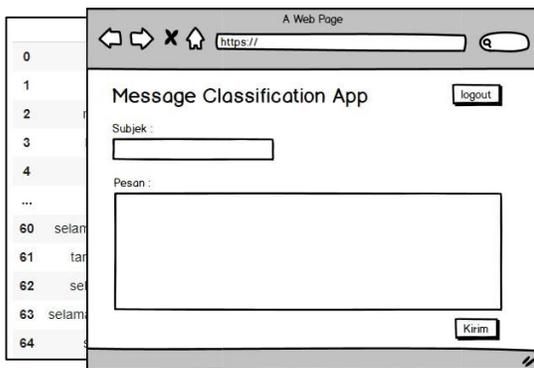
Pada iterasi dua ini akan dibangun tampilan dari aplikasi ini berdasarkan kebutuhan aplikasi dan kebutuhan penelitian. Bagian inti dari aplikasi yang sudah dibuat di iterasi sebelumnya akan dijadikan sebagai dasar landasan pembuatan tampilan.

Di halaman *login* terdapat 2 input teks masing– masing untuk alamat Gmail dan *password* pengguna lalu tombol *login* yang melakukan proses *login*. Di dalam halaman penulisan pesan terdapat tempat *input* teks untuk menulis subjek dari pesan dan area teks untuk tempat pengguna menulis pesan serta tombol kirim yang akan melakukan proses klasifikasi pesan serta mengirimnya dan tombol keluar akan melakukan proses *logout* dari akun yang digunakan saat itu.

Lalu pada tahap desain, dibuat purwarupa yang memberi gambaran dari tampilan aplikasi seperti pada Gambar 6 dan Gambar 7.



Gambar 6 Halaman login



Gambar 7 Halaman penulisan pesan

III. HASIL DAN PEMBAHASAN

Pada tahap iterasi satu, hal yang dilakukan terlebih dahulu adalah pra-pemrosesan terhadap data pesan mahasiswa. Berikut adalah hasil pra-pemrosesan terhadap data pesan mahasiswa yang meliputi tahap *case folding*, *remove punctuation*, *stemming*, dan *stopword removal* yang ditunjukkan pada Gambar 8.

Gambar 8 Hasil pra-pemrosesan

Setelah melewati tahap pra-pemrosesan, dilakukan pembagian terhadap data pesan dengan rasio 90 % untuk data latih dan 10 % untuk data uji. Lalu dilakukan pembobotan terhadap data yang sudah dibagi menjadi data latih dan data uji. Berikut adalah hasil pembobotan terhadap data latih dan data uji ditunjukkan pada Gambar 9.

Gambar 6 dan 7 adalah dua halaman utama dari aplikasi. Kemudian purwarupa yang telah dibuat diimplementasikan ke dalam bentuk kode pada tahap implementasi. *Library* Python yang digunakan untuk membangun tampilan dari aplikasi adalah Flask. Selain Flask, digunakan juga HTML, CSS, dan Javascript sehingga pengguna dapat mengakses aplikasi ini dalam bentuk *web app*. Lalu pada tahap pengujian objek-objek yang ada dalam tampilan akan diuji dengan metode *Black Box*.

