MIND (Multimedia Artificial Intelligent Networking Database) Journal

MIND Journal | ISSN (p): 2528-0015 | ISSN (e): 2528-0902 | Vol. 7 | No. 2| Halaman 111 - 123
DOI : https://doi.org/10.26760/mindjournal.v7i2.111-123 Desember 2022

Penerapan *Data Standardization* dan Multilayer Perceptron pada Identifikasi *Website Phishing*

YUSUP MIFTAHUDDIN, MOHAMAD MUQIIT FATURRAHMAN

Program Studi Informatika, Institut Teknologi Nasional Bandung
Email: yusufm@itenas.ac.id

Received 1 Agustus 2022 **|** *Revised* 30 Agustus 2022 **|** *Accepted* 8 September 2022

ABSTRAK

Website phishing adalah salah satu masalah utama dalam bidang keamanan website. Website phishing dibuat oleh orang yang tidak bertanggungjawab untuk mengambil informasi pribadi seseorang contohnya seorang hacker atau cracker. Teknik umum yang digunakan pada phishing yaitu manipulasi Uniform Resource Locator (URL), pemalsuan halaman situs web, dan pop up window. Pada tahun 2019, APWG (Anti-Phishing Working Group) mendeteksi kasus phishing sebanyak 162.155 kasus di dunia. Pada penelitian ini, melakukan eksperimen dengan menerapkan metode Data Standardization dan Multilayer Perceptron (MLP) untuk mendeteksi website phishing. Eksperimen dilakukan menggunakan 2 model yaitu model A dan model B. Untuk melihat performa dari model MLP yang dihasilkan dapat dilihat menggunakan tingkat accuracy, recall, precision, f1-score dan specificity. Selain itu juga dapat dilihat menggunakan confusion matrix untuk melihat kinerja pada model MLP. Pada penelitian ini menghasilkan bahwa model B merupakan model terbaik dengan mendapatkan tingkat accuracy 95.7% , recall 97.3%, precision 94.0%, f1-score 95.6% dan specificity 97.3%.

***Kata kunci**: multilayer perceptron, data standardization, website phishing*

ABSTRACT

Phishing websites are one of the main problems in the field of website security. Phishing websites are created by people who are not responsible for taking someone's personal information. Common techniques used in phishing are Uniform Resource Locator (URL) manipulation, website page spoofing, and pop up windows. In 2019, APWG (Anti-Phishing Working Group) detected 162,155 cases of phishing in the world. In this study, conducting experiments by using Data Standardization and Multilayer Perceptron (MLP) methods to detect phishing websites. Experiments were carried out using 2 models, namely model A and B. To see the performance of MLP model, it can be seen using score of accuracy, recall, precision, f1-score and specificity. In addition, it can also be seen using the confusion matrix to see the performance of the MLP model. This research shows that model B is the best model with 95.7% accuracy, 97.3% recall, 94.0% precision, 95.6% f1-score and 97.3% specificity.

***Keywords**: multilayer perceptron, data standardization, website phishing*

1. PENDAHULUAN

Website phishing merupakan salah satu masalah utama dalam keamanan *website,* yang dibuat oleh orang tidak bertanggungjawab untuk mengambil informasi pribadi dari seseorang seperti nama, alamat, kata sandi, nomor kartu kredit, dan lain-lain **(Ali, 2017)**. Selain itu, pelaku *website phishing* dapat berpura-pura menjadi organisasi yang dapat dipercaya sehingga orang-orang dapat mengunjungi *website* dengan mengirmkan *Uniform Resource Locator (URL)* tersebut agar mendapatkan informasi pribadi **(Abusaimeh & Alshareef, 2021)**.

Halaman *website phishing* dapat diidentifikasi menggunakan *Uniform Resource Locator (URL)*. Penyerang menggunakan *URL* yang berisi *path* dan komponen file untuk tujuan *phishing* **(Tripathy dkk., 2021)**. Terdapat beberapa teknik yang dilakukan oleh penyerang untuk melancarkan serangan *website phishing* yaitu manipulasi tautan (*URL*), pemalsuan halaman situs web, dan *pop up* **(Kalaharsha & Mehtre, 2021)**. Deteksi *website phishing* dapat dilakukan dengan memperbarui daftar hitam (*blacklist*) *URL* yang dilarang, Protokol Internet (*IP Address*), hingga basis data antivirus (Mahajan & Siddavatam, 2018)**.** APWG (*Anti-Phishing Working Group*) pada 2019 mendeteksi sebanyak 162.155 sebagai *website phishing* pada kuartal keempat tahun 2019. Studi statistik dari Kaspersky Lab pada 2019, 19,8% pengguna komputer ditargetkan untuk diserang oleh situs *malware*.

