PENGEMBANGAN MODEL *MACHINE LEARNING* UNTUK MENDETEKSI URL *PHISHING*

IF5100 – PEMROGRAMAN UNTUK DATA ANALITIK

Disusun oleh

Kelompok 9:

Clement Nathanael Lim	**/ 18222032**
Najmi I T Kertasafari	**/ 23324005**
Batrisyia Zahrani Ananto	**/ 23525042**

PROGRAM STUDI MAGISTER INFORMATIKA

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

2025

DAFTAR ISI

1 Penjelasan *Dataset*

Dataset yang digunakan dalam Tugas Besar kali ini adalah *dataset* PhiUSILL *Phising* URL. PhiUSILL *Phishing* URL merupakan *dataset* yang terdiri dari deskripsi suatu URL dan juga fitur – fitur yang terkait dengan URL tersebut beserta *label* URL *legitimate* dan URL *phishing*. *Dataset* terdiri atas 235.975 *rows* dan 55 kolom / fitur. Dari 235.975 *rows* tersebut, 134.850 *rows* merupakan URL *legitimate* dan sekitar 100.945 *rows* merupakan URL *phishing*. URL yang digunakan dalam *dataset* ini adalah URL – URL terbaru. *Link* dari *dataset* dapat diakses dari tautan berikut: https://www.kaggle.com/datasets/ndarvind/phiusiil-phishing-url-dataset

Berikut ini merupakan daftar kolom yang ada di *dataset* PhiUSILL *Phishing* URL *dataset*.

#	Column	Non-Null Count	Dtype
0	URL	235795 non-null	object
1	URLLength	235795 non-null	int64
2	Domain	235795 non-null	object
3	DomainLength	235795 non-null	int64
4	IsDomainIP	235795 non-null	int64
5	TLD	235795 non-null	object
6	URLSimilarityIndex	235795 non-null	float64
7	CharContinuationRate	235795 non-null	float64
8	TLDLegitimateProb	235795 non-null	float64
9	URLCharProb	235795 non-null	float64
10	TLDLength	235795 non-null	int64
11	NoOfSubDomain	235795 non-null	int64
12	HasObfuscation	235795 non-null	int64
13	NoOfObfuscatedChar	235795 non-null	int64
14	ObfuscationRatio	235795 non-null	float64
15	NoOfLettersInURL	235795 non-null	int64
16	LetterRatioInURL	235795 non-null	float64
17	NoOfDegitsInURL	235795 non-null	int64
18	DegitRatioInURL	235795 non-null	float64
19	NoOfEqualsInURL	235795 non-null	int64
20	NoOfQMarkInURL	235795 non-null	int64
21	NoOfAmpersandInURL	235795 non-null	int64
22	NoOfOtherSpecialCharsInURL	235795 non-null	int64
23	SpacialCharRatioInURL	235795 non-null	float64
24	IsHTTPS	235795 non-null	int64
25	LineOfCode	235795 non-null	int64
26	LargestLineLength	235795 non-null	int64
27	HasTitle	235795 non-null	int64
28	Title	235795 non-null	object
29	DomainTitleMatchScore	235795 non-null	float64
30	URLTitleMatchScore	235795 non-null	float64
31	HasFavicon	235795 non-null	int64
32	Robots	235795 non-null	int64

```
33  IsResponsive               235795 non-null  int64
34  NoOfURLRedirect            235795 non-null  int64
35  NoOfSelfRedirect           235795 non-null  int64
36  HasDescription             235795 non-null  int64
37  NoOfPopup                  235795 non-null  int64
38  NoOfiFrame                 235795 non-null  int64
39  HasExternalFormSubmit      235795 non-null  int64
40  HasSocialNet               235795 non-null  int64
41  HasSubmitButton            235795 non-null  int64
42  HasHiddenFields            235795 non-null  int64
43  HasPasswordField           235795 non-null  int64
44  Bank                       235795 non-null  int64
45  Pay                        235795 non-null  int64
46  Crypto                     235795 non-null  int64
47  HasCopyrightInfo           235795 non-null  int64
48  NoOfImage                  235795 non-null  int64
49  NoOfCSS                    235795 non-null  int64
50  NoOfJS                     235795 non-null  int64
51  NoOfSelfRef                235795 non-null  int64
52  NoOfEmptyRef               235795 non-null  int64
53  NoOfExternalRef            235795 non-null  int64
54  label                      235795 non-null  int64
```

Pada *dataset* ini, kebanyakan kolom / fitur merupakan *fields* yang bersifat biner (0 atau 1). Terdapat juga *fields* yang bersifat numerikal maupun kategorikal. Beberapa fitur – fitur penting yang ada di dalam *dataset* ini (yang menjadi fitur yang paling berkorelasi tinggi dengan *label* – akan dijelaskan lebih lanjut di bagian berikutnya), antara lain:

Nama Fitur	**Deskripsi Fitur**
DomainTitleMatchScore	Sebuah skor yang mengukur seberapa mirip nama domain dengan judul halaman *web*.
URLTitleMatchScore	Skor yang mengukur seberapa mirip keseluruhan URL dengan judul halaman *web*.
SpacialCharRatioInURL	Sebuah rasio nilai yang dihitung dengan membagi jumlah karakter spesial dengan total panjang URL.
TLD	Sebuah kolom yang berisi *string* yang merupakan bagian akhir dari nama domain.
HasSocialNet	Sebuah kolom dengan nilai biner yang menunjukkan apakah *link* tersebut mengarah ke jaringan sosial.

HasCopyrightInfo	Sebuah kolom dengan nilai biner yang menunjukkan apakah halaman *web* tersebut berisi informasi hak cipta.
HasDescription	Sebuah kolom dengan nilai biner yang menunjukkan apakah halaman *web* tersebut memiliki *tag meta* deskripsi.
IsHTTPS	Sebuah kolom dengan nilai biner yang menunjukkan apakah halaman *web* menggunakan protokol HTTPS atau tidak.
HasSubmitButton	Sebuah kolom dengan nilai biner yang menunjukkan apakah halaman *web* tersebut berisi tombol *submit* di dalam sebuah *form* atau tidak.
IsResponsive	Sebuah kolom dengan nilai biner yang menunjukkan apakah desain halaman *web* tersebut responsif.

2 *Data Preparation*

Pada bagian *data preparation*, kami akan melakukan beberapa *step* yang bertujuan untuk membersihkan data, melakukan analisis data yang bertujuan untuk eksplorasi data, mengenal lebih lanjut terkait data yang akan digunakan sebelum digunakan untuk pembangunan model *machine learning*.

2.1 *Data Exploration* (*Exploratory Data Analysis*)

Untuk tugas besar, kelompok kami akan melakukan *import dataset* terlebih dahulu, serta membagi data menjadi *data training* dan juga *data testing*. Pembagian kami lakukan dengan membagi menjadi 30% *data testing* dan 70% *data training*. Pembagian dilakukan dengan rasionalisasi bahwa jumlah *rows* cukup banyak sehingga diperlukan data lebih untuk *training*.

Setelah melakukan *import dataset*, kami akan melakukan *exploratory data analysis* (EDA) yang bertujuan untuk memahami bagaimana kondisi awal dari data tersebut. Beberapa hal yang dilakukan, dalam proses EDA, antara lain:

1) Melihat bentuk dan informasi umum dari *dataset*.

- Pada bagian ini, kami akan melihat info dari *dataset*, antara lain dengan menunjukkan jumlah *rows* dan *column* / fitur dari *dataset*. Didapatkan bahwa *dataset* terdiri atas 235.795 baris dan 55 kolom.
- Selain itu, kami juga mencari info dari tiap *dataset*, antara lain nama dari tiap kolom beserta tipe dari kolom tersebut.

2) Melihat deskripsi dari *dataset*.
    - Pada bagian ini, kami akan melihat nilai statistik dari tiap kolom, antara lain terdiri atas *count*, *mean*, *standard deviation*, dan sebagainya.
3) Mencari nilai *missing values*, NaN *values*, dan nilai duplikasi.
    - Pada bagian ini, kami akan melihat nilai dari *missing*, NaN, dan nilai duplikasi. Berdasarkan hasil yang didapat, *dataset* ini pada dasarnya merupakan *dataset* yang cukup bersih, karena tidak ada *missing values* maupun duplikasi baris.
4) Mencari URL dengan format tidak standar.
    - Pada bagian ini, kami akan mencari URL yang tidak standar. Beberapa kriteria URL yang tidak standar yang kami definisikan, antara lain adalah tidak diawali dengan HTTPS / HTTP / WWW. Didapatkan bahwa semua URL yang ada di dalam *dataset* tersebut semuanya standar.
5) Mencari URL yang memiliki karakter aneh.
    - Pada bagian ini, kami akan mencari URL yang memiliki karakter tidak sesuai. Contoh karakter yang tidak sesuai adalah karakter selain dari karakter alfabet maupun simbol pada umumnya. Didapatkan bahwa terdapat 2 URL yang memiliki karakter aneh, yang bisa mengindikasikan adanya URL *phishing*.

2.2 *Data Cleaning*

Setelah melakukan EDA, tahap berikutnya adalah melakukan *data cleaning*. Karena pada EDA didapatkan bahwa tidak ada *missing values* maupun *null values*, kami memfokuskan proses *data cleaning* ke bagian standarisasi, antara lain adalah dengan mencari URL yang memiliki huruf besar, karena pada dasarnya tidak ada URL yang menggunakan huruf besar. Setelah dilakukan proses *cleaning*, didapatkan bahwa sekarang semua URL sudah standar, yakni menggunakan huruf kecil semua.

