IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 11, No. 4, December 2022, pp. 1232~1238
ISSN: 2252-8938, DOI: 10.11591/ijai.v11.i4.pp1232-1238

CekUmpanKlik: an artificial intelligence-based application to detect Indonesian clickbait

Muhammad Noor Fakhruzzaman, Sie Wildan Gunawan
Faculty of Advanced Technology and Multidiscipline, Universitas Airlangga, Surabaya, Indonesia

Article Info

Article history:

Received Aug 2, 2021
Revised Jun 14, 2022
Accepted Jul 5, 2022

Keywords:

Adult literacy
Clickbait
Digital divide
Indonesian
Natural language processing

ABSTRACT

This study attempted to deploy a high performing natural language processing model which specifically trained on flagging clickbait Indonesian news headline. The deployed model is accessible from any internet-connected device because it implements representational state transfer application programming interface (RESTful API). The application is useful to avoid clickbait news which often solely purposed to rack money but not delivering trustworthy news. With many online news outlets adopting the click-based advertising, clickbait headline become ubiquitous. Thus, newsworthy articles often cluttered with clickbait news. Leveraging state-of-the-art bidirectional encoder representation from transformers (BERT), a lightweight web application is developed. This study offloaded the computing resources needed to train the model on a separate instance of virtual server and then deployed the trained model on the cloud, while the client-side application only needs to send a request to the API and the cloud server will handle the rest, often known as three-layer architecture. This study designed and developed a web-based application to detect clickbait in Indonesian using IndoBERT as its language model. The application usage and potentials were discussed. The source code and running application are available for public with a performance of mean receiver operating characteristic-area under the curve (ROC-AUC) of 89%.

Corresponding Author:

Muhammad Noor Fakhruzzaman
Faculty of Advanced Technology and Multidiscipline, Universitas Airlangga
Gedung Kuliah Bersama, Kampus C, Universitas Airlangga, Surabaya, Indonesia
Email: ruzza@ftmm.unair.ac.id, ruzza@alumni.iastate.edu

1. INTRODUCTION

Clickbait headlines have becoming more prominent since the advertiser decide to count popularity based on clicks. Some news that were deemed as less newsworthy used clickbait to attract readers, to ensure that the news still makes money [1]. Since then, a lot of scientific articles proposed ways to detect clickbait using artificial intelligence (AI), one of them was fine tuned for Indonesian headlines [2]–[6]. However, those articles often propose methods only but did not create any usable tool for the readers. Some Indonesian people simply cannot distinguish clickbait and normal headlines and often get disappointed after clicking the news, and it may be correlated to the inequality of internet access in Indonesia, widely known as the digital divide [7]. People with less internet exposure may have a hard time at pointing out clickbaits, simply because they seldom encountered it.

A study found that human curiosity takes part in detecting clickbait, such with high curiosity, people are more likely to get trapped on clickbait news. The connection between curiosity and clickbait persisted on all age [8]. Indonesian is mostly curious people, the high engagement on celebrity related news and overly friendly Indonesian culture, which often followed by overly curious question (or kepo), one of the examples

is how Indonesian really like to dig into other's private lives, such as asking why a couple still haven't had kids or so. This culture of high curiosity is common among Asians, naturally seen as an act of care toward others and a sense of togetherness [9]–[11].

With the increasing usage of clickbait headline in Indonesian news outline, there must be a reason. Clickbait news often met with superb engagement on all platform, this is mainly due to the nature of clickbait that are dramatic, full of teases, often sexually explicit, and incite over curiosity [12]. High engagement from the advertising perspective means good money, disregarding how the news was written or how newsworthy the news is. Thus, online news outlet is faced with dilemma, either write a lot of sloppy news with clickbait headline and high advertisement money, or staying true to the high-quality journalism ethics of publishing the truth without hiding anything [13], [14].

Previous studies proposed a plethora of machine learning method to point out clickbait headline, but often undeployed and remained as a report of evaluation metrics [2]–[6], [15]. Such model also needs a high computing power to run, with the addition of knowledge on how to actually run the model with user input [16], [17]. However, most Indonesian did not receive higher education or did not have required knowledge on how to run specific program with complicated access [18]–[21]. Although, Indonesian internet users are among the highest in South-East Asia and rapidly grows in the last 5 years. Given a ready to use application, with mobile-accessibility and user-friendly user interaction (UI), the clickbait detection AI model will be highly accessible to all Indonesian.