Pada deteksi *website phishing* dapat digunakan metode *machine learning* ataupun *deep learning*. Fitur untuk deteksi *website phishing* disusun dalam beberapa format yang dapat berupa *URL-based feature*, *content-based feature* dan *external service feature* **(Hannousse & Yahiouche, 2021).** Fitur-fitur tersebut memiliki rentang nilai dan satuan unit yang berbeda-beda, maka diperlukan proses *feature scaling*. Pada penelitian ini, memanfaatkan salah satu proses *feature scaling* yaitu *data standardization*. *Data standardization* dapat meningkatkan akurasi klasifikasi dengan melakukan standarisasi nilai karakteristik ke dalam rentang yang sempit **(Hosseinzadeh dkk., 2021).** Pada proses klasifikasi *website phishing* dapat menggunakan metode *artificial neural network (ANN)* karena memiliki potensi fenomenal untuk mecapai akurasi yang optimal dibandingkan dengan metode *machine learning* **(M dkk., 2020)**. *Multilayer perceptron* termasuk dalam metode *ANN* yang banyak digunakan karena arsitekturnya merupakan lapisan *perceptron* yang digabungkan dan membentuk arsitektur *multilayer*, hal ini memberikan kompleksitas yang diperlukan dari pemrosesan *neural network* **(Sen dkk., 2020)**.

Pada penelitian yang dilakukan oleh **(Hannousse & Yahiouche, 2021)** melakukan penelitian mengenai *benchmark* dari 5 algoritma *machine learning* yaitu *Decision Tree*, *Random Forest*, *Support Vector Machine (SVM)*, *Naïve Bayes* dan *Logistic Regression*. Selain itu juga membandingkan dengan 4 algoritma *feature selection* yaitu *Chi-Square*, *Pearson's correlation*, *Information gain* dan *relief*. Hasilnya algoritma *Random forest* memiliki tingkat akurasi tertinggi akurasi di atas 96% dengan menggunakan 4 *feature selection* yang berbeda.

Kemudian pada penelitian yang dilakukan oleh **(Al-Ahmadi & Lasloum, 2020)** mengenai deteksi *phishing* menggunakan teknik *Multilayer Perceptron* dengan pendekatan memetakan *URL* menjadi beberapa atribut tertentu yang tersedia. Selain itu, menggunakan fitur seleksi *Correlation Matrix* yang nantinya akan memilih fitur mana saja yang tidak diperlukan. Penelitian ini menggunakan beberapa metode yaitu *PDMLP*, *K-Nearest Neighbour, SVM, C4.5 Decision Tree, Random Forest* dan *Recurrent Neural Network*. Hasil pada penelitian ini mengungkapkan bahwa *PDMLP* merupakan metode terbaik dengan 96% tingkat akurasi.

Kemudian pada penelitian yang dilakukan oleh **(Abusaimeh & Alshareef, 2021)** mengenai deteksi *website phishing* dengan menggabungkan 3 algoritma sekaligus yaitu *Decision Tree*,

Random Forest, dan *Support Vector Machine*. Selain itu, pada penelitian ini juga membandingkan hasil dari metode yang diusulkan dengan masing-masing metode (*Decision Tree*, *Random Forest*, dan *Support Vector Machine*) dengan tingkat akurasi Random Forest 97,259%, Support Vector Machine sebesar 95,3597%, Decision Tree sebesar 95,8752%, dan model yang diusulkan mendapat nilai akurasi tertinggi sebesar yaitu 98,5256%.

Berdasarkan hasil penelitian di atas, maka pada penelitian ini menggunakan metode *preprocessing* yaitu *data standardization* untuk mengubah rentang nilai fitur, berbeda dengan penelitian **(Al-Ahmadi & Lasloum, 2020)** yang hanya menggunakan *multilayer perceptron* tanpa melakukan *preprocessing*.

2. METODE PENELITIAN

2.1. Data Standardization

Data scaling perlu dilakukan untuk memastikan validitas pemodelan prediktif, terutama ketika variabel atau fitur memiliki skala yang berbeda-beda. Salah satunya yaitu metode *data standardization*. *Data standardization* memusatkan angka-angka di sekitar rata-rata dan menggunakan deviasi standar satuan. Akibatnya, rata-rata atribut menjadi mendekati nol, dan distribusi yang dihasilkan memiliki simpangan baku satuan, seperti yang ditunjukan pada Persamaan (1) **(Fan dkk., 2021)**.

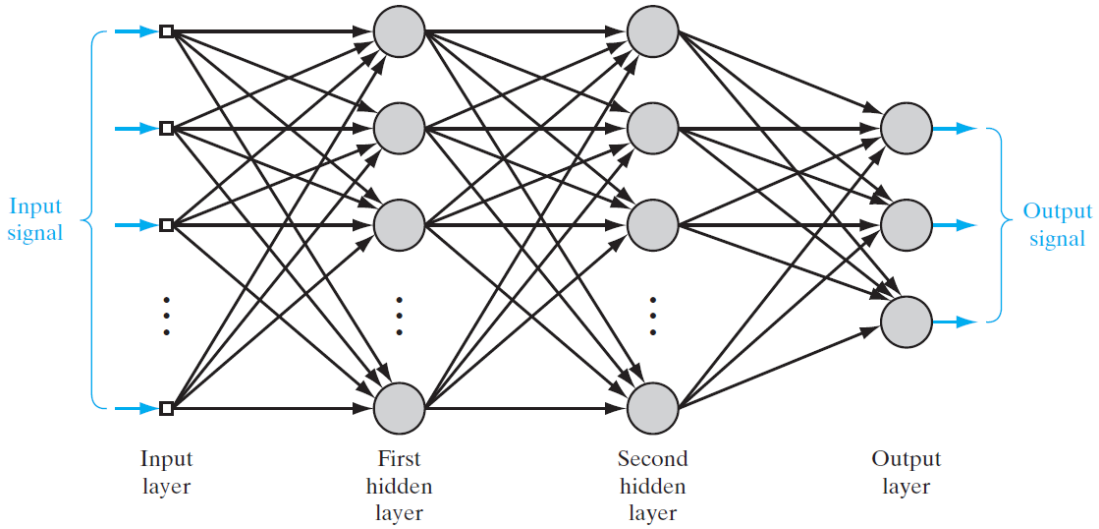
$$x' = \frac{x - \mu}{\sigma} \quad (1)$$