2.3 *Data Validation*

Proses validasi merupakan suatu proses Beberapa proses validasi yang dilakukan, antara lain adalah:

1) Validasi apakah panjang URL sama dengan kolom URLLength.
    - Kami melakukan validasi apakah panjang kolom URL sudah sesuai dengan URLLength. Didapatkan bahwa terdapat sekitar 187.151 URL yang tidak sesuai dengan kolom URLLength.
2) Validasi apakah domain dari URL sudah ada di URL atau belum.
    - Kami melakukan validasi apakah domain yang ada di kolom domain sudah ada di URL atau belum. Didapatkan bahwa semua domain sudah sesuai dengan yang ada di URL.
3) Validasi apakah domain sudah sesuai format.
    - Kami melakukan validasi apakah domain sudah sesuai format, yakni dimulai dengan – dan diakhiri dengan –. Didapatkan bahwa semua URL yang ada sudah memenuhi format domain yang sesuai.
4) Validasi apakah domain IP dan TLD sudah mengikuti format.
    - Kami melakukan validasi apakah *format* IP sudah sesuai dengan format atau belum. Validasi berupa format IP berupa minimal 3 angka dalam 4 sub–kelompok (XXX.XXX.XXX.XXX) dan apakah IP *web* tersebut sama dengan kolom IsDomainIP. Terdapat 39 URL yang tidak memiliki format IP yang sesuai.
    - Selain itu, dilakukan validasi juga untuk TLD apakah sudah sesuai dengan standar TLD pada umumnya ('com', 'org', 'net', 'gov', 'edu', 'uk', 'de', 'io', 'co', 'info', 'xyz'). Didapatkan sekitar 71.161 URL dengan TLD yang tidak sesuai dengan standar.

2.4 *Data Integration*

Pada bagian *data integration*, terdapat beberapa kolom yang dapat diintegrasikan untuk dapat memperkaya fitur dari *dataset* PhiUSILL *Phishing* URL, antara lain:

1) *Content Suspicious Features*
    - Dilakukan penjumlahan untuk kolom – kolom dengan nilai 0/1 untuk aktivitas yang mencurigakan, antara lain terdiri atas 'NoOfPopup', 'NoOfiFrame',

'HasExternalFormSubmit', 'HasHiddenFields', 'HasPasswordField', 'NoOfSelfRedirect'.

2) *Financial Keyword Score*
    - Dilakukan penjumlahan untuk kolom yang berhubungan dengan keuangan, antara lain adalah ‘Bank’, ‘Pay’, dan ‘Crypto’.
3) *Ratio Feature Self Reference*
    - Fitur yang berupa rasio halaman fitur yang mengacu ke URL sendiri atau tidak.
4) *Ratio Empty Reference*
    - Fitur yang berupa rasio *link* kosong pada suatu URL.
5) *Insecure Password Fields*
    - Fitur yang mendeteksi apakah suatu URL aman atau tidak, yang dapat dicek berdasarkan kolom IsHTTPS dan HasPasswordFields.
6) *Feature Mismatched Financial Page*
    - Fitur yang bisa mendeteksi jika skor kecocokan rendah tapi halaman ada *keyword* finansialnya, yang bisa mengarahkan ke penipuan, yang dapat dilihat dari nilai DomainTitleMatchScore yang lebih rendah dari 0,5 tetapi memiliki *feature financial keyword score* lebih dari 0.

2.5 *Feature Encoding*

Feature encoding digunakan untuk *encoding* fitur kategorikal, antara lain seperti TLD, Robots, dan beberapa fitur lainnya. *Feature encoding* bertujuan untuk memastikan data – data kategorikal juga dapat digunakan seperti penggunaan data numerikal. *Feature encoding* akan menggunakan OneHotEncoder yang cukup mudah untuk digunakan.

2.5 *Feature Engineering*

Pada bagian *feature engineering*, kami menggunakan *feature selector* untuk dapat melakukan pemilihan fitur. *Feature selector* akan menggunakan metode Cramer’s V dengan *threshold* sebesar 0,2. Untuk membuat *class feature selector* ini, akan dilakukan inisialisasi, lalu akan dilakukan *fitting* berdasarkan kolom numerikal. Selanjutnya, untuk fitur kategorikal, akan digunakan fitur Cramer’s V yang akan menghitung berdasarkan *contingency table*. Selanjutnya, akan dipilih sekitar 10 fitur numerikal dan 5 fitur kategorikal yang akan digunakan untuk melatih model.