By the availability of an easy-to-use application, digital literacy among readers may be increased and misinformation can be hindered [12], [22]. Furthermore, many methods in detecting clickbait from scientific articles can be implemented with high level of abstraction, so that people can use it easily without hassle. Therefore, this study aims to propose and develop an easy-to-use application to detect Indonesian clickbait headline, leveraging state-of-the-art language model, while still being easy on the client's computing resource.

2. METHODS

This study uses an existing model and annotated dataset of clickbait headlines. The model architecture used multilingual bidirectional encoder representations from transformers (BERT) and topped with a hidden layer of 100 neurons and one output layer [6]. While the dataset consisted of 6,000 annotated headlines with balanced class of clickbait and non-clickbait, also with absolute reliability [23]. Figure 1 shows the application architecture, depicting how user requests flow through the application. It leverages the representational state transfer application programming interface (ReST API) architecture.

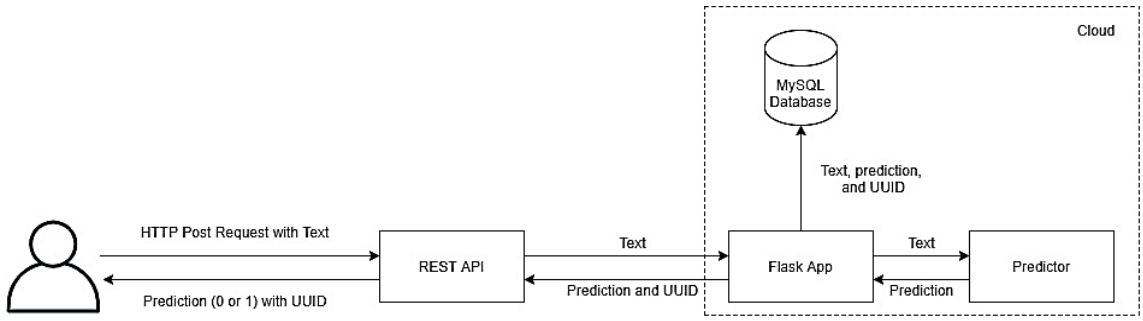


Figure 1. User request flow diagram

REST API architecture offers flexibility that it can run on multiple front-end versions at the same time without breaking its clients [24]. Using hypertext transfer protocol (HTTP) request, it leverages the light protocol to send standardized data format, often in javascript object notation (JSON), on a secure way because it still travels through secure HTTP (HTTPS). This way, when handling requests from mobile applications, or from any client-side interface, the data still flows seamlessly to the cloud server.

In order to reduce memory usage, we train the model using IndoBERT lite base by IndoNLU instead of using multilingual BERT model used in the previous study [6]. This pre-trained model has 12 layers and around 11.7 million parameters. Compared to other pre-trained models, IndoBERT lite base performs well with f1-score average of 85%. Other models in IndoBERT family performs better but they require more parameters, hence higher memory usage [25].

This study also use MySQL as the database engine due to its compatibility with a lot of server-side environments [26]. The database is used to store user feedback of the prediction for later retraining purposes. Figure 2 shows the feedback flow diagram and how the prediction is stored into the database. Furthermore, this study use flask to build the API, fronted by a gunicorn webserver gateway interface (WSGI) and engine x (nginx) webserver to serve the endpoint through the cloud, as depicted in Figure 1. Flask is used because it is not only simple and straightforward, but also very flexible, it provides high level abstraction to build API easily. Additionally, flask integrates seamlessly with gunicorn and nginx [27]. Moreover, nginx is used for its reverse-proxy capability, so that in the future, other microservices can be added to the same server and contained easily.

REST API (or sometimes called RESTful API) is an API that uses REST architecture to handle a request sent from a user in the front-end. This allows users to interact with RESTful web services. In short, REST API works like a mediator between user and server [28]. The API was built to communicate the front-end with a trained model, while the training iteration was executed on a separate google colaboratory platform. A snippet of the source code is shown on Figure 3. All respective codes are stored in https://github.com/ruzcmc/ClickbaitIndo-textclassifier and accessible to public.