Di mana :
$\boldsymbol{x}$ = input data
$\boldsymbol{\mu}$ = rata-rata data sampel
$\boldsymbol{\sigma}$ = standar deviasi data sampel

2.2. Multilayer Perceptron

Multilayer Perceptron (MLP) adalah salah satu metode di bidang *Artifical Neural Network*. Arsitektur *MLP* dapat terdiri dari satu atau lebih dari *hidden layer*. Proses *training* pada *MLP* terdiri dari dua bagian utama yaitu *feed forward* dan *backward*. *Feed forward* digunakan untuk menghitung *output* dari setiap *hidden layer* berdasarkan nilai masukan, nilai *weight* saat ini, dan berdasarkan fungsi aktivasi yang digunakan. Sedangkan backward digunakan untuk mengupdate nilai bobot mengikuti nilai error yang telah ditentukan. Proses pelatihan akan berhenti ketika nilai *Mean Square Error* dapat diterima **(Hartono dkk., 2020).**

Pada Gambar 1 arsitektur *MLP* terdapat *input layer, hidden layer* dan *output*. Di setiap *hidden layer* dan output terdapat fungsi aktivasi. Hasil penjumlahan dari perkalian bobot neuron dan sinaptik akan ditransformasikan dengan menggunakan fungsi aktivasi yang digunakan untuk menentukan keluaran pada masing-masing neuron. Contoh fungsi aktivasi yang biasa digunakan adalah *sigmoid* dan *ReLU* **(Vanneschi & Castelli, 2018)**.



Gambar 1. Arsitektur Multilayer Perceptron

Pada *multilayer perceptron*, bobot (ω) setiap koneksi antar neuron menunjukkan seberapa kuat hubungan antara dua neuron. Selain itu, *hidden* dan *output* neuron memiliki bias (β), yang merupakan ambang batas untuk mengkondisikan output *neuron* untuk mengubah prediksi. Proses pembelajaran *multilayer perceptron* memerlukan penentuan kumpulan bobot dan bias terbaik **(Rojas dkk., 2022)**.

2.3. Evaluasi Model

Pada penelitian ini, dilakukan percobaan dengan 2 teknik yaitu klasifikasi *multilayer perceptron* dengan *data standardization* dan tanpa *data standardization*. Hasil penelitian ini dapat disajikan berupa *confusion matrix* yaitu terdapat *accuracy*, *precision*, *recall*, dan *f1-score*. *Accuracy* secara umum, mewakili proporsi prediksi yang tepat dengan jumlah total data yang diperiksa. *Precision* merupakan jumlah prediksi positif yang benar (TP) dibagi dengan jumlah total prediksi positif (TP + FP). *Recall* menghasilkan Jumlah prediksi positif yang benar (TP) dibagi dengan jumlah total prediksi positif (P). *F1-score* nilai *harmonic mean* (rata-rata harmonik) dari *precision* dan *recall*. Sedangkan *specificity* dilakukan dengan membagi jumlah total negative (TN) dengan jumlah hasil negatif (N). Persamaan untuk *accuracy*, *precision*, *recall* dan *f1-score* dapat dilihat pada Persamaan (2), (3), (4), (5) dan (6) **(Vujović, 2021)**.

$$accuracy = \frac{tp+tn}{tp+fp+fn+tn} \quad (2)$$

$$precision = \frac{tp}{tp+fp} \quad (3)$$

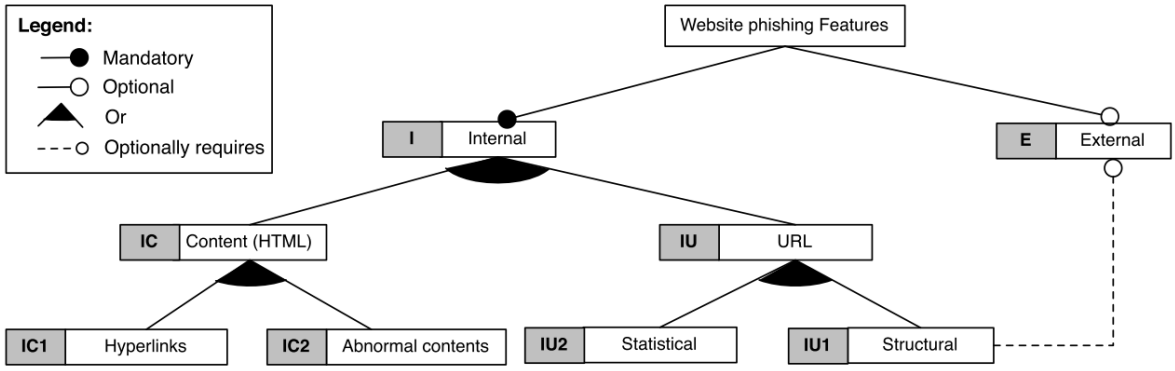
$$recall = \frac{tp}{tp+fn} \quad (4)$$

$$f1-score = \frac{2 \times precision \times recall}{precision + recall} \quad (5)$$

