CONTEN : Computer and Network Technology
Vol. 4, No. 1, Juni 2024, hlm. 21-30

e-ISSN: 2797-5274
p-ISSN: 2809-9788

Menggunakan Binary Classification untuk Mendeteksi SPAM pada SMS dengan Metode Logistic Regression

Hendra [1], B. T. Sutrisno. SP. [2], Andy Arief Setyawan [3]
[1] Institut Teknologi dan Bisnis Adias
[2] Institut Teknologi dan Bisnis Adias
[3] Institut Teknologi dan Bisnis Adias
Email: [1]camp.hendra@gmail.com, [2] denbambang@gmail.com, [3] andyariefsetyawan@gmail.com
*** korespondensi**

Abstrak

Spam biasanya digunakan baik oleh perusahaan maupun perseorangan kepada pelanggannya baik pelanggan lama maupun calon pelanggan untuk memberikan berbagai macam informasi secara terus-menerus. Umumnya informasi yang diberikan adalah promosi produk atau penawaran produk baru. Namun informasi yang diberikan tersebut adalah informasi yang bisa jadi tidak penting sehingga dapat mengganggu orang yang menerima informasi tersebut. Masalah spam ini bisa diantisipasi dengan menciptakan sebuah *Machine Learning* yang mampu mendeteksi apakah informasi yang masuk itu adalah spam atau bukan dengan melakukan klasifikasi terhadap isi pesan SMS. Karena label yang terdapat pada penelitian ini hanya ada dua variabel, maka penelitian ini menggunakan tipe *Binary Classification* dengan menerapkan metode *Logistic Regression* untuk melakukan klasifikasi pesan SMS. Tujuan dalam penelitian ini adalah selain untuk menciptakan sebuah tools pendeteksi spam, penelitian ini juga akan mengdapatkan hasil uji terhadap metode Logistic Regression berdasarkan akurasi pendeteksian yang dapat dicapai. Hasil dari penelitian ini adalah untuk mendapatkan teknik klasifikasi SPAM SMS dengan tingkat akurasi yang tinggi.

Kata kunci: *Binary Classification, Logistic Regression, SMS, Spam*

Abstract

Spam is usually used both by companies and individuals to their customers, both existing customers and potential customers, to provide various kinds of information continuously. Generally, the information provided is product promotion or new product offers. However, the information provided is information that may not be important so that it can disturb the person who receives the information. This spam problem can be anticipated by creating a Machine Learning that is able to detect whether the incoming information is spam or not by classifying the content of SMS messages. Because there are only two variables in the labels contained in this study, this study uses the Binary Classification type by applying the Logistic Regression method to classify SMS messages. The purpose of this study is not only to create a spam detection tool, this research will also obtain test results on the Logistic Regression method based on the detection accuracy that can be achieved. The result of this study is to obtain SMS SPAM classification techniques with a high level of accuracy.

Keywords: *Binary Classification, Logistic Regression, SMS, Spam*

1. PENDAHULUAN

Spam merupakan penggunaan elektronik yang mengirimkan pesan secara bertubi-tubi tanpa dikehendaki oleh penerimanya. Berbagai macam bentuk spam antara lain melalui E-Mail, pesan (melalui social media atau SMS), social media, dan lain sebagainya. Spam dikirim oleh seseorang dengan biaya yang rendah karena tidak memerlukan klasifikasi terhadap penerima sehingga pesan yang dikirim dapat menjangkau bahkan kepada pelanggan yang tidak menginginkan pesan tersebut [1]. Dalam

e-ISSN: 2797-5274
p-ISSN: 2809-9788

perkembangannya spam diawali melalui surat elektronik (e-mail), namun seiring perkembangan teknologi komunikasi, selain melalui e-mail, spam juga dikirim melalui pesan singkat (SMS) yang langsung menjangkau secara personal. Meski demikian, bukan berarti spam adalah sesuatu yang merugikan, karena spam banyak digunakan oleh perusahaan maupun pelaku usaha kecil untuk memberikan informasi yang mungkin bermanfaat suatu saat nanti. Penelitian ini akan bertujuan memberikan klasifikasi terhadap pesan SMS yang masuk untuk membedakan mana pesan SMS yang termasuk spam dan mana pesan SMS yang bukan spam.