(0, 10)	0.1339092532805305	(0, 46)	0.37005778788815666
(0, 11)	0.11059495522776713	(0, 47)	0.11059495522776713
(0, 115)	0.11059495522776713	(0, 343)	0.31155375496488913
(0, 116)	0.11059495522776713	(0, 344)	0.31155375496488913
(0, 117)	0.11059495522776713	(0, 345)	0.31155375496488913
(0, 118)	0.11059495522776713	(0, 346)	0.31155375496488913
(0, 119)	0.11059495522776713	(0, 347)	0.31155375496488913
(0, 120)	0.11059495522776713	(0, 348)	0.31155375496488913
(0, 121)	0.11059495522776713	(0, 349)	0.31155375496488913
(0, 122)	0.11059495522776713	(0, 350)	0.31155375496488913
(0, 123)	0.1339092532805305	(0, 256)	0.17571160725381705
(0, 124)	0.1339092532805305	(0, 257)	0.17571160725381705
(0, 125)	0.1339092532805305	(0, 154)	0.2530497220406216
(0, 126)	0.1339092532805305	(0, 155)	0.2530497220406216
(0, 127)	0.1339092532805305	(0, 156)	0.2530497220406216
(0, 128)	0.1339092532805305	(0, 157)	0.2530497220406216
(0, 129)	0.1339092532805305	(0, 158)	0.2530497220406216
(0, 130)	0.1339092532805305	(0, 159)	0.2530497220406216
(0, 131)	0.1339092532805305	(0, 160)	0.2530497220406216
(0, 132)	0.1339092532805305	(0, 161)	0.2530497220406216
(0, 133)	0.1339092532805305	(0, 162)	0.2530497220406216
(0, 134)	0.1339092532805305	(0, 163)	0.2530497220406216
(0, 135)	0.1339092532805305	(0, 164)	0.2530497220406216
(0, 136)	0.1339092532805305	(0, 165)	0.2530497220406216
(0, 137)	0.1339092532805305	(0, 166)	0.2530497220406216
(0, 138)	0.1339092532805305	(0, 167)	0.2530497220406216
(0, 139)	0.1339092532805305	(0, 168)	0.2530497220406216
(0, 140)	0.1339092532805305	(0, 169)	0.2530497220406216
(0, 141)	0.1339092532805305	(0, 170)	0.2530497220406216
(0, 142)	0.1339092532805305	(0, 171)	0.2530497220406216
(0, 143)	0.1339092532805305	(0, 172)	0.2530497220406216
(0, 144)	0.1339092532805305	(0, 173)	0.2530497220406216
(0, 145)	0.1339092532805305	(0, 174)	0.2530497220406216
(0, 146)	0.1339092532805305	(0, 175)	0.2530497220406216
(0, 147)	0.1339092532805305	(0, 176)	0.2530497220406216
(0, 148)	0.1339092532805305	(0, 177)	0.2530497220406216
(0, 149)	0.1339092532805305	(0, 178)	0.2530497220406216
(0, 150)	0.1339092532805305	(0, 179)	0.2530497220406216
(0, 151)	0.1339092532805305	(0, 180)	0.2530497220406216
(0, 152)	0.1339092532805305	(0, 181)	0.2530497220406216
(0, 153)	0.1339092532805305	(0, 182)	0.2530497220406216
(0, 154)	0.1339092532805305	(0, 183)	0.2530497220406216
(0, 155)	0.1339092532805305	(0, 184)	0.2530497220406216
(0, 156)	0.1339092532805305	(0, 185)	0.2530497220406216
(0, 157)	0.1339092532805305	(0, 186)	0.2530497220406216
(0, 158)	0.1339092532805305	(0, 187)	0.2530497220406216
(0, 159)	0.1339092532805305	(0, 188)	0.2530497220406216
(0, 160)	0.1339092532805305	(0, 189)	0.2530497220406216
(0, 161)	0.1339092532805305	(0, 190)	0.2530497220406216
(0, 162)	0.1339092532805305	(0, 191)	0.2530497220406216
(0, 163)	0.1339092532805305	(0, 192)	0.2530497220406216
(0, 164)	0.1339092532805305	(0, 193)	0.2530497220406216
(0, 165)	0.1339092532805305	(0, 194)	0.2530497220406216
(0, 166)	0.1339092532805305	(0, 195)	0.2530497220406216
(0, 167)	0.1339092532805305	(0, 196)	0.2530497220406216
(0, 168)	0.1339092532805305	(0, 197)	0.2530497220406216
(0, 169)	0.1339092532805305	(0, 198)	0.2530497220406216
(0, 170)	0.1339092532805305	(0, 199)	0.2530497220406216
(0, 171)	0.1339092532805305	(0, 200)	0.2530497220406216
(0, 172)	0.1339092532805305	(0, 201)	0.2530497220406216
(0, 173)	0.1339092532805305	(0, 202)	0.2530497220406216
(0, 174)	0.1339092532805305	(0, 203)	0.2530497220406216
(0, 175)	0.1339092532805305	(0, 204)	0.2530497220406216
(0, 176)	0.1339092532805305	(0, 205)	0.2530497220406216
(0, 177)	0.1339092532805305	(0, 206)	0.2530497220406216
(0, 178)	0.1339092532805305	(0, 207)	0.2530497220406216
(0, 179)	0.1339092532805305	(0, 208)	0.2530497220406216
(0, 180)	0.1339092532805305	(0, 209)	0.2530497220406216
(0, 181)	0.1339092532805305	(0, 210)	0.2530497220406216
(0, 182)	0.1339092532805305	(0, 211)	0.2530497220406216
(0, 183)	0.1339092532805305	(0, 212)	0.2530497220406216
(0, 184)	0.1339092532805305	(0, 213)	0.2530497220406216
