Volume 4, No.1 2023

# JISE

Journal of Internet and Software Engineering

ISSN 2797-9016

https://ugm.id/jise

The journal published by
Department of Electrical Engineering and Informatics
Vocational College, Universitas Gadjah Mada

JISE
Journal of Internet and Software Engineering
Volume 4, No.1 2023

# EDITORIAL TEAM

## Journal of Internet and Software Engineering (JISE)



https://ugm.id/jise

The journal published by
Department of Electrical Engineering and Informatics
Vocational College, Universitas Gadjah Mada
Sekip unit III, Caturtunggal, Terban,
Kec. Gondokusuman, Kab. Sleman, D.I. Yogyakarta 55281

JISE
Journal of Internet and Software Engineering
Volume 4, No.1 2023



https://ugm.id/jise

The journal published by
Department of Electrical Engineering and Informatics
Vocational College, Universitas Gadjah Mada

# Perancangan *Federated Learning* Berbasis *Homomorphic Encryption* untuk Perangkat *Internet of Things*

Yuris Mulya Saputra[1], Ganjar Alfian[1,*], Muhammad Qois Huzyan Octava[1]
[1]Departemen Teknik Elektro dan Informatika, Sekolah Vokasi, Universitas Gadjah Mada;
ym.saputra@ugm.ac.id
qoisoctava@mail.ugm.ac.id
[*]Korespondensi: ganjar.alfian@ugm.ac.id;

***Abstract – The growth of big data market for intelligence-based Internet-of-Things (IoT) users has attracted both industry and academia. Through using local data from various IoT devices, the service provider can produce valuable information for its users via maching learning (ML) such as centralized learning with a cloud server and local learning with the IoT devices. However, due to privacy leakage risk when the IoT users send the local data to the cloud server and limited computation resources of IoT devices, federated learning (FL) can be the efficient solution to solve the above problems. FL approach is a collaborative ML in which each IoT device can first conduct the individual training process and then share the local model only to the cloud server without data sharing. In this case, this approach can not only improve the training process performance, but also protect data privacy for the IoT users. This research focuses on FL system design with privacy-awareness for IoT users. Particularly, a homomorphic encryption based-encryption method is used to encrypt data from IoT devices during the local training process of the FL as the data privacy protection fom IoT malicious attackers. From this research, we can analyze the model accuracy performance between FL without and with the above encryption method.***

***Keywords – Federated Learning, Data Privacy, Encryption, IoT, Artificial Intelligence***

**Intisari – Semakin berkembangnya pasar *big data* yang digunakan oleh pengguna khususnya *Internet of Things* (IoT) berbasis kecerdasan buatan telah menarik banyak pihak baik dari industri maupun akademisi. Melalui penggunaan data lokal dari berbagai perangkat IoT, pemberi layanan aplikasi dapat menghasilkan informasi berguna melalui pendekatan *machine learning* (ML) seperti *centralized learning* dengan menggunakan *cloud server* dan *local learning* pada perangkat IoT langsung. Namun, dengan adanya risiko bocornya privasi pengguna ketika mengirim data lokal ke *cloud server* dan sumber daya komputasi yang terbatas pada IoT, penggunaan *federated learning* (FL) dapat menjadi solusi efisien. Pendekatan FL merupakan sebuah pendekatan ML kolaboratif di mana setiap perangkat IoT dapat melakukan proses *training* secara independen dan kemudian hanya mengirimkan model *local* kepada *cloud server* tanpa melakukan *data sharing*. Secara khusus, penggunaan FL untuk layanan aplikasi pada perangkat IoT tidak hanya memperbaiki kinerja untuk proses *training*, namun juga dapat melindungi privasi data bagi penggunanya. Penelitian ini berfokus pada perancangan sistem FL dengan *privacy-awareness* yang dapat digunakan oleh para pengguna perangkat IoT. Dalam hal ini, teknik enkripsi yang berbasis *homomorphic encryption* untuk mengenkripsi data dari perangkat IoT ketika proses *training* dari FL dapat diimplementasikan sebagai bentuk perlindungan privasi pengguna IoT dari *malicious attackers*. Dari penelitian ini, dapat dianalisis perbandingan tingkat akurasi model dari berbagai pendekatan baik tanpa dan dengan teknik enkripsi tersebut.**

**Kata kunci – *Federated Learning,* Keamanan Data, Enkripsi, IoT, Kecerdasan Buatan**

## I. PENDAHULUAN

Kebutuhan yang tinggi terhadap penggunaan *big data* untuk berbagai macam aplikasi yang berbasis kecerdasan buatan (misalnya untuk layanan kesehatan, *crowdsensing*, dan aplikasi jaringan yang menggunakan pendekatan *machine learning)* saat ini menjadi topik hangat untuk revolusi teknologi internet pada masa yang akan datang [1]. Hal ini didasari oleh adanya pandemi COVID-19 di mana masyarakat harus beradaptasi untuk bekerja dari rumah, sehingga menghasilkan data internet yang sangat besar untuk proses analisis khususnya ketika masyarakat secara umum menggunakan perangkat *mobile*. Dalam hal ini, informasi data yang disimpan dari perangkat *mobile* seperti *smartphone*, *smartwatch*, dan perangkat IoT selanjutnya dapat digunakan untuk membantu pemberi layanan dalam membuat layanan aplikasi dengan tingkat akurasi yang tinggi. Berdasarkan *Allied Market Research* dalam hal pasar *big data,* perkembangan layanan *big data* secara global pada Tahun 2030 akan meningkat lebih dari 3 kali lipat dari Tahun 2020 baik dalam hal perangkat lunak, perangkat keras, dan layanan data disebabkan kebutuhan tinggi pengguna terhadap kecerdasan buatan [2].

Adanya kebutuhan yang tinggi terhadap kecerdasan buatan memberikan motivasi terhadap pemberi layanan aplikasi untuk dapat mengumpulkan data dari berbagai perangkat pengguna khususnya perangkat IoT (*embedded sensors*) untuk keperluan ekstraksi informasi yang berguna melalui pemanfaatan *machine learning* (ML). Hal ini dapat dilakukan melalui *centralized learning* di mana semua data dari pengguna IoT akan diproses di *cloud server,* serta *local learning* di mana perangkat IoT pengguna akan memproses data lokal sendiri tanpa adanya *cloud server* [3, 4]. Namun, ada dua tantangan utama ketika dua hal tersebut dilakukan. Pertama, pengguna IoT mungkin tidak ingin melakukan *data sharing* yang disebabkan oleh risiko bocornya privasi pengguna ketika mengirim data lokal ke *cloud server*. Kedua, perangkat IoT biasanya memiliki data lokal yang tidak banyak dan sumber daya komputasi yang terbatas secara inheren,

sehingga penggunaan perangkat IoT untuk proses ML akan mengakibatkan kualitas *training* yang tidak efektif dan tingkat akurasi yang rendah.

*Federated learning* telah dianggap sebagai pendekatan yang sangat efektif untuk mengatasi dua tantangan tersebut seperti yang ditampilkan. Dengan menggunakan FL, proses *training* dapat dilakukan oleh banyak perangkat IoT tanpa adanya *data sharing* yang mungkin mengandung informasi pribadi pengguna [5, 6]. Secara spesifik, setiap perangkat IoT dapat melakukan proses *training* dengan menggunakan data lokal serta sumber daya komputasinya untuk menghasilkan sebuah model *training* lokal secara independen. Kemudian, pemberi layanan aplikasi atau *cloud server* dapat meminta setiap perangkat IoT yang berpartisipasi untuk mengirimkan model lokal tersebut untuk memperbarui model global yang nantinya dapat digunakan untuk ekstraksi informasi berguna bagi seluruh pengguna IoT.

Akan tetapi penggunaan FL secara konvensional masih dapat membuat data pribadi pengguna IoT bocor (walaupun proses *training* dilakukan secara lokal) ketika model lokal dikirimkan ke *cloud server.* Untuk mengatasi hal tersebut sebelum proses *training* dilakukan, sebuah teknik enkripsi tanpa harus melakukan deskripsi terhadap data dan model lokal serta dapat melakukan operasi matematis seperti perkalian dapat diimplementasikan.

Tujuan dan kontribusi yang ingin dicapai melalui penelitian ini yaitu 1) membuat rancangan bangun sistem keamanan *big data* berbasis kecerdasan buatan melalui pendekatan *federated learning* (FL) yang sederhana dengan menggunakan perangkat IoT dari pengguna; dan 2) memberikan nilai kemudahan dalam melindungi data pengguna IoT dengan menggunakan teknik enkripsi tingkat lanjut berupa *fully homomorphic encryption* (FHE) ketika terjadi komunikasi antar perangkat IoT pada proses *training* untuk menghasilkan informasi yang berguna. Batasan masalah yang terdapat dalam penelitian ini yaitu rancang bangun FL dibatasi pada metode *logistic regression* dengan klasifikasi biner pada dataset yang digunakan.

## II. DASAR TEORI

Adanya kebutuhan yang tinggi terhadap kecerdasan buatan memberikan motivasi terhadap pemberi layanan aplikasi untuk dapat mengumpulkan data dari berbagai perangkat IoT (*embedded sensors*) untuk keperluan ekstraksi informasi yang berguna melalui pemanfaatan *machine learning* (ML). Pada umumnya, hal ini dapat dilakukan melalui *centralized learning (CL)* di mana semua data dari pengguna IoT akan diproses di *cloud server* [3, 7, 8, 9, 10]. Namun, melalui metode di atas, pengguna IoT mungkin tidak ingin melakukan *data sharing* yang disebabkan oleh risiko bocornya privasi pengguna ketika mengirim data lokal ke *cloud server*. Selain itu, perangkat IoT biasanya memiliki data lokal yang tidak banyak dan sumber daya komputasi yang terbatas secara inheren jika *local training* [4] tanpa adanya kolaborasi antar perangkat IoT dilakukan. Hal ini menyebabkan penggunaan perangkat IoT untuk proses ML akan mengakibatkan kualitas *training* yang tidak efektif dan tingkat akurasi yang rendah.

Untuk mengatasi dua masalah di atas, *federated learning* (FL) [5, 6] menjadi solusi yang sangat efektif yang tidak hanya melindungi data pengguna perangkat IoT, namun juga memungkinkan antar perangkat IoT untuk melakukan kolaborasi proses *training* dengan adanya keterbatasan sumber daya komputasi. Pada akhirnya, model global yang digunakan untuk ekstraksi informasi penting bagi pemberi layanan aplikasi dapat mencapai tingkat akurasi yang tinggi. Penelitian terkait FL banyak dilakukan untuk optimisasi jumlah perangkat pengguna yang digunakan untuk proses *training*. Pada penelitian [11], sebuah sistem bernama *FedCS* dibuat untuk memilih partisipan yang akan berkontribusi dalam proses *training* yang sesuai dengan kemampuan komputasinya. Pada penelitian [12], sebuah sistem gabungan berbasis FL untuk memilih perangkat pengguna sebagai partisipan *training* berdasarkan data yang disimpan oleh perangkat yang bersifat *independently and identically distributed* (i.i.d) dikembangkan. Penelitian terkait dengan FL berlanjut untuk melakukan optimisasi jaringan *mobile edge* termasuk untuk aplikasi perangkat IoT. Pada penelitian [13], penggunaan FL yang digabungkan dengan *deep reinforcement learning* (DRL) dirancang untuk optimisasi *caching* dan *computation offloading* pada sebuah sistem *mobile edge computing* (MEC). Pada penelitian [14], sistem FL dengan DRL juga dikembangkan untuk *computation offloading* khusus untuk perangkat IoT. Di sisi lain, pengembangan sistem FL dengan menggunakan *stacked autoencoder* diteliti pada penelitian [15]. Kemudian, penelitian [16] membuat sistem FL yang berbasis algoritma *greedy* untuk melakukan optimisasi penempatan layanan yang tepat pada *proactive caching*.

Dari semua sistem FL yang dibuat, semua penelitian di atas tidak mempertimbangkan adanya masalah pada privasi dan keamanan data ketika proses *training* dilakukan dan model lokal dikirimkan dari perangkat pengguna ke *cloud server.* Pada penelitian [17], sebuah teknik yang disebut dengan *differentially private stochastic gradient descent* dibuat untuk menambah kan beberapa gangguan pada parameter yang sudah di-*training* pada sistem FL. Kemudian, penelitian [18] mengembangkan sebuah pendekatan yang dapat mencapai perlindungan privasi yang lebih baik dengan mengacak partisipan yang berkontribusi dalam proses *training* dan menambahkan distribusi *Gaussian*. Pada penelitian [19], sebuah mekanisme kolaboratif untuk membuat banyak partisipan mempelajari model global tanpa mengunggah semua parameter dari model lokal mereka ke *cloud server* dirancang.

Penelitian-penelitian di atas hanya berfokus pada perlindungan privasi dari parameter model lokal yang dikirimkan ke *cloud server* pada proses *training* dari sistem FL. Namun, sejauh ini belum ada penelitian yang berfokus pada gabungan dari perlindungan data pribadi perangkat pengguna IoT yang digunakan untuk proses *training* dan

model lokal untuk proses agregasi model global. Oleh karena itu, penelitian ini dimaksudkan untuk meningkatkan perlindungan data pengguna perangkat IoT dengan melakukan enkripsi data dan juga enkripsi model lokal selama proses *training* pada sistem FL berbasis *logistic regression* sedang berlangsung. Hal ini cukup sulit dilakukan karena terdapat dua proses enkripsi yang harus diselesaikan selama proses *training* pada sistem FL berjalan sampai dengan selesai.

## III. METODOLOGI

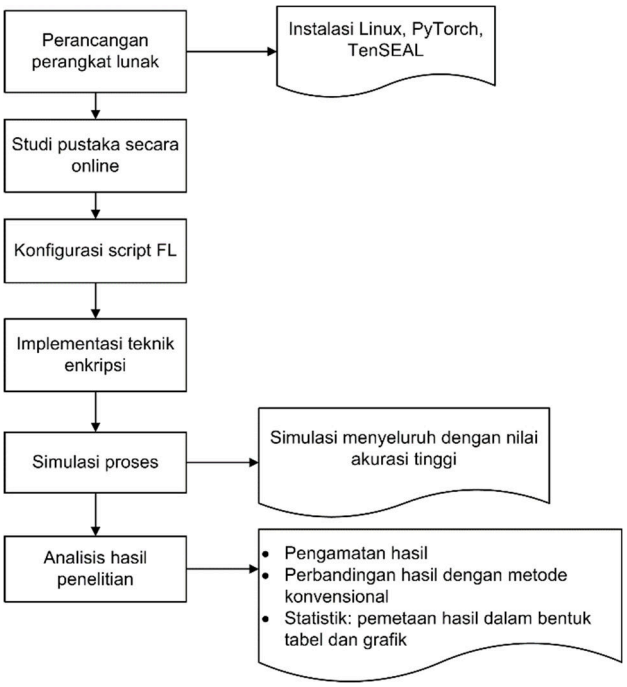
Pada penelitian ini, metode penelitian yang dilakukan berfokus pada perancangan pendekatan FL dengan tambahan metode enkripsi FHE. Berikut ini merupakan langkah-langkah proses penelitian yang dilakukan dengan diagram alir pada Gambar 1.

A. Perancangan perangkat lunak seperti melakukan instalasi Linux, PyTorch, dan *library* TenSEAL.
B. Studi literatur secara *online* yaitu dengan mengumpulkan literatur-literatur yang berkaitan dengan perangkat lunak seperti konfigurasi PyTorch, FL, dan metode enkripsi menggunakan FHE.
C. Konfigurasi *script* FL seperti penentuan dataset, data *pre-processing*, konfigurasi model, proses *training* dan *testing*, serta tampilan hasil.
D. Implementasi teknik enkripsi dengan menggunakan FHE dengan metode Cheon-Kim-Kim-Song (CKKS).
E. Simulasi proses melalui hasil integrasi antara FL dan FHE untuk mendapatkan hasil berupa nilai akurasi.
F. Perbandingan hasil penelitian dengan metode konvensional FL, sehingga diperoleh keunggulan dan kelemahan menggunakan hasil penelitian.

## IV. HASIL DAN PEMBAHASAN

Untuk melakukan evaluasi terhadap rancang bangun FL dengan keamanan data melalui metode enkripsi FHE, dataset yang bersifat *random* dengan jumlah sampel dan fitur yang beraneka ragam dapat digunakan. Sebagai label, dua nilai klasifikasi biner dengan menggunakan pendekatan *logistic regression* yaitu 0 dan 1 diimplementasikan. Pada konfigurasi digunakan jumlah sampel sebanyak 1000, 5000, dan 10000 sampel dengan 2, 3, dan 5 fitur. Semua eksperimen dilakukan dengan menggunakan perangkat lunak PyTorch CPU 1.10.1 dan TenSEAL 0.3.12. Untuk mengaplikasikan pendekatan FL, diasumsikan 10 pengguna IoT di mana setiap pengguna memiliki jumlah sampel yang sama.

Selanjutnya perbandingan kinerja akurasi antara FL dengan FHE dan FL konvensional dapat dijelaskan sebagai berikut. Sesuai dengan ekspektasi seperti yang tampak pada Tabel 1-3, akurasi yang diperoleh dari pendekatan FL dengan tambahan enkripsi akan menghasilkan kinerja yang sedikit lebih rendah dibandingkan dengan FL konvensional. Hal ini dikarenakan dengan adanya penggunaan enkripsi FHE, maka akan sulit untuk menggunakan fungsi aktivasi *Sigmoid* secara langsung. Untuk mengatasi hal tersebut digunakan bentuk aproksimasi dari fungsi aktivasi *Sigmoid* dengan derajat *polynomial* yang lebih rendah untuk mengurangi banyaknya operasi perkalian pada proses *training*.




Gambar 1. Diagram alir metode penelitian FL dengan FHE

Tabel 1. Kinerja akurasi dengan 1000 sampel

| Metode | 2 fitur | 3 fitur | 5 fitur |
|---|---|---|---|
| FL | 97.8% | 99.2% | 97.4% |
| FL + FHE | 96.4% | 97.8% | 97% |

Tabel 2. Kinerja akurasi dengan 5000 sampel

| Metode | 2 fitur | 3 fitur | 5 fitur |
|---|---|---|---|
| FL | 99.8% | 99.6% | 99.5% |
| FL + FHE | 96.4% | 97% | 97% |

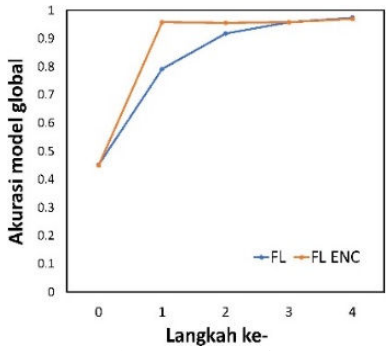
Tabel 3. Kinerja akurasi dengan 10000 sampel

| Metode | 2 fitur | 3 fitur | 5 fitur |
|---|---|---|---|
| FL | 99.7% | 99.8% | 99.7% |
| FL + FHE | 99.5% | 95.5% | 99.7% |

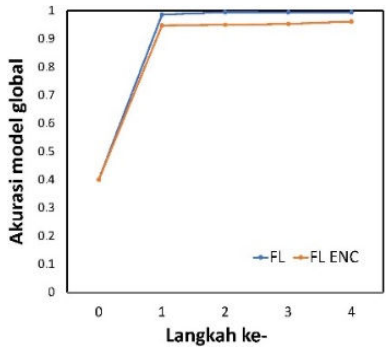
Secara umum dapat dilihat pada skenario 1000 sampel di Tabel 1 bahwa perbedaan akurasi antara FL konvensional dan FL dengan FHE adalah sampai dengan 1.43%. Namun, pada saat 5000 sampel digunakan di Tabel 2, ternyata perbedaan akurasi menjadi lebih besar yaitu sampai dengan 3.41%. Hal ini dikarenakan FL konvensional dapat melakukan *training* dengan baik sedangkan FL dengan FHE mengalami kendala dalam melakukan aproksimasi fungsi *Sigmoid*.

Sebagai bentuk solusi, penggunaan sampel yang lebih banyak yaitu 10000 sampel yang tambak pada Tabel 3 dapat memperbaiki perbedaan akurasi antara keduanya menjadi 0.22% saja khususnya untuk penggunaan 2 dan 5 fitur.

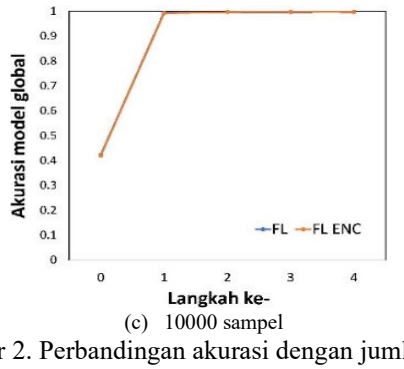
Untuk dapat melihat bagaimana proses *training* berjalan, hasil dalam bentuk grafik dengan jumlah 5 langkah dapat dilihat pada Gambar 2 dan 3 berikut ini. Untuk penggunaan jumlah sampel yang berbeda pada Gambar 2, semakin banyak sampel yang digunakan, maka perbedaan akurasi antara FL konvensional dan FL dengan FHE semakin kecil. Hal ini menunjukkan bahwa FL dengan tambahan keamanan data dapat digunakan untuk mengurangi adanya kebocoran data tanpa harus mengurangi nilai akurasi yang signifikan. Kemudian untuk penggunaan jumlah fitur yang berbeda pada Gambar 3, tidak dapat disimpulkan tren perbedaan nilai akurasi ketika fitur diperbanyak. Hal ini menunjukkan bahwa jumlah fitur yang semakin banyak tidak akan selalu menghasilkan nilai akurasi model yang lebih besar pada pendekatan FL dengan FHE.