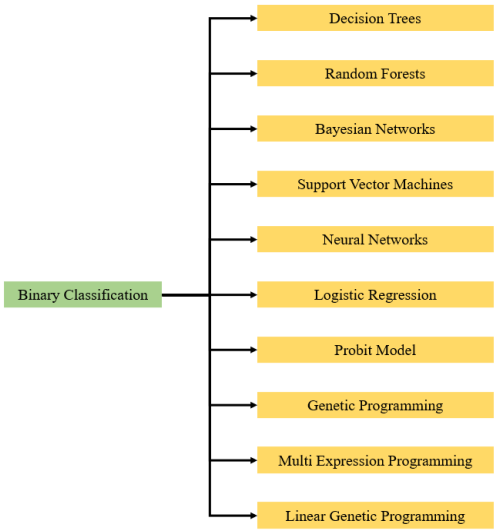
2. METODE PENELITIAN

Secara umum, penelitian ini menggunakan *binary classification* dengan menggunakan metode *logistic regression*. Di dalam metode *logistic regression* terdapat metode yang disebut *Term Frequency-Inverse Document Frequency* (TF-IDF) yang berfungsi mengecek kemunculan sebuah term pada sebuah dokumen. Hasil dari TF-IDF ini yang kemudian akan digunakan pada metode *logistic regression*.

2.1. Binary Classification

Klasifikasi biner adalah sebuah studi yang dipelajari pada *machine learning* yang kategori-kategorinya telah ditentukan sebelumnya dan digunakan untuk mengkategorikan pengamatan probabilistik baru ke dalam kategori-kategori tersebut. Dalam banyak permasalahan, klasifikasi menerapkan dikotomisasi pada situasi praktis dimana masing-masing kelompok tidak simetris sehingga yang menjadi perhatian utama adalah proposi relatif dari berbagai jenis kesalahan. Misalnya dalam pengujian medis yang bertujuan untuk mendeteksi penyakit padahal penyakit itu tidak ada *(false positive)* akan dianggap berbeda dengan tidak mendeteksi suatu penyakit padahal penyakit itu ada *(false negative).*

Klasifikasi biner pada dasarnya memiliki fungsi yang sama dengan klasifikasi yang digunakan pada pengambilan keputusan. Hanya saja pada klasifikasi yang digunakan pada pengambilan keputusan *(decision support system)* biasanya output yang dihasilkan bisa lebih dari dua (peringkat dari probabilitas hasil berdasarkan *score* yang didapatkan dari input [2][3]. Sedangkan pada klasifikasi biner output keputusan yang dihasilkan hanya berupa bilangan biner yaitu “0” dan “1”. Pada gambar 2.1 menunjukkan metode-metode yang umumnya digunakan untuk klasifikasi biner yang mana setiap klasifikasi memiliki fungsi domain tertentu berdasarkan jumlah pengamata, fitur dimensi vektor, gangguan dalam data dan faktor lainnya [4][5].



Gambar 2.1 Metode *Binary Classification*

2.2. Logistic Classification

Logistic Regression merupakan suatu teknik umum yang digunakan dalam statistika dan *machine learning* untuk melakukan *binary classification* [6]. Walaupun nama metode ini mengandung *regression* (yang merujuk pada model yang umum digunakan untuk melakukan *forecasting* atau mencari hubungan antara variabel terikat terhadap variabel bebas untuk mendapatkan *analytics* [7]), metode ini sebenarnya

e-ISSN: 2797-5274
p-ISSN: 2809-9788

adalah metode yang digunakan untuk melakukan klasifikasi. Dalam statistik, model logistik adalah model statistik yang memodelkan *log-odds* suatu peristiwa sebagai kombinasi linier dari satu atau lebih variabel bebas dengan memperkirakan parameter model logistik [8]. Dalam *logistic regression* terdapat satu variabel biner (label) yang nilainya "0" dan "1", sedangkan variabel bebasnya (features) dapat berupa variabel biner ataupun variabel riil. Probabilitas label dapat bervariasi antara nilai "0" dan "1", oleh karena itu diberi label yang mana kemudian fungsi inilah yang mengubah *log-odds* menjadi probabilitas dan menjadi fungsi logistik [9].