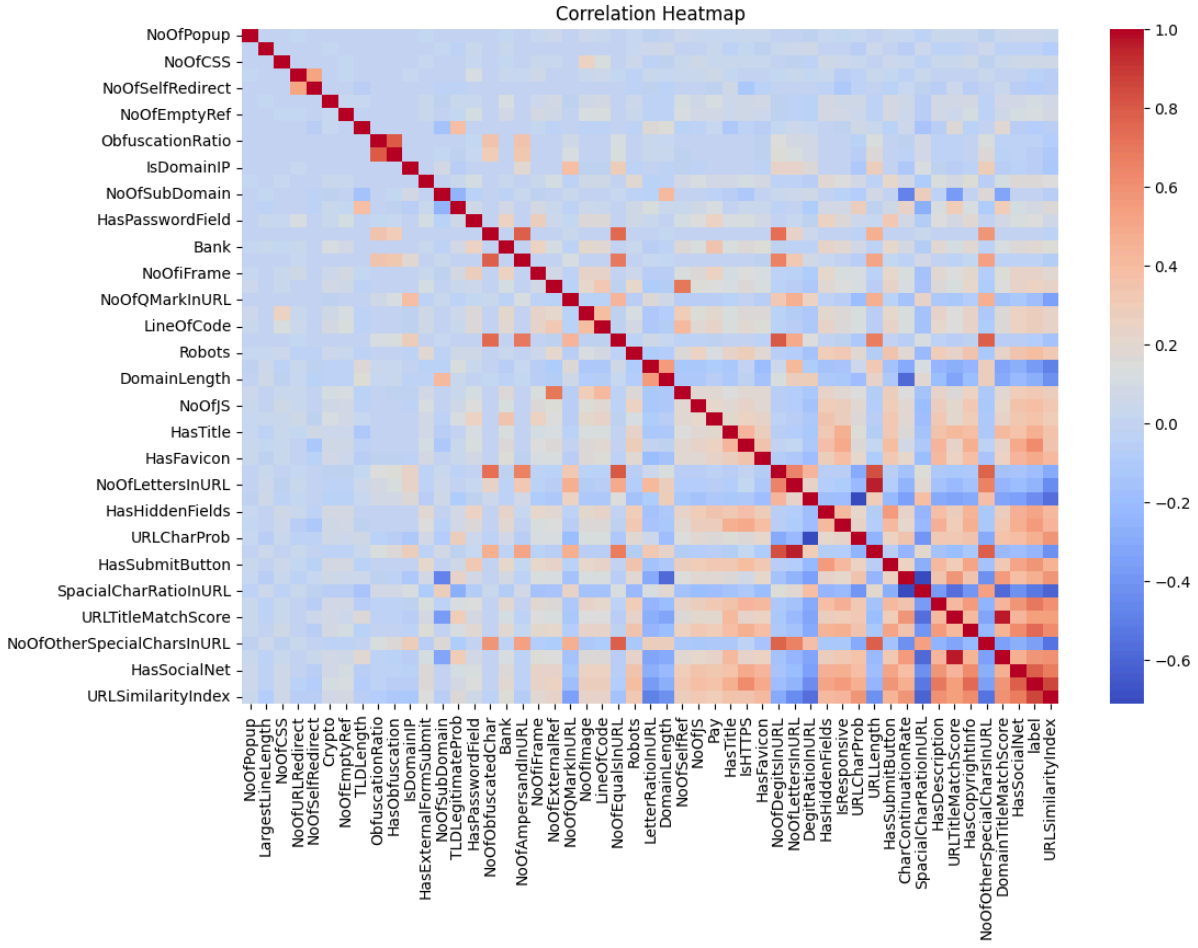
Selanjutnya, akan dilakukan *feature scaling* untuk melakukan *scaling* / pemerataan. Untuk fitur kategorikal akan digunakan *standard scaler* untuk memastikan bahwa data numerikal memiliki nilai rata – rata 0 dan memiliki standar deviasi 1.

3 *Data Visualization*

Akan dilakukan beberapa visualisasi pada data yang digunakan untuk melihat keterhubungan antar fitur yang ada pada dataset yang digunakan dan persebaran nilai pada tiap-tiap fitur.