(a) 1000 sampel

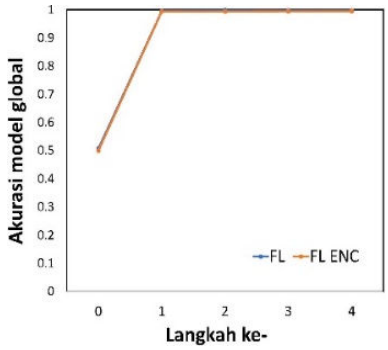



(b) 5000 sampel

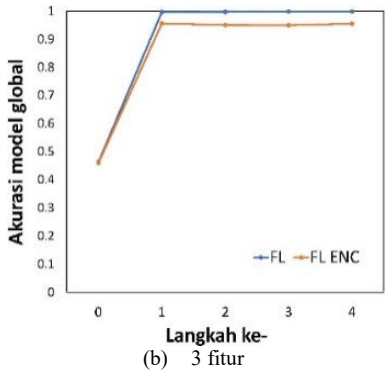



(c) 10000 sampel

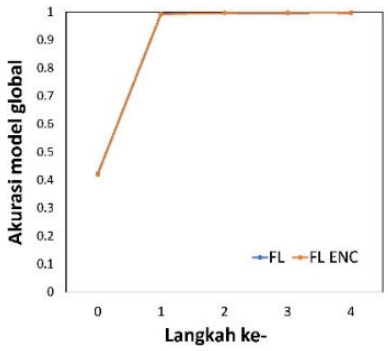
Gambar 2. Perbandingan akurasi dengan jumlah sampel yang berbeda (5 fitur)




(a) 2 fitur




(b) 3 fitur




(c) 5 fitur

Gambar 3. Perbandingan akurasi dengan jumlah fitur yang berbeda (10000 sampel)

## V. SIMPULAN

Pada penelitian ini telah dibuat sebuah rancang bangun sistem FL dengan tambahan keamanan data yang dapat digunakan oleh pengguna perangkat IoT. Teknik enkripsi FHE dengan metode *training logistic regression* digunakan sebagai bentuk perlindungan privasi pengguna IoT dari *malicious attackers*. Dari penelitian ini dapat disimpulkan bahwa, pendekatan FL dengan FHE menghasilkan nilai kinerja akurasi yang sedikit lebih rendah dengan adanya aproksimasi fungsi aktivasi *Sigmoid*. Namun, penggunaan jumlah sampel yang lebih banyak dapat digunakan untuk mengurangi perbedaan kinerja akurasi. Oleh karena itu, pendekatan FL dengan FHE dapat digunakan untuk mengurangi adanya kebocoran data pada perangkat IoT tanpa harus mengurangi nilai akurasi model global yang signifikan.

## REFERENSI


[1] I. H. Sarker, M. M. Hoque, Md. K. Uddin, and T. Alsanoosy, "Mobile Data Science and Intelligent Apps: Concepts, AI-Based Modeling and Research Directions," *Mob. Netw. Appl.*, vol. 26, no. 1, pp. 285–303, Feb. 2021, doi: 10.1007/s11036-020-01650-z.

[2] "Big Data and Business Analytics Market 2027," 2021. [Online]. Available: https://www.alliedmarketresearch.com/big-data-and-business-analytics-market

[3] C. Zhang, P. Patras, and H. Haddadi, "Deep Learning in Mobile and Wireless Networking: A Survey," *IEEE Commun. Surv. Tutor.*, vol. 21, no. 3, pp. 2224–2287, 2019, doi: 10.1109/COMST.2019.2904897.

[4] Y. Sun, M. Peng, Y. Zhou, Y. Huang, and S. Mao, "Application of Machine Learning in Wireless Networks: Key Techniques and Open Issues," *IEEE Commun. Surv. Tutor.*, vol. 21, no. 4, pp. 3072–3108, 2019, doi: 10.1109/COMST.2019.2924243.

[5] Q. Yang, Y. Liu, Y. Cheng, Y. Khang, T. Chen, and H. Yu, "Federated Learning: Synthesis Lectures on Artificial Intelligence and Machine Learning," *Learning*, vol. 13, no. 3, pp. 1–207, 2019.

[6] W. Y. B. Lim *et al.*, "Federated Learning in Mobile Edge Networks: A Comprehensive Survey," *IEEE Commun. Surv. Tutor.*, vol. 22, no. 3, pp. 2031–2063, 2020, doi: 10.1109/COMST.2020.2986024.

[7] E. Zeydan *et al.*, "Big Data Caching for Networking: Moving from Cloud to Edge," *IEEE Commun. Mag.*, vol. 54, no. 9, pp. 36–42, Sep. 2016, doi: 10.1109/MCOM.2016.7565185.

[8] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep Learning based Recommender System: A Survey and New Perspectives," *ACM Comput. Surv.*, vol. 52, no. 1, pp. 1–38, Jan. 2020, doi: 10.1145/3285029.

[9] W. Nie, V. C. S. Lee, D. Niyato, Y. Duan, K. Liu, and S. Nutanong, "A Quality-Oriented Data Collection Scheme in Vehicular Sensor Networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 5570–5584, Jul. 2018, doi: 10.1109/TVT.2018.2818190.

[10] A. Mulyani and U. Y. Oktiawati, "Implementasi Arsitektur Serverless Internet of Things pada Monitoring Cold Chain," 2022.

[11] T. Nishio and R. Yonetani, "Client Selection for Federated Learning with Heterogeneous Resources in Mobile Edge," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, Shanghai, China: IEEE, May 2019, pp. 1–7. doi: 10.1109/ICC.2019.8761315.

[12] N. Yoshida, T. Nishio, M. Morikura, K. Yamamoto, and R. Yonetani, "Hybrid-FL for Wireless Networks: Cooperative Learning Mechanism Using Non-IID Data," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, Dublin, Ireland: IEEE, Jun. 2020, pp. 1–7. doi: 10.1109/ICC40277.2020.9149323.

[13] X. Wang, Y. Han, C. Wang, Q. Zhao, X. Chen, and M. Chen, "In-Edge AI: Intelligentizing Mobile Edge Computing, Caching and Communication by Federated Learning," *IEEE Netw.*, vol. 33, no. 5, pp. 156–165, Sep. 2019, doi: 10.1109/MNET.2019.1800286.

[14] J. Ren, H. Wang, T. Hou, S. Zheng, and C. Tang, "Federated Learning-Based Computation Offloading Optimization in Edge Computing-Supported Internet of Things," *IEEE Access*, vol. 7, pp. 69194–69201, 2019, doi: 10.1109/ACCESS.2019.2919736.

[15] Z. Yu *et al.*, "Federated Learning Based Proactive Content Caching in Edge Computing," in *2018 IEEE Global Communications Conference (GLOBECOM)*, Abu Dhabi, United Arab Emirates: IEEE, Dec. 2018, pp. 1–6. doi: 10.1109/GLOCOM.2018.8647616.

[16] Y. Qian, L. Hu, J. Chen, X. Guan, M. M. Hassan, and A. Alelaiwi, "Privacy-aware Service Placement for Mobile Edge Computing via Federated Learning," *Inf. Sci.*, vol. 505, pp. 562–570, Dec. 2019, doi: 10.1016/j.ins.2019.07.069.

[17] M. Abadi *et al.*, "Deep Learning with Differential Privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, Oct. 2016, pp. 308–318. doi: 10.1145/2976749.2978318.

[18] R. C. Geyer, T. Klein, and M. Nabi, "Differentially Private Federated Learning: A Client Level Perspective." arXiv, Mar. 01, 2018. Accessed: May 10, 2023. [Online]. Available: http://arxiv.org/abs/1712.07557

[19] R. Shokri and V. Shmatikov, "Privacy-Preserving Deep Learning," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, Denver Colorado USA: ACM, Oct. 2015, pp. 1310–1321. doi: 10.1145/2810103.2813687.

# Pengembangan Aplikasi Pembelajaran *Online* dengan Metode Gamifikasi Berbasis Web

Clara Putri Andini Sukran[1], Irkham Huda[1*]
[1]Departemen Teknik Elektro dan Informatika, Sekolah Vokasi, Universitas Gadjah Mada
clara.p@mail.ugm.ac.id
*Korespondensi: irkham@ugm.ac.id;

***Abstract – Technological advances have had a major impact on progress in the field of education. As technology advances, many people are interested in studying the field of technology because the field of information technology is starting to be widely needed in various sectors. There are many sources that are widespread and easy to obtain by anyone and can be accessed anywhere as material for learning. However, independent learning with the internet is vulnerable to a decrease in learning motivation. Based on the existing problems, an Online Learning Application with Web-Based Gamification Method was built to help provide organized learning materials to facilitate the learning process. In this application, you will have a variety of material related to software development which is compiled using the gamification method in an effort to increase individual learning motivation. Existing material will be grouped into learning paths to assist users in determining the direction of learning and arranged sequentially. Online Learning Applications with Web-Based Gamification Methods are built using PHP programming languages, Laravel framework, JavaScript, and MySQL as database management tools. The results of developing Online Learning Applications with the Web-Based Gamification Method can help in carrying out a more organized and easy learning process.***

***Keywords : Gamification, Laravel, PHP, Web, Online Learning***

**Intisari – Kemajuan teknologi memberikan dampak yang besar dalam kemajuan di bidang pendidikan. Seiring kemajuan teknologi, banyak orang yang tertarik untuk mempelajari bidang teknologi karena bidang informasi teknologi mulai banyak dibutuhkan dalam berbagai sektor. Banyak sumber yang tersebar luas dan mudah didapatkan oleh siapapun serta dapat diakses di mana saja sebagai bahan untuk belajar. Namun, pembelajaran secara mandiri dengan internet rentan dengan penurunan motivasi belajar. Berdasarkan masalah yang ada, maka dibangun sebuah Aplikasi Pembelajaran Online dengan Metode Gamifikasi Berbasis Web untuk membantu menyediakan materi belajar yang tertatata untuk mempermudah proses belajar. Dalam aplikasi ini akan memiliki beragam materi yang terkait seputar pengembangan perangkat lunak yang disusun dengan menggunakan metode gamifikasi dalam upaya meningkatkan motivasi belajar pada individu. Materi yang ada akan dikelompokkan ke dalam *learning path* untuk membantu pengguna dalam menentukan arah belajar dan disusun secara berurutan. Aplikasi Pembelajaran *Online* dengan Metode Gamifikasi Berbasis Web dibangun dengan menggunakan bahasa pemograman PHP, *framework* Laravel, JavaScript, serta MySQL sebagai sarana dalam pengelolaan basis data. Hasil dari pengembangan Aplikasi Pembelajaran *Online* dengan Metode Gamifikasi Berbasis Web dapat membantu dalam melakukan proses belajar yang lebih tertata dan mudah.**

**Kata kunci : Gamifikasi, Laravel, PHP, Web, *Online Learning***

## I. PENDAHULUAN

Teknologi berkembang dengan sangat cepat pada abad ke-21. Kehadiran internet memberikan dampak yang sangat besar terhadap kemajuan teknologi. Teknologi telah dapat dirasakan hampir di seluruh dunia pada berbagai lapisan masyarakat. Salah satu kemajuan teknologi yang saat ini banyak dirasakan oleh masyarakat pada saat ini yakni dalam bidang informasi teknologi. Perkembangan teknologi ini menjadi perhatian bagi banyak orang. Tidak sedikit orang yang mencoba untuk mempelajari hal yang berkaitan dengan bidang ini. Hal ini dikarenakan bidang informasi teknologi mulai dibutuhkan dalam berbagai sektor yang ada saat ini.

Pada masa ini banyak orang yang memanfaatkan sumber internet dalam mempelajari suatu hal sebagai upaya meningkatkan kemampuan yang dimiliki, salah satunya peningkatan keahlian dalam bidang teknologi. Peningkatkan kemampuan bagi setiap individu diperlukan untuk meningkatkan kepercayaan diri dan dalam upaya untuk menghadapi persaingan antar individu yang semakin ketat. Banyak sumber di internet memberikan materi yang dapat diakses oleh siapapun dan dimanapun. Namun, kemudahan yang diberikan untuk melakukan pembelajaran secara mandiri yang dilakukan seseorang ketika memulai bidang baru rentan untuk tidak diselesaikan secara tuntas akibat adanya penurunan motivasi untuk belajar ataupun terjadi kebosanan selama proses belajar berlangsung.

Dalam upaya untuk meningkatkan keinginan menyelesaikan materi yang dipilih pengguna, mengurangi rasa bosan, serta upaya untuk meningkatkan motivasi belajar, maka aplikasi ini akan menerapkan metode gamifikasi. Gamifikasi adalah penerapan teknik desain dan mekanisme permainan ke dalam konteks *non-game* dalam mengikat pengguna untuk dapat mencapai suatu tujuan [1]. Pemberian gamifikasi akan dapat membuat pembelajaran menjadi lebih interaktif dan meningkatkan motivasi dalam menyelesaikan materi yang ada [2]. Permainan dapat memberikan tiga keuntungan psikologis bagi pengguna, yakni kognitif, emosional, dan sosial, serta dapat meningkatkan motivasi pengguna [3].

Penerapan gamfikasi diberlakukan dengan pemberian poin ketika pengguna menyelesaikan tugas, pemberian *badges* ketika materi selesai dipelajari, dan *leader board* untuk memberikan daftar teratas pengguna lain yang mengumpulkan poin tertinggi.

Aplikasi pembelajaran *online* ini akan memiliki nama sebagai SETUP (*software engineering technology upgrading skill*) dan dapat berisikan berbagai materi yang terkait dalam bidang *software engineering* yang dapat diakses dengan mudah oleh penggunanya dalam rentang waktu kapanpun, dan dimanapun. Pengembangan terhadap aplikasi pembelajaran *online* berbasis web akan membantu dalam menyediakan materi yang tertata dan interaktif terhadap penggunanya. Materi akan disusun berdasarkan preferensi pengguna terhadap keahlian yang akan dipelajari, sehingga akan terbentuk *learning path* dalam memudahkan pengguna mengetahui arah dari keahlian yang akan diambil. Fitur gamifikasi yang terdapat di dalam aplikasi adalah pemberian poin setiap peserta menyelesaikan tugas dan pemberian *badges* setiap peserta menyelesaikan kelas, yang berfungsi untuk menambah motivasi peserta. Juga terdapat *leaderboard* untuk melihat perbandingan pencapaian antar peserta.

## II. DASAR TEORI

### A. *E-Learning*

Sebuah *e-learning* merupakan pembelajaran yang merujuk terhadap penggunaan internet untuk dapat mengirimkan rangkaian solusi yang sesuai, sehingga dapat meningkatkan pengetahuan dan keterampilan [4].

### B. Aplikasi

Aplikasi merupakan suatu program yang siap pakai dan dapat digunakan untuk menjalankan perintah dari pengguna dengan bertujuan untuk menghadirkan hasil [5].

### C. Web

Web atau *website* adalah suatu kumpulan dari berbagai macam halaman yang dapat menampilkan beragam informasi media berupa gambar, teks, video ataupun gabungan dari semua, baik bersifat statis maupun dinamis, dimana semua dihubungkan melalui jaringan halaman atau *hyperlink* [6].

### D. Gamifikasi

Gamifikasi dapat didefinisikan sebagai penggunaan elemen permainan dan mekanik ke dalam konteks *non-game* yang memiliki dua unsur kunci yang digunakan untuk tujuan non-hiburan dan menarik inspirasi dari permainan yang terutama elemen penyusun tanpa menciptakan permainan secara utuh [7].

### E. Metode *Scrum*

Metode *Scrum* merupakan kerangka kerja suatu siklus pengembangan sistem yang digunakan dalam pengelolaan produk dengan beragam teknik dan proses, sehingga dapat terus meningkatkan kinerja dari tim, produk, dan lingkungan kerja. *Scrum* menggunakan pendekatan *agile* dengan didasarkan pada pengembangan berulang dan bertahap untuk dapat mengoptimalkan kemampuan prediksi dan pengendalian risiko [8].

### F. PHP

PHP atau *Hypertext preprocessor* merupakan bahasa pemrograman *server side,* dimana penggunaan PHP sendiri dalam web adalah untuk menyesuaikan tampilan konten dengan sebuah situasi dan akan memberikan web yang dinamis.

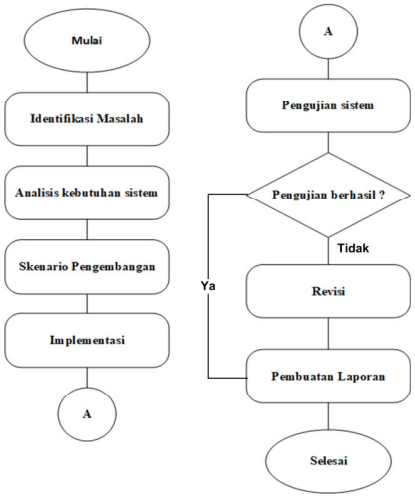
### G. Laravel

Laravel adalah sebuah *framework* pada bahasa pemrograman PHP yang diciptakan dan disebarkan di bawah lisensi MIT dan dibangun menggunakan konsep MVC (*model view controller*).

## III. METODOLOGI

### A. Tahapan Penelitian

Pada pengembangan Aplikasi Pembelajaran *Online* dengan Metode Gamifikasi akan digunakan *Scrum*. *Scrum* akan menggunakan pendekatan *agile* dimana pengembangan perangkat lunak dengan didasarkan pada pengembangan yang berulang. Pengembangan akan dimulai dengan pendefinisian bisnis proses, analisis kebutuhan sistem, desain antarmuka, perencanaan pengembangan, implementasi dan diakhiri dengan pengujian sistem atau seperti pada Gambar 1.




Gambar 1. Diagram alir penelitian

### B. Kebutuhan Sistem

1) Pengguna Sistem

Dalam pengembangan aplikasi pembelajaran *online* ini akan terdapat dua komponen pengguna, yaitu admin yang akan bertindak sebagai pengelola data utama dalam aplikasi dan pengguna aplikasi.

- Admin
- *Student* (*user*)
- Pengunjung

2) Kebutuhan fungsional

Kebutuhan fungsional terhadap Sistem akan dibagi berdasarkan peran yang terdapat pada sistem.

a) Admin
- Admin dapat melakukan manajemen pengguna
- Admin dapat melakukan manajemen kelas
- Admin dapat melakukan manajemen materi belajar
- Admin dapat menyusun *learning path*
- Admin dapat melakukan tinjauan tugas
- Admin dapat melakukan manajemen tugas
- Admin dapat melakukan manajemen *badges*
- Admin dapat memberikan poin

b) *Student*
- *Student* dapat membuat akun
- *Student* dapat masuk ke akun
- *Student* dapat melakukan pendaftaran kelas
- *Student* dapat melihat materi
- *Student* dapat memperoleh poin dan *badges*
- *Student* dapat melihat *learning path*
- *Student* dapat melihat kelas
- *Student* dapat mengerjakan tugas
- *Student* dapat melakukan perubahan profil dirinya
- *Student* dapat melakukan pencarian kelas

c) Pengunjung
- Pengunjung dapat melihat *learning path*
- Pengunjung dapat melihat kelas
- Pengunjung dapat melakukan pencarian kelas

C. Kebutuhan Non-fungsional

Kebutuhan non-fungsional dari sistem merupakan kebutuhan yang dimiliki sistem di luar dari kebutuhan fungsional.

1) Aplikasi pembelajaran berbasis web
2) Aplikasi dapat diakses melalui desktop ataupun mobile dengan menggunakan *browser*
3) Aplikasi minimal dapat berjalan pada *browser* Google Chrome dan Mozilla Firefox
4) Aplikasi membutuhkan koneksi internet

D. Skenario Pengembangan

Skenario pengembangan dilakukan dengan menerapkan metode *scrum*, dimana akan disusun dengan penyusunan *product backlog* berdasarkan *story.*

*1) Story*

*Story* merupakan istilah lain di dalam *scrum* untuk menyebutkan fitur yang diceritakan dalam perspektif pengguna aplikasi. Penulisan *story* dapat dilihat pada Tabel 1.