Logistic regression termasuk dalam algoritma *supervised machine learning* yang banyak digunakan untuk tugas klasifikasi biner seperti mendeteksi sebuah email termasuk kategori spam atau bukan dan juga dapat mendiagnosis penyakit dengan menilai ada atau tidak sebuah kondisi tertentu berdasarkan hasil tes yang dilakukan oleh pasien. Pendekatan ini menggunakan fungsi logistik untuk mengubah kombinasi linier input menjadi nilai probabilitas yang berkisar antara 0 dan 1 yang mana probabilitas ini akan menunjukkan kemungkinan bahwa input sesuai dengan salah satu dari dua kategori yang sudah ditentukan sebelumnya. Kurva yang dihasilkan pada fungsi logistik ini akan berbentuk huruf S yang khas yang mana secara fungsi logistik ini sangat efektif dalam memetakan bilangan bernilai riil apapun ke nilai dalam interval 0 hingga 1. Dengan menghitung probabilitas bahwa variabel dependen (terikat) akan dikategorikan ke dalam kelompok tertentu, regresi logistik memberikan kerangka probabilistik yang mendukung pengambilan keputusan yang tepat [10].

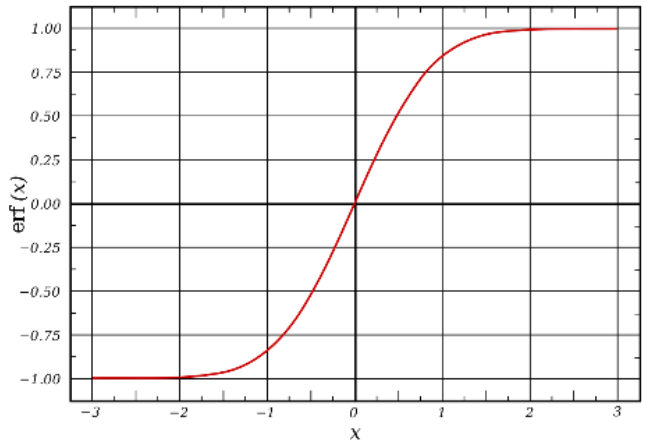
Model *logistic* dapat dilihat pada persamaan 1 dibawah ini.

$$g(X) = sigmoid(\alpha + \beta X) \quad (1)$$

Dimana fungsi *sigmoid* dapat diekspresikan pada persamaan 2 berikut ini.

$$sigmoid(x) = \frac{1}{1+exp(-x)} \quad (2)$$

Fungsi sigmoid adalah fungsi matematika yang grafiknya mempunyai ciri khas berbentu S atau kurva sigmoid [11]. Contoh umum fungsi sigmoid adalah fungsi logistik yang ditunjukkan pada gambar 2.2 berikut ini.



Gambar 2.2 *Curva Sigmoid*

Dari gambar *curva sigmoid* diatas dapat dilihat karakteristik penting yaitu kurva mendekati 0 ketika *x* menuju negatif tak terhingga dan mendekati 1 ketika *x* menuju positif tak terhingga (*asimptotik*); kurva selalu meningkat tetapi tidak pernah turun (*monotinik*); dan titik di mana kurva mencapai 0.5 adalah ketika *x*=0 [12].

Persamaan 3 adalah bentuk alternatif dari fungsi sigmoid dimana $\sigma(x)$ adalah *output* dari fungsi sigmoid untuk input *x*.

e-ISSN: 2797-5274
p-ISSN: 2809-9788

$$\sigma(x) = \frac{1}{1+e^{-x}} = \frac{e^{x}}{1+e^{x}} = 1 - \sigma(-x) \quad (3)$$

Metode *Term Frequency-Inverse Document Frequency* (TF-IDF) adalah teknik yang digunakan dalam pemrosesan bahasa alami (*Natural Language Processing*) dan penambangan teks untuk mengukur seberapa penting sebuah kata bagi sebuah dokumen dalam sebuah kumpulan dokumen. TF-IDF terdiri dari dua jenis statistik: yaitu *Term Frequency* (TF) dan *Inverse Document Frequency* (IDF).