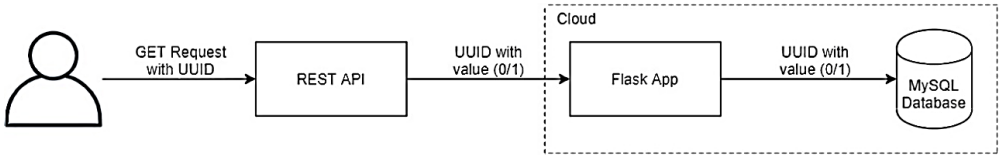


Figure 2. User feedback flow diagram

```
--app.py--

from flask import Flask, request, jsonify, make_response
from flask_limiter import Limiter
from flask_limiter.util import get_remote_address
from predict import predict

@app.route("/predict", methods=["POST"])
@limiter.limit("2 per minute")
def predictText():
request_data = request.get_json()
uuid = uuid4()
text = request_data['text']
user_ip = request.remote_addr
prediction = predict(text)
data = UserRequest(
uuid = uuid,
text = text,
prediction = prediction,
ip_address = user_ip
)
db.session.add(data)
db.session.commit()
response = make_response(
jsonify(
{
"id": uuid,
"prediction": prediction
}
),
200
)
response.headers["Content-Type"]="application/json"
return response

--predict.py--

import tensorflow as tf

filename = "model/model.h5"
model = keras.models.load_model(filename)


def predict(input):
input_id, attn_mask = np.array(encodeText(input))
data = [input_id, attn_mask]

prediction = model.predict(data)
prediction = (prediction > 0.5).astype("int32")
prediction = prediction[0].item()

return prediction
```

Figure 3. Code snippet

In Figure 3, app.py shows how the API handled HTTP POST request, especially for the prediction. When the API receive a HTTP POST request from the frontend, it redirects the json data to another program that loads the trained IndoBERT model, passing through the whole processing pipeline. The prediction value is then saved into a variable then returned as json data with 200 code, including its unique id for archiving purpose. Finally, the frontend will receive json data from the API then display it to the user.

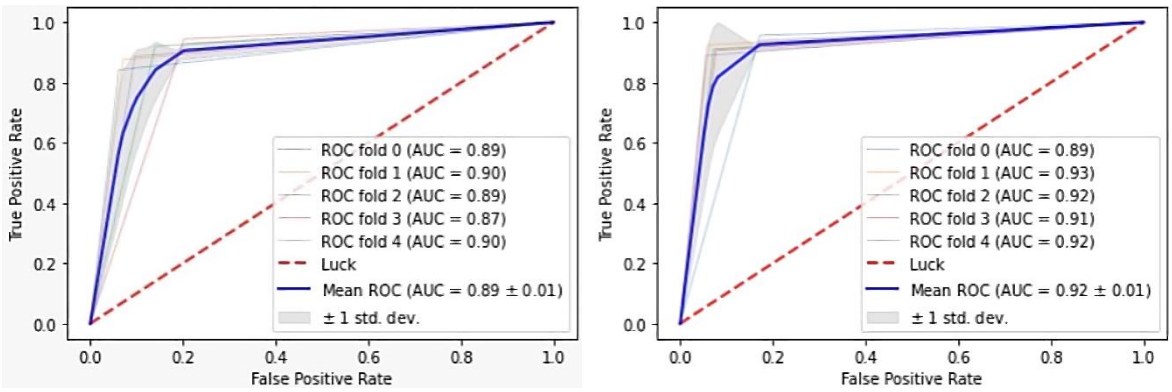
While predict.py is the core of the prediction. The predict() function includes text pre-processing and BERT encoding from its respective functions, then the trained model is loaded to receive encoded text. Its output takes form as an integer from final sigmoid activation function, if the probability of the classification reached more than 0.5, it is classified as a clickbait (positive). Finally, the prediction value is returned to be caught by the API.

The scripts are then hosted on a DigitalOcean droplet with Linux Ubuntu 20 as the operating system. Then, a static page hosted on github is used to invoke the API via HTTPS. To secure the endpoint from any unwanted actions, this study used encrypted HTTPS protocol to serve both the API and the front-end. We also use a request limiter in the front-end and back-end to prevent spamming and disk operating system (DoS) attacks. The limiter limits requests to one request per minute from the fronted but set to be two requests per minute in the back-end. The limit difference is due to Axios' default behaviour to send pre-flight request before sending the actual request.

3. RESULTS AND DISCUSSION

After training the model, IndoBERT performs well with a receiver operating characteristic-area under the curve (ROC-AUC) average of 89%. Using this approach, memory usage is reduced by around 800 MB. Less memory usage means our model is more efficient and will use less resources when processing a prediction request. Therefore, the cost of the server that hosts the model can be significantly reduced.