$$specificity = \frac{tn}{tn+fp} \quad (6)$$

2.4. Deskripsi Datasets

Datasets yang digunakan mempunyai 11430 baris dan 87 kolom fitur dengan 2 label yaitu *legitimate* dan *phishing*. *Datasets* ini diambil melalui penelitian yang dilakukan oleh **(Hannousse & Yahiouche, 2021)**.



Gambar 2. Diagram Fitur Datasets

Pada *datasets* ini ditunjukkan bahwa pada Gambar 3 terdapat 2 pembagian kategori utama yaitu fitur*Internal* dan *External*. Pada fitur tersebut dibagi menjadi 3 sub-bagian penjelasan fitur yaitu *URL-based features*, *Content-based features* dan *External-based features* **(Hannousse & Yahiouche, 2021)**.

2.4.1. URL-based features

URL-based features adalah termasuk fitur internal dan dilambangkan dengan (IU). Fitur ini hanya cukup mempelajari teks yang ada pada *URL* yang dibagi menjadi dua bagian yaitu *Structural-based features* (IU1) yang memperhatikan keberadaan, posisi, dan sifat elemen dasar URL dan *Statistical-based features* (IU2) yang memperhatikan pada jumlah dan distribusi elemen dasar *URL*, kata-kata tertentu, dan karakter dalam konten *URL.*

2.4.2. Content-based features

Fitur berbasis konten (IC) diperoleh dengan memuat dan mengevaluasi konten *HTML* halaman web *URL*. Fitur ini diklasifikasikan ke dalam dua kategori: *hyperlink* dan *aberrant content*. Fitur *hyperlink* (IC1) menangani kuantitas, status, dan karakter hyperlink (internal/eksternal) yang digunakan dalam tag HTML. Fitur *abnormal content* (IC2) memfokuskan pada identifikasi materi atau *script* yang mencurigakan yang melakukan tindakan mencurigakan.

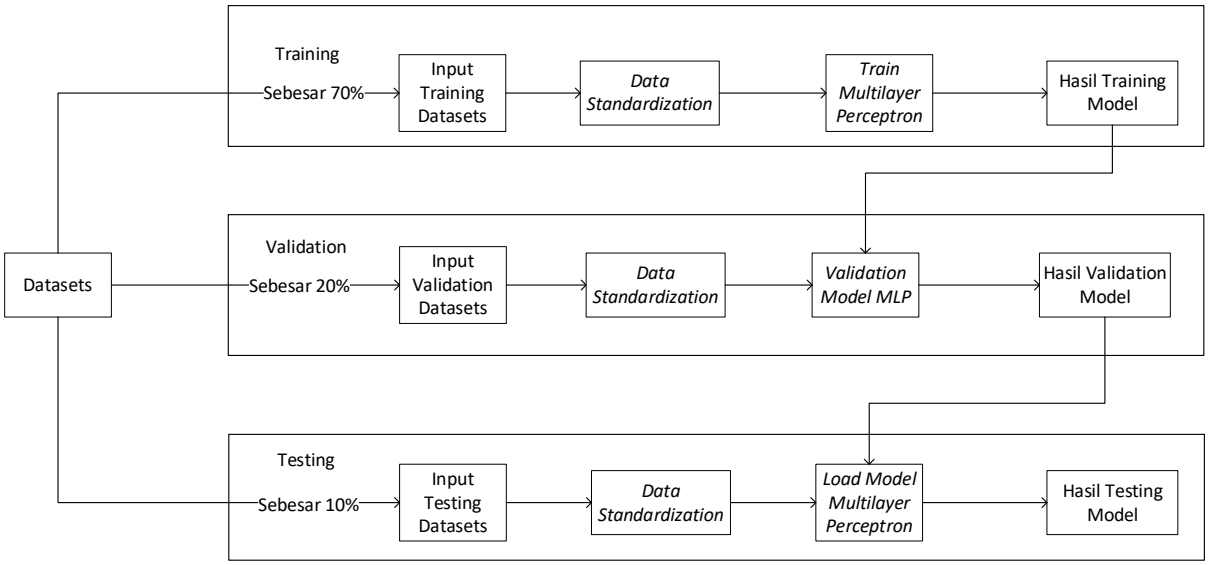
2.4.3. External-based features

External-based features dilambangkan dengan (E) diperoleh dengan menggunakan referensi dari layanan pihak ketiga dan mesin pencari *WHOIS*, *Alexa*, *Openpagerank* dan *Google*.