Term Frequency merupakan frekuensi kemunculan term *i* pada dokumen *j* dibagi dengan total term pada dokumen *j*. *Term Frequency* dapat ditulis dalam bentuk sebagai berikut.

$$tf_{ij} = \frac{f_d(i)}{\max_{j \in d} f_d(j)} \quad (4)$$

Inverse Document Frequency (IDF) berfungsi untuk mengurangi bobot suatu *term* jika kemunculannya banyak tersebar diseluruh dokument. IDF dapat ditulis dalam bentuk persamaan sebagai berikut.

$$idf(t, D) = log \frac{N}{|\{d \in D : t \in d\}|} \quad (5)$$

Dimana N adalah jumlah total dokument dalam *corpus*, N=|D|. $|\{d \in D : t \in d\}|$ adalah jumlah dokumen yang mengandung *term t*. IDF juga dapat dituliskan dalam bentuk persamaan sebagai berikut.

$$idf(t, D) = log\left(\frac{N}{df(t)+1}\right) \quad (6)$$

Penambahan 1 untuk menghindari pembagian terhadap 0 jika *df(t)* tidak ditemukan pada corpus.

2.3. Evaluasi Matriks

Hasil dari penelitian ini akan diuji menggunakan *confusion matrix*, *accuracy*, *precission & recall*, *F1-Score* dan *Receiver Operating Characteristic* (ROC). *Confusion matrix* merupakan sebuah alat yang digunakan untuk mengukur kinerja algoritma klasifikasi dalam *machine learning*. Matriks ini adalah tabel yang merangkum hasil prediksi model klasifikasi dibandingkan dengan label sebenarnya dari data uji. *Confusion matrix* terdiri dari empat elemen utama yang mencerminkan berbagai kemungkinan hasil dari klasifikasi yaitu *true positif* (TP) yang berarti jumlah kasus di mana model memprediksi positif, dan sebenarnya memang positif; *true negatif* (TN) yang berarti jumlah kasus dimana model memprediksi negatif, dan sebenarnya memang negatif; *false positif* (FN) yang berarti jumlah kasus di mana model memprediksi positif, tetapi sebenarnya negatif; dan *false negatif* (FN) yang bermakna jumlah kasus di mana model memprediksi negatif, tetapi sebenarnya positif [13]. *Accuracy, precission, recall,* dan *F1-Score* dihitung berdasarkan hasil matriks dari *confusion matrix* berikut ini.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (7)$$

$$Precission = \frac{TP}{TP+FP} \quad (8)$$

$$Recall = \frac{TP}{TP+FN} \quad (9)$$

$$F1Score = 2 \times \frac{Precission \times Recall}{Precission+Recall} \quad (10)$$

Receiver Operating Characteristic (ROC) merupakan alat grafis yang digunakan untuk mengevaluasi kinerja model klasifikasi biner. Grafik ROC menggambarkan hubungan antara *True Positive Rate* (TPR) dan *False Positive Rate* (FPR) pada berbagai *threshold* klasifikasi. TPR juga dikenal sebagai *recall* atau sensitivity yang bertujuan mengukur proporsi contoh positif yang benar-benar diidentifikasi sebagai positif oleh model [14].

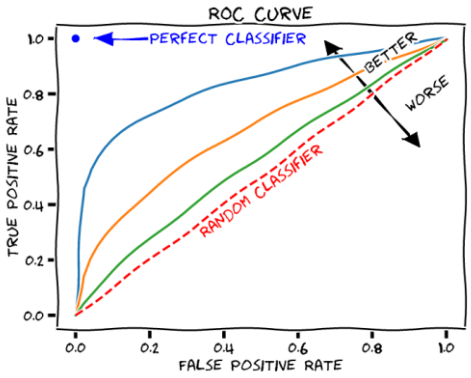
e-ISSN: 2797-5274
p-ISSN: 2809-9788

$$TPR = \frac{TP}{TP+FN} \quad (11)$$

FPR mengukur proporsi contoh negatif yang salah diidentifikasi sebagai positif oleh model. Berikut adalah persamaan FPR.