However, switching to lighter model sacrificed prediction performance. Figure 4 (a)-(b) shows the ROC-AUC plot of IndoBERT model used in the developed application compared to multilingual BERT model used in previous study [6]. The developed application is accessible through our github page https://ruzcmc.github.io/ClickbaitIndo-textclassifier. Figure 5 shows the interface for user input, and request prediction button.



(a) (b)

Figure 4. Language model performance comparison (a) IndoBERT lite base and (b) Multilingual BERT

After the prediction is shown to the user, the interface pop up button for user feedback. The feedback is used to verify the prediction. Figure 6 shows the interface after user received the prediction and the app request for user feedback.

The web-based application for detecting Indonesian clickbait headlines was developed successfully. Although the training performance was slightly waned, the finished product still delivers most of the time. Future study should aim to evaluate the performance of the app, either from the UI/ user experience (UX) perspective or from the memory efficiency perspective. The prediction model also need to be evaluated and retrained in the future to enhance its capability.

The limitation of the finished product is the lack of security on its endpoints. It should be possible to add extra protection such as Google reCAPTCHA or API-key based authentication. Future study needs to integrate authentication protocol to enhance the endpoint security. Although the finished product is easy

enough to use, sometimes it is a hassle to open new website only for checking clickbaits. Future study can further develop this application into a more integrated solution, such as browser extension capable of flagging clickbaits on-the-fly.