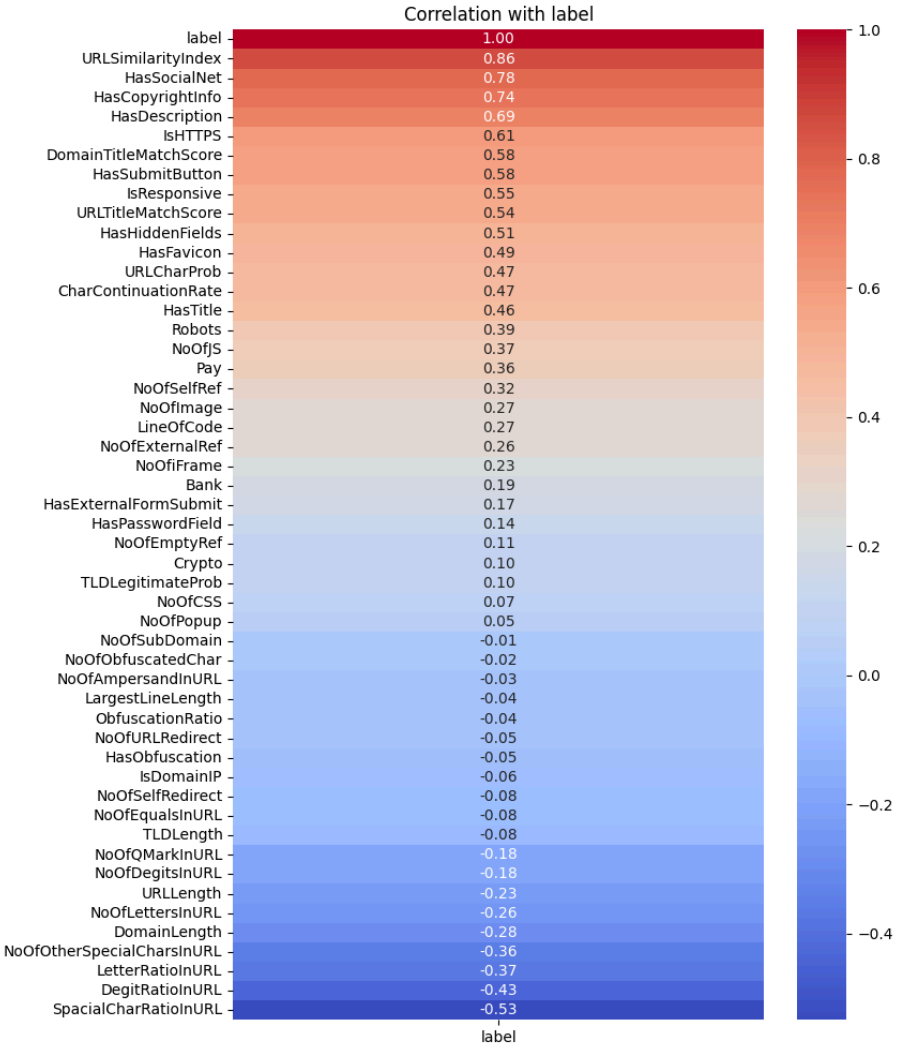
3.1 *Heatmap*

Heatmap digunakan untuk melihat kekuatan hubungan atau korelasi antar fitur. *Heatmap* sudah diurutkan berdasarkan fitur yang paling berkorelasi positif atau bernilai 1 hingga fitur yang paling berkorelasi negatif atau bernilai -1.



Gambar 3.1 *Correlation Heatmap*

Karena banyaknya jumlah fitur yang ada pada dataset, angka detail pasti korelasi tiap fiturnya tidak tercantum dengan jelas. Maka dari itu, digunakanlah *heatmap* khusus untuk melihat korelasi fitur-fitur yang ada dengan label atau targetnya. Berdasarkan gambar 3.2, dapat dilihat bahwa tiga fitur dengan korelasi paling positif dengan label adalah URL Similarity Index, Has Social Net, dan Has Copyright Info. Sedangkan korelasi paling negatifnya adalah Spacial Char Ratio In URL, Degit Ratio In URL, dan Letter Ratio In URL.