2.5. Perancangan Umum

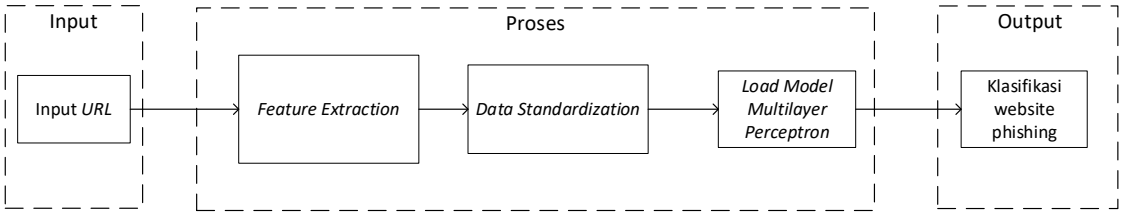
Kehati-hatian dalam penulisan sumber dan Daftar Rujukan merupakan satu keharusan agar penulis dapat terhindar dari plagiarisme.

2.5.1. Blok Diagram



Gambar 3. Blok Diagram *Training, Validation* dan *Testing*

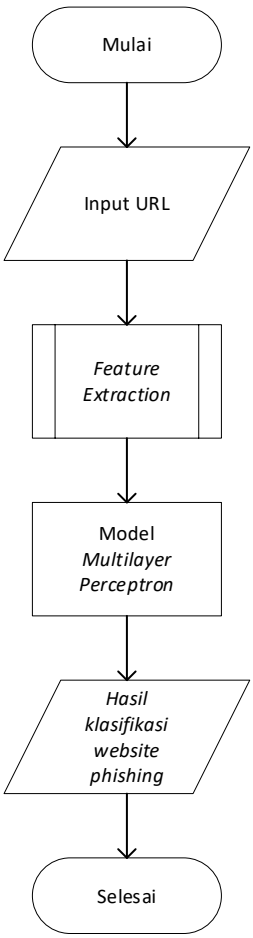
Sebelum melakukan proses *training, validation* dan *testing*, *datasets* harus dibagi menjadi 70% untuk *training,* 20% untuk *validation* dan 10% untuk *testing* seperti yang ditunjukan pada Gambar 4. Pada setiap tahapan dilakukan proses *data standardization* untuk mengubah rentang nilai tertentu. Pada tahap *training*, dilakukan beberapa kali agar mendapatkan hasil model terbaik yang nantinya merupakan *output* dari proses *training*. Model pada saat *training* dievaluasi menggunakan *datasets validation* untuk melihat performa model. Model dari hasil *training* akan disimpan dan nantinya akan dipanggil ketika melakukan *testing*. Pada tahapan *testing*, *datasets* sebesar 10% tadi langsung diproses menggunakan model yang disimpan dan akan muncul evaluasi dari performa model yang telah dibuat dan hasil klasifikasi *website phishing*.



Gambar 4. Blok Diagram Sistem

Pada sistem, proses yang dilakukan pertama kalo merupakan *input URL* berdasarkan Gambar 4. Dari *URL* tersebut akan diekstraksi ke dalam 87 fitur pada *datasets* dan menghasilkan *feature vector* yang berupa masukan untuk model. Sebelum proses klasifikasi oleh model, dilakukan tahapan *data standardization* untuk mengubah rentang nilai. Setelah itu, model akan mempelajari data tersebut dan menghasilkan klasifikasi *website phishing* atau *legitimate*.

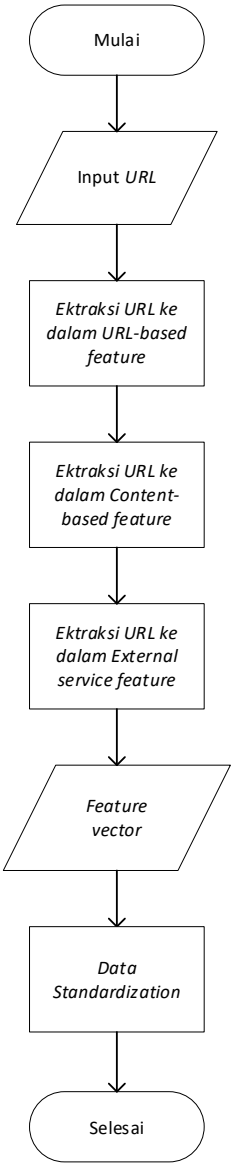
2.5.2. Flowchart Umum



Gambar 5. *Flowchart* Umum Sistem

Pada Gambar 4 merupakan *flowchart* umum dari sistem. Tahapan selanjutnya dijelaskan mengenai *flowchart* tersebut sebagai berikut :

1. Pengguna memasukkan *URL* dari *website* yang diinginkan.
2. Dari *URL* tersebut, dilakukan proses *feature extraction* agar *URL* tersebut dapat dipetakan menjadi beberapa fitur yang diperlukan oleh model.
3. Kemudian fitur-fitur tersebut dimasukkan pada model *Multilayer Perceptron* yang telah dibuat pada proses *training*.
4. Maka, akan muncul hasil klasifikasi apakah *website* tersebut termasuk *phishing* atau *legitimate*.