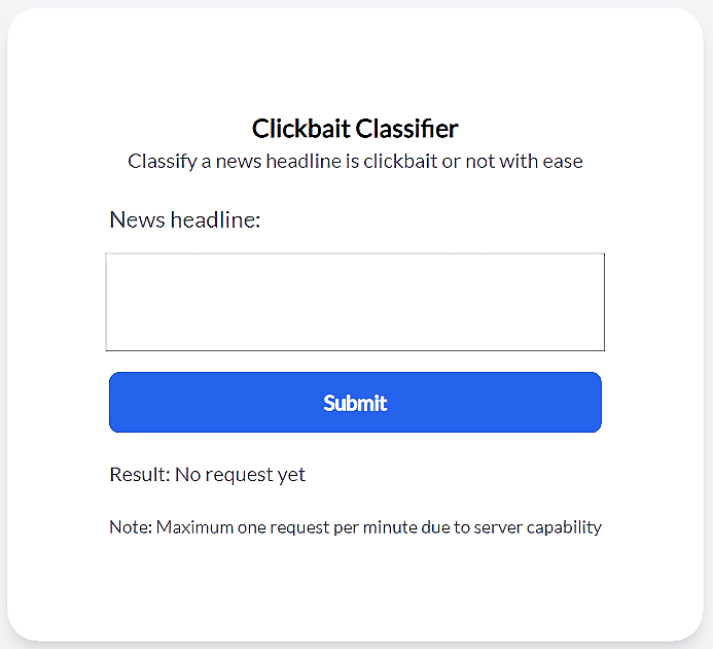


Figure 5. Initial application UI

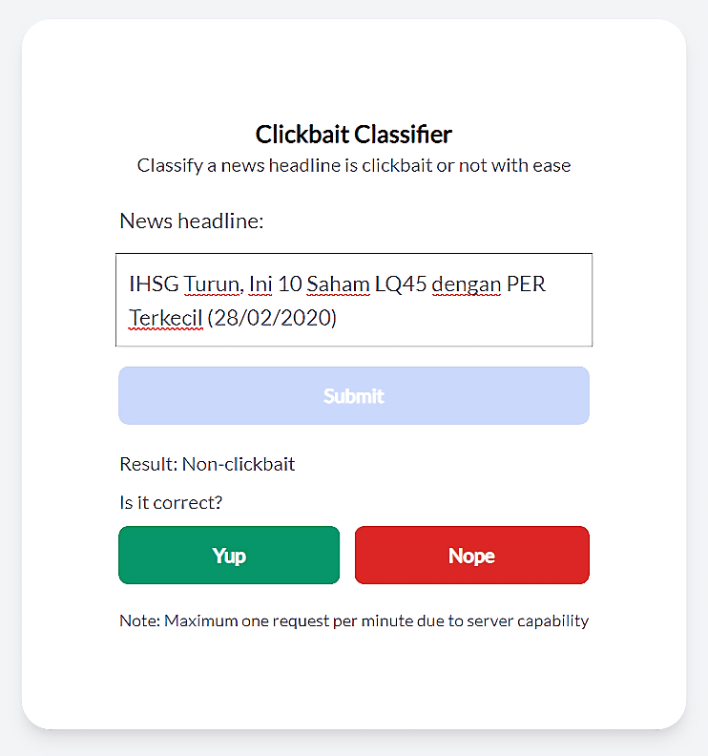


Figure 6. AI prediction result and user feedback prompt

4. CONCLUSION

The deployment of previously trained natural language processing model to detect Indonesian clickbait is successful. Using limited resource and leveraging the three layer service-oriented-architecture, this study can offload heavy natural language processing to an established cloud server. The usage of the app was not tested to the public even though the web application is publicly accessible and hosted on Github Page. Future work should enhance the endpoint security, especially if they want to transport private data through the APIs and also evaluate the user feedback on how useful and how easy to use the application is. Finally, future researcher should also consider integrating the endpoint into a media monitoring dashboard. A smart dashboard will help stakeholders to monitor their media quality and provide insights to enhance its public relations. This integration should also help news outlet to increase their journalism quality.

ADDITIONAL RESOURCE

The complete Python notebook, Scripts, Dockerfile, APIs and datasets are stored on https://github.com/ruzcmc/ClickbaitIndo-textclassifier and are available for public. The web application can be accessed and used on https://ruzcmc.github.io/ClickbaitIndo-textclassifier/

ACKNOWLEDGEMENTS

Training dataset provided by A. William and Y. Sari as cited. This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

REFERENCES

[1] Y. Chen, N. K. Conroy, and V. L. Rubin, "News in an online world: The need for an 'automatic crap detector," *Proceedings of the Association for Information Science and Technology*, vol. 52, no. 1, pp. 1–4, Jan. 2015, doi: 10.1002/pra2.2015.145052010081.

[2] A. Agrawal, "Clickbait detection using deep learning," in *2016 2nd International Conference on Next Generation Computing Technologies (NGCT)*, 2016, pp. 268–272.

[3] A. Anand, T. Chakraborty, and N. Park, "We used neural networks to detect clickbaits: You won't believe what happened next!," in *European Conference on Information Retrieval*, 2017, pp. 541–547.

[4] P. Biyani, K. Tsioutsiouliklis, and J. Blackmer, "8 amazing secrets for getting more clicks": detecting clickbaits in news streams using article informality," in *AAAI'16: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, Feb. 2016, pp. 94–100.

[5] A. Chakraborty, B. Paranjape, S. Kakarla, and N. Ganguly, "Stop clickbait: Detecting and preventing clickbaits in online news media," in *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 2016, pp. 9–16.

[6] M. N. Fakhruzzaman, S. Z. Jannah, R. A. Ningrum, and I. Fahmiyah, "Clickbait headline detection in Indonesian news sites using multilingual bidirectional encoder representations from transformers (M-BERT)," *arXiv preprint*, pp. 1–19, Feb. 2021.

[7] L. Puspitasari and K. Ishii, "Digital divides and mobile Internet in Indonesia: Impact of smartphones," *Telematics and Informatics*, vol. 33, no. 2, pp. 472–483, May 2016, doi: 10.1016/j.tele.2015.11.001.

[8] L. Venneti and A. Alam, "How curiosity can be modeled for a clickbait detector," *arXiv preprint*, Jun. 2018, doi: 10.48550/arXiv.1806.04212.

[9] T. Novák, *Go Hungary-Go Indonesia*. Budapest Business School, 2017.

[10] C. Geertz, *The religion of Java*. University of Chicago Press, 1976.

[11] W. Sucher, A. K. Pusiran, N.-T. Dhevabanchachai, and K. Chon, "The influences of Asian cultural values in the Asian hospitality services," in *The 11th APacCHRIE Conference*, 2013, pp. 21–24.

[12] N. Hurst, "To clickbait or not to clickbait? an examination of clickbait headline effects on source credibility," University of Missouri - Columbia, 2016.

[13] K. Munger, "All the news that's fit to click: The economics of clickbait media," *Political Communication*, vol. 37, no. 3, pp. 376–397, Dec. 2019, doi: 10.1080/10584609.2019.1687626.

[14] J. A. Braun and J. L. Eklund, "Fake news, real money: Ad tech platforms, profit-driven hoaxes, and the business of journalism," *Digital Journalism*, vol. 7, no. 1, pp. 1–21, 2019.