(0, 185)	0.1339092532805305	(0, 214)	0.2530497220406216
(0, 186)	0.1339092532805305	(0, 215)	0.2530497220406216
(0, 187)	0.1339092532805305	(0, 216)	0.2530497220406216
(0, 188)	0.1339092532805305	(0, 217)	0.2530497220406216
(0, 189)	0.1339092532805305	(0, 218)	0.2530497220406216
(0, 190)	0.1339092532805305	(0, 219)	0.2530497220406216
(0, 191)	0.1339092532805305	(0, 220)	0.2530497220406216
(0, 192)	0.1339092532805305	(0, 221)	0.2530497220406216
(0, 193)	0.1339092532805305	(0, 222)	0.2530497220406216
(0, 194)	0.1339092532805305	(0, 223)	0.2530497220406216
(0, 195)	0.1339092532805305	(0, 224)	0.2530497220406216
(0, 196)	0.1339092532805305	(0, 225)	0.2530497220406216
(0, 197)	0.1339092532805305	(0, 226)	0.2530497220406216
(0, 198)	0.1339092532805305	(0, 227)	0.2530497220406216
(0, 199)	0.1339092532805305	(0, 228)	0.2530497220406216
(0, 200)	0.1339092532805305	(0, 229)	0.2530497220406216
(0, 201)	0.1339092532805305	(0, 230)	0.2530497220406216
(0, 202)	0.1339092532805305	(0, 231)	0.2530497220406216
(0, 203)	0.1339092532805305	(0, 232)	0.2530497220406216
(0, 204)	0.1339092532805305	(0, 233)	0.2530497220406216
(0, 205)	0.1339092532805305	(0, 234)	0.2530497220406216
(0, 206)	0.1339092532805305	(0, 235)	0.2530497220406216
(0, 207)	0.1339092532805305	(0, 236)	0.2530497220406216
(0, 208)	0.1339092532805305	(0, 237)	0.2530497220406216
(0, 209)	0.1339092532805305	(0, 238)	0.2530497220406216
(0, 210)	0.1339092532805305	(0, 239)	0.2530497220406216
(0, 211)	0.1339092532805305	(0, 240)	0.2530497220406216
(0, 212)	0.1339092532805305	(0, 241)	0.2530497220406216
(0, 213)	0.1339092532805305	(0, 242)	0.2530497220406216
(0, 214)	0.1339092532805305	(0, 243)	0.2530497220406216
(0, 215)	0.1339092532805305	(0, 244)	0.2530497220406216
(0, 216)	0.1339092532805305	(0, 245)	0.2530497220406216
(0, 217)	0.1339092532805305	(0, 246)	0.2530497220406216
(0, 218)	0.1339092532805305	(0, 247)	0.2530497220406216
(0, 219)	0.1339092532805305	(0, 248)	0.2530497220406216
(0, 220)	0.1339092532805305	(0, 249)	0.2530497220406216
(0, 221)	0.1339092532805305	(0, 250)	0.2530497220406216
(0, 222)	0.1339092532805305	(0, 251)	0.2530497220406216
(0, 223)	0.1339092532805305	(0, 252)	0.2530497220406216
(0, 224)	0.1339092532805305	(0, 253)	0.2530497220406216
(0, 225)	0.1339092532805305	(0, 254)	0.2530497220406216
(0, 226)	0.1339092532805305	(0, 255)	0.2530497220406216
(0, 227)	0.1339092532805305	(0, 256)	0.2530497220406216
(0, 228)	0.1339092532805305	(0, 257)	0.2530497220406216
(0, 229)	0.1339092532805305	(0, 258)	0.2530497220406216
(0, 230)	0.1339092532805305	(0, 259)	0.2530497220406216
(0, 231)	0.1339092532805305	(0, 260)	0.2530497220406216
(0, 232)	0.1339092532805305	(0, 261)	0.2530497220406216
(0, 233)	0.1339092532805305	(0, 262)	0.2530497220406216
(0, 234)	0.1339092532805305	(0, 263)	0.2530497220406216
(0, 235)	0.1339092532805305	(0, 264)	0.2530497220406216
(0, 236)	0.1339092532805305	(0, 265)	0.2530497220406216
(0, 237)	0.1339092532805305	(0, 266)	0.2530497220406216
(0, 238)	0.1339092532805305	(0, 267)	0.2530497220406216
(0, 239)	0.1339092532805305	(0, 268)	0.2530497220406216
(0, 240)	0.1339092532805305	(0, 269)	0.2530497220406216
(0, 241)	0.1339092532805305	(0, 270)	0.2530497220406216
(0, 242)	0.1339092532805305	(0, 271)	0.2530497220406216
(0, 243)	0.1339092532805305	(0, 272)	0.2530497220406216
(0, 244)	0.1339092532805305	(0, 273)	0.2530497220406216
(0, 245)	0.1339092532805305	(0, 274)	0.2530497220406216
(0, 246)	0.1339092532805305	(0, 275)	0.2530497220406216
(0, 247)	0.1339092532805305	(0, 276)	0.2530497220406216
(0, 248)	0.1339092532805305	(0, 277)	0.2530497220406216
(0, 249)	0.1339092532805305	(0, 278)	0.2530497220406216
(0, 250)	0.1339092532805305	(0, 279)	0.2530497220406216
(0, 251)	0.1339092532805305	(0, 280)	0.2530497220406216
(0, 252)	0.1339092532805305	(0, 281)	0.2530497220406216
(0, 253)	0.1339092532805305	(0, 282)	0.2530497220406216
(0, 254)	0.1339092532805305	(0, 283)	0.2530497220406216
(0, 255)	0.1339092532805305	(0, 284)	0.2530497220406216
(0, 256)	0.1339092532805305	(0, 285)	0.2530497220406216
(0, 257)	0.1339092532805305	(0, 286)	0.2530497220406

