

Implementasi dan Deployment Model Machine Learning Menggunakan App Engine pada Google Cloud Platform

Fathin Cahyo Ramadhan¹, Wiga Maulana Baihaqi²

^{1,2}Teknologi Informasi, Fakultas Ilmu Komputer, Universitas Amikom Purwokerto, Purwokerto, Indonesia
surel: ¹fatincahya69@gmail.com, ²wiga@amikompurwokerto.ac.id

Info Artikel

Sejarah artikel:

Diterima 08-01-2024

Revisi 15-11-2024

Diterima 26-12-2024

Kata kunci:

Google Cloud Platform
Convolutional Neural Network
Machine Learning
Cloud Computing
Mobile Development
Google App Engine

ABSTRAK

Penelitian ini bertujuan menganalisis keberhasilan implementasi dan deployment aplikasi mobile berbasis cloud dalam konteks industri peternakan ayam. Fokus utama penelitian adalah mengevaluasi faktor-faktor kritis yang mempengaruhi kinerja aplikasi dalam meningkatkan produktivitas dan efisiensi produksi telur ayam. Tim multidisiplin terdiri dari anggota dari divisi Machine Learning, Cloud Computing, dan Mobile Development yang bekerja sama untuk mengembangkan aplikasi deteksi keberhasilan telur dengan menggunakan teknologi Convolutional Neural Network (CNN) dan Google Cloud Platform (GCP). Proses implementasi melibatkan pengembangan model machine learning menggunakan TensorFlow Lite, deployment pada Google Cloud Console dan integrasi dengan aplikasi mobile berbasis Android. Langkah-langkah melibatkan perancangan arsitektur aplikasi, otentikasi pengguna, pengelolaan dataset, dan pengujian fungsionalitas aplikasi. Sebagai tambahan, analisis biaya dan evaluasi dampak ekonomi dijalankan untuk memahami keefektifan solusi yang diusulkan. Banyak kemudahan yang ditawarkan dalam google app engine salah satunya terdapat fitur free trial dimana tidak perlu membayar apapun jika ingin menggunakan platform ini.

Penulis yang sesuai:

Fathin Cahyo Ramadhan

Program Studi Teknologi Informasi Fakultas Ilmu Komputer Universitas Amikom Purwokerto

Email: fatincahya69@gmail.com

1. PENDAHULUAN

Industri peternakan telur di Indonesia mengalami pertumbuhan signifikan pada tahun 2023, mencerminkan peran vital dalam memenuhi kebutuhan protein hewani masyarakat. Namun, tantangan dalam efisiensi produksi dan manajemen risiko masih menjadi perhatian utama. Sebagai respons terhadap tantangan ini, inovasi teknologi menjadi sangat penting untuk meningkatkan produktivitas dan efisiensi dalam sektor ini.

Penelitian ini memfokuskan pada pengembangan dan implementasi aplikasi mobile berbasis cloud yang menggunakan teknologi Convolutional Neural Network (CNN) dan Google Cloud Platform (GCP) [1], [2]. Tujuan utama dari aplikasi ini adalah untuk meningkatkan efisiensi produksi dan pengelolaan data peternakan

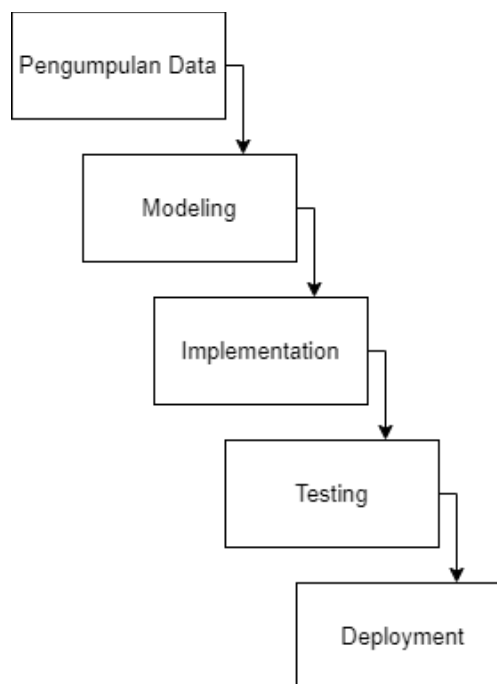
ayam telur, dengan solusi yang dapat memantau kesehatan telur secara otomatis dan memberikan analisis yang relevan kepada peternak. Solusi ini diharapkan dapat memberikan dampak positif, baik dalam konteks peningkatan produktivitas maupun efisiensi operasional di sektor peternakan [3], [4]

Teknologi Cloud Computing menjadi komponen penting dalam solusi ini. Cloud Computing memungkinkan penyimpanan data, pengelolaan, dan pengolahan secara efisien melalui internet, sehingga pengguna dapat mengakses layanan dari mana saja. Selain itu, teknologi ini menawarkan fleksibilitas dan skalabilitas yang sesuai dengan kebutuhan berbagai jenis pengguna, mulai dari peternak kecil hingga skala industri besar [4].