Tabel 1. Penyusunan *Story*

| **Peran** | ***Story*** |
|---|---|
| Admin | Saya dapat melakukan login |
| Admin | Saya dapat melakukan tambah pengguna |
| Admin | Saya dapat melakukan hapus pengguna |
| Admin | Saya dapat melakukan unggah materi |
| Admin | Saya dapat melakukan pembaruan materi |
| Admin | Saya dapat melakukan penghapusan materi |
| Admin | Saya dapat melakukan penambahan kelas |
| Admin | Saya dapat melakukan pembaruan kelas |
| Admin | Saya dapat melakukan penghapusan kelas |
| Admin | Saya dapat mengubah *password* |
| Admin | Saya dapat melakukan penyusunan *learning path* |
| Admin | Saya dapat melakukan tinjauan tugas |
| Admin | Saya dapat melakukan pemberian *point* |
| Admin | Saya dapat melakukan pemberiana badges |
| Pengguna | Saya dapat melakukan sign up |
| Pengguna | Saya dapat melakukan login |
| Pengguna | Saya dapat melakukan logout |
| Pengguna | Saya dapat mengubah password |
| Pengguna | Saya dapat mendaftar kelas |
| Pengguna | Saya dapat melakukan unggah tugas |
| Pengguna | Saya dapat melihat materi |
| Pengguna | Saya dapat mengerjakan tugas |
| Pengguna | Saya dapat melihat *point* |
| Pengguna | Saya dapat mendapatkan *badges* |
| Pengguna | Saya dapat memilih *learning path* |
| Pengguna | Saya dapat melihat *leaderboard* |
| Pengguna | Saya dapat mencari kelas |

*2) Product backlog*

*Product backlog* merupakan daftar fungsi dan kebutuhan yang akan dikerjakan selama pengembangan berlangsung. Penyusunan produk *backlog* dilakukan berdasarkan *story.* Tabel 2 akan menampilkan daftar *product backlog.*

Tabel 2. Penyusunan Backlog

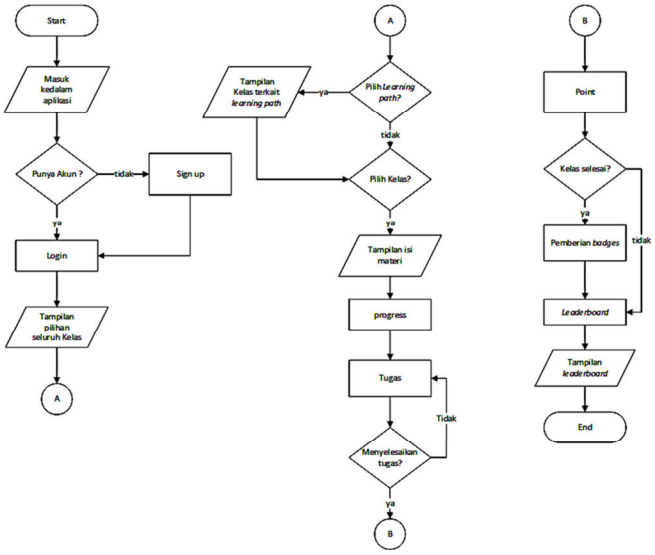
| *Story* | *Priority* | *Estimation* |
|---|---|---|
| Melakukan *login* | *Must* | 1 |
| Melakukan *logout* | *Must* | 1 |
| Melakukan tambah pengguna | *Must* | 1 |
| Melakukan hapus pengguna | *Must* | 1 |
| Melakukan unggah materi | *Must* | 3 |
| Melakukan pembaruan materi | *Should* | 2 |
| Melakukan penghapusan materi | *Should* | 2 |
| Melakukan penambahan kelas | *Must* | 3 |
| Melakukan pembaruan kelas | *Should* | 2 |
| Melakukan penghapusan kelas | *Should* | 2 |
| Mengubah *password* | *Should* | 1 |
| Melakukan penyusunan *learning path* | *Must* | 4 |
| Dapat melakukan tinjauan tugas | *Must* | 4 |
| Dapat melakukan pemberian poin | *Must* | 2 |
| Dapat melakukan pemberian *badges* | *Must* | 2 |
| Melakukan *sign up* | *Must* | 1 |
| Melalukan *logout* | *Must* | 1 |
| Melakukan *login* | *Must* | 1 |
| Mengubah *password* | *Should* | 1 |
| Mendaftar kelas | *Must* | 2 |
| Melakukan unggah tugas | *Must* | 3 |
| Melihat materi | *Must* | 2 |
| Mengerjakan tugas | *Must* | 4 |
| Melihat poin | *Must* | 1 |
| Mendapatkan *badges* | *Must* | 1 |
| Memilih *learning path* | *Must* | 2 |
| Melihat *leaderboard* | *Must* | 4 |
| Mencari kelas | *Should* | 1 |

## IV. HASIL DAN PEMBAHASAN

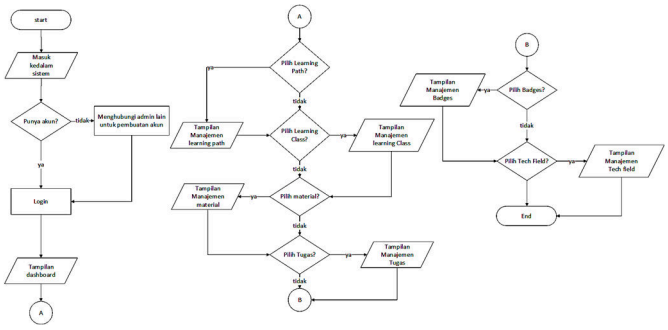
Analisis terhadap sistem dilakukan sebelum melakukan pengembangan. Analisis dilakukan untuk melakukan identifikasi terhadap masalah yang ada pada aplikasi dan menetapkan kebutuhan dari aplikasi yang akan dilakukan selama pengembangan.

### A. Ilustrasi Sistem

Aplikasi ini akan diaplikasikan pada lingkup informasi dan teknologi sebagai media pembelajaran *online* dengan penerapan metode gamifikasi. Penggambaran dari ilustrasi sistem akan terbagi menjadi dua dimana disesuaikan dengan peran yang terdapat di dalam sistem. Ilustrasi sistem untuk *student* ditunjukkan pada Gambar 2 dan Ilustrasi sistem untuk admin dapat dilihat pada Gambar 3.



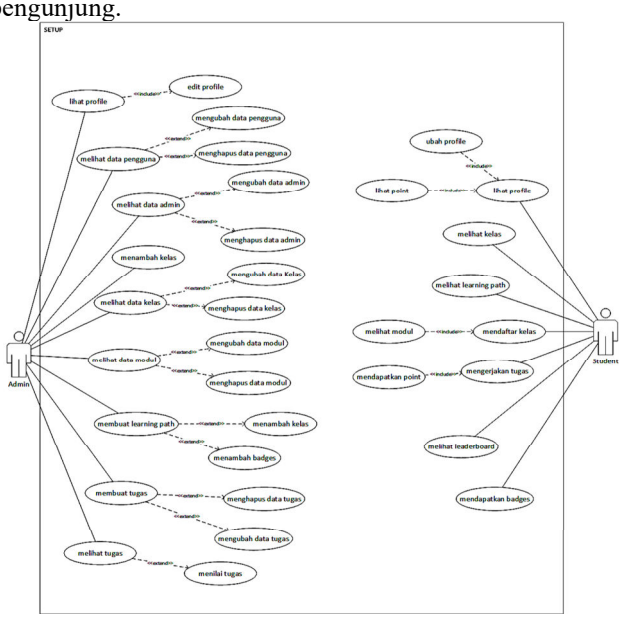

Gambar 2. Ilustrasi sistem (*student*)




Gambar 3. Ilustrasi sistem (admin)

### B. Perancangan Proses

Perancangan proses akan menjelaskan bagaimana fase dari alur yang ada di dalam aplikasi. Perancangan proses akan digambarkan menggunakan *use case diagram* yang dapat dilihat pada Gambar 4. *Use case* akan menjelaskan fitur-fitur yang dapat diakses oleh setiap peran yang terdapat di dalam sistem. Untuk peran yang terdapat di dalam sistem akan terbagi menjadi tiga yakni, admin, *student*, dan pengunjung.




Gambar 4. *Use case diagram*

C. Spesifikasi Lingkungan pengembangan

Dalam lingkungan pengembangan aplikasi akan menggunakan spesifikasi perangkat lunak sebagai berikut:

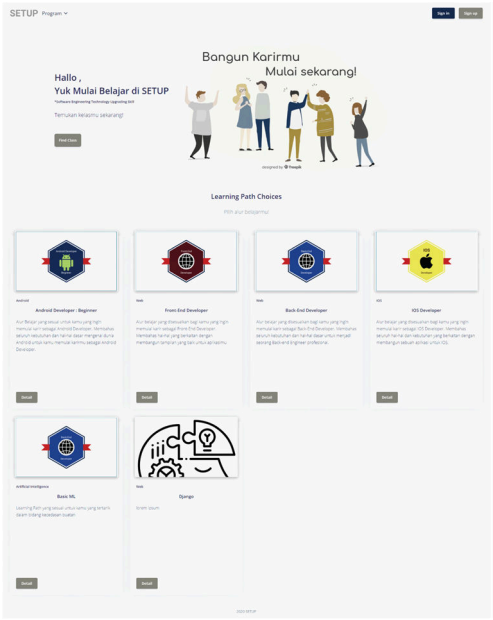
- Windows 10 64 bit sebagai sistem operasi
- Laravel 7 sebagai framework bahasa pemrograman PHP
- PHP 7.4.10 sebagai bahasa pemrograman
- HTML, CSS, dan JavaScript
- XAMPP 3.2.4 sebagai paket aplikasi Apache, PHP, dan MySQL
- Google Chrome sebagai web browser
- Visual Studio Code sebagai text editor

Implementasi yang dilakukan untuk aplikasi pembelajaran *online* akan menggunakan perangkat keras sebagai berikut:

- Prosesor Intel® Core i7-8850H, Nvdia Quadro P600
- RAM 8GB
- SSD 256GB
- Monitor LCD 16”
- Keyboard
- Touchpad
- Mouse

D. Implementasi Sistem

Implementasi dari sistem yang menggunakan Laravel akan menggunakan konsep MVC (*model, view, controller*). Konsep MVC akan memisahkan bagian untuk mengatur tampilan dan fungsi sistem. Untuk melakukan implementasi tampilan dalam Laravel akan menggunakan *Blade.* Tampilan dari halaman utama sistem dapat dilihat pada Gambar 5. Halaman utama sistem akan berisikan informasi umum dan daftar alur belajar yang dimiliki oleh sistem.

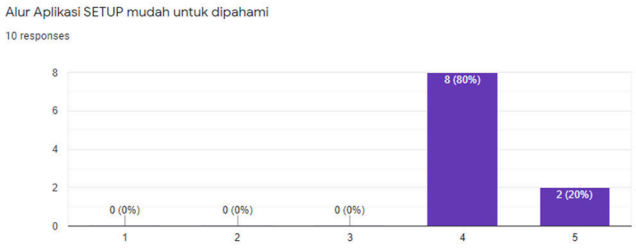



Gambar 5. Tampilan halaman utama

E. Pengujian

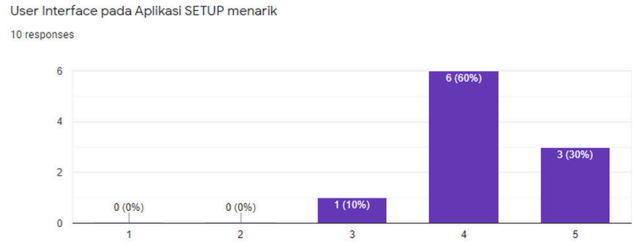
Pengujian dilakukan terhadap sistem untuk mengetahui respon pengguna terhadap sistem. Pengujian dilakukan terbatas dengan menggunakan kuesioner yang disebarkan kepada 10 responden berasal dari mahasiswa dan pekerja. Pengujian dilakukan meliputi alur dari sistem, tampilan dari sistem, serta fitur yang terdapat dalam sistem.

Diagram hasil kuesioner menunjukkan sumbu Y jumlah responden dan X penilaian responden dimana nilai semakin tinggi semakin baik.




Gambar 6. Pengujian alur sistem

Gambar 6 di atas menunjukan hasil dari pengujian yang dilakukan terhadap alur yang terdapat pada sistem. Gambar 7 akan menampilkan pengujian yang dilakukan terhadap tampilan antarmuka yang dimiliki oleh sistem.

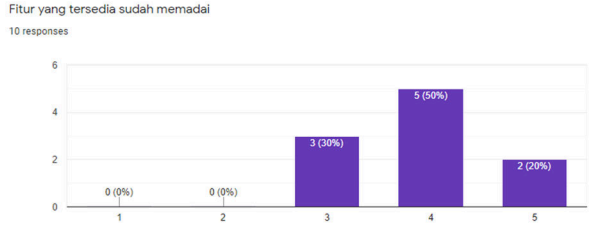



Gambar 7. Pengujian tampilan antarmuka

Pengujian fitur gamifikasi yang dimiliki oleh sistem pembelajaran *online* dilakukan untuk mengetahui apakah fitur tersebut dapat digunakan sesuai dengan rencana. Hasil dari pengujian terhadap fitur gamifikasi dapat dilihat pada Gambar 8.


Gambar 8. Pengujian fitur gamifikasi

Kemudian, pengujian selanjutnya adalah pengujian untuk fitur yang terdapat di dalam sistem apakah sudah memadai. Pengujian dapat dilihat pada Gambar 9.




Gambar 9. Pengujian kesediaan fitur

## V. SIMPULAN

Berdasarkan dari tahapan yang telah dijalani terhadap Aplikasi Pembelajaran *Online* dengan Metode Gamifikasi Berbasi Web pada penelitian ini, dapat diperoleh kesimpulan sebagai berikut:

1. Aplikasi Pembelajaran *online* dengan metode gamifikasi berhasil dibangun dan dapat berjalan dengan baik.
2. Sistem dapat digunakan untuk melakukan pembelajaran secara mandiri pada *software engineering*.
3. Hasil pengujian yang telah dilakukan menunjukan fitur yang terdapat di dalam aplikasi pembelajaran *online* dapat digunakan sesuai dengan kebutuhan.
4. Implementasi dari Gamifikasi pada aplikasi dapat berhasil dilakukan dan dapat membantu pengguna untuk meningkatkan motivasi belajar.

## REFERENSI

[1] G. Zichermann and C. Cunningham, *Gamification by Design: Implementing Game Mechanics in Web and Mobile Apps*. O'Reilly Media, Inc., 2011.

[2] F. H. Romdhoni and R. P. Wibowo, "Penerapan Gamification pada Aplikasi Interaktif Pembelajaran SQL Berbasis Web," *J. Tek. Pomits*, vol. 1, no. 1, pp. 1–6, 2014.

[3] J. J. Lee and J. Hammer, "Gamification in Education: What, How, Why Bother?," *Acad. Exch. Q.*, vol. 15, no. 2, p. 146, 2011.

[4] M. J. Rosenberg, *E-learning: Strategies for Delivering Knowledge in the Digital Age*. New York: McGraw-Hill, 2001.

[5] H. Abdurahman and A. R. Riswaya, "Aplikasi Pinjaman Pembayaran secara Kredit pada Bank Yudha Bhakti," *J. Comput. Bisnis E-J.*, vol. 8, no. 2, pp. 61–69, 2014.

[6] A. M. Rudianto, *Pemrograman Web Dinamis menggunakan PHP dan MySQL*. Yogyakarta: Andi Offset, 2011.

[7] K. Seaborn and D. I. Fels, "Gamification in theory and action: A Survey," *Int. J. Hum.-Comput. Stud.*, vol. 74, pp. 14–31, 2015.

[8] K. Schwaber and J. Sutherland, "The Scrum Guide," 2011.

# Analisis Perbedaan Pengaruh Penggunaan *Iptables Chains* dalam Mencegah *Denial of Service* (DoS) pada Jaringan IoT

Hanifatun Nida[1], Ronald Adrian[1*]
[1]Departemen Teknik Elektro dan Informatika, Sekolah Vokasi, Universitas Gadjah Mada
hanifatun.nida@mail.ugm.ac.id
*Korespondensi: ronald.adr@ugm.ac.id;

***Abstract – The growth of home IoT device is aligned with the growth of security vulnerabilities in home network. IoT has limited resources which makes them an ideal target to Denial of Service (DoS) attack, including SYN flood. This attack tends to targeting crucial resources, such as CPU. When this attack continuously happen massively, this attack can exhaust the resources and make IoT device lost their functionality. Iptables implementation on Raspberry Pi comes to rescue to reduce SYN flood effects. Iptables has various chains and tables with their own function. This research aims to analyze the difference effect of iptables chains usage against Raspberry Pi CPU usage. The lower CPU usage, the lower DoS will likely to happen. This results then will be compared to when Raspberry Pi use previous solution by other researcher. The results show no significant difference in CPU usage for both rules.***
***Keywords : Internet of Things (IoT), Raspberry Pi, Iptables, Denial of Service (DoS)***

**Intisari – Peningkatan penggunaan perangkat *Internet of Things* (IoT) rumahan dibarengi dengan berkembangnya berbagai kerentanan keamanan pada jaringan rumah. Keterbatasan sumber daya yang dimiliki perangkat IoT menjadikannya target ideal untuk diluncurkan serangan *Denial of Service* (DoS), termasuk *SYN flood*. Serangan ini cenderung menargetkan sumber daya IoT yang krusial, termasuk CPU. Apabila dilakukan secara terus menerus, serangan ini mampu menguras sumber daya IoT dan membuatnya kehilangan fungsionalitasnya. Penerapan *iptables* pada *Raspberry Pi* mampu meminimalkan dampak serangan *SYN flood*. *Iptables* memiliki berbagai *chains* dan *tables* dengan kemampuan dan fungsi yang berbeda. Penelitian ini bertujuan untuk menganalisis perbedaan pengaruh penggunaan *iptables chains* terhadap penggunaan CPU *Raspberry Pi*. Penggunaan CPU yang semakin rendah memperkecil kemungkinan terjadinya DoS. Hasil penelitian ini kemudian akan dibandingkan dengan penggunaan CPU ketika aturan *iptables* pada penelitian terdahulu diterapkan. Hasilnya, kedua aturan mengonsumsi CPU dalam persentase yang hampir sama.**
**Kata kunci : *Internet of Things* (IoT), *Raspberry Pi, Iptables*, *Denial of Service* (DoS)**

## I. PENDAHULUAN

*Internet of Things* (IoT) merupakan teknologi yang memungkinkan terjadinya komunikasi antara perangkat elektronik dengan sensor melalui internet. Berdasarkan data Statistik pada Oktober 2021, terdapat sekitar 11,3 miliar perangkat IoT pada Tahun 2021 dan akan bertambah menjadi 19,1 miliar pada Tahun 2025 [1]. Sementara itu, berdasarkan prediksi yang dilakukan Insider Intelligence pada Oktober 2021, sekitar 60 juta rumah tangga di Amerika Serikat akan menggunakan perangkat IoT pada Tahun 2021 [2].

Masifnya penggunaan IoT ternyata dibarengi dengan meningkatnya berbagai ancaman keamanan, termasuk *Denial of Service/Distributed Denial of Service* (DoS/DDoS). Berdasarkan analisis Tahun 2016 terkait insiden siber terdahulu, disimpulkan bahwa 96% perangkat yang terkena DDoS adalah perangkat IoT. Salah satu kasus serangan DDoS terjadi pada Oktober 2016 yang disebabkan terinfeksinya perangkat IoT oleh *malware* Mirai. Hal ini berimplikasi pada kegagalan akses ke sebagian besar *platform* dan layanan di Amerika Utara dan Eropa. Serangan DoS/DDoS sendiri cenderung menargetkan sumber daya IoT yang krusial, salah satunya CPU [3].

Banyaknya kasus serangan terhadap perangkat IoT dikarenakan perangkat IoT sendiri memiliki kemampuan yang terbatas dan datang tanpa fitur keamanan. Keterbatasan sumber daya IoT seperti rendahnya memori, komputasi, dan konsumsi baterai membuatnya rentan terhadap serangan yang bertujuan menghabiskan sumber daya itu sendiri. Hal ini menjadikannya target yang ideal untuk dilakukan penyerangan seperti DoS [4].

Berdasarkan pada keterbatasan dan kerentanan perangkat IoT tersebut, maka diperlukan suatu cara untuk memproteksi perangkat IoT rumahan dari serangan DoS seperti *SYN flood*. *Salah* satu cara yang dapat digunakan untuk menangani *SYN flood* adalah dengan menggunakan *iptables* sebagai *firewall*. Penerapan *iptables* sebagai *firewall* pernah dilakukan oleh Marek Majkowski dalam menganalisis perbandingan jumlah paket yang ditolak ketika menggunakan *chain* PREROUTING tabel RAW dan *chain* INPUT tabel FILTER dalam rentang waktu yang sama [5]. Hasilnya, *chain* PREROUTING tabel RAW lebih banyak menolak paket dibanding *chain* INPUT tabel FILTER. Sementara itu, Dmitrij Melkov berfokus pada pengaruh jumlah aturan yang diterapkan pada tiap *chain* terhadap *throughput* [6]. Hasilnya, *chain* INPUT tabel FILTER mampu menangani lebih banyak aturan dengan *throughput* lebih besar dibanding *chain* PREROUTING tabel RAW. Berbeda dengan keduanya, penelitian ini akan membandingkan pengaruh kinerja tiap *chain* dalam menangani *SYN flood* terhadap penggunaan CPU Raspberry Pi. Penggunaan CPU dijadikan tolok ukur keefektifan suatu *iptables chains* dalam menangani *SYN flood.* Aturan yang pernah digunakan AL-Musawi [7] dalam memitigasi DoS/DDoS pun dianalisis.