Gambar 6. *Flowchart* Subproses *Feature Extraction*

Pada Gambar 5 terdapat *flowchart* dari sub proses *feature extraction*. Berikut merupakan penjelasan dari *flowchart* tersebut :

1. Menerima masukan dari *input URL*.
2. *URL* tersebut akan dipetakkan menjadi beberapa fitur yang ada pada *URL-based feature*.
3. *URL* tersebut akan dipetakkan menjadi beberapa fitur yang ada pada *Content-based feature*.
4. *URL* tersebut akan dipetakkan menjadi beberapa fitur yang ada pada *External service feature* dengan bantuan dari layanan *WHOIS*, *Google*, *Alexa,*dan *Openpagerank* .
5. Kemudian hasil dari ekstraksi fitur tersebut menghasilkan *feature vector* atau daftar fitur yang dibutuhkan untuk model.
6. *Feature vector* tersebut perlu dilakukan *data standardization* agar data *input* dapat distandardisasikan karena memiliki skala atau unit yang berbeda-beda tiap fiturnya yang dibutuhkan untuk model.

3. HASIL DAN PEMBAHASAN

3.1. Penggunaan Datasets

Datasets yang digunakan pada penelitian ini berbentuk *file csv* dengan jumlah 11430 baris dengan 87 kolom fitur. Berdasarkan **(Nguyen dkk., 2021)** pembagian *datasets* 70% untuk *training* dan 30% untuk *testing* merupakan ratio yang ideal. Dengan menggunakan *datasets validation* model akan mempelajari *datasets* yang belum pernah dilihat sebelumnya sehingga akan mengubah *weight* setiap *epoch* hingga *loss* menurun. Maka dari itu, *datasets testing* akan dibagi kembali menjadi 20% untuk *validation* dan 10% untuk *testing* agar model dapat belajar dengan baik pada proses *training*. Sehingga *datasets* dilakukan proses pemisahan untuk proses *training, validation* dan *testing* dengan masing-masing pembagian yaitu 70%, 20% dan 10% seperti pada Tabel 1.

Tabel 1. Pembagian *datasets training*, *validation* dan *testing*

	Training	***Validation***	***Testing***
Phishing	3975	1170	570
Legitimate	4025	1116	574
Jumlah	8000	2286	1144

Pada tahapan *training* terdapat 8000 data dengan jumlah data *phishing* 3974 dan *legitimate* 4026. Tahapan *validation* terdapat 1715 dengan jumlah data *phishing* 866 dan *legitimate* 849. Sedangkan tahapan *testing* terdapat 1715 dengan jumlah data *phishing* 875 dan *legitimate* 840.

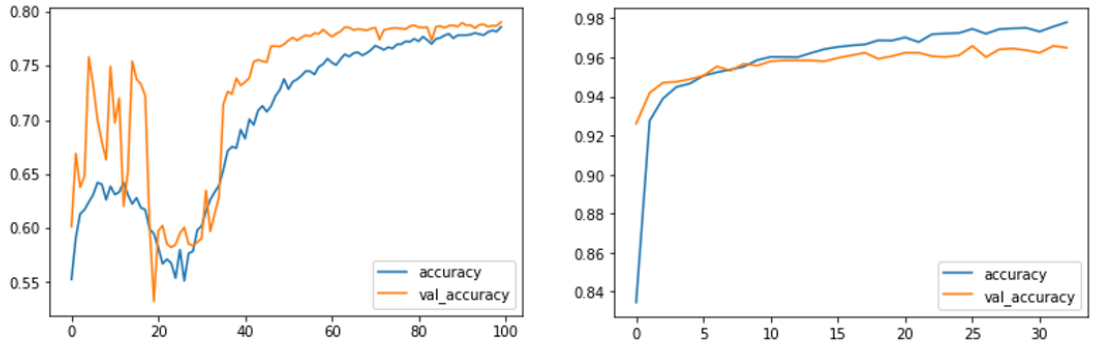
3.2. Training Model

Pada tahapan *training* pada model, dapat mengatur beberapa *hyperparameter* seperti yang ditunjukkan pada Tabel 2.

Tabel 2. Hyperparameter

Hyperparameter	***Training***
Batch Size	32
Hidden Layer	2
Hidden Unit	256
Dropout	0.2
Optimizer	*Adam*
Learning Rate	0.0001

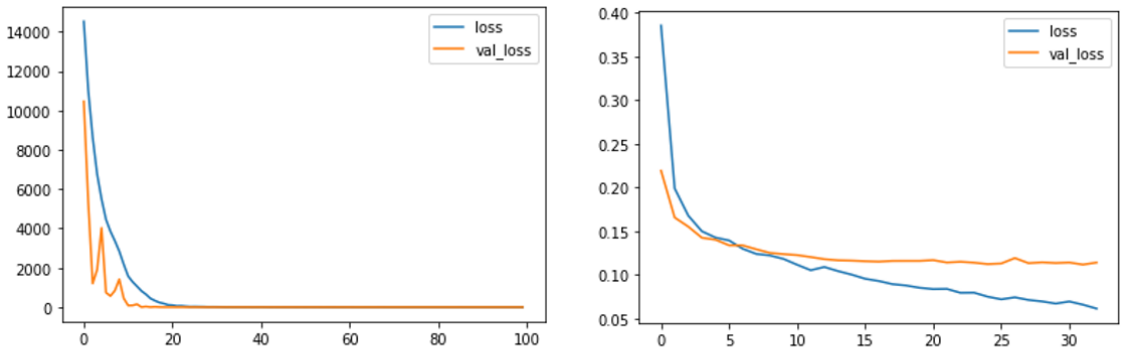
Eksperimen pada tahapan ini menggunakan *data standardization* dan tanpa *data standardization* dengan menggunakan *hyperparameter* yang sama yaitu mengatur *batch size*, *hidden layer*, *hidden unit*, *dropout*, *optimizer*, dan *learning rate* seperti yang ditunjukkan pada Tabel 2. Eksperimen dilakukan dengan 2 model yang berbeda yaitu model A dan model B seperti pada Gambar 7.