$$FPR = \frac{FP}{FP+TN} \quad (12)$$

Gambar 2.3 adalah kurva ROC yang terdapat sumbu *Y* yang berarti *True Positive Rate* (TPR) dan sumbu *X* yang berarti *False Positive Rate* (FPR). Garis diagonal dari (0,0) ke (1,1) merepresentasikan model yang melakukan tebakan acak. Model yang lebih baik dari tebakan acak akan berada di atas garis ini. Semakin tinggi kurva ROC dari model klasifikasi (mendekati sudut kiri atas), semakin baik kinerja model. Gambar 2.3 Kurva ROC



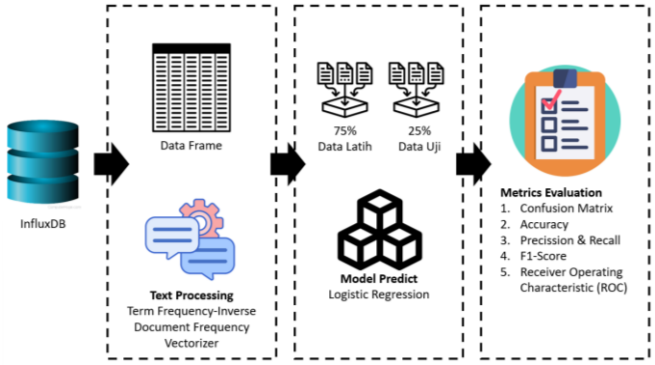
Gambar 2.3 Curva ROC

3. HASIL DAN PEMBAHASAN

3.1. Pembahasan

Penelitian ini dikerjakan menggunakan bahasa pemrograman *Python* dengan menggunakan database tidak terstruktur InfluxDB. InfluxDB adalah layanan database berbasis *open source* yang menyediakan penyimpanan data dalam bentuk deret waktu (*timeseries database*) untuk instrumen, pengamatan, pembelajaran dan mengotomatiskan berbagai jenis aplikasi maupun proses bisnis dengan berbagai macam tujuan [15].

Data SMS yang tersimpan pada database InfluxDB kemudian akan bertransformasi menjadi bentuk *data frame* dengan memanfaatkan metode ETL (*extract, transform, load*). Gambar 3.1 menjelaskan tentang skema penelitian ini yang membagi kedalam tiga bagian utama yaitu melakukan *text processing* menggunakan TF-IDF, kemudian melakukan prediksi dengan metode *logistic regression* yang mana membagi data kedua bagian yaitu data latih sebanyak 75% dan data uji sebanyak 25% dan bagian yang terakhir adalah melakukan evaluasi terhadap hasil klasifikasi dari model *logistic regression*.



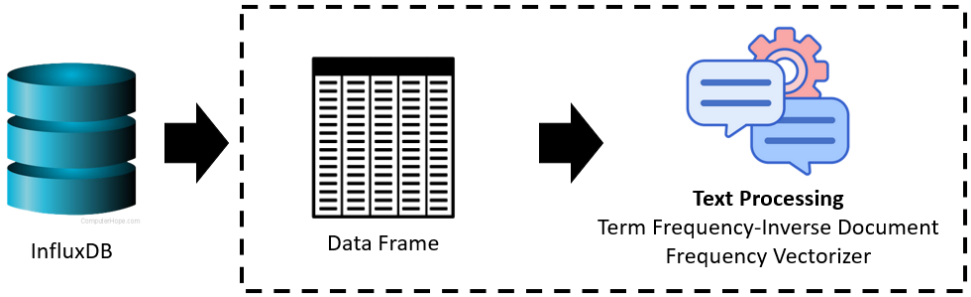
Gambar 3.1 Skema Penelitian

e-ISSN: 2797-5274
p-ISSN: 2809-9788

Gambar 3.2 adalah hasil dari *extract* data yang telah diubah kedalam bentuk *data frame*. Dari *data frame* inilah yang kemudian pada gambar 3.3 dilakukan proses *text processing* menggunakan TF-IDF.

	label	sms
0	ham	Go until jurong point, crazy.. Available only ...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I don't think he goes to usf, he lives aro...