## II. DASAR TEORI

### A. Raspberry Pi

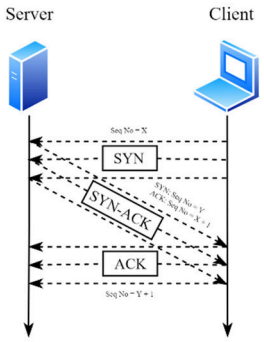
Raspberry Pi merupakan sebuah *single-board* komputer yang dapat digunakan bersama perangkat keras, seperti *mouse, keyboard,* dan monitor. Raspberry Pi memiliki CPU yang kompatibel dengan ARM, *on-chip graphics,* dan slot untuk memasukkan *SD card. SD card* ini digunakan sebagai penyimpanan memori bagi sistem operasi dan aplikasi secara keseluruhan [8].

Ukuran Raspberry Pi yang sekecil kartu kredit dilengkapi dengan *wireless chip* yang mampu menjadikannya *Wireless Access Point* (WAP) bagi perangkat-perangkat IoT, empat *USB port,* sebuah *ethernet port,* sebuah *HDMI port,* dan empat puluh pin *General-purpose input/output* (GPIO). Fitur-fitur resebut mampu menjadikan Raspberry Pi sebagai perangkat koneksi IoT, karena kemampuannya mengumpulkan, menyimpan, memproses, hingga mengunggah data yang dihasilkan perangkat-perangkat IoT yang terhubung padanya [9].

### B. Serangan *SYN Flood*

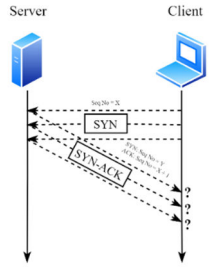
Serangan DoS maupun Serangan *SYN flood* merupakan salah satu tipe serangan *Denial of Service* (DoS) yang mampu menyibukkan seluruh *port* target [10]. Serangan ini dilakukan dengan mengirimkan paket SYN dalam jumlah besar dengan memanfaatkan kerentanan dari *TCP connection sequence,* yaitu *three-way handshake*. *Three-way handshake* adalah suatu metode yang digunakan klien dan server untuk bertukar paket SYN *(synchronization)* dan ACK *(acknowledgement)* sebelum komunikasi data antar keduanya terjadi [11].

Normalnya, untuk memulai koneksi TCP, klien akan mengirim paket SYN ke server. Server kemudian membalasnya dengan mengirim paket SYN-ACK. Paket ACK kemudian dikirimkan oleh klien ke server, yang menandakan koneksi TCP antara klien dengan server telah terbangun [12].




Gambar 1. Proses *TCP three-way handshake*

Sementara itu, serangan *SYN flood* terjadi ketika klien mengirimkan sejumlah besar paket SYN ke tiap *port* mesin target dan tidak merespons paket SYN-ACK server dengan ACK [13]. Akibatnya, server terus menunggu paket ACK yang akhirnya menyebabkan koneksi setengah terbuka *(half-open connection)* antara klien dan server. Koneksi ini akan terus terbuka dan menyebabkan server kebanjiran paket SYN hingga membuatnya kehabisan sumber daya. Habisnya sumber daya pada server mampu menyebabkan terjadinya DoS, yakni kondisi ketika server kehilangan fungsionalitasnya dalam menyediakan sumber daya jaringan bagi pengguna yang sah [14].




Gambar 2. Proses *TCP half-open connection*

### C. Iptables

Iptables merupakan *default firewall* di Linux dan beroperasi di atas *Netfilter*. Dengan kata lain, *iptables* merupakan *front end* dari *Netfilter* [15]. *Netfilter* merupakan *framework* yang digunakan pada pemrosesan paket di kernel Linux [16]. *Framework Netfilter* menyediakan *hooks* penyaringan paket *(packet filtering hooks)* pada *network stack* kernel Linux. Untuk memproses penyaringan paket, iptables berinteraksi dengan *hooks* ini. *Hooks* kemudian akan mencegat paket dan meneruskannya ke aturan-aturan pemrosesan paket. Aturan-aturan ini sendiri dikelola oleh tabel Netfilter yang memiliki beberapa *chains* bawaan *(built-in chains)* [12]. Beberapa tabel pada iptables di antaranya FILTER, NAT, MANGLE, dan RAW. Nama tiap *chain* dari tabel-tabel Netfilter sendiri mencerminkan nama *Netfilter hook* yang berasosiasi dengannya.

Paket atau lalu lintas yang datang dapat memicu aktifnya *Netfilter hooks*. Aktifnya *hook* bergantung pada kondisi paket atau lalu lintas seperti dijelaskan pada Tabel 1 [17]. Setiap *Netfilter hook* mampu mengaktifkan satu *chain.*

Tabel 1. *Netfilter hooks*

| ***Netfilter hook*** | **Kondisi diaktifkannya *hook*** | ***Chain* yang diaktifkan** |
|---|---|---|
| NF_IP_PRE_ROUTING | *Hook* ini akan diaktifkan segera ketika terdeteksi adanya lalu lintas yang masuk *(incoming traffic)* pada antarmuka jaringan. *Hook* ini diproses sebelum dibuatnya keputusan perutean *(routing decision)* mengenai ke mana paket ini harus dikirim. | PREROUTING |
| NF_IP_LOCAL_IN | *Hook* ini akan diaktifkan setelah paket yang masuk telah dirutekan dan ditujukan ke sistem lokal. | INPUT |
| NF_IP_FORWARD | *Hook* ini akan diaktifkan setelah paket yang masuk telah dirutekan dan akan diteruskan ke *host* lain. | FORWARD |
| NF_IP_LOCAL_OUT | *Hook* ini akan diaktifkan ketika terdeteksi ada lalu lintas atau paket yang dihasilkan sistem lokal yang akan dikirim ke luar. | OUTPUT |
| NF_IP_POST_ROUTING | *Hook* ini akan diaktifkan oleh lalu lintas keluar *(outgoing traffic)* atau lalu lintas yang diteruskan *(forwarded traffic)* setelah dirutekan dan sesaat sebelum meninggalkan *host*. | POSTROUTING |

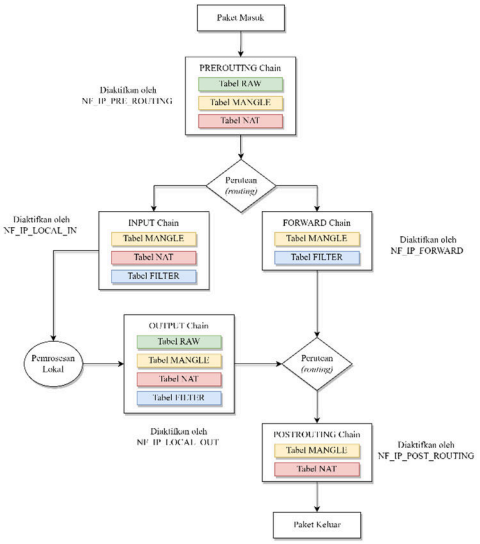
Gambar 3 menunjukkan arsitektur *Netfilter* yang merangkum pemrosesan paket di *Netfilter*.

D. Ipset

Ipset digunakan untuk mengelola sekelompok atau kumpulan *(set)*. Kumpulan tersebut dapat berupa alamat IP, alamat jaringan, alamat MAC, nomor *port,* dan antarmuka jaringan. Ipset dapat digunakan bersama iptables [18].

E. Hping3

*Hping3* merupakan alat jaringan yang dapat mengirim paket TCP/IP secara *custom* dan mampu menampilkan jawaban dari target. Alat ini tersedia di Kali Linux secara *pre-installed* [19]. Beberapa hal yang dapat dilakukan menggunakan *hping3* adalah menguji aturan *firewall,* melakukan *port scanning,* dan menguji performa jaringan. *Hping3* dapat juga digunakan untuk mengirim paket secepat mungkin, yakni dengan menggunakan opsi *flood* [20].




Gambar 3. Arsitektur Netfilter

F. Ksoftirqd

*Ksoftirqd* merupakan suatu utas kernel *(kernel thread)* yang menangani *softirqs* yang tertunda *(pending)*. *Softirqs* atau *software interrupt request* sendiri digunakan untuk menangani penjadwalan suatu proses yang terjadi pada sistem [21]. Sibuknya pemrosesan yang dilakukan oleh kernel dapat terjadi salah satunya ketika antarmuka jaringan dibanjiri oleh paket. Hal ini akan mengaktifkan *softirqs* dengan frekuensi yang sangat tinggi. Ketika terdapat proses-proses yang masih tertunda untuk ditangani, *softirqs* akan mengaktifkan *ksoftirqd* dan membuatnya mengantrikan proses-proses yang masih tertunda tersebut, serta menanganinya seefisien mungkin [22].

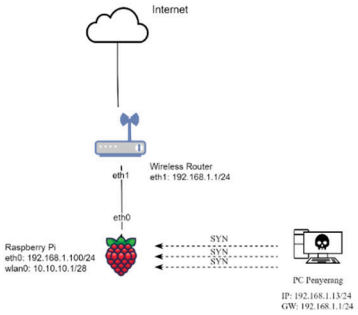
Setiap CPU dalam suatu mesin memiliki *ksoftirqd/n*-nya masing-masing, yang *n* sendiri merupakan angka logis CPU. *Ksoftirqd/0* artinya *ksoftirqd* pada CPU 0. Setiap utas *ksoftirqd/n* mendapatkan bagian untuk menangani *softirqs* yang tertunda, sehingga menyebabkan aktivitas pada penggunaan CPU, baik peningkatan maupun penurunan, tergantung pada jumlah proses yang perlu ditangani [22].

## III. METODOLOGI

Penelitian ini dilakukan dengan mengidentifikasi masalah keamanan yang marak terjadi, melakukan studi literatur terhadap masalah yang ditemukan beserta solusinya, melakukan perancangan terhadap sistem yang akan dibuat sebagai solusi atas permasalahan yang ditemukan, pengaplikasian sistem, pengujian sistem, serta pencatatan dan analisis hasil. Perancangan sistem meliputi perancangan dan pengaplikasian topologi jaringan. Apabila Raspberry Pi sebagai objek yang akan dikenai serangan telah siap digunakan, *firewall* kemudian diterapkan padanya dan diuji dengan serangan *SYN flood*. Kegagalan *firewall* dapat berupa penulisan aturan yang salah maupun koneksi jaringan yang tidak stabil. Lalu pada tahap akhir, dilakukan pencatatan dan analisis terhadap hasil pengujian *firewall.*

A. Perancangan Topologi

Gambar 4 menunjukkan topologi jaringan yang digunakan dalam penelitian. Raspberry Pi terhubung ke WAP rumah menggunakan kabel *ethernet* untuk dapat terhubung ke jaringan internet. Sementara itu, mesin penyerang ditempatkan juga dalam jaringan rumah yang akan membanjiri Raspberry Pi dengan paket SYN.




Gambar 4. Topologi jaringan

B. Pembuatan Aturan Iptables

Aturan iptables yang akan dibandingkan adalah aturan iptables yang mengutilisasi ipset dan aturan iptables yang dibuat oleh AL-Musawi [7] yang mengutilisasi penggunaan *limit* dan *limit burst*. Seluruh *chains* menggunakan *default policy* ACCEPT.

1. Aturan 1: Penggunaan ipset

   Aturan ini berfokus pada penolakan seluruh paket SYN dari perangkat yang tidak didefinisikan dalam *set* WHITELIST.

```
# ipset -N WHITELIST iphash

# ipset -A WHITELIST <alamat IP>

# iptables -N SYNFLOOD

# iptables -t <filter/mangle/raw> -A <INPUT/
  PREROUTING> -m set ! --match-set WHITELIST src
  -p tcp --syn -j SYNFLOOD

# iptables <filter/mangle/raw> -A SYNFLOOD -j
  DROP
```

2. Aturan 2: Penggunaan limitasi (*limit* dan *limit burst*)

   Aturan ini berfokus pada pembatasan paket SYN yang diterima *firewall* Raspberry Pi.

Ini ditunjukkan pada *limit 1/s limit-burst 3,* yang berarti aturan tersebut hanya akan menerima maksimal tiga paket SYN dalam satu detik.

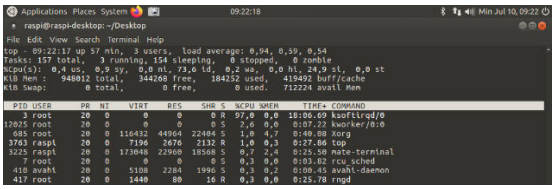
```
# iptables -N SYNFLOOD
# iptables <filter/mangle/raw> -A <INPUT/PREROUTING> -p tcp --syn -j SYNFLOOD
# iptables <filter/mangle/raw> -A SYNFLOOD -m limit --limit 1/s --limit-burst 3 -j RETURN
# iptables <filter/mangle/raw> -A SYNFLOOD -j DROP
```

C. Pengujian Aturan Iptables

Pengujian dilakukan dengan metode *black box testing*. *Black box testing* merupakan metode yang digunakan untuk menguji fungsionalitas tiap aturan yang dibuat. Tiap *chain* diuji satu per satu untuk mengetahui dampaknya terhadap penggunaan CPU Raspberry Pi ketika dikenai *SYN flood*. *SYN flood* dalam penelitian ini dilakukan dengan membanjiri Raspberry Pi dengan paket SYN dari Kali Linux dengan perintah berikut.

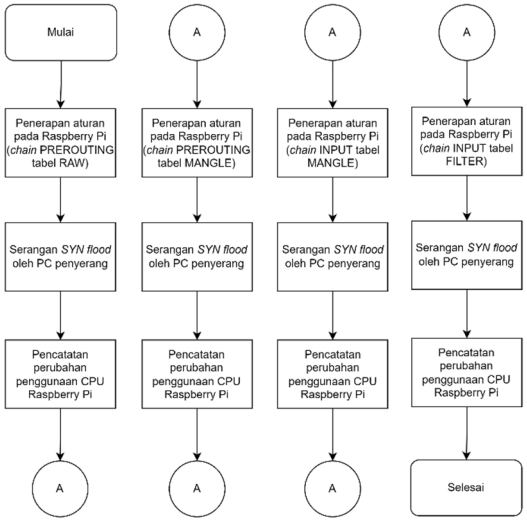
```
hping3 -S --flood -V 192.168.1.100
```

*Ksoftirqd/0* dijadikan sebagai indikator kenaikan penggunaan CPU ketika terjadi *SYN flood.* Gambar 5 menunjukkan contoh penggunaan CPU Raspberry Pi mencapai 97% ketika terjadi serangan. Pengamatan terhadap CPU dilakukan dalam satu menit atau dua puluh perubahan nilai *ksoftirqd/0.*




Gambar 5. Penggunaan CPU Raspberry Pi saat *SYN flood*

Gambar 6 menunjukkan alur pengujian setiap aturan iptables yang dibuat, baik aturan 1 maupun aturan 2. Untuk setiap aturan yang diterapkan, serangan dilakukan selama satu menit. Selama satu menit pula perubahan penggunaan CPU diamati.




Gambar 6. Metode Pengujian Aturan Iptables

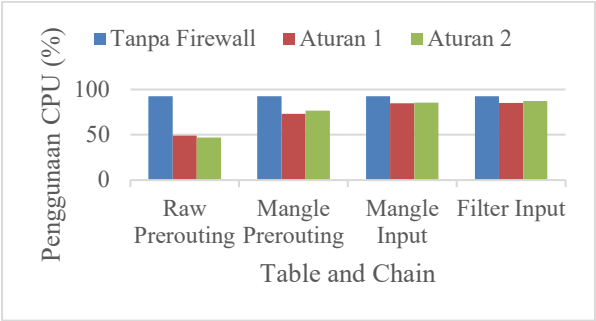
## IV. HASIL DAN PEMBAHASAN

Tabel 2 menunjukkan perbandingan rata-rata penggunaan CPU Raspberry Pi ketika digunakan aturan dan *chain* dari tabel berbeda.

Tabel 2. Perbandingan penggunaan *chain* terhadap penggunaan CPU

| Kondisi Raspberry Pi | Penggunaan CPU (%) |
|---|---|
| Tidak diproteksi *firewall* | 92,41 |
| Diproteksi *firewall*: Aturan 1 | |
| *Chain* PREROUTING tabel RAW | 49,04 |
| *Chain* PREROUTING tabel MANGLE | 72,82 |
| *Chain* INPUT tabel MANGLE | 84,62 |
| *Chain* INPUT tabel FILTER | 84,89 |
| Diproteksi *firewall*: Aturan 2 | |
| *Chain* PREROUTING tabel RAW | 46,81 |
| *Chain* PREROUTING tabel MANGLE | 76,45 |
| *Chain* INPUT tabel MANGLE | 85,34 |
| *Chain* INPUT tabel FILTER | 86,83 |

Sementara itu, Gambar 7 menunjukkan grafik pengaruh penggunaan *chain* terhadap penggunaan CPU Raspberry Pi. Berdasarkan persentase penggunaan CPU tersebut, dapat diamati bahwa tidak terdapat perbedaan yang signifikan maupun drastis pada penggunaan CPU Raspberry Pi ketika diterapkan aturan 1 maupun aturan 2. Selain itu, terlihat juga bahwa *chain* PREROUTING tabel RAW menekan penggunaan CPU paling besar dibanding *chain* dari tabel lainnya, yakni sekitar 43-45% dari penggunaan CPU ketika Raspberry Pi tidak diproteksi *firewall.*



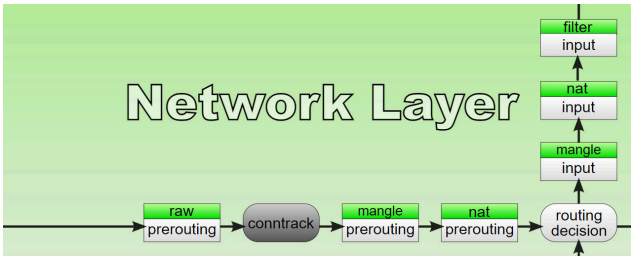

Gambar 7. Grafik pengaruh penggunaan *chain* terhadap penggunaan CPU saat *SYN flood*

Arsitektur *Netfilter* pada Gambar 3 menunjukkan bahwa aturan yang terdapat pada *chain* PREROUTING diproses sebelum dirutekan, tidak seperti *chain* INPUT. Perbedaan ini ternyata membawa dampak terhadap penggunaan CPU Raspberry Pi ketika diterapkan aturan pada salah satu dari *chain* tersebut seperti terlihat pada Tabel 2.

Selain itu, meski sama-sama menggunakan *chain* PREROUTING, penggunaan CPU ketika diterapkan aturan pada tabel RAW dan MANGLE memiliki perbedaan yang cukup besar, yakni 23-30%. Hal ini mengindikasikan adanya perbedaan pemrosesan paket antara dua tabel tersebut. Gambar 8 menunjukkan alur pemrosesan paket pada *Netfilter* lapisan jaringan.

Berdasarkan Gambar 8, terlihat bahwa *chain* PREROUTING tabel RAW diproses sebelum *connection tracking (conntrack),* sementara *chain* PREROUTING tabel MANGLE diproses tepat setelahnya.

*Connection tracking* sendiri digunakan untuk menyimpan informasi terkait koneksi yang masuk. Ini memungkinkan kernel untuk melacak semua koneksi atau alur logis dari koneksi jaringan, sehingga dapat ditangani bersamaan secara konsisten [23].




Gambar 8. Alur pemrosesan paket pada Netfilter di lapisan jaringan
Sumber: diambil dari [24]

Gambar 9 dan Gambar 10 membuktikan bahwa aturan pada *chain* PREROUTING tabel RAW menginspeksi paket terlebih dahulu sebelum *chain* PREROUTING tabel MANGLE.