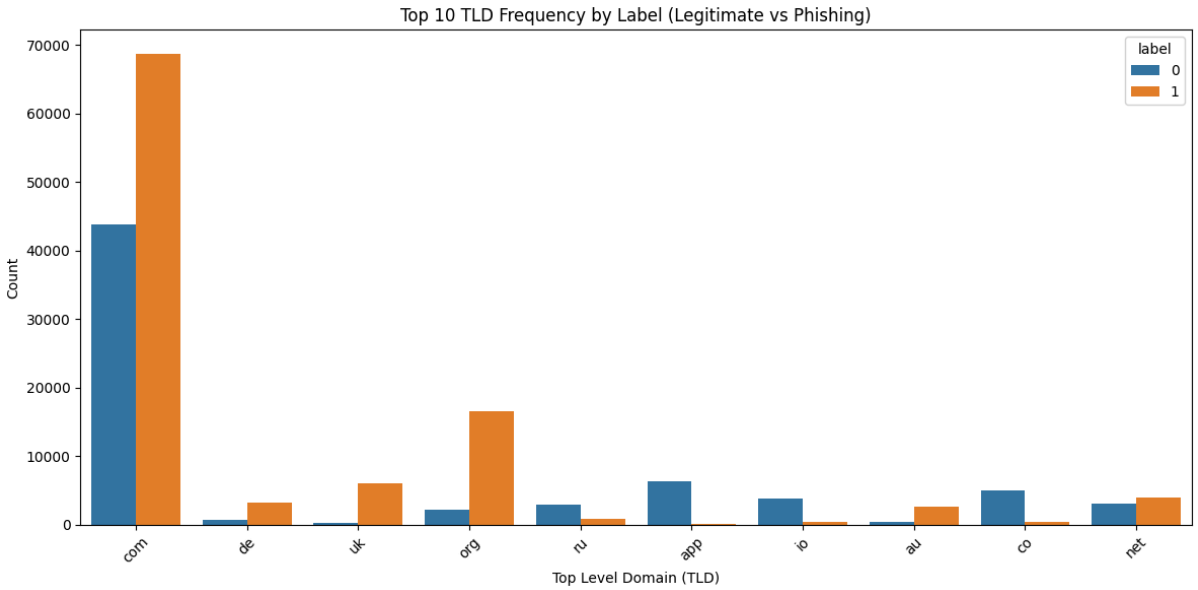


Gambar 3.2 *Correlation Heatmap* Terhadap Label

3.2 TLD *Frequency Plot*

TLD *frequency plot* digunakan untuk menunjukkan berapa banyak domain yang menggunakan TLD tertentu, dipisahkan berdasarkan label 1 (*legitimate*) dan label 0 (*phishing*). Berdasarkan hasil yang didapatkan, TLD .com adalah yang paling banyak digunakan, namun jumlah *phising*–nya lebih dari setengah jumlah *legitimate*–nya. Untuk

TLD seperti .app, .io, .co, .ru yang memiliki rasio *phishing*–nya lebih tinggi dari pada *legitimate*–nya menunjukkan bahwa TLD tersebut sangatlah berisiko.