[15] B. Siregar, I. Habibie, and E. B. Nababan, "Identification of Indonesian clickbait news headlines with long short-term memory recurrent neural network algorithm," in *Journal of Physics: Conference Series*, 2021, vol. 1882, no. 1, p. 012129.

[16] G. Yi and V. Loia, "High-performance computing systems and applications for AI," *The Journal of Supercomputing*, vol. 75, no. 8, pp. 4248–4251, 2019.

[17] J. Welser, J. W. Pitera, and C. Goldberg, "Future computing hardware for AI," in *2018 IEEE International Electron Devices Meeting (IEDM)*, 2018, p. 1.3. 1--1.3. 6.

[18] M. E. Susilo, S. Afifi, and S. Yustitia, "Hoax as a reflection on the low digital literacy in Indonesia," in *Proceedings of the Second International Conference on Social, Economy, Education and Humanity (ICoSEEH 2019) - Sustainable Development in Developing Country for Facing Industrial Revolution 4.0*, 2020, pp. 165–174. doi: 10.5220/0009100201650174.

[19] I. R. Katz and A. Macklin-Smith, "Information and communication technology (ICT) literacy: Integration and assessment in higher education," *Journal on Systemics, Cybernetics and Informatics*, vol. 5, no. 4, pp. 50–55, 2007.

[20] A. Rahmah, "Digital literacy learning system for Indonesian citizen," *Procedia Computer Science*, vol. 72, pp. 94–101, Jan. 2015, doi: 10.1016/j.procs.2015.12.109.

[21] R. Rakimahwati and Z. Ardi, "An alternative strategy for increasing Indonesian student digital literacy skills through interactive game," *Journal of Physics: Conference Series*, vol. 1339, p. 12122, Dec. 2019, doi: 10.1088/1742-6596/1339/1/012122.

[22] Y. Chen, N. J. Conroy, and V. L. Rubin, "Misleading online content: recognizing clickbait as" false news"," in *Proceedings of the 2015 ACM on Workshop on Multimodal Deception Detection*, Nov. 2015, pp. 15–19. doi: 10.1145/2823465.2823467.
[23] A. William and Y. Sari, "CLICK-ID: A novel dataset for Indonesian clickbait headlines," *Data in Brief*, vol. 32, p. 106231, Oct. 2020, doi: 10.1016/j.dib.2020.106231.
[24] L. Li, W. Chou, W. Zhou, and M. Luo, "Design patterns and extensibility of REST API for networking applications," *IEEE Transactions on Network and Service Management*, vol. 13, no. 1, pp. 154–167, 2016.
[25] B. Wilie *et al*., "IndoNLU: Benchmark and resources for evaluating Indonesian natural language understanding," in *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, 2020, pp. 843–857.
[26] M. Kofler, "What is MySQL?," in *MySQL*, Berkeley, CA: Apress, 2001, pp. 3–19. doi: 10.1007/978-1-4302-0853-2_1.
[27] D. Ghimire, "Comparative study on Python web frameworks: Flask and Django," Metropolia University of Applied Sciences, 2020.
[28] M. Masse, *REST API design Rulebook: designing consistent RESTful web service interfaces*. O'Reilly Media, Inc., 2011.

BIOGRAPHIES OF AUTHORS

Muhammad Noor Fakhruzzaman was born and raised in Surabaya. He holds an interdisciplinary Master's Degree in Human-Computer Interaction and Journalism & Mass Communication from Iowa State University. His current research interests fall between Data Science and Mass Communication, mainly automated media monitoring using natural language processing. He currently teaches at Data Science Technology Study Program in Universitas Airlangga. His research includes Brain-Computer Interface, Indonesian natural language processing, and media monitoring. He is a member of Kappa Tau Alpha, an honor society of Journalism and Mass Communication studies. He also loves to train in grappling sport: Wrestling, Brazilian Jiu-jitsu (2nd-degree blue belt) and fanatically watch pro-wrestling shows. He can contact at email: ruzza@ftmm.unair.ac.id or ruzza@alumni.iastate.edu.

Sie Wildan Gunawan was born and raised in Samarinda. He is now an undergraduate student in Universitas Airlangga pursuing a degree in Robotics and Artificial Intelligence. His current research interests are computer vision and natural language processing. He can be contacted at email: sie-wildan-gunawan-2020@ftmm.unair.ac.id.