Pada gambar 9, terlihat sekitar 1.077 ribu paket berukuran total 43 MB ditolak oleh aturan pada *chain* PREROUTING tabel MANGLE. Sejumlah paket ini tampaknya juga melintasi *chain* PREROUTING tabel RAW *(policy ACCEPT 1077k packets, 43M bytes).*

```
root@raspi-desktop:/home/raspi# iptables -t mangle -vL
Chain PREROUTING (policy ACCEPT 184 packets, 10672 bytes)
 pkts bytes target     prot opt in     out     source
1077K   43M SYNFLOOD   tcp  --  any    any     anywhere

Chain INPUT (policy ACCEPT 184 packets, 10672 bytes)
 pkts bytes target     prot opt in     out     source

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source

Chain OUTPUT (policy ACCEPT 191 packets, 13092 bytes)
 pkts bytes target     prot opt in     out     source

Chain POSTROUTING (policy ACCEPT 191 packets, 13092 bytes)
 pkts bytes target     prot opt in     out     source

Chain SYNFLOOD (1 references)
 pkts bytes target     prot opt in     out     source
   71  2840 RETURN     all  --  any    any     anywhere
    5   200 LOG        all  --  any    any     anywhere
1077K   43M DROP       all  --  any    any     anywhere
root@raspi-desktop:/home/raspi# iptables -t raw -vL
Chain PREROUTING (policy ACCEPT 1077K packets, 43M bytes)
 pkts bytes target     prot opt in     out     source
```

Gambar 9. Paket ditolak pada *chain* PREROUTING tabel MANGLE

Sementara itu pada Gambar 10, terlihat sekitar 1.615 ribu paket berukuran total 65 MB ditolak oleh aturan pada *chain* PREROUTING tabel RAW. Kemudian pada *chain* PREROUTING tabel MANGLE, jumlah paket yang melintasi *chain* tersebut berkurang drastis *(policy ACCEPT 179 packets, 10036 bytes).* Hal ini menunjukkan paket telah berhasil ditolak oleh aturan sebelumnya, yang dalam hal ini aturan pada *chain* PREROUTING tabel RAW.

```
root@raspi-desktop:/home/raspi# iptables -t raw -vL
Chain PREROUTING (policy ACCEPT 140 packets, 7727 bytes)
 pkts bytes target     prot opt in     out     source
1615K   65M SYNFLOOD   tcp  --  any    any     anywhere

Chain OUTPUT (policy ACCEPT 138 packets, 8836 bytes)
 pkts bytes target     prot opt in     out     source

Chain SYNFLOOD (1 references)
 pkts bytes target     prot opt in     out     source
   69  2760 RETURN     all  --  any    any     anywhere
    1    40 LOG        all  --  any    any     anywhere
1615K   65M DROP       all  --  any    any     anywhere
root@raspi-desktop:/home/raspi# iptables -t mangle -vL
Chain PREROUTING (policy ACCEPT 179 packets, 10036 bytes)
 pkts bytes target     prot opt in     out     source
```

Gambar 10. Paket ditolak pada *chain* PREROUTING tabel RAW

Berdasarkan alur pemrosesan paket tersebut dan hasil perbandingan penggunaan CPU antara *chain* PREROUTING tabel RAW, *chain* PREROUTING tabel MANGLE, *chain* INPUT tabel MANGLE, dan *chain* INPUT tabel FILTER, dapat dikatakan bahwa *chain* yg digunakan mempengaruhi bagaimana CPU Raspberry Pi dikonsumsi. Semakin cepat paket SYN ini ditangani, maka semakin rendah penggunaan CPU Raspberry Pi ketika terjadi *SYN flood.*

## V. SIMPULAN

Berdasarkan hasil penelitian tersebut, diketahui bahwa penggunaan ipset maupun *limit* dan *limit-burst* pada iptables sama-sama efektif dalam menyaring paket *SYN* ketika terjadi *SYN flood.* Selain itu, *chain* PREROUTING tabel RAW terbukti mampu menekan penggunaan CPU Raspberry Pi paling besar, yakni sekitar 43-45% dari penggunaan CPU ketika Raspberry Pi tidak diproteksi *firewall.* Hal ini dikarenakan *chain* PREROUTING tabel RAW merupakan *chain* yang pertama kali dilintasi paket ketika paket memasuki jaringan. Berikutnya, penggunaan CPU berturut-turut meningkat ketika aturan diterapkan pada *chain* PREROUTING tabel MANGLE, *chain* INPUT tabel MANGLE, dan terakhir *chain* INPUT tabel FILTER yang hanya mengurangi penggunaan CPU sekitar 5-7% dari penggunaan CPU ketika Raspberry Pi tidak diproteksi *firewall.* Hal ini menunjukkan bahwa semakin cepat paket SYN ditangani, maka semakin rendah penggunaan CPU Raspberry Pi ketika terjadi *SYN flood.* Semakin rendah penggunaan CPU, maka semakin kecil pula kemungkinan terjadinya DoS ketika perangkat IoT terserang *SYN flood*.

## REFERENSI

[1] Statista, “IoT connected devices worldwide 2019-2030,” *Statista*, 2022. https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/ (accessed May 22, 2023).

[2] J. Lis, “Smart Home Forecast 2021,” *Insider Intelligence*. https://www.insiderintelligence.com/content/smart-home-forecast-2021 (accessed May 22, 2023).

[3] N. N. Thilakarathne, “Security and privacy issues in IoT Environment,” *Int. J. Eng. Manag. Res.*, vol. 10, 2020.

[4] C. Wheelus and X. Zhu, “IoT Network Security: Threats, Risks, and a Data-Driven Defense Framework,” *IoT*, vol. 1, no. 2, pp. 259–285, 2020.

[5] M. Majkowski, “How to drop 10 million packets per second,” *Cloudflare*, 2018.

[6] D. Melkov, A. Šaltis, and Š. Paulikas, “Performance Testing of Linux Firewalls,” in *2020 IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream)*, IEEE, 2020, pp. 1–4.

[7] B. Q. M. AL-Musawi, “Mitigating DoS/DDoS attacks using iptables,” *Int. J. Eng. Technol.*, vol. 12, no. 3, pp. 101–111, 2012.

[8] R. Karunamoorthi *et al.*, “Design and Development of IoT based Home Computerization using Raspberry Pi,” *Mater. Today Proc.*, 2020.

[9] W. J. McBride and J. R. Courter, “Using Raspberry Pi Microcomputers to Remotely Monitor Birds and Collect Environmental Data,” *Ecol. Inform.*, vol. 54, p. 101016, 2019.

[10] T. A. Ahanger, A. Aldaej, M. Atiquzzaman, I. Ullah, and M. Yousufudin, “Federated Learning-Inspired Technique for Attack Classification in IoT Networks,” *Mathematics*, vol. 10, no. 12, p. 2141, 2022.

[11] D. Nashat and F. A. Hussain, “Multifractal detrended fluctuation analysis based detection for SYN flooding attack,” *Comput. Secur.*, vol. 107, p. 102315, 2021.

[12] Techopedia, "Three-Way Handshake," *Techopedia*, Nov. 10, 2020. https://www.techopedia.com/definition/10339/three-way-handshake (accessed May 22, 2023).

[13] R. Nagai, W. Kurihara, S. Higuchi, and T. Hirotsu, "Design and implementation of an openflow-based tcp syn flood mitigation," in *2018 6th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud)*, IEEE, 2018, pp. 37–42.

[14] D. Kshirsagar, S. Sawant, A. Rathod, and S. Wathore, "CPU load analysis & minimization for TCP SYN flood detection," *Procedia Comput. Sci.*, vol. 85, pp. 626–633, 2016.

[15] P. Likhar and R. S. Yadav, "Impacts of Replace Venerable Iptables and Embrace Nftables in a new futuristic Linux firewall framework," in *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, IEEE, 2021, pp. 1735–1742.

[16] L. Ceragioli, P. Degano, and L. Galletta, "Can my firewall system enforce this policy?," *Comput. Secur.*, vol. 117, p. 102683, 2022.

[17] J. Ellingwood, "A Deep Dive into Iptables and Netfilter Architecture | DigitalOcean," 2015. https://www.digitalocean.com/community/tutorials/a-deep-dive-into-iptables-and-netfilter-architecture (accessed May 22, 2023).

[18] "Man page of IPSET." https://ipset.netfilter.org/ipset.man.html (accessed May 22, 2023).

[19] W. B. W. Mariam and Y. Negash, "Performance evaluation of machine learning algorithms for detection of SYN flood attack," in *2021 IEEE AFRICON*, IEEE, 2021, pp. 1–6.

[20] "hping3(8) - Linux man page." https://linux.die.net/man/8/hping3 (accessed May 22, 2023).

[21] S. Van Rossem, W. Tavernier, D. Colle, M. Pickavet, and P. Demeester, "Vnf performance modelling: From stand-alone to chained topologies," *Comput. Netw.*, vol. 181, p. 107428, 2020.

[22] D. P. Bovet and M. Cesati, *Understanding the Linux Kernel: from I/O ports to process management*. O'Reilly Media, Inc., 2005.

[23] A. Pollitt, "Linux Conntrack: Why it breaks down and avoiding the problem," *Tigera*, Apr. 26, 2019. https://www.tigera.io/blog/when-linux-conntrack-is-no-longer-your-friend/ (accessed May 22, 2023).

[24] J. Engelhardt, *Schematic for the packet flow paths through Linux networking and Xtables*. 2019. Accessed: May 22, 2023. [Online]. Available: https://commons.wikimedia.org/wiki/File:Netfilter-packet-flow.svg.

# *Content Retrieval* dengan *FastText Word Embedding* pada *Learning Management System* Olimpiade

Rochana Prih Hastuti[1,*], Vellya Riona[1], Margareta Hardiyanti[1]
[1]Departemen Teknik Elektro dan Informatika, Sekolah Vokasi, Universitas Gadjah Mada
vellya.riona@mail.ugm.ac.id
margareta.hardiyanti@ugm.ac.id
[*]Korespondensi: rochana.prih.h@ugm.ac.id;

***Abstract – Learning Management System (LMS) is a type of learning media that can be relied upon for students at various levels. Its use for competition purposes, specifically the Olympics, has its own characteristics compared to LMS for common learning purposes. One of them is the ability of the system to manage and retrieve the collection of problem sets which is relevant to users. Users on the Olympic LMS are segmented according to the field of science they want to be engaged in. Even so, each field has different types of learning topics and tends to grow over time. Pre process steps involving topic annotations needs experts and time-consuming. While to make search feature without using metadata information is of course also difficult to do. The semantic search feature becomes important in such a system. A content-based scheme is needed that is able to return problem sets relevant to the topic in each field. The search feature is built using an information retrieval scheme, namely the vector space model. The results of the experiment and the evaluation of the respondents showed that the word embedding representation with the best performance was the FastText word embedding. The efficiency of the model size is carried out by using the compressed version. This representation is not only able to accommodate the results according to the context of the query but also to solve problem of out-of-vocabulary.***

***Keywords – Information Retrieval, Vector Space Model, FastText Word Embedding***

**Intisari – *Learning Management System* (LMS) merupakan jenis media pembelajaran daring yang digunakan siswa di berbagai tingkat. Penggunaannya pada keperluan kompetisi, secara khusus olimpiade, memiliki karakteristik tersendiri dibanding LMS untuk keperluan pembelajaran sehari-hari. Salah satunya adalah kemampuan sistem mengelola bank soal dan menyajikan kategori yang relevan kepada *user*. *User* pada LMS Olimpiade tersegmentasi sesuai bidang ilmu yang ingin ditekuni. Meski begitu, tiap bidang memiliki topik pembelajaran yang beragam jenisnya dan bahkan cenderung berkembang seiring waktu. Manajemen bank soal dengan anotasi topik di awal memerlukan tenaga ahli dan memakan waktu. Sedangkan fitur pencarian tanpa menggunakan informasi metadata tersebut tentu juga sulit dilakukan. Fitur pencarian semantik menjadi penting pada sistem seperti ini. Dibutuhkan skema pencarian berdasarkan konten yang mampu mengembalikan soal-soal yang relevan dengan topik di masing-masing bidang. Fitur pencarian dibangun menggunakan skema *information retrieval* yakni *vector space model*. Hasil eksperimen dan evaluasi responden menunjukkan representasi *word embedding* dengan performa pencarian terbaik adalah FastText *word embedding*. Efisiensi ukuran model dilakukan dengan menggunakan *compressed version*. Representasi ini selain dapat mengakomodasi hasil pencarian sesuai konteks kueri juga dapat mengatasi permasalahan *out-of-vocabulary*.**

**Kata kunci – Information Retrieval, Vector Space Model, FastText Word Embedding**

## I. PENDAHULUAN

Penggunaan *Learning Management System* (LMS) sebagai media pembelajaran daring semakin meningkat di masa pandemi Covid-19. Siswa dari berbagai tingkat dasar, menengah, atas bahkan sampai mahasiswa menjadi sangat bergantung dengan keberadaan LMS. Penelitian tentang membangun *engagement* pada suatu LMS terus dilakukan untuk dijadikan rujukan pembangunan LMS yang lebih baik pada masa mendatang. Misalnya [1] dan [2] keduanya menganalisis bagaimana pengaruh penggunaan LMS dan bagaimana tantangan yang harus dipecahkan untuk membuat LMS yang lebih baik.

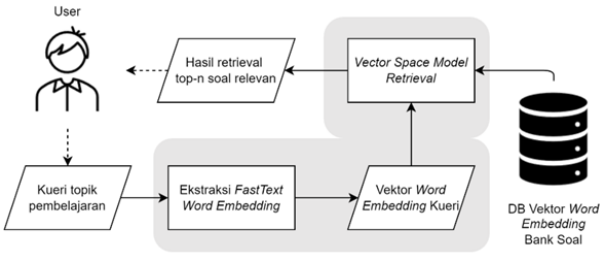
Di sisi lain, sistem serupa lain yang juga bisa dikategorikan sebagai LMS adalah sistem pembelajaran untuk domain kompetisi, misalnya berupa situs-situs https://www.hackerrank.com/ dan https://tlx.toki.id/. Situs-situs ini menyediakan berbagai persoalan untuk keperluan mengasah *skill* di bidang *competitive programming*. Jumlah soal yang disediakan terus bertambah, begitu pula topik bahasannya. Fitur penting yang dibutuhkan pada sistem seperti ini adalah *tagging* kategori soal-soal. Untuk sistem dengan enumerator yang memadai, maka metadata dapat dimanfaatkan untuk membangun indeks yang sesuai. Akan tetapi, untuk sistem lokal yang digunakan di sekolah-sekolah, di mana soal dikumpulkan secara sukarela, maka fitur *tagging* kategori akan sulit dibuat. Begitu pula dari sisi fitur pencarian, maka biasanya akan menggunakan metode *string matching* sederhana sebagai logika utamanya. Sistem informasi dengan menggunakan pendekatan *retrieval* banyak dibuat, misalnya adalah pada domain Al-Quran [3] [4], yang tentu memiliki jumlah data tekstual yang mencukupi. Kebanyakan dari metode yang digunakan pada penelitian tersebut mengandalkan algoritma *stemming* atau menggunakan korpus dan *thesaurus*.

Penggunaan algoritma-algoritma tersebut masih memiliki kekurangan berupa kueri *user* yang terbatas dengan kosakata yang ada pada korpus dan harus seragam. Selain itu, hasil pencarian yang diharapkan tentu adalah yang relevan dengan kueri yang diberikan. Oleh karenanya, pengembangan fitur pencarian dengan pendekatan *content retrieval* akan sangat membantu pada LMS dengan domain kompetisi seperti ini.

Penggunaan representasi fitur *FastText word embedding* akan menghasilkan representasi semantik yang akan menghasilkan hasil pencarian yang memiliki konten yang serupa dengan kueri yang diinputkan [5]. Selain itu, fitur *subword* dari *FastText,* juga dapat membantu mengatasi permasalahan *typo* dan *out-of-vocabulary*. Penggunaan *word embedding* ini didukung dengan adanya *pre-trained* model bahasa Indonesia yang dapat dengan mudah diimplementasikan [6].

## II. METODOLOGI

Modul *content retreival* akan digunakan sebagai mesin utama dari fitur pencarian soal pada *Learning Management System* (LMS) Olimpiade berbasis web, yakni SOLVE. SOLVE memiliki fungsionalitas sebagai bank soal untuk beberapa kategori yang diunggah oleh *user*. Fitur pencarian sebelumnya menggunakan *string matching* digantikan oleh modul *content retrieval* berbasis *FastText word embedding*. Modul *content retrieval* akan dibuat dengan skema *Representational State Transfer Application Programming Interface* (REST API) tersusun dari beberapa metode terlihat pada Gambar 1 pada bagian berbayang abu-abu.




Gambar 1. Ilustrasi *searching* soal pada LMS dengan API *content retrieval*

### A. Dataset

Data yang digunakan berasal dari bank soal aplikasi LMS SOLVE. Terdapat total 515 soal dari 9 mata pelajaran yang digunakan dalam eksperimen penelitian ini terlihat pada Tabel 1. Jumlah ini cukup terbatas dikarenakan aplikasi tersebut masih dalam fase awal rilis, sehingga pengguna yang melakukan penyimpanan soal juga terbatas.

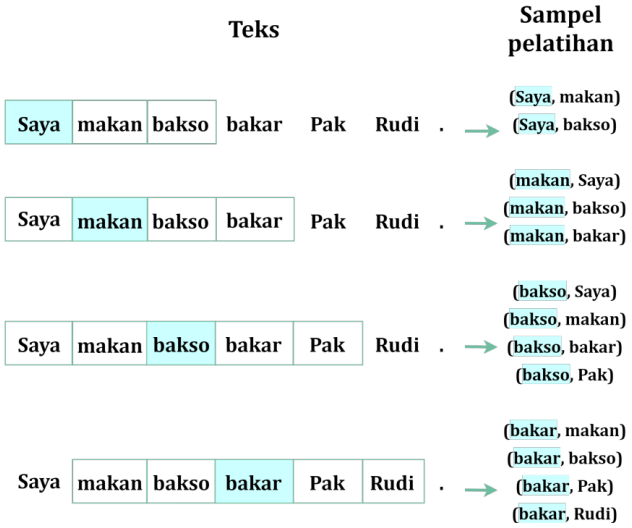
Tabel 1. Jumlah soal tiap mata pelajaran

| No | Mata pelajaran | Jumlah soal |
|---|---|---|
| 1 | Informatika | 125 |
| 2 | Biologi | 87 |
| 3 | Kebumian | 50 |
| 4 | Geografi | 117 |
| 5 | Astronomi | 59 |
| 6 | Ekonomi | 16 |
| 7 | Kimia | 20 |
| 8 | Matematika | 21 |
| 9 | Fisika | 20 |

### B. *FastText Word Embedding*

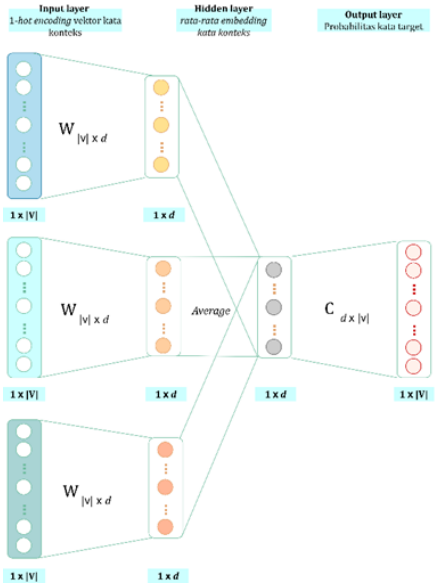
*FastText* merupakan metode ekstraksi fitur dari kata ke dalam bentuk bilangan rill dengan konsep *prediction-based word embedding* [5]. *FastText* adalah pengembangan dari metode continuous *bag-of-words* (CBOW) [7], dengan skema input *one-hot encoding* dari kata konteks dan menghasilkan output *one-hot encoding* dari kata target.

Pengambilan kata konteks dilakukan dengan menggunakan *sliding window* dengan ukuran tertentu. Setiap kata yang berada dalam *window* dari kata target akan menjadi kata konteks. Berikut merupakan contoh *window* dengan ukuran 2 untuk teks "Saya makan bakso bakar Pak Rudi." yang dapat dilihat pada Gambar 2. Kata yang berbayang biru adalah kata target. Dalam satu sampel, sebuah kata target dapat memiliki beberapa kata konteks.




Gambar 2. Kata konteks dari window berukuran 2

Arsitektur CBOW terdiri dari tiga *layer*, yaitu *layer input* dengan neuron sejumlah banyak *unique vocabulary* $\|V\|$ dari data pelatihan, *hidden layer* dengan neuron sejumlah dimensi fitur yang diinginkan $d$, dan *layer output* dengan ukuran $\|V\|$. Layer input akan menerima vektor *one-hot encoding* dari kata konteks, sedangkan *layer output* dilatih agar dapat menghasilkan vektor *one-hot encoding* dari kata target. Setiap kata target dapat memiliki lebih dari satu kata konteks dalam satu sampel. Sehingga, untuk menghitung nilai pada *hidden layer*, dihitung terlebih dahulu rata-rata hasil *layer* sebelumnya dari semua kata konteks dalam satu sampel. Vektor fitur yang akan digunakan dari proses pelatihan CBOW ini adalah nilai hasil neuron pada bagian *hidden layer* CBOW. Ilustrasi ditampilkan pada Gambar 3.



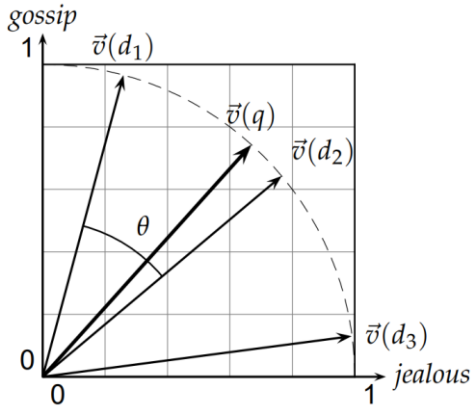

Gambar 3. Arsitektur skema CBOW secara umum
Sumber: berasal dari [8]

Perbedaan *FastText* terletak pada *input* berupa *character n-gram subword*, sehingga ekstraksi fitur dapat dilakukan pada level prefiks dan sufiks di luar data pelatihan atau *out-of-vocabulary* (OOV). Pada penelitian ini digunakan model *pre-trained FastText* Bahasa Indonesia dengan skema CBOW [6].