Gambar 3.2 Data Frame



Gambar 3.3 Skema ETL

Sebelum melakukan *text processing* dengan menggunakan TF-IDF, data yang sudah diubah dalam bentuk *data frame* akan dipecah menjadi dua bagian yaitu data latih dan data uji. Script untuk membagi data menjadi dua bagian adalah sebagai berikut:

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X,
                                                    y,
test_size=0.25,
random_state=0)
```

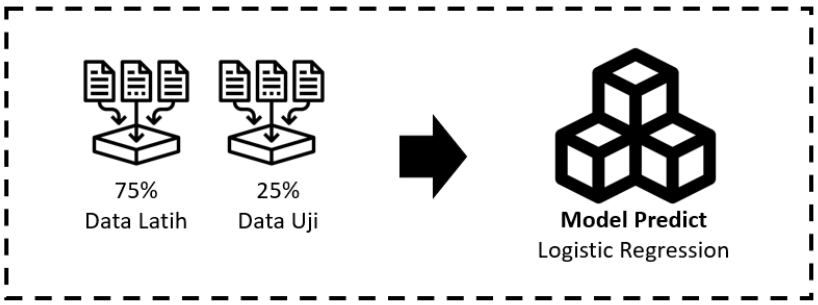
Setelah mendapatkan data yang akan digunakan sebagai model dan data yang akan digunakan sebagai data uji, maka selanjutnya adalah melakukan *text processing*. Script *Python* untuk melakukan *text processing* dengan metode TF-IDF adalah sebagai berikut:

```
from sklearn.feature_extraction.text import
TfidfVectorizer

vectorizer = TfidfVectorizer(stop_words='english')

X_train_tfidf = vectorizer.fit_transform(X_train)
X_test_tfidf = vectorizer.transform(X_test)
```

Setelah proses *text processing* telah dilakukan, maka proses yang terakhir seperti yang digambarkan pada gambar 3.3 dimana setelah data dibagi menjadi dua bagian maka akan dilakukan *binary classification* dengan metode *logistic regression*. *Script Python* untuk metode *logistic regression* adalah sebagai berikut:

e-ISSN: 2797-5274
p-ISSN: 2809-9788

```
from sklearn.linear_model import LogisticRegression

model = LogisticRegression()
model.fit(X_train_tfidf, y_train)
y_pred = model.predict(X_test_tfidf)

for pred, sms in zip(y_pred[:5], X_test[:5]):
  print(f'PRED: {pred} - SMS: {sms}\n')
```



Gambar 3.3 Pembagian Data

Hasil dari *script python* yang memproses metode *logistic regression* adalah sebagai berikut:

```
PRED: 0 - SMS: Storming msg: Wen u lift d phne, u say "HELLO" Do u knw wt is d real meaning of HELLO?? . . . It's d nam
e of a girl..! . . . Yes.. And u knw who is dat girl?? "Margaret Hello" She is d girlfrnd f Grahmbell who invnted telph
one... . . . . Moral:One can 4get d name of a person, bt not his girlfrnd... G o o d n i g h t . . .@

PRED: 0 - SMS: <Forwarded from 448712404000>Please CALL 08712404000 immediately as there is an urgent message waiting f
or you.

PRED: 0 - SMS: And also I've sorta blown him off a couple times recently so id rather not text him out of the blue look
ing for weed

PRED: 0 - SMS: Sir Goodmorning, Once free call me.

PRED: 0 - SMS: All will come alive.better correct any good looking figure there itself..
```

Gambar 3.4 Hasil *Logistic Regression*

Dari hasil proses data dengan metode *logistic regression* dapat dilihat bahwa ada dua variabel yaitu “PRED: 0” dan kalimat isi dari SMS. Variabel “PRED: 0” adalah sebuah label yang diberikan kepada SMS dimana 0 berarti bahwa SMS masuk kategori bukan SPAM.

3.2. Hasil

Berdasarkan model yang telah dikerjakan sebelumnya, maka tahap selanjutnya adalah mendapatkan hasil uji berdasarkan model yang telah didapatkan. Uji pada penelitian ini yaitu *confusion matrix*, *accuracy*, *precission & recall*, *F1-Score*, dan *receiver operating characteristic (ROC)*.

A. Confusion Matrix

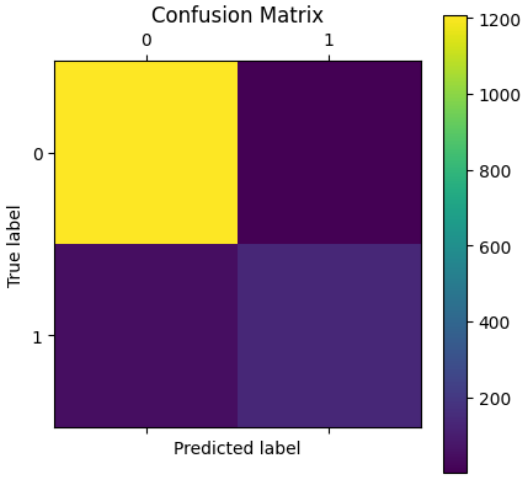
Untuk mendapatkan *confusion matrix* maka perlu menggunakan *class confusion_matrix* yang sudah disediakan oleh *sklearn*. Berikut adalah *source code* untuk mendapatkan data *true positive, true negative, false positive,* dan *false negative*. *Script Python* untuk mendapatkan *confusion matrix* adalah sebagai berikut:

```
from sklearn.metrics import confusion_matrix
matrix = confusion_matrix(y_test, y_pred)
matrix

tn, fp, fn, tp = matrix.ravel()
```

e-ISSN: 2797-5274
p-ISSN: 2809-9788

Dari *script* diatas didapatkan hasil *True Negative* sebesar 1.207, *False Positive* sebesar 1, *False Negative* sebesar 48, dan *True Positive* sebesar 137. Selanjutnya adalah membuat visualisasi dari *confusion matrix* dengan menggunakan *script Python* berikut ini.

```
import matplotlib.pyplot as plt
plt.matshow(matrix)
plt.colorbar()

plt.title('Confusion Matrix')
plt.ylabel('True label')
plt.xlabel('Predicted label')
plt.show()
```



Gambar 3.5 Confusion Matrix

B. *Accuracy, Precission, Recall & F1-Score*

Script Python berikut ini adalah untuk mendapatkan hasil pengujian *accuracy, precission, recall* dan *F1-Score.*

```
from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
from sklearn.metrics import f1_score

accuracy_score(y_test, y_pred)
precision_score(y_test, y_pred)
recall_score(y_test, y_pred)
f1_score(y_test, y_pred)
```

Dari script diatas, maka akan langsung didapatkan hasil dari masing-masing pengujian dimana hasil *accuracy* adalah 0.964824120603015, hasil *precission* adalah 0.9927536231884058, hasil *recall* adalah 0.7405405405405405, dan hasil *F1-Score* adalah 0.8482972136222909.

C. *Receiver Operating Characteristic (ROC)*

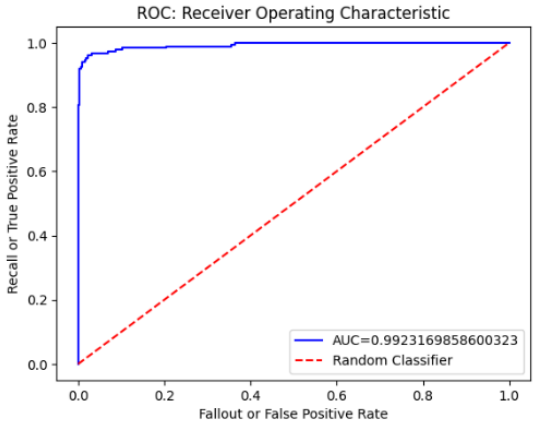
Dari *SKLearn* yaitu *roc_curve* dan *auc*. *Script Python* untuk mendapatkan nilai ROC dan memvisualisasikan kedalam bentuk grafik adalah sebagai berikut:

```
from sklearn.metrics import roc_curve, auc

prob_estimates =
model.predict_proba(X_test_tfidf)
```

e-ISSN: 2797-5274
p-ISSN: 2809-9788