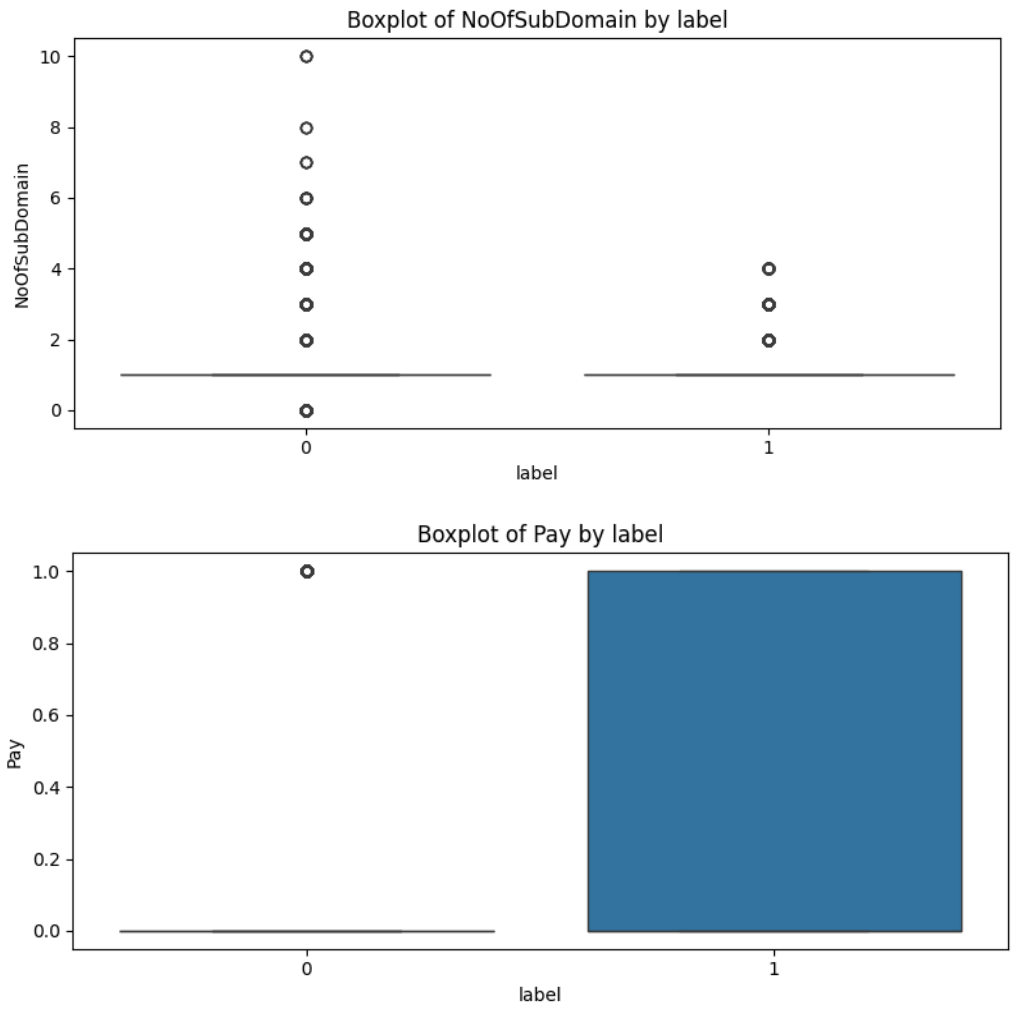


Gambar 3.3 Top 10 Frekuensi TLD berdasarkan Label

3.3 *Boxplot*

Boxplot digunakan untuk melihat persebaran data dan *outlier*. Pada visualisasi ini akan dilakukan pada dua fitur yang dengan *outlier* terbanyak, yaitu No Of Sub Domain dan Pay. Pada *boxplot* NoOfSubDomain by Label, tampak bahwa label 0 memiliki distribusi yang bervariasi, ditunjukkan oleh banyaknya nilai *outlier*, terutama pada rentang 2 hingga 10 subdomain. Ini menggambarkan bahwa data pada label 0 seringkali memiliki jumlah subdomain yang lebih banyak dan lebih beragam. Sebaliknya, label 1 memperlihatkan sebaran yang lebih sempit, dengan *outlier* hanya pada kisaran 2 hingga 4 subdomain, sehingga variasinya lebih kecil. Pola ini menunjukkan bahwa fitur NoOfSubDomain dapat membedakan kedua label secara lebih jelas, karena data dengan label 0 cenderung memiliki subdomain lebih banyak dibandingkan label 1.

Sementara itu, pada *boxplot* Pay by Label, terlihat bahwa data dengan label 0 memiliki nilai Pay yang hampir seluruhnya berada pada angka 0, kecuali satu titik *outlier* yang bernilai 1.0. Hal ini menunjukkan bahwa nilai Pay pada label 0 sangat tidak bervariasi dan cenderung konstan. Sedangkan pada label 1, nilai Pay memiliki rentang yang lebih luas, yaitu dimulai dari 0 hingga mendekati 1, yang menandakan bahwa fitur ini lebih beragam pada label tersebut. Perbedaan pola distribusi ini menunjukkan bahwa fitur Pay dapat berpotensi menjadi pembeda antara label 0 dan label 1, karena label 1 memiliki variasi yang lebih tinggi dibandingkan label 0.



Gambar 3.4 Visualisasi *boxplot* berdasarkan Label

4 *Machine Learning Inference*

4.1 K – *Nearest Neighbour* (KNN)

Model KNN dipilih untuk tugas ini karena algoritma ini bekerja berdasarkan prinsip kesamaan fitur (*feature similarity*). Karena jumlah fitur yang memiliki korelasi tinggi dengan label cukup banyak, data dengan banyak kesamaan karakteristik akan cenderung berkelompok sesuai dengan kelasnya. Kondisi ini ideal untuk model KNN dalam klasifikasi data. Dengan cara kerjanya yang mencari kesamaan, salah satu penerapan model KNN adalah memberikan rekomendasi berdasarkan data historis dari konsumer.

Fitur dipilih menggunakan nilai korelasi fitur terhadap label untuk fitur numerik dan Cramer's V untuk fitur kategorikal. Hasil tersebut diambil 10 tertingginya. Sehingga didapat `'DomainTitleMatchScore', 'URLTitleMatchScore', 'SpacialCharRatioInURL'` untuk fitur numerik dan `'HasSocialNet', 'HasCopyrightInfo', 'HasDescription',`

`'IsHTTPS', 'HasSubmitButton', 'IsResponsive', 'TLD'` untuk fitur kategorikal. Setelah dipilih, fitur-fitur tersebut direkayasa agar siap digunakan oleh model. *StandardScaler* digunakan untuk menormalisasi nilai dari fitur numerik, dan *One-Hot Encoding* digunakan untuk menormalisasi nilai fitur kategorikal menjadi biner. Dengan demikian, model dapat memproses nilai data secara matematis.