Model yang telah dilatih dengan data latih sebelumnya akan diuji menggunakan data uji yang sudah melewati proses pembobotan. Data uji akan diklasifikasi oleh model berdasarkan pada data latih sebelumnya. Hasil klasifikasi pada data uji ditunjukkan pada Gambar 10.

Gambar 10 Hasil klasifikasi data uji

Kemudian untuk mengetahui performa dari model *machine learning* dalam melakukan klasifikasi maka akan ditampilkan tabel *confusion matrix* pada Gambar 11 dan hasil performa model pada Gambar 12.

(0, 4)	Actual Label	Academic-Operation	4	0				
(0, 3)								
(0, 3)		Finance	Academic Operation	1	2			
(0, 3)								
(0, 3)			Finance	Academic Operation				
(0, 3)								
(0, 3)				Finance	Academic Operation			
(0, 3)								
(0, 2)					Finance	Academic Operation		
(0, 2)								
(0, 2)	Finance	Academic Operation						
(0, 2)								
(0, 2)	Finance	Academic Operation						
(0, 2)								
(0, 215)	Predicted Label	Academic Operation	0.14817954039915499					
(0, 210)			Finance	0.2447319570716214				
(0, 169)		Finance		0.13746594423591285				

Gambar 11 Hasil confusion matrix

	precision	recall	f1-score
Academic Operation	0.80	1.00	0.89
Finance	1.00	0.67	0.80
accuracy			0.86

Gambar 12 Hasil performansi model

Pada Gambar 12 dapat dilihat model memiliki akurasi sebesar 86%. Sedangkan nilai *precision*, *recall*, dan *f1-score* dari *Academic Operation*: 80%, 100%, 89% dan *Finance* : 100%, 67%, 80%.