Akan tetapi, ukuran model *FastText* berkisar antara 3-7 Gb, sehingga sulit untuk diproses dengan spesifikasi RAM pada server biasa. *Compressed FastText* adalah pendekatan untuk mengurangi ukuran model *FastText* dengan beberapa proses, yakni penghilangan *negative-vector* untuk *extend* pelatihan, pengurangan *matrix vocabulary*, dan penentuan ukuran matriks baru yang efisien dengan memperhatikan *collision uniformity* [9]. Ukuran model yang dihasilkan berkurang jauh yakni pada kisaran 15-20 Mb, sehingga lebih *feasible* untuk digunakan pada tahap *deployment* aplikasi.

C. *Vector Space Model*

Representasi sekumpulan dokumen, dalam hal ini bank soal, sebagai vektor pada sebuah ruang vektor disebut dengan *vector space model* [10]. Kueri dapat direpresentasikan sebagai vektor pada ruang vektor yang sama. Misalkan vektor sebuah dokumen $d$ dinotasikan dengan $\vec{V}(d)$, yakni vektor *FastText sentence embedding* dengan dimensi tertentu. Dua vektor, vektor dokumen dan vektor kueri, terlepas dari ukuran *vocabulary*-nya dapat diukur tingkat kemiripannya dengan melihat besarnya sudut yang terbentuk. *Cosine similarity* adalah besarnya sudut yang dibentuk dari kedua vektor tersebut terlihat pada Gambar 4 dapat dihitung dengan rumus (1).




Gambar 4. Ilustrasi *cosine similarity*. $sim(d_1, d_2) = \cos\theta$

Sumber: berasal dari [10]

Sebelum menggunakan besarnya sudut untuk kuantifikasi kemiripan, perlu dihitung *magnitude* dari kedua vektor. Akan tetapi, pendekatan ini lemah jika panjang vektor dari kedua dokumen signifikan berbeda, meskipun konten yang dimiliki mirip. Ini berarti meskipun distribusi relatif kedua dokumen serupa, namun nilai absolut dari frekuensi *vocab* yang dimiliki berbeda jauh. Untuk menormalkan hal ini, cara standar mengkuantifikasi similaritas dari dua dokumen yang direpresentasikan dengan vektor numerik adalah dengan menghitung *cosine similarity*-nya. Efek dari pembagian dengan penyebut pada (1) adalah untuk menormalkan vektor $\vec{V}(d_1)$ dan $\vec{V}(d_2)$ menjadi unit vektor $\vec{v}(d_1) = \vec{V}(d_1)/|\vec{V}(d_1)|$ dan $\vec{v}(d_2) = \vec{V}(d_2)/|\vec{V}(d_2)|$.

$$sim(d_1, d_2) = \frac{\vec{V}(d_1).\vec{V}(d_2)}{|\vec{V}(d_1)||\vec{V}(d_2)|} \quad (1)$$

Sehingga, (1) dapat ditulis ulang menjadi (2), yang merupakan *dot product* dari kedua vektor dokumen yang sudah dinormalkan.

$$sim(d_1, d_2) = \vec{v}(d_1).\vec{v}(d_2) \quad (2)$$

Perhitungan ini adalah *cosine* dari sudut $\theta$ yang diapit kedua vektor pada Gambar 4. Nilai *similarity* ini dapat digunakan jika sistem memiliki sekumpulan dokumen dalam hal ini bank soal, maka kueri yang juga merupakan dokumen $d$ dapat dicari nilai similaritas tertingginya dengan dokumen lain dalam bank soal. Pencarian seperti ini bermanfaat jika *user* mencari tidak hanya satu dokumen, melainkan beberapa dokumen lain dengan nilai similaritas tertinggi. Sehingga, perhitungan *dot product* antara $\vec{v}(d)$ dihitung terhadap setiap dokumen dalam koleksi $\vec{v}(d_1), \ldots, \vec{v}(N)$, kemudian pilih beberapa dokumen dengan nilai similaritas tertinggi sesuai yang dikehendaki.

Tabel 2. Hasil relevansi *precision* terhadap *groundtruth* dokumen

| No | Skema kueri | Compressed FastText | SQL |
|---|---|---|---|
| 1 | 1 kata | 0.32 | 0.28 |
| 2 | 2 kata | 0.28 | 0.16 |
| 3 | 1 kata *typo* | 0.12 | 0 |
| 4 | 2 kata *typo* | 0.11 | 0 |
| 5 | 1 kata OOV | 0.15 | 0 |
| 6 | 2 kata OOV | 0.13 | 0 |

Tabel 3. Hasil relevansi *precision* terhadap penilaian responden

| No | Skema kueri | Compressed FastText | SQL |
|---|---|---|---|
| 1 | 1 kata | 0.5 | 0.32 |
| 2 | 2 kata | 0.48 | 0.17 |
| 3 | 1 kata *typo* | 0.19 | 0 |
| 4 | 2 kata *typo* | 0.20 | 0 |
| 5 | 1 kata OOV | 0.25 | 0 |
| 6 | 2 kata OOV | 0.30 | 0 |

D. Evaluasi

Pengukuran efektivitas sebuah sistem *retrieval* memerlukan beberapa komponen yakni koleksi dokumen, kueri, dan standar pengujian [10]. Fokus utama pengujian adalah relevansi dokumen yang dihasilkan untuk tiap kueri.

1. Pengujian
   Tingkat relevansi pada penelitian ini diukur menggunakan dua jenis acuan, kategori asal dokumen kemudian disebut relevansi *groundtruth* dan pengujian oleh pengguna sistem selanjutnya disebut relevansi responden. Pencarian sistem dilakukan pada seluruh dokumen di bank soal yakni sejumlah 515 dokumen.
2. Kueri
   Seluruh kategori dokumen terdiri dari 9 mata pelajaran akan diuji dengan beberapa kueri bertipe kata dan frasa (dua kata), serta tiga jenis kueri yakni standar, *typo*, dan *out-of-vocabulary* (OOV). Sehingga, total diujikan 12 kueri untuk tiap kategori dokumen. Jumlah dokumen yang dihasilkan 10 buah, menyesuaikan ukuran terkecil dari kategori mata pelajaran yakni 20, agar perhitungan dengan matriks evaluasi tidak rancu karena mengembalikan terlalu banyak dokumen.
3. Matriks evaluasi
   Matriks evaluasi yang digunakan adalah nilai *precision*. Nilai ini digunakan untuk mengukur efektivitas sistem pencarian dalam menampilkan konten paling relevan pada halaman pertama, yang menjadi tipikal pengguna sistem pencarian. *Precision* (P) adalah bagian dari hasil pencarian yang dinilai relevan, dihitung dengan (3).

$$Precision = \frac{\#(relevant\ items\ retrieved)}{\#(retrieved\ items)} \quad (3)$$

## III. HASIL DAN PEMBAHASAN

Hasil *precision* untuk pengujian dengan *groundtruth* dokumen dan penilaian responden berturut-turut ditampilkan pada Tabel 2 dan Tabel 3. Setiap jenis kueri diujikan untuk seluruh 9 mata pelajaran masing-masing dengan dua kueri. Nilai merupakan rata-rata persentase dari total 10 dokumen yang dikembalikan untuk tiap kueri. Tampak pada kedua tabel *Compressed FasText* memiliki nilai *precision* lebih tinggi dibanding kueri menggunakan SQL biasa.

Tabel 4. Kueri dan hasil pencarian kategori Informatika

| No | Skema kueri | Kueri | Compressed FastText |
|---|---|---|---|
| 1 | 1 kata *typo* | mthod | *Method overriding* berbeda dengan *method*… |
| 2 | 2 kata *typo* | kywrd fnal | *Keyword this* dapat digunakan untuk membedakan *variable*… |
| 3 | 1 kata OOV | undefined | *Method Accessor* tidak menggunakan parameter… |
| 4 | 2 kata OOV | switch case | Konstruktor pada suatu *class* boleh menggunakan… |

Contoh hasil pencarian untuk beberapa kueri sampel dari kategori Informatika dapat dilihat pada Tabel 4. Contoh nomor 1 dan 2 adalah dua contoh kata yang mengalami *typo* atau kesalahan penulisan (seharusnya "method" namun ditulis "mthod" pada kueri). Sedangkan contoh nomor 3 dan 4 adalah kueri yang kosakatanya tidak ada pada data.

Pada Tabel 4, terlihat bahwa model representasi vektor dengan *Compressed FasText word embedding*, dapat mengembalikan hasil pencarian dengan konten yang sesuai dengan kueri yang diujikan meskipun menggunakan kueri bertipe *typo* maupun *out-of-vocabulary*. Hal ini sesuai dengan dugaan awal bahwa representasi vektor dengan skema *word embedding* memuat konteks dan dapat mengembalikan hasil pencarian dengan lebih baik. Selain itu, representasi ini juga dapat mengatasi permasalahan *out-of-vocabulary*. Pencarian menggunakan kueri yang sama pada SQL tidak akan memberikan hasil.

## IV. SIMPULAN

Modul *content retrieval* telah dibuat dan diintegrasikan sebagai opsi fitur pencarian di aplikasi LMS Olimpiade SOLVE. Performa model berbasis *FastText word embedding* mengungguli metode awal *string matching* pada dua acuan evaluasi: *automatic* (*groundtruth*) dan *human evaluation* (responden). Penggunaan *Compressed FastText word embedding* mengatasi permasalahan ukuran model, sehingga modul siap untuk digunakan di tahap produksi.

## REFERENSI


[1] S. A. Raza, W. Qazi, K. A. Khan, and J. Salam, "Social isolation and acceptance of the learning management system (LMS) in the time of COVID-19 pandemic: an expansion of the UTAUT model," *J. Educ. Comput. Res.*, vol. 59, no. 2, pp. 183–208, 2021.

[2] U. Alturki and A. Aldraiweesh, "Application of learning management system (Lms) during the covid-19 pandemic: A sustainable acceptance model of the expansion technology approach," *Sustainability*, vol. 13, no. 19, p. 10991, 2021.

[3] I. Humaini, T. Yusnitasari, L. Wulandari, D. Ikasari, and H. Dutt, "Informatian Retrieval of Indonesian Translated version of Al Quran and Hadith Bukhori Muslim," in *2018 International Conference on Sustainable Energy, Electronics, and Computing Systems (SEEMS)*, IEEE, 2018, pp. 1–5.

[4] A. Aulia, D. Khairani, and N. Hakiem, "Development of a retrieval system for Al Hadith in Bahasa (case study: Hadith Bukhari)," in *2017 5th International Conference on Cyber and IT Service Management (CITSM)*, IEEE, 2017, pp. 1–5.

[5] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Trans. Assoc. Comput. Linguist.*, vol. 5, pp. 135–146, 2017.

[6] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, and T. Mikolov, "Learning word vectors for 157 languages," *ArXiv Prepr. ArXiv180206893*, 2018.

[7] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *ArXiv Prepr. ArXiv13013781*, 2013.

[8] R. P. Hastuti, "Q-Learning untuk Pembentukan Tree dalam Sentence Generation Bahasa Indonesia dengan Algoritme Tree Long Short-Term Memory," PhD Thesis, Universitas Gadjah Mada, 2020.

[9] D. Dale, "Compress-fastText." May 06, 2023. Accessed: May 24, 2023. [Online]. Available: https://github.com/avidale/compress-fasttext

[10] H. Schutze, C. D. Manning, and P. Raghavan, *Introduction to information retrieval*. Cambridge University Press, 2008.

# Pengembangan Purwarupa Laboratorium Virtual Berbasis VMWare dengan Terraform

Michael Putra Kusuma[1], Nur Rohman Rosyid[1,*]
[1]Departemen Teknik Elektro dan Informatika, Sekolah Vokasi, Universitas Gadjah Mada;
michael.p.k@mail.ugm.ac.id
[*]Korespondensi: nrohmanr@ugm.ac.id;

***Abstract – In recent years, Virtual Laboratory have increasingly been used by organizations around the world as learning media and a place to practice skills in the field of information technology by providing disposable virtual machines labs that can be created and terminated using buttons found on learning websites. With a Virtual Laboratory prototype that implements Infrastructure as Code, it will make things easier for infrastructure Administrators to manage virtual infrastructure. We need a website based interface that can create virtual machine lab, view the status of the created virtual machine lab and delete the virtual machine lab that have been created if they no longer want to use the virtual machine lab. The development of Virtual Laboratory prototype website uses PHP as server side scripting to run scripts for each of its features. Infrastructure as Code automation inthis prototype functions as an automation of the management and provision of virtual machine lab on VMWare ESXi using Terraform. Before being able to access the website, users who are registered in the database are required to login first so that they can be directed to the folders that have been provided for each user. The hope with this website application is that it can make it easier for laboratory infrastructure Administrators to manage virtual machines labs, so there is no need to bother manually configuring each virtual machine for each user.***

***Keywords – Virtual Labroratory, VMWare ESXi, Terraform, Infrastructure as Code***

**Intisari – Pada tahun-tahun terakhir, Laboratorium Virtual telah semakin marak digunakan oleh organisasi di seluruh dunia sebagai media pembelajaran dan tempat melatih *skill* di bidang teknologi informasi dengan menyediakan *virtual machine lab* yang dapat dibuat dan diberhentikan menggunakan tombol yang terdapat di *website* pembelajaran. Dengan adanya purwarupa Laboratorium Virtual yang menerapkan *Infrasructure as Code*, maka hal tersebut akan memudahkan *Administrator* infrastruktur dalam mengelola infrastruktur virtual. Diperlukannya sebuah antarmuka berbasis *website* yang dapat melakukan pembuatan *virtual machine lab*, melihat status *virtual machine lab* yang dibuat serta menghapus *virtual machine lab* yang telah dibuat jika sudah tidak ingin menggunakan *virtual machine lab* tersebut. Pengembangan purwarupa *website* Laboratorium Virtual ini mengunakan PHP sebagai *server side scripting* untuk menjalankan *script* dari tiap fiturnya. *Infratruktur as Code* pada purwarupa ini berfungsi sebagai otomatisasi pengelolaan dan penyediaan *virtual machine lab* pada VMWare ESXi menggunakan Terraform. Sebelum dapat mengakses *website* tersebut, *user* yang terdaftar di *database* diharuskan untuk *login* terlebih dahulu sehingga dapat diarahkan ke *folder* yang telah disediakan untuk masing-masing *user*. Harapan dengan adanya aplikasi *website* ini adalah dapat memudahkan *Administrator* infrastruktur laboratorium dalam melakukan manajemen *virtual machine lab*, sehingga tidak perlu repot-repot lagi untuk melakukan konfigurasi tiap *virtual machine* secara manual bagi tiap *user*.**

**Kata kunci – Laboratorium Virtual, VMWare ESXi, Terraform, *Infrastructure as Code***

## I. PENDAHULUAN

Pada era revolusi industri 4.0 ini, perkembangan teknologi yang digunakan oleh umat manusia terus mengalami kemajuan dan perkembangan yang sangat cepat dan pesat terutama di bidang infrastruktur dan virtualisasi. Akibat dari hal tersebut adalah semakin banyaknya penggunaan infrastruktur virtual pada organisasi di seluruh dunia. Salah satu contoh penerapan infrastruktur virtual adalah Laboratorium Virtual. Organisasi di dunia akhir–akhir ini banyak yang menggunakan Laboratorium Virtual pada *website* pembelajaran mereka sebagai salah satu media pembelajaran untuk melatih keterampilan penggunanya, khususnya dalam di bidang teknologi informasi [1]. Dalam bidang virtualisasi, VMWare ESXi merupakan produk *Hypervisor* gratis milik VMWare yang dapat digunakan untuk men-*deploy Virtual Machines* (VM) dalam rangka menciptakan infrastruktur virtual untuk organisasi [2]. Sebelum Laboratorium Virtual terkenal, *Administrator* infrastruktur laboratorium secara manual dan berulang mengelola infrastruktur virtual. *Administrator* membuat VM baru, mengkonfigurasi VM, menyalakan VM, memberi IP *Address* untuk VM, dan menghapus VM yang sudah tidak digunakan. Semakin bertambahnya VM yang terdapat pada *Hypervisor,* maka akan semakin bertambah pula waktu dan beban kerja yang dimiliki oleh *Administrator* infrastruktur, bahkan hal tersebut berpotensi menimbulkan terjadinya *human error*. Maka dari itu, diperlukan pengelolaan infrastruktur virtual berbasis kode atau yang biasa disebut *Infrastructure as Code* (IaC) [3]. IaC sangat cocok untuk Laboratorium Virtual yang melakukan hal-hal berulang dalam mengelola VM, IaC juga sangat penting karena dapat mengurangi waktu yang dihabiskan untuk melakukan hal–hal berulang tersebut, serta dapat memastikan konfigurasi VM yang akan dibuat, sehingga dapat mengurangi terjadinya *human error* [4].

Untuk mempermudah *Administrator* infrastruktur laboratorium dalam mengelola Laboratorium Virtual, pada VMWare ESXi. dibuatlah IaC dengan menggunakan Terraform. Selain IaC, diperlukan sebuah *website* sebagai antarmuka untuk pengguna laboratorium virtual sebagai aplikasi untuk membuat VM, menampilkan status berupa informasi dari VM yang dibuat pada antarmuka *website* serta menghapus VM tersebut jika sudah tidak digunakan. Sebelum dapat melakukan hal tersebut pada *website*, diperlukan fitur keamanan berupa halaman *login* yang akan mengarahkan penggunanya ke halamannya masing-masing, sehingga pengguna tidak dapat mengakses dan menghapus VM milik pengguna lainnya. Mengelola infrastruktur virtual akan semakin sulit dilakukan seiring bertambahnya jumlah pengguna. Semakin banyak VM yang dibuat, maka akan memakan waktu yang banyak pula untuk mencari satu persatu VM yang digunakan oleh pengguna beserta IP *Address*-nya.

Diharapkan dengan adanya purwarupa Laboratorium Virtual ini akan dapat membantu *Administrator* infrastruktur laboratorium agar tidak perlu lagi secara manual melakukan pembuatan VM, mengkonfigurasi IP *Address* ke VM, menginfokan IP *Address* ke setiap pengguna, mengkonfigurasi VM, menyalakan VM, dan menghapus VM yang sudah tidak digunakan, karena hal tersebut dapat dilakukan secara berulang menggunakan otomatisasi dari kode, sehingga lebih hemat waktu dan efisien dalam mengelola infrastruktur virtual.

## II. DASAR TEORI

### A. Laboratorium Virtual

Laboratorium Virtual merupakan sistem yang dapat digunakan untuk mendukung praktikum yang berjalan secara konvensional. Laboratorium Virtual dapat menjadi media pembelajaran yang digunakan sebagai solusi keterbatasan perangkat komputer [1].

Salah satu penerapan Laboratorium Virtual dilakukan oleh [5] mengenai "*The Ball and Beam System: A Case Study of Virtual and Remote Lab Enhancement With Moodle"*. Penelitian tersebut mendemonstrasikan manfaat dan pentingnya Laboratorium Virtual pada pendidikan jarak jauh, serta menunjukan dampak positif dari *Virtual* dan/atau *Remote Lab* (VRL) dan penerapan *Moodle* terhadap pembelajaran siswa.

### B. VMWare ESXi

*Hypervisor* merupakan lapisan virtualisasi yang berfungsi sebagai fondasi untuk lini produk lainnya. Di vSphere versi 5 dan yang lebih baru, *Hypervisor* dari produk vSphere hanya hadir dalam bentuk VMWare ESXi. VMWare ESXi adalah *Type* 1 *bare-metal Hypervisor* yang berjalan langsung pada perangkat keras sistem [2].

Salah satu penerapan VMWare ESXi dilakukan oleh [6] mengenai "*Virtualization and cyber security: arming future security practitioners"*. Penelitian tersebut menunjukan bahwa VMWare ESXi keluar sebagai pemenang dari ketiga *platform* virtualisasi yang telah dipelajari (VMWare ESXi, Xenserver dan Hyper-V) setelah menjalankan lingkungan lab virtual untuk mengajar kursus keamanan di tingkat universitas. Dalam lingkungan lab virtual, fleksibilitas adalah kualitas penting karena perlu menyiapkan banyak skenario dengan parameter yang bervariasi. Konfigurasi dan opsi implementasi di VMWare adalah yang paling banyak intuitif, opsi manajemen pengguna lebih terperinci dan komprehensif, bahkan saat menjalankan ESXi sebagai *server* mandiri pada saat sistem lain membutuhkan konfigurasi yang jauh lebih kompleks untuk mencapai kontrol akses granular.

### C. *Infrastructure as Code*

*Infrastructure as Code* (*IaC*) adalah pendekatan otomatisasi infrastruktur berdasarkan praktik dari pengembangan perangkat lunak. IaC menekankan rutinitas yang konsisten dan berulang untuk penyediaan (*provisioning*) dan perubahan sistem beserta konfigurasinya. *Administrator* membuat perubahan pada kode, lalu menggunakan otomatisasi untuk menguji dan menerapkan perubahan tersebut ke sistem yang dimiliki [3].