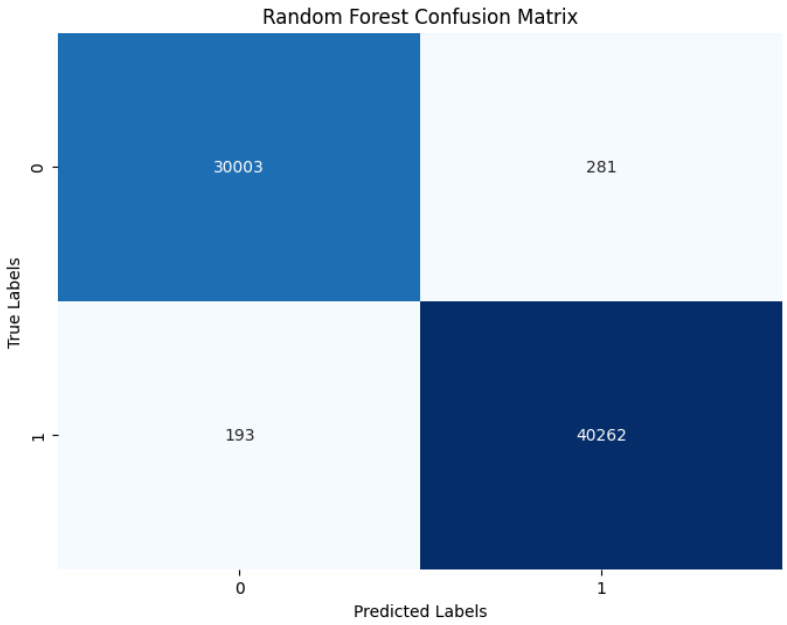
Algoritma dari KNN yang dibangun adalah algoritma KNN yang menggunakan metrik *Euclidean.* Untuk parameternya, dipilih nilai K sebesar 200. Nilai ini cukup besar untuk membuat *decision boundary* lebih halus, sehingga model lebih *robust* terhadap *noise* dan *outlier*. Pada model KNN, didapatkan hasil pendeteksian yang baik karena terdapat sedikit jumlah *false negative* (FN) dan *false positive* (FP) yang berarti model jarang salah mengklasifikasikan *phishing* sebagai *legitimate* ataupun sebaliknya. Selain itu, model ini juga memiliki nilai F1 – *score euclidean* sebesar 0.9874. Nilai ini sudah cukup baik untuk performa klasifikasi.

4.2 *Random Forest*

Random Forest dipilih sebagai pembanding model KNN. Cara kerja dari model ini adalah dengan membuat banyak *decision tree* untuk mengurangi variansi. Salah satu penerapan Random Forest adalah pada *spam filter*. Karena setiap email atau URL spam memiliki karakteristik unik (seperti penggunaan simbol aneh, domain yang panjang, atau kata-kata mendesak), Random Forest bekerja dengan membangun serangkaian aturan keputusan (decision rules) berdasarkan karakteristik tersebut untuk mengklasifikasikan apakah pesan tersebut Spam atau Legitimate.

Pada model random forest, didapatkan hasil pendeteksian yang baik karena terdapat sedikit jumlah *false negative* (FN) dan *false positive* (FP) yang berarti model jarang salah mengklasifikasikan *phishing* sebagai *legitimate* ataupun sebaliknya. Selain itu, model ini juga memiliki nilai F1 – *score macro* sebesar 0.9931. Hasil ini lebih baik dibandingkan performa KNN. Ini membuktikan bahwa metode *random forest* lebih baik dalam menangani hubungan yang kompleks pada dataset PhiUSIIL daripada KNN.

Selain itu, pada *random forest* terdapat fungsi *feature importance* yang menjadikan model ini lebih dapat dipahami. Dengan menjalankan fungsi ini, didapat fitur-fitur yang menjadi pemeran utama dalam klasifikasi URL *Phishing* yaitu `HasCopyrightInfo`, `HasSocialNet`, dan `HasDescription`. Hal ini menunjukkan bahwa URL *Phishing* pada umumnya tidak memiliki informasi hak cipta, media sosial, dan deskripsi yang jelas.



4.3 Kesimpulan dan *Lesson – Learned*

Berdasarkan hasil model yang sudah dicapai, didapatkan bahwa kedua model sudah berhasil memprediksi Label URL *Phishing* dengan tingkat akurasi diatas 0.98 untuk kedua model. Beberapa hal yang dapat mempengaruhi akurasi dari model tersebut, antara lain dari kualitas data itu sendiri, di mana sebelum kami mengolah data sudah relatif bersih, tidak ada *missing values*, dan minim *outliers*. Selain itu, beberapa teknik data *preparation* seperti *feature encoding* dan juga *feature engineering* membantu men – *enhance* model menjadi lebih baik. Model yang sudah dikembangkan dapat dilakukan pengetesan lebih lanjut untuk memastikan bahwa memang model dapat menangani data diluar *dataset* dengan hasil yang baik.

Pembagian Tugas

Anggota	Deskripsi
Clement Nathanael Lim / 18222032	Mengerjakan bagian EDA, *data cleaning*, *data preparation*, dan *inference*.
Batrisyia Zahrani Ananto / 23525042	Mengerjakan bagian visualisasi data, *confusion matrix*.
Najmi I T Kertasafari / 23324005	Mengerjakan bagian *inference*.

***Link* video presentasi:** https://youtu.be/urONwT56WZA