Kemudian dilakukan tahap pengujian dengan data kasus setelah

dilakukan pelatihan dan pengujian terhadap model *machine learning*. Berikut adalah hasil pramrosesan terhadap data kasus ditunjukkan pada Tabel 1.

Tabel 1 Hasil pra-pemrosesan data kasus

Sebelum Pra-pemrosesan	Setelah Pra-pemrosesan
Dear student service, Saya Daniel Alexander Philip dari jurusan informatika 2017. Saya hari lupa untuk tap in untuk mata kuliah natural language programming praktikum pkl 16.00, dikarenakan pada saat saya ingin tap in pkl 16.00 sessionnya belum ada. Nanti demikian saya sudah menyelesaikan modul yang diberikan dosen pada pertemuan ini dan tetap melakukan tap out tepat seperti waktu yang terdapat dalam jadwal mata kuliah ini. Atas perhatiannya saya ucapkan terimakasih.	dear student service daniel alexander philip jurus informatika 2017 hari lupa tap in mata kuliah natural language programming praktikum pkl 16 00 tap in pkl 16 00 sessionnya ada demikian sudah selesai modul beri dosen pada tap in dan tap out tepat waktu dapat jadwal mata kuliah atas perhatian saya ucap terimakasih

Lalu dilakukan tahap pembobotan terhadap data kasus. Berikut adalah hasil pembobotan yang ditunjukkan pada Gambar 13.

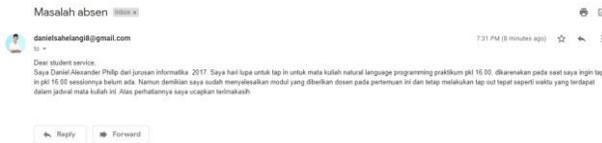
Gambar 13 Hasil pembobotan data kasus

Model *machine learning* yang sudah terlatih akan mengklasifikasi data kasus yang sudah diperoleh matriks bobotnya. Berikut adalah hasil klasifikasi pada data kasus yang ditunjukkan pada Gambar 14.

Gambar 14 Hasil klasifikasi data kasus

Seperti pada Gambar 14, model *machine learning* mengklasifikasi data kasus ke dalam kategori pesan yang ditujukan untuk *Academic Operation*. Maka munculah pesan yang mengatakan bahwa pesan akan dikirim ke *Academic Operation*. Berikut adalah bukti pesan tersebut ke kirim ke *Academic Operation* yang ditunjukkan pada Gambar 15.

pengguna salah maka *alert* bahwa *e-mail* dan *password* pengguna salah atau langkah – langkah yang perlu dilakukan sebelum melakukan *login*. Namun jika *e-mail* dan *password* benar



Gambar 15 Pesan terkirim ke *Academic Operation*

Hasil dari iterasi dua adalah tampilan dari aplikasi web. Tampilan dari aplikasi dibangun menggunakan kerangka kerja Flask dari Python. Dengan kerangka kerja ini, *library* yang berkaitan dengan *machine learning* mudah ditambahkan. Berikut adalah halaman *login* dari aplikasi ini yang ditunjukkan pada Gambar 16.



Gambar 16 Halaman login

Jika pada saat tombol *login* ditekan *e-mail* dan *password*

maka *alert* bahwa pengguna berhasil *login* dan pengguna akan diarahkan ke halaman penulisan pesan. Kedua *alert* akan ditunjukkan pada Gambar 17 dan Gambar 18.