a) Non Data Standardization b) Data Standardization

Gambar 7. *Accuracy* Model Training

Pada Gambar 7 menampilkan akurasi dari model A tanpa *data standardization* dan model B dengan *data standardization* pada saat proses *training*. Terlihat konsistensi akurasi dan tingkat akurasi dari *training* berbeda cukup jauh sekitar 78,5% untuk model A dan 97,8% untuk model B. Kedua model dilakukan optimasi menggunakan fungsi *early stopping* karena untuk melihat apakah model terjadi peningkatan *validation loss* pada saat proses *training* atau tidak. Jika tidak mengalami peningkatan *validation loss* maka proses pelatihan akan dihentikan. Terlihat pada model B berhenti pada *epoch* ke 32 sedangkan untuk model A berhenti pada *epoch* ke 100.



a) Non Data Standardization b) Data Standardization

Gambar 8. *Loss* Model Training

Pada Gambar 8 menampilkan tingkat *loss* dari model A tanpa *data standardization* dan model B dengan *data standardization* pada saat proses *training*. Terlihat pada model A memiliki *loss* yang sangat tinggi pada saat awal proses *training* yaitu hingga di atas 14000. Ini menunjukkan semakin tinggi *loss* maka semakin buruk performa dari model tersebut. Pada model A mempunyai tingkat *loss* terbaik yaitu 0,447 sedangkan pada model B mempunyai tingkat *loss* terbaik yaitu 0,06.

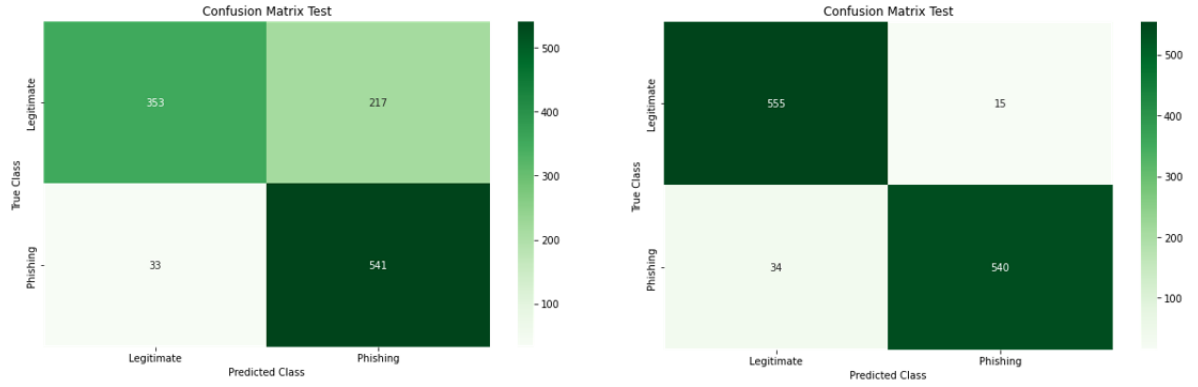
3.3. Testing Model

Pada tahapan *testing* atau pengujian model menggunakan 10% dari *datasets* yang ada seperti pada Tabel 1. Pengujian ini menghasilkan *confusion matrix* untuk mengukur performa model. Selain itu juga terdapat tingkat *accuracy*, *recall*, *precision*, *f1-score* dan *specificity*. Performa model dapat dilihat pada Tabel 3.

Tabel 3. Performa Pengujian Model

Model	***Accuracy***	***Precision***	***Recall***	***F1-Score***	***Specificity***
Model A	79.0%	71.4%	95.1%	81.6%	63.6%
Model B	95.7%	97.3%	94.0%	95.6%	97.3%

Pada Gambar 9 ditunjukkan *confusion matrix* pada kedua model. Terlihat pada model A terdapat kesalahan prediksi sebesar 217 data yang seharusnya *legitimate* tetapi dideteksi sebagai *phishing*. Tetapi, pada model B hanya terdapat prediksi sebesar 15 data yang seharusnya *legitimate* tetapi dideteksi sebagai *phishing*. Ini menunjukkan bahwa *data standardization* mempengaruhi performa model *MLP*.