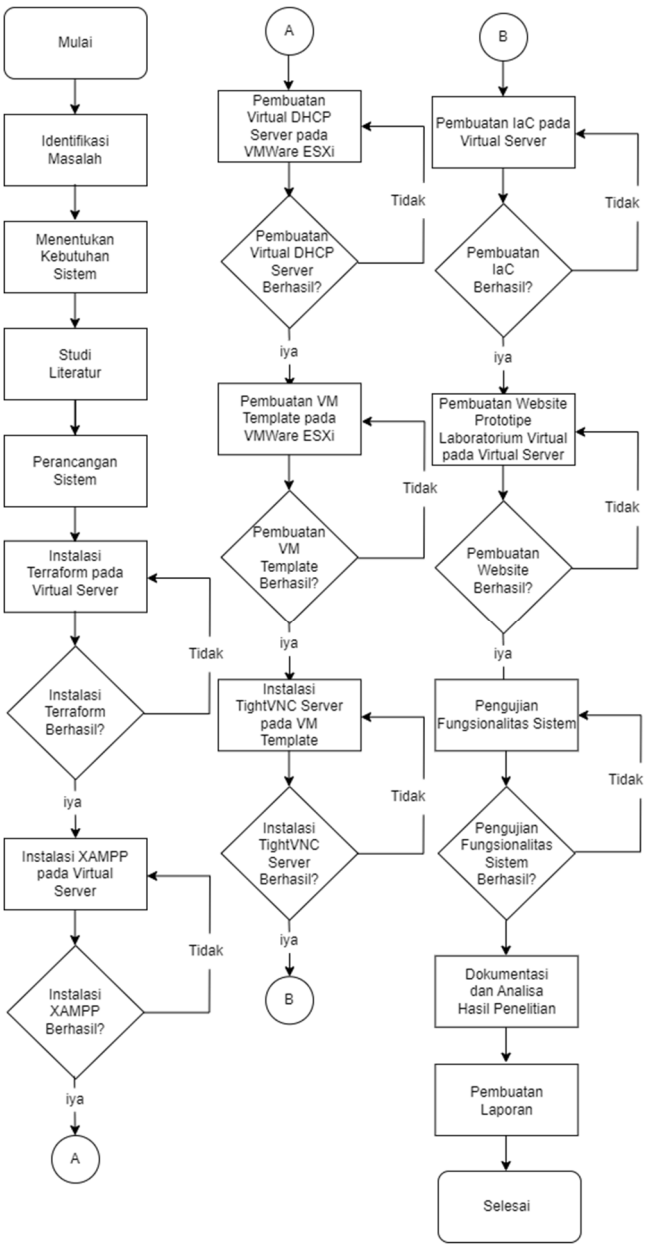
Terraform adalah *tool* IaC yang memungkinkan penggunanya untuk menentukan sumber daya *cloud* dan lokal dalam *file* konfigurasi yang dapat dibaca manusia, dapat dibuat versi, digunakan kembali, dan dibagikan. Pengguna kemudian dapat menggunakan alur kerja yang konsisten untuk menyediakan dan mengelola semua infrastruktur pengguna sepanjang siklus hidupnya. Terraform dapat mengelola komponen tingkat rendah, seperti komputasi, penyimpanan, dan sumber daya jaringan [7].

Salah satu penerapan IaC dilakukan oleh [8] mengenai "*Infrastructure as Code for Business Continuity in Institutions of Higher Learning"*. Penelitian tersebut mendapatkan kesimpulan bahwa IaC memiliki manfaat yang sangat besar seperti penghematan waktu, konfigurasi awal dan pembaruan konfigurasi, investasi awal pada perangkat keras, infrastruktur yang andal dan terukur, serta *Recovery Time Objective* (RTO) yang lebih baik.

## III. METODOLOGI

### A. Tahap Penelitian

Tahapan penelitian dimulai dari identifikasi masalah, menentukan kebutuhan sistem, studi literatur, perencanaan sistem, hingga instalasi Terraform pada Virtual Server. Tahapan penelitian secara jelas disajikan pada Gambar 1 Diagram Alir Penelitian.




Gambar 1. Diagram Alir Penelitian

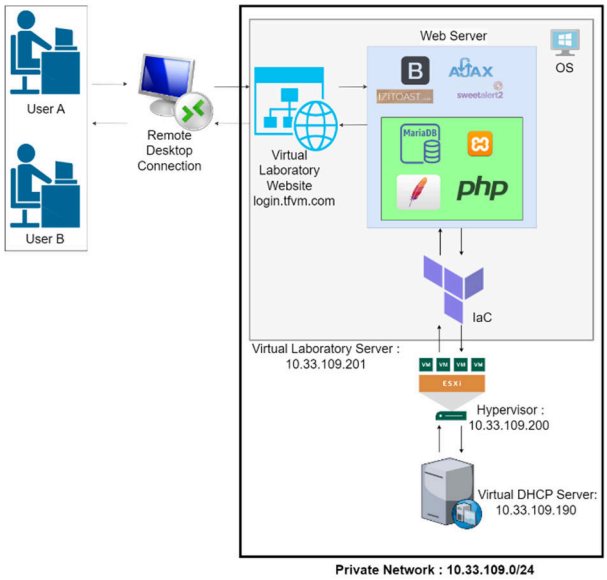
B. Alat dan Bahan

Terdapat beberapa bahan perangkat lunak dan perangkat keras yang diperlukan untuk menjalankan Proyek Akhir ini. Bahan-bahan yang digunakan untuk Proyek Akhir ini adalah sebagai berikut:

1) Perangkat Keras
- Laptop
- VMWare ESXi
- Server purwarupa Laboratorium Virtual

2) Perangkat Lunak
- Sistem Operasi Windows 10 Home
- Sistem Operasi Kali Linux 2022.2
- Sistem Operasi CentOS 7
- Terraform 1.2.2
- VMWare Tools
- VMWare ESXi 7.0 U2
- XAMPP 7.2.34
- Terraform *provider* esxi *plugin* 1.10.2
- OVF Tool 4.4.3
- GO 1.18.3
- TightVNC (1.3.10 Kali Linux, 2.8.63 Windows)
- OpenVPN Connect 3.3.5
- Remote Desktop Connection
- jQuery 3.6.0
- Izitoast 1.4.0
- Sweetalertt2 11.4.23
- Bootstrap 5.2
- OpenVPN Connect 3.3.5
- Dhcpd
- Wireshark
- Nmap

C. Perancangan Sistem

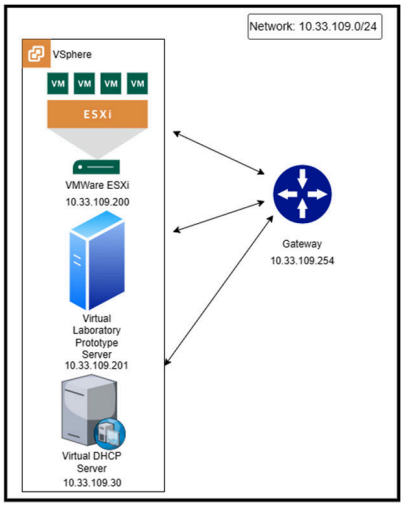



Gambar 2. Diagram Sistem Arsitektur Purwarupa

Alir kerja sistem purwarupa Laboratorium Virtual dapat dilihat pada Gambar 2. Proses dimulai dari *user* melakukan *remote access* ke *server* dari purwarupa Laboratorium Virtual menggunakan *Remote Desktop Connection* [9], aplikasi bawaan Windows untuk melakukan RDP. Pada *server* purwarupa Laboratorium Virtual inilah *web server* dari *website* purwarupa Laboratorium Virtual yang menggunakan XAMPP [10]. Hal yang dilakukan saat sudah dapat mengakses halaman *website* purwarupa Laboratorium Virtual adalah melakukan *login* dengan memasukkan *username* dan *password* agar *user* terverifikasi untuk dapat mengakses halaman utama dari *website* purwarupa Laboratorium Virtual milik setiap *user*. *Website* kemudian menggunakan Ajax [11] untuk melakukan verifikasi terhadap *user*name dan *password* yang telah dimasukkan oleh *user* dengan data akun yang sudah ada pada *database* MariaDB [12]. Izitoast dan Sweetalert2 berfungsi sebagai notifikasi dan *alert* saat proses memverifikasi *user* [13], *website* akan menampilkan *alert* jika user salah memasukkan *username* atau *password*,

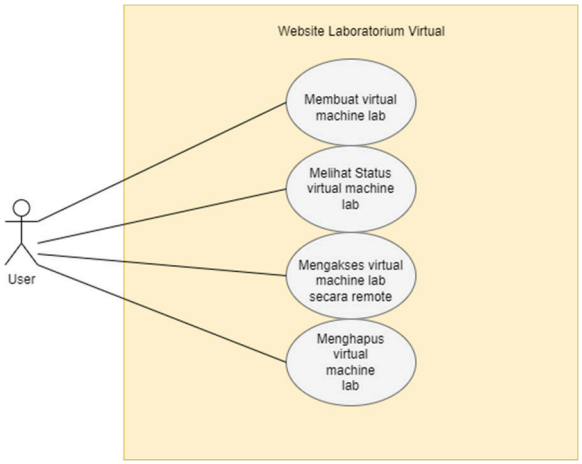
tidak memasukkan *username* atau *password* dan menampilkan notifikasi berhasil *login* jika *username* dan *password* yang dimasukkan benar. Setelah berhasil *login*, *user* dapat memilih salah satu dari empat modul purwarupa Laboratorium Virtual sebagai pilihan yang tersedia pada *submenu* dari menu *Deploy* yang terdapat di *navigation bar*. Keempat modul tersebut adalah "1 Linux", "2 Linux", "1 Windows", dan "2 Windows". Setiap modul tersebut berisi *virtual machine lab* sesuai dengan namanya. *Navigation bar* tersebut dibuat dengan menggunakan Bootstrap [14].

Pada setiap halaman dari keempat modul purwarupa Laboratorium Virtual terdapat tiga tombol yang tersedia untuk *user*. Saat tombol ditekan oleh *user*, PHP [15] akan mengeksekusi perintah Terraform [7] pada *Command Prompt* di *background* dari *server* purwarupa Laboratorium Virtual untuk menjalankan Terraform sesuai dengan fungsi Terraform yang ada di setiap tombolnya. Pertama, *user* menekan tombol "Start VM" untuk membuat *virtual machine lab*, saat tombol "Start VM" tombol ditekan VMWare ESXi akan membuat *virtual machine lab* menggunakan VM *template* yang telah tersedia. *Virtual machine lab* tersebut kemudian akan mendapatkan IP *Address* dari *Virtual DHCP Server* [16], setelah berhasil dibuat. Setelah *virtual machine lab* berhasil di-*deploy* Terraform akan mengeluarkan *output* pada kotak *output* di bawah tombol "Start VM" berisi durasi waktu untuk membuat *virtual machine lab* tersebut dan *output* berupa status dari *virtual machine lab*. Setelah itu, *user* menekan tombol "VM Resource" untuk melihat status dari *virtual machine lab* yang telah di-*deploy*, kemudian *user* dapat mengakses *virtual machine lab* tersebut secara *remote* lewat aplikasi TightVNC Viewer [17]. Terakhir, *user* menekan tombol "Stop VM" untuk menghapus *virtual machine lab* yang telah digunakan dari VMWare ESXi. *Virtual machine lab* yang telah di-*deploy* ini adalah hasil *clone* oleh Terraform dari VM *Template* yang telah dibuat pada VMWare ESXi. Terdapat dua VM *Template* pada penelitian ini, *Linux VM Template* dengan OS Kali Linux dan *Windows VM Template* dengan OS Windows 10 Home. *User* dapat keluar dari akun yang digunakan dengan menekan menu *Logout* pada *navigation bar*.




Gambar 3. Topologi Jaringan Sistem

Topologi jaringan sistem dapat dilihat pada Gambar 3. Pada rancangan sistem terdapat satu buah *server* untuk *website* purwarupa Laboratorium Virtual, satu buah VMWare ESXi, dan satu buah *Virtual DHCP Server. Server* purwarupa Laboratorium Virtual, *Virtual DHCP Server,* dan VMWare ESXi tersebut berada dalam jaringan yang sama yaitu "10.33.109.0/24" dengan *gateway* "10.33.109.254".




Gambar 4. *User Case Diagram*

Dari pembahasan *website* purwarupa Laboratorium Virtual di atas, *user* memiliki akses untuk dapat membuat, melihat *status,* mengakses secara *remote* dan menghapus *virtual machine lab*. Seperti *user case diagram* pada Gambar 4, tidak ada *role user* secara khusus dan setiap *user* dapat menjalankan empat fitur utama yang ada pada *website* purwarupa Laboratorium Virtual.

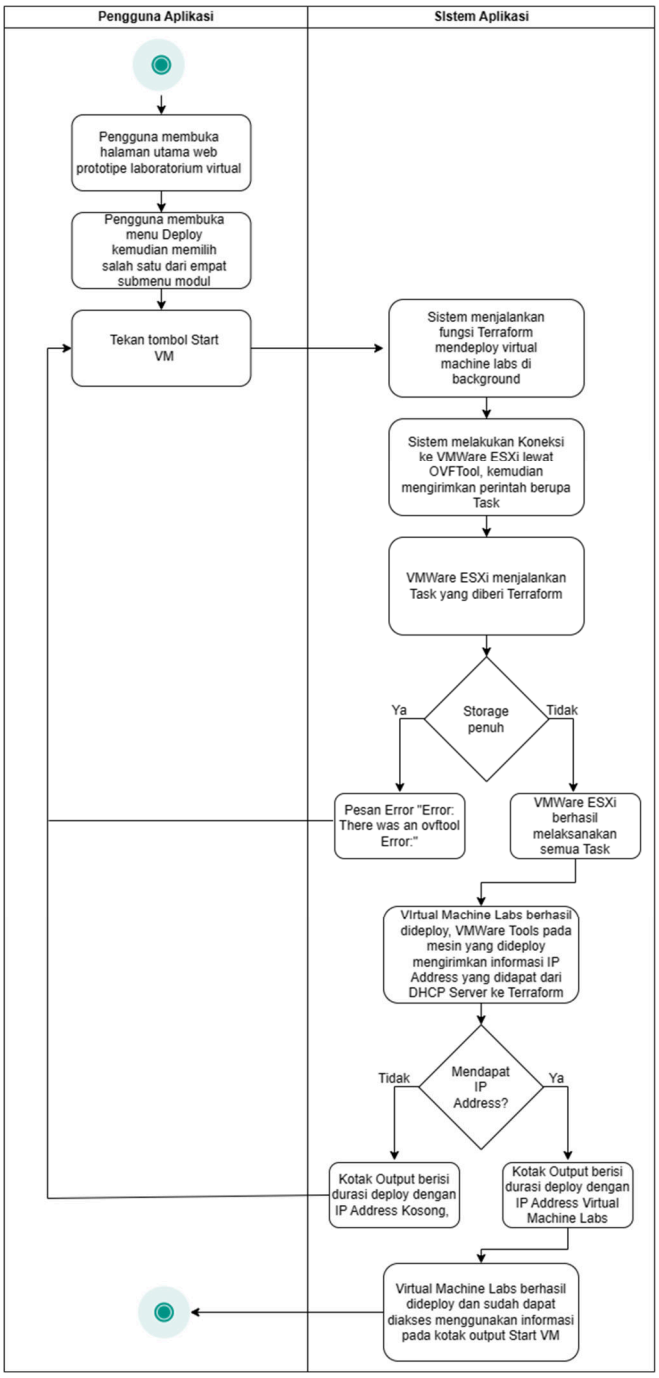
IaC pada penelitian ini digunakan untuk otomatisasi pengelolaan dan penyediaan *virtual machine lab*. *Tools* IaC yang digunakan pada penelitian ini adalah Terraform. Terraform memiliki tiga fungsi utama pada penelitian ini sebagai IaC [7]. Fungsi tersebut adalah *deploy virtual machine lab* menggunakan *command* "terraform ini" dan "terraform *apply -auto-approve*", melihat status *virtual machine lab* menggunakan *command* "terraform *output*" dan menghapus *virtual machine lab* menggunakan *command* "terraform *destroy -auto-approve*". *Provider* yang digunakan pada penelitian ini adalah terraform provider esxi [18].

D. Skenario Pengujian

Pengujian yang dilakukan pada penelitian ini menggunakan metode *Black Box Testing*. Pengujian ini dilakukan sebagai pengujian fungsionalitas setiap fitur pada sistem purwarupa Laboratorium Virtual yang telah dikembangkan. Proses pengujian sistem aplikasi dengan metode *Black Box Testing* tidak terlalu perlu mengetahui proses internal aplikasi dan kode program secara detail, tetapi cukup diketahui proses *testing* dari sisi bagian luar saja [19]. Metode tersebut akan digunakan untuk menguji fungsionalitas ketujuh fitur yang terdapat pada purwarupa Laboratorium Virtual.

1) Pengujian Fitur *Start VM*

Fitur *Start VM* berfungsi untuk membuat *virtual machine lab* dengan menjalankan fungsi *deploy* Terraform, PHP akan mengeksekusi *script* untuk menjalankan fungsi *deploy* Terraform di *background* saat tombol "Start VM" ditekan oleh *user*. Setelah *virtual machine lab* berhasil di-*deploy,* maka *output* fungsi Terraform yang berjalan di *background* akan ditampilkan oleh PHP di kotak *output* "Start VM". Proses fitur *Start VM* dapat dilihat pada Gambar 5.

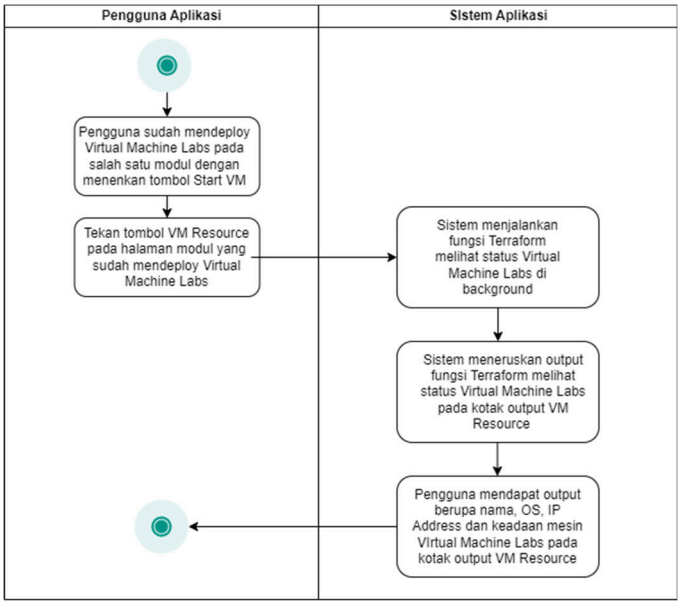



Gambar 5. *Activity Diagram* Fitur *Start VM*

Pengujian fitur ini bertujuan untuk memastikan sistem purwarupa Laboratorium Virtual dapat membuat *virtual machine lab* sesuai dengan modulnya dan *website* purwarupa Laboratorium Virtual dapat mengeluarkan *output* dari fungsi Terraform yang dijalankan pada kotak *output* di bawah tombol "Start VM".

1) Pengujian Fitur *VM Resource*

Fitur *VM Resource* berfungsi untuk menampilkan *resource* berupa status dari *virtual machine lab* yang telah di-*deploy* pada halaman modul tersebut. PHP akan menjalankan fungsi Terraform melihat status *virtual machine lab* setelah tombol "VM Resource*"* ditekan oleh *user*. Setelah fungsi Terraform melihat status berhasil dieksekusi, maka *output* fungsi Terraform yang berjalan di *background* akan ditampilkan oleh PHP di kotak *output* "VM Resource*"*. Proses fitur *VM Resource* dapat dilihat pada Gambar 6.

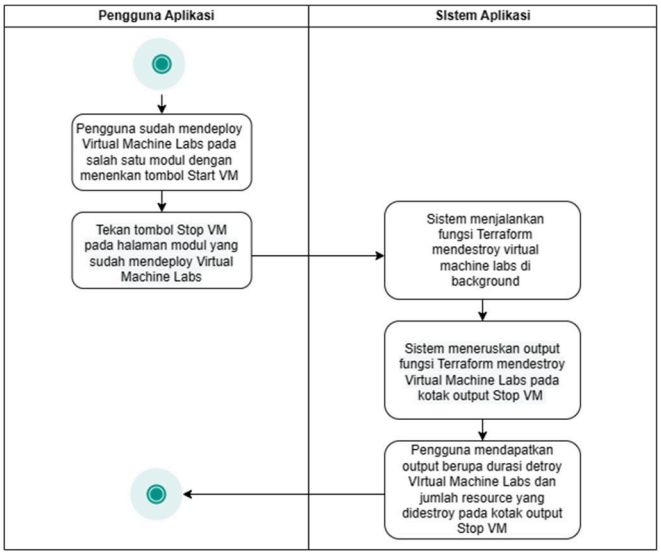



Gambar 6. *Activity Diagram* Fitur *VM Resource*

Pengujian fitur ini bertujuan untuk memastikan sistem purwarupa Laboratorium Virtual dapat menampilkan informasi status dari *virtual machine lab* yang sebelumnya telah di-*deploy* pada kotak *output* dibawah tombol "VM *Resource*".