Gambar 17 Alert e-mai atau password keliru



Gambar 18 Alert pengguna berhasil login



Gambar 19 Halaman penulisan pesan

Pada Gambar 19 adalah halaman penulisan pesan yang dapat diakses jika proses *login* berhasil dilakukan. Di dalam halaman penulisan pesan terdapat *input* teks subjek dan area teks untuk menulis pesan beserta dengan tombol kirim untuk melakukan klasifikasi pesan dan mengirimnya ke tujuan yang sesuai hasil klasifikasi. Sebagai bentuk konfirmasi kepada pengguna bahwa pesan telah dikirim ke bagian yang dituju, maka muncul *alert* yang mengatakan bahwa pesan telah dikirim ke *Academic Operation* atau *Finance*. Ada juga *alert* yang memberi tahu bahwa pengguna

sudah *logout* pada saat tombol *logout* ditekan. Berikut adalah *alert* yang sudah dijelaskan sebelumnya pada Gambar 20 sampai Gambar 22



Gambar 20 Alert academic operation



Gambar 21 Alert finance



Gambar 22 Alert pengguna logout

IV. SIMPULAN

Berdasarkan penelitian dan pengembangan aplikasi yang telah dilakukan maka dirumuskan kesimpulan sebagai berikut :

1. Aplikasi klasifikasi teks dapat melakukan klasifikasi pesan yang ditulis oleh mahasiswa Kalbis ke dalam kategori *Academic Operation* dan *Finance*.
2. Aplikasi juga dapat melakukan pengiriman pesan ke tujuan sesuai dengan hasil klasifikasi yang dihasilkan oleh model *machine learning*.
3. Aplikasi klasifikasi teks yang dibuat dengan model Inkremental ini pada iterasi satu menghasilkan logika pemrograman dengan fungsi – fungsi yang berjalan sesuai harapan, sedangkan model *Multinomial Naïve Bayes* dapat melakukan klasifikasi dengan tingkat akurasi sebesar 86 % beserta dengan

nilai *precision*, *recall* dan *f1-score* yang dapat dilihat di Gambar 12. Di iterasi dua dihasilkan tampilan aplikasi dengan objek dalam tampilan yang dapat melakukan aksi sesuai harapan.

4. Dengan aplikasi klasifikasi teks, pesan yang berkaitan dengan keuangan dapat langsung tersampaikan ke *Finance* tanpa harus ada yang tersampaikan ke *Academic Operation* lagi.
5. Aplikasi klasifikasi teks dapat menjadi wadah untuk segala macam pesan pertanyaan atau keluhan yang disampaikan oleh mahasiswa, karena aplikasi yang akan melakukan

klasifikasi terhadap pesan yang ditulis oleh mahasiswa.

DAFTAR RUJUKAN

- [1] H. Haryono, P. Palupiningsih, Y. Asri, dan A. N. S. Handayani, "Klasifikasi Pesan Gangguan Pelanggan Menggunakan Metode Naive Bayes Classifier," *Kilat*, vol. 7, no. 2, hal. 100–108, 2018, doi: 10.33322/kilat.v7i2.354.
- [2] L. Li, W. Li, dan D. Gong, "Naive bayesian automatic classification of railway service complaint text based on eigenvalue extraction," *Teh. Vjesn.*, vol. 26, no. 3, hal. 778–785, 2019, doi: 10.17559/TV-20190420161815.
- [3] A. Chavez, "TF-IDF classification based Multinomial Naïve Bayes model for spam filtering MSc Cybersecurity Alan Chavez School of Computing National College of Ireland Supervisor :," 2019.
- [4] A. Kossiakoff, S. Biener, S. Seymour, dan D. Flanigan, *Systems Engineering Principles and Practice*, Third. John Wiley & Sons, 2020.
- [5] A. Habib dan A. Kartika W. H., "Development of an Online Sales Information System for SMEs Using Incremental Methods," *INTENSIF J. Ilm. Penelit. dan Penerapan Teknol. Sist. Inf.*, vol. 4, no. 1, hal. 51–62, 2020, doi: 10.29407/intensif.v4i1.13524.