a) Non Data Standardization b) Data Standardization

Gambar 9. *Confusion Matrix*

4. KESIMPULAN

Pada penelitian yang telah dilakukan, telah menerapkan model *MLP* untuk deteksi *website phishing*. Model *MLP* diatur *hyperparameter* berdasarkan *batch size*, *hidden layer, hidden unit, dropout*, *optimizer* mengunakan *Adam* dan *learning rate* 0.0001. Pengujian model dilakukan dengan jumlah data *training* 70% yaitu sebesar 8000 data, data *validation* 20% yaitu sebesar 2286 data dan *data testing* 15% yaitu sebesar 1144. Selain itu, pengujian dibagi menjadi 2 model *MLP* yaitu model A dan model B. Berdasarkan hasil pengujian pada penelitian ini, mendapatkan tingkat *accuracy* terbaik yaitu 95,7% dan *loss* terbaik yaitu 0,06 untuk model B yang menggunakan *data standardization*. Kedua model *MLP* yang dilakukan pengujian menunjukkan bahwa penggunaan *data standardization* dapat meningkatkan tingkat *accuracy* pada model *MLP*.

DAFTAR RUJUKAN

Abusaimeh, H., & Alshareef, Y. (2021). Detecting the Phishing Website with the Highest Accuracy. *TEM Journal*, *10*(2), 947–953. https://doi.org/10.18421/TEM102-58

Al-Ahmadi, S., & Lasloum, T. (2020). PDMLP: Phishing Detection using Multilayer Perceptron. *International Journal of Network Security & Its Applications*, *12*(3), 59–72. https://doi.org/10.5121/ijnsa.2020.12304

Ali, W. (2017). Phishing Website Detection based on Supervised Machine Learning with Wrapper Features Selection. *International Journal of Advanced Computer Science and Applications*, *8*(9), 72–78. https://doi.org/10.14569/ijacsa.2017.080910

Fan, C., Chen, M., Wang, X., Wang, J., & Huang, B. (2021). A Review on Data Preprocessing Techniques Toward Efficient and Reliable Knowledge Discovery From Building Operational Data. *Frontiers in Energy Research*, *9*(March), 1–17. https://doi.org/10.3389/fenrg.2021.652801

Hannousse, A., & Yahiouche, S. (2021). Towards benchmark datasets for machine learning based website phishing detection: An experimental study. *Engineering Applications of Artificial Intelligence*, *104*, 1–21. https://doi.org/10.1016/j.engappai.2021.104347

Hartono, Sadikin, M., Sari, D. M., Anzelina, N., Lestari, S., & Dari, W. (2020). Implementation of Artifical Neural Networks with Multilayer Perceptron for Analysis of Acceptance of Permanent Lecturers. *Jurnal Mantik*, *4*(4), 1389–1396.

Hosseinzadeh, M., Ahmed, O. H., Ghafour, M. Y., Safara, F., hama, H. kamaran, Ali, S., Vo, B., & Chiang, H. Sen. (2021). A multiple multilayer perceptron neural network with an adaptive learning algorithm for thyroid disease diagnosis in the internet of medical things. *Journal of Supercomputing*, *77*(4), 3616–3637. https://doi.org/10.1007/s11227-020-03404-w

Kalaharsha, P., & Mehtre, B. M. (2021). *Detecting Phishing Sites -- An Overview*. 1–13. http://arxiv.org/abs/2103.12739

M, S., R V, J., Blessy Ida Gla, & Priyadharshini. (2020). A REVIEW ON PHISHING WEBSITE DETECTION USING MACHINE LEARNING. *Journal of Critical Reviews*, *7*(19), 4847–4853.

Mahajan, R., & Siddavatam, I. (2018). Phishing Website Detection using Machine Learning Algorithms. *International Journal of Computer Applications*, *181*(23), 45–47. https://doi.org/10.5120/ijca2018918026

Nguyen, Q. H., Ly, H. B., Ho, L. S., Al-Ansari, N., Van Le, H., Tran, V. Q., Prakash, I., & Pham, B. T. (2021). Influence of data splitting on performance of machine learning models in prediction of shear strength of soil. *Mathematical Problems in Engineering*, *2021*.

https://doi.org/10.1155/2021/4832864

Rojas, M. G., Olivera, A. C., & Vidal, P. J. (2022). Optimising Multilayer Perceptron weights and biases through a Cellular Genetic Algorithm for medical data classification. *Array*, *14*(April), 100173. https://doi.org/10.1016/j.array.2022.100173

Sen, S., Sugiarto, D., & Rochman, A. (2020). Komparasi Metode Multilayer Perceptron (MLP) dan Long Short Term Memory (LSTM) dalam Peramalan Harga Beras. *Ultimatics*, *XII*(1), 35.

Tripathy, A. K., Sarkar, M., Sahoo, J. P., Li, K.-C., & Chinara, S. (2021). Advances in and Machine Computing Distributed Learning. In *Lecture Notes in Networks and Systems* (Vol. 127). Springer International Publishing. https://doi.org/10.1007/978-981-15-4218-3_15

Vanneschi, L., & Castelli, M. (2018). Multilayer perceptrons. In *Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics* (Vols. 1–3). https://doi.org/10.1016/B978-0-12-809633-8.20339-7

Vujović, Ž. (2021). Classification Model Evaluation Metrics. *International Journal of Advanced Computer Science and Applications*, *12*(6), 599–606. https://doi.org/10.14569/IJACSA.2021.0120670