```
fpr, tpr, threshhold = roc_curve(y_test,
prob_estimates[:,1])
nilai_auc = auc(fpr, tpr)

plt.plot(fpr, tpr, 'b',
label=f'AUC={nilai_auc}')
plt.plot([0,1], [0,1], 'r--', label='Random
Classifier')

plt.title('ROC: Receiver Operating
Characteristic')
plt.xlabel('Fallout or False Positive Rate')
plt.ylabel('Recall or True Positive Rate')
plt.legend()
plt.show()
```

Gambar 3.6 adalah hasil dari ROC dalam bentuk grafik dimana berdasarkan *script Python* diatas didapatkan nilai ROC sebesar 0.9923169858600323.



Gambar 3.6 Grafik ROC

4. KESIMPULAN

Berdasarkan hasil uji yang didapatkan dengan menggunakan 5 jenis evaluasi matriks yang diterapkan pada *Binary Classification* dengan menggunakan metode *Logistic Regression*, dapat diketahui bahwa metode ini memiliki akurasi yang cukup tinggi yaitu sebesar 0.96 atau 96%, nilai presisi sebesar 0.99 atau 99%, dan nilai ROC sebesar 0.99 atau 99% sehingga metode ini sangat baik ketika digunakan untuk mendeteksi SPAM pada pesan singkat SMS. Karena penelitian ini dikerjakan pada sisi *backend* atau bisa di bilang pada sisi *server* pada *provider* jaringan SMS, maka sistem hasil penelitian ini bisa disisipkan sebelum SMS dikirim ke tujuan dan akan menambahkan informasi *header* kepada user yang membertahu SMS yang diterima termasuk kategori SPAM atau bukan.

REFERENSI

[1] P. Gerbaudo. 2014. Spam: a shadow history of the internet. Information Communication and Society, 17 (7), doi: 10.1080/1369118x.2013.873475.
[2] B. T. Sutrisno and W. Andriyani. 2021. Penerapan Madm Dengan Metode Saw Untuk Menentukan Target Promosi Berdasarkan Asal Jurusan Di Sekolah. *Jurnal Simetris*, 11 (2), 480-492, doi: 10.24176/simet.v11i2.4784.
[3] H. Muhrial, B. Purnomosidi.D.P, W. Andriyani, and H. Hamdani. 2022. Data Warehouse to Support the Decision Using Vikor Method. *Journal of Intelligent Software Systems,* 1 (2), 153-176, doi: 10.26798/jiss.v1i2.767.
[4] R. Zhang and A. Zakhor. 2014. Automatic identification of window regions on indoor point clouds using LiDAR and cameras, doi: 10.1109/WACV.2014.6836112.
[5] Y. Lu and C. Rasmussen. 2012. Simplified markov random fields for efficient semantic labeling of 3D point clouds, doi: 10.1109/IROS.2012.6386039.
[6] S. Menard. 2014. Logistic Regression: From Introductory to Advanced Concepts and Applications.

e-ISSN: 2797-5274
p-ISSN: 2809-9788

[7] K. Iriyanta, B. P. D. Putranto, and W. Andriyani. 2023. IOT Based Soil Moisture Monitoring And Soil Moisture Prediction Using Linear Regression (Case Study of Vinca Plants). *Journal of Intelligent Software Systems,* 2 (1), doi: 10.26798/jiss.v2i1.929.

[8] J. Tolles and W. J. Meurer. 2016. Logistic regression: Relating patient characteristics to outcomes," JAMA. Journal of the American Medical Association, 316 (5), doi: 10.1001/jama.2016.7653.

[9] A. J. Scott, D. W. Hosmer, and S. Lemeshow. 1991. Applied Logistic Regression. *Biometrics*, 47 (4), doi: 10.2307/2532419.

[10] M. Pohar, M. Blas, and S. Turk. 2004. Comparison of Logistic Regression and Linear Discriminant Analysis: A Simulation Study. *Metodološki zvezki*, 1 (1).

[11] J. Han and C. Moraga. 1995. The influence of the sigmoid function parameters on the speed of backpropagation learning," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 930, doi: 10.1007/3-540-59497-3_175.

[12] I. Goodfellow, Y. Bengio, and A. Courville. 2016. Deep Learning. MIT Press.

[13] K. P. Murphy. 1991. Machine Learning: A Probabilistic Perspective.

[14] T. Fawcett. 2006. An introduction to ROC analysis. *Pattern Recognition Letters*, 27 (8), doi: 10.1016/j.patrec.2005.10.010.

[15] Hendra and W. Andriyani. 2010. Studi Komparasi Menyimpan dan Menampilkan Data Histori Antara Database Terstruktur Mariadb dan Database Tidak Terstruktur Influxdb. *Jurnal Teknologi Technoscientia*, 12 (2), 168-174.