2) Pengujian Fitur *Stop VM*

Fitur *Stop VM* berfungsi untuk menghapus *virtual machine lab* yang telah di-*deploy* pada halaman modul. Fitur ini hanya berlaku untuk modul pada halaman modulnya sendiri, sehingga fitur ini tidak dapat men-*destroy resource* milik halaman modul lain, apalagi men-*destroy* milik *user* lain. PHP akan menjalankan fungsi Terraform menghapus *virtual machine lab* setelah tombol "Stop VM" ditekan oleh *user*. Setelah fungsi Terraform menghapus *virtual machine lab* berhasil dieksekusi, *output* fungsi Terraform yang berjalan di *background* akan ditampilkan oleh PHP di kotak *output* "Stop VM". Proses fitur *Stop VM* dapat dilihat pada Gambar 7.

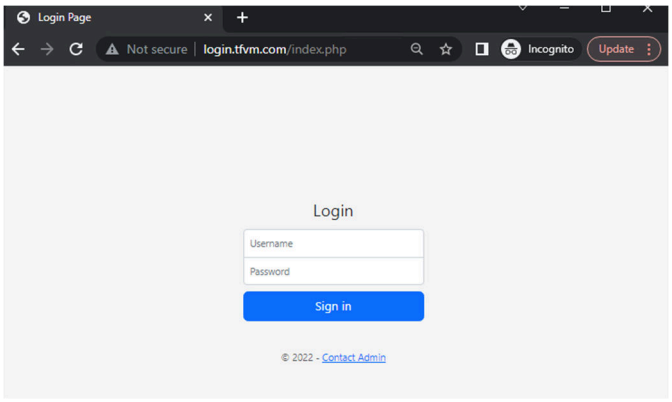



Gambar 7. *Activity Diagram* Fitur *Stop VM*

Pengujian fitur ini bertujuan untuk memastikan bahwa sistem purwarupa Laboratorium Virtual dapat menghapus *virtual machine lab* yang telah di-*deploy* dan *website* purwarupa Laboratorium Virtual dapat mengeluarkan *output* dari fungsi Terraform yang dijalankan pada kotak *output* di bawah tombol "Stop VM".

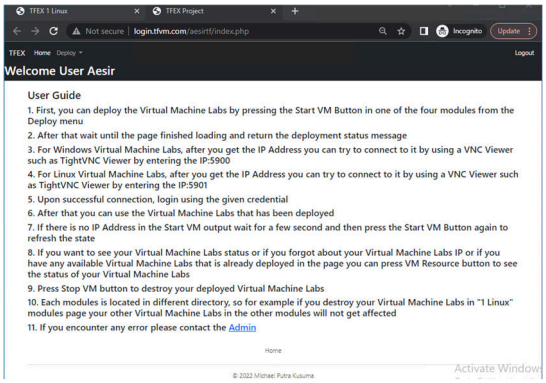
## IV. HASIL DAN PEMBAHASAN

### A. Tampilan Aplikasi



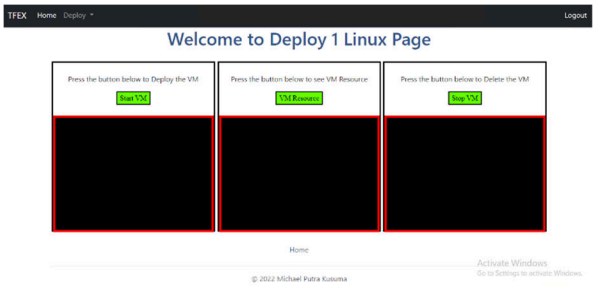

Gambar 8. Tampilan Halaman *Login*

Pengembangan antarmuka *website* purwarupa Laboratorium Virtual ini dibuat menggunakan Ajax, Bootstrap, Sweeralert2 dan iziToast, serta PHP dan MariaDB untuk *backend.* Pada Gambar 8, halaman utama *website* purwarupa Laboratorium Virtual terdiri dari *field username* dan *password* untuk memasukkan krendensial milik *user*.




Gambar 9. Tampilan Halaman Utama

Pada Gambar 9, halaman utama *website* purwarupa Laboratorium Virtual terdapat tiga menu yaitu "*Home*", "*Deploy*" dan "*Logout*", serta *user guide* dari purwarupa Laboratorium Virtual. Menu "*Home*" berfungsi untuk mengembalikan *user* ke halaman utama. Menu "*Deploy*" terdiri dari empat sub-menu berisi halaman modul purwarupa Laboratorium Virtual, keempat modul tersebut yaitu modul "1 Linux" untuk membuat satu *virtual machine lab* Linux, modul "2 Linux" untuk membuat 2 *virtual machine lab* Linux, modul "1 Windows" untuk membuat satu *virtual machine lab* Windows dan modul "2 Windows" untuk membuat 2 *virtual machine lab* Windows. Menu "*Logout*" berfungsi untuk menjalankan fitur *Logout*. *User guide* pada halaman utama merupakan panduan kepada *user* tentang purwarupa Laboratorium Virtual dan berisi cara *user* untuk dapat mengakses *virtual machine lab*.




Gambar 10. Tampilan Halaman Modul

Pada Gambar 10, halaman modul pada *menu* "*Deploy*" memiliki tampilan yang sama, kecuali bagian judul halaman di tengah. Pada halaman modul terdapat tiga fitur dengan tombol dan kotak *output* dibawahnya. Ketiga fitur tersebut adalah fitur *Start VM*, fitur S*top VM* dan fitur *VM Resource*.

### B. Hasil *Black Box Testing*

Seluruh proses pengujian dilakukan dengan membuat tabel berisikan skenario dan kasus pengujian yang akan digunakan sebagai acuan pengujian dari sistem purwarupa Laboratorium Virtual. Hasil *Black Box Testing* dapat dilihat pada Tabel 1.

Tabel 1. Hasil *Black Box Testing*

| No | Skenario | Kasus Pengujian | Hasil |
|---|---|---|---|
| 1. | Fitur *Start VM* | User aesir modul 1 Linux | Berhasil |
| | | User aesir modul 1 Windows | Berhasil |
| | | User aesir modul 2 Linux | Berhasil |
| | | User aesir modul 2 Windows | Berhasil |
| | | User busir modul 1 Linux | Berhasil |
| | | User busir modul 1 Windows | Berhasil |
| | | User busir modul 2 Linux | Berhasil |
| | | User busir modul 2 Windows | Berhasil |
| | | User aesir modul 1 Linux dan User busir modul 1 Windows secara bersamaan | Berhasil |
| | | User aesir modul 1 Windows dan User busir modul 1 Windows secara bersamaan | Berhasil |
| | | User aesir modul 1 Linux dan User busir modul 1 Linux secara bersamaan | Berhasil |
| | | User aesir modul 1 Windows dan User busir modul 1 Linux secara bersamaan | Berhasil |
| 2. | Fitur *VM Resource* | User aesir modul 1 Linux | Berhasil |
| | | User aesir modul 1 Windows | Berhasil |
| | | User aesir modul 2 Linux | Berhasil |
| | | User aesir modul 2 Windows | Berhasil |
| | | User busir modul 1 Linux | Berhasil |
| | | User busir modul 1 Windows | Berhasil |
| | | User busir modul 2 Linux | Berhasil |
| | | User busir modul 2 Windows | Berhasil |
| | | User aesir modul 1 Linux dan User busir modul 1 Windows secara bersamaan | Berhasil |
| | | User aesir modul 1 Windows dan User busir modul 1 Windows secara bersamaan | Berhasil |
| | | User aesir modul 1 Linux dan User busir modul 1 Linux secara bersamaan | Berhasil |
| | | User aesir modul 1 Windows dan User busir modul 1 Linux secara bersamaan | Berhasil |
| 3. | Fitur *Stop VM* | User aesir modul 1 Linux | Berhasil |
| | | User aesir modul 1 Windows | Berhasil |
| | | User aesir modul 2 Linux | Berhasil |
| | | User aesir modul 2 Windows | Berhasil |
| | | User busir modul 1 Linux | Berhasil |
| | | User busir modul 1 Windows | Berhasil |
| | | User busir modul 2 Linux | Berhasil |
| | | User busir modul 2 Windows | Berhasil |
| | | User aesir modul 1 Linux dan User busir modul 1 Windows secara bersamaan | Berhasil |
| | | User aesir modul 1 Windows dan User busir modul 1 Windows secara bersamaan | Berhasil |
| | | User aesir modul 1 Linux dan User busir modul 1 Linux secara bersamaan | Berhasil |
| | | User aesir modul 1 Windows dan User busir modul 1 Linux secara bersamaan | Berhasil |

Dari hasil yang didapat pada Tabel 1, dapat disimpulkan bahwa pengujian seluruh fitur *website* purwarupa tersebut berhasil, karena seluruh fitur berfungsi dengan baik dan mendapatkan hasil sesuai dengan yang diharapkan untuk pengujian setiap kasusnya. Pada skenario fitur *Login*, *user* berhasil mendapatkan *pop-up* berupa notifikasi atau *alert* sesuai dengan rancangan pada setiap kasusnya. Pada skenario fitur *Redirect Unverified User*, *user* berhasil dialihkan sesuai dengan rancangan pada setiap kasusnya. Pada skenario fitur *VM Resource*, user berhasil melihat informasi status *virtual machine lab* sesuai dengan rancangan pada setiap kasusnya. Pada skenario fitur *Remote Access*, *user* berhasil mengakses *virtual machine lab* yang telah di-*deploy* dan dapat mengerjakan modul tugas yang ada pada *virtual machine lab*. *User* dapat menjalankan *command* Nmap [20] pada *virtual machine lab* Linux dan menjalankan Wireshark untuk menangkap *traffic ethernet* [21] pada *virtual machine lab* Windows. Pada skenario fitur *Logout*, *user* berhasil mengakhiri sesi pada *website* purwarupa Laboratorium Virtual sesuai dengan rancangan pada setiap kasusnya.

Tabel 2. Durasi *Deploy* Pengujian Fitur *Start VM*

| Kasus Pengujian | Durasi *Deploy* |
|---|---|
| *User* aesir *deploy* modul "1 Linux" | 17 menit 16 detik |
| *User* aesir *deploy* modul "1 Windows" | 17 menit 12 detik |
| *User* aesir *deploy* modul "2 Linux" | 19 menit 32 detik |
| *User* aesir *deploy* modul "2 Windows" | 17 menit 45 detik |
| *User* busir *deploy* modul "1 Linux" | 15 menit 38 detik |
| *User* busir *deploy* modul "1 Windows" | 15 menit 34 detik |
| *User* busir *deploy* modul "2 Linux" | 15 menit 45 detik |
| *User* busir *deploy* modul "2 Windows" | 17 menit 37 detik |
| *User* aesir *deploy* modul "1 Linux" bersamaan dengan *user* busir *deploy* modul "1 Windows" | 17 menit 20 detik untuk *user* aesir dan 18 menit 26 detik untuk *user* busir |
| *User* aesir *deploy* modul "1 Windows" bersamaan dengan *user* busir *deploy* modul "1 Windows" | 17 menit 21 detik untuk *user* aesir dan 18 menit 29 detik untuk *user* busir |
| *User* aesir *deploy* modul "1 Linux" bersamaan dengan *user* busir *deploy* modul "1 Linux" | 17 menit 57 detik untuk *user* aesir dan 19 menit 48 detik untuk *user* busir |
| *User* aesir *deploy* modul "1 Windows" bersamaan dengan *user* busir *deploy* modul "1 Linux" | 18 menit 3 detik untuk *user* aesir dan 18 menit 5 detik untuk *user* busir |

Dari hasil yang didapat pada Tabel 1 dan data durasi *Deploy* pada Tabel 2, dapat disimpulkan bahwa durasi dari *deployment virtual machine lab* pada fitur *Start VM* stabil dan tidak ada perbedaan durasi *deployment* yang signifikan antara membuat satu *virtual machine lab* dengan membuat dua *virtual machine lab*.

Data durasi yang didapat dari dua *user* melakukan *deploy* secara bersamaan membuktikan bahwa VMWare ESXi dapat men-*deploy virtual machine lab* dari dua *user* sekaligus tanpa adanya peningkatan durasi *deployment* yang signifikan.

Dari pengujian ini didapat rata-rata durasi *deployment* dari delapan *virtual machine lab* Linux sebesar 17 menit 40 detik dan rata–rata durasi *deployment* dari delapan *virtual machine lab* Windows sebesar 17 menit 33 detik. Dari data rata-rata *deployment* tersebut dapat disimpulkan bahwa waktu durasi *deploy virtual machine lab* Windows lebih cepat dari pada waktu durasi *deploy virtual machine lab* Linux. Hal ini terjadi karena proses *Task* "*Import* VApp" pada VMWare ESXi yang berfungsi untuk *deploy* VM dengan OS Linux memakan waktu lebih banyak.

Tabel 3. Durasi *Destroy* Pengujian Fitur *Stop VM*

| Kasus Pengujian | Durasi *Destroy* |
|---|---|
| *User* aesir *deploy* modul "1 Linux" | 23 detik |
| *User* aesir *deploy* modul "1 Windows" | 39 detik |
| *User* aesir *deploy* modul "2 Linux" | 23 detik |
| *User* aesir *deploy* modul "2 Windows" | 49 detik |
| *User* busir *deploy* modul "1 Linux" | 23 detik |
| *User* busir *deploy* modul "1 Windows" | 43 detik |
| *User* busir *deploy* modul "2 Linux" | 22 detik |
| *User* busir *deploy* modul "2 Windows" | 48 detik |
| *User* aesir *deploy* modul "1 Linux" bersamaan dengan *user* busir *deploy* modul "1 Windows" | 23 detik untuk *user* aesir dan 40 detik untuk *user* busir |
| *User* aesir *deploy* modul "1 Windows" bersamaan dengan *user* busir *deploy* modul "1 Windows" | 40 detik untuk *user* aesir dan 41 detik untuk *user* busir |
| *User* aesir *deploy* modul "1 Linux" bersamaan dengan *user* busir *deploy* modul "1 Linux" | 22 detik untuk *user* aesir dan 22 detik untuk *user* busir |
| *User* aesir *deploy* modul "1 Windows" bersamaan dengan *user* busir *deploy* modul "1 Linux" | 39 detik untuk *user* aesir dan 21 detik untuk *user* busir |

Dari hasil yang didapat pada Tabel 1 dan data durasi *Destroy* pada Tabel 3, dapat disimpulkan bahwa durasi dari *destroy virtual machine lab* pada fitur *Stop* VM stabil dan tidak ada perbedaan durasi *destroy* yang signifikan antara menghapus satu *virtual machine lab* dengan menghapus dua *virtual machine lab*. Data durasi yang didapat dari dua *user* melakukan *destroy* secara bersamaan membuktikan bahwa VMWare ESXi dapat menghapus *virtual machine lab* dari dua *user* sekaligus tanpa adanya peningkatan durasi *destroy* yang signifikan. Dari pengujian ini didapat rata-rata durasi *destroy* dari delapan *virtual machine lab* Linux sebesar 22 detik dan rata-rata durasi *destroy* dari delapan *virtual machine lab* Windows sebesar 42 detik. Dari data rata-rata *destroy* tersebut dapat disimpulkan bahwa waktu durasi *destroy virtual machine lab* Linux lebih cepat dari pada waktu durasi *destroy virtual machine lab* Windows. Hal ini terjadi karena proses *Task* "*Shutdown Guest*" pada VMWare ESXi yang berfungsi untuk melakukan *shutdown* terhadap VM dengan OS Windows memakan waktu lebih banyak.

## V. SIMPULAN

Berdasarkan data yang diperoleh serta analisis yang dilakukan dari Proyek Akhir Pengembangan Purwarupa Laboratorium Virtual Berbasis VMWare dengan Terraform, dapat ditarik kesimpulan sebagai berikut:

- Telah dikembangkan sistem purwarupa Laboratorium Virtual yang dapat membuat *virtual machine lab*, melihat status *virtual machine lab* dan menghapus *virtual machine lab* lewat *website* purwarupa Laboratorium Virtual berbasis VMWare ESXi dengan menggunakan Terraform sebagai *Infrastructure as Code*.
- Pengembangan sistem menghasilkan fitur-fitur lainnya yaitu fitur *Login* yang dapat memverifikasi kemudian mengarahkan *user* ke halaman utama *website* purwarupa Laboratorium Virtual milik masing-masing *user*, fitur *Redirect Unverified User* yang mengalihkan *user* ke halaman *login* jika belum melakukan *login*, fitur *Remote Access* yang membuat *user* dapat mengakses *virtual machine lab* secara *remote* dan fitur *Logout* yang dapat mengakhiri sesi *user* kemudian mengalihkan *user* ke halaman *login*.
- *Infrastructure as Code* pada purwarupa Laboratorium Virtual digunakan sebagai otomatisasi pengelolaan dan penyediaan *virtual machine lab* dapat membantu meringankan beban kerja *Administrator* infrastruktur laboratorium yang dilakukan secara berulang–ulang.
- Proses pembuatan *virtual machine lab* memerlukan waktu rata-rata 17 menit 40 untuk Linux dan 17 menit 33 detik untuk Windows. Sedangkan proses penghapusan *virtual machine lab* memerlukan waktu rata-rata 22 detik untuk Linux dan 42 detik untuk Windows. Sehingga proses pembuatan dan penghapusan *virtual machine lab* membutuhkan waktu sebanyak 18 menit 2 detik untuk Linux dan 18 menit 15 detik untuk Windows.
- Metode *Black Box Testing* sebagai uji fungsionalitas sistem menunjukkan bahwa seluruh fungsi yang terdapat pada sistem dapat bekerja dengan baik.

## REFERENSI


[1] W. Susanti *et al.*, *Model Virtual Lab dan Remote Lab melalui Inkuiri dan Kolaborasi Berbantuan Android*. Penerbit Lakeisha, 2022.

[2] N. Marshall, M. Brown, G. B. Fritz, and R. Johnson, *Mastering VMware VSphere 6.7*. John Wiley & Sons, 2018.

[3] K. Morris, *Infrastructure as code*. O'Reilly Media, 2020.

[4] V. HROMÁDKA, "Infrastrucure As Code in Agile Software Development," 2022.

[5] L. De La Torre, M. Guinaldo, R. Heradio, and S. Dormido, "The ball and beam system: A case study of virtual and remote lab enhancement with moodle," *IEEE Trans. Ind. Inform.*, vol. 11, no. 4, pp. 934–945, 2015.

[6] M. B. Tharayanil, G. Whitney, M. Aiash, and C. Benzaid, "Virtualization and cyber security: arming future security practitioners," in *2015 IEEE Trustcom/BigDataSE/ISPA*, IEEE, 2015, pp. 1398–1402.

[7] T. Terraform, "Documentation | Terraform | HashiCorp Developer," *Documentation | Terraform | HashiCorp Developer*, 2022. https://developer.hashicorp.com/terraform/docs (accessed May 29, 2023).

[8] S. Muthoni, G. Okeyo, and G. Chemwa, "Infrastructure as Code for Business Continuity in Institutions of Higher Learning," in *2021 International Conference on Electrical, Computer and Energy Technologies (ICECET)*, IEEE, 2021, pp. 1–6.

[9] QuinnRadich, "Remote Desktop Protocol - Win32 apps," Aug. 19, 2020. https://learn.microsoft.com/en-us/windows/win32/termserv/remote-desktop-protocol (accessed May 29, 2023).
[10] A. Apache, "About the XAMPP project," 2022. https://www.apachefriends.org/about.html (accessed May 29, 2023).
[11] L. Babin, *Beginning Ajax with PHP: From novice to professional*. Apress, 2007.
[12] M. MariaDB, "About MariaDB Server," *MariaDB.org*, 2022. https://mariadb.org/about/ (accessed May 29, 2023).
[13] I. udiawan Sitorus, "Cara Membuat Login Multi User Ajax Menggunakan PHP, Sweetalert Dan IziToast," *BelajarwithIB*, Apr. 09, 2021. https://www.belajarwithib.my.id/2021/04/cara-membuat-login-multi-user-ajax.html (accessed May 29, 2023).
[14] J. Spurlock, *Bootstrap: responsive web development*. O'Reilly Media, Inc., 2013.
[15] M. McGrath, *PHP & MySQL in easy steps*. In Easy Steps, 2012.
[16] Hewlett-Packard Development Company, "DHCPv4 server," 2022. https://techhub.hpe.com/eginfolib/networking/docs/switches/WB/15-18/5998-8162_wb_2920_mcg/content/ch06s04.html (accessed May 29, 2023).
[17] TightVNC, "TightVNC for Windows: Installation and Getting Started." 2012.
[18] J. Senkerik, "Terraform Provider." May 29, 2023. Accessed: May 29, 2023. [Online]. Available: https://github.com/josenk/terraform-provider-esxi
[19] A. Nordeen, *Learn Software Testing in 24 Hours: Definitive Guide to Learn Software Testing for Beginners*. Guru99, 2020.
[20] Nmap, "Nmap: the Network Mapper - Free Security Scanner," 2022. https://nmap.org/ (accessed May 29, 2023).
[21] R. Sharpe, E. Warnicke, and U. Lamping, "Wireshark User's Guide: Version 4.1.0."