Dalam konteks penelitian ini, proses implementasi melibatkan langkah-langkah strategis seperti perancangan arsitektur aplikasi, otentikasi pengguna, pengelolaan dataset, dan pengujian fungsionalitas aplikasi. Selain itu, analisis biaya dan evaluasi dampak ekonomi dilakukan untuk memahami keefektifan solusi yang diusulkan. Dengan kombinasi teknologi CNN dan GCP, penelitian ini bertujuan untuk menjawab tantangan utama dalam industri peternakan telur di Indonesia serta mendukung inovasi dalam sektor pertanian yang terus berkembang [5].

2. METODE

Penelitian ini menggunakan metode pengembangan perangkat lunak Waterfall yang terdiri dari lima tahapan utama: pengumpulan data, pemodelan, implementasi, pengujian, dan deployment. Setiap tahapan dirancang untuk memastikan kelengkapan proses pengembangan aplikasi Fertile Egg Detector (FED) berbasis cloud.



Gambar 1 Metode Waterfall

2.1. Pengumpulan Data

Tahap ini berfokus pada identifikasi kebutuhan data yang relevan untuk melatih dan menguji model machine learning. Data dikumpulkan dari sumber yang terpercaya, seperti Kaggle.com, untuk memastikan

kelengkapan dan kualitas dataset yang digunakan. Proses ini dimulai dengan menentukan parameter dataset, termasuk jumlah data, atribut yang relevan, dan format data yang sesuai untuk model machine learning [6]. Selain itu, dilakukan validasi terhadap dataset untuk menghindari adanya data yang tidak lengkap atau bias. Dataset yang dikumpulkan kemudian dianalisis untuk memastikan bahwa informasi yang terkandung mendukung pengembangan model secara efektif [7].

2.2. Pemodelan

Dalam tahap pemodelan, desain aplikasi dirancang oleh Tim Mobile Development dengan menggunakan Figma. Proses ini mencakup pembuatan tata letak antarmuka pengguna, seperti halaman onboarding, login, register, homepage, kamera deteksi, profil, dan pengaturan. Setelah desain selesai, Tim Mobile Development menerjemahkan desain tersebut menjadi kode program menggunakan Android Studio dan Kotlin [8]. Selain itu, Tim Cloud Computing merancang Application Programming Interface (API) yang menghubungkan aplikasi dengan backend berbasis Google Cloud Console. API dirancang untuk menangani permintaan data, otentikasi pengguna, serta penyimpanan dan pengambilan data dari cloud. Proses ini memastikan bahwa aplikasi memiliki struktur backend yang kuat dan aman untuk digunakan [9].

2.2.1. Implementasi

Tahap implementasi merupakan proses penerapan semua elemen yang telah dirancang pada tahap sebelumnya. Tim Machine Learning menggunakan TensorFlow Lite untuk melatih model machine learning di Google Colab, dengan memanfaatkan dataset yang telah dikumpulkan [10]. Proses ini melibatkan pelatihan model untuk mengenali pola dan karakteristik dari data telur. Sementara itu, Tim Cloud Computing menggunakan Visual Studio Code dan Postman untuk mengembangkan dan menguji API yang sebelumnya dirancang. Di sisi lain, Tim Mobile Development mengintegrasikan model machine learning yang telah dilatih ke dalam aplikasi Android menggunakan Android Studio. Setiap elemen diuji untuk memastikan kompatibilitas dan kinerja yang optimal dalam system [11].

2.2.2. Pengujian

Pada tahap pengujian, seluruh sistem diuji secara menyeluruh untuk memastikan bahwa semua fungsi bekerja dengan baik sesuai dengan spesifikasi. Pengujian dilakukan dengan menggunakan Postman untuk memverifikasi API, memastikan bahwa setiap endpoint dapat diakses dan merespons dengan benar. Selain itu, pengujian aplikasi dilakukan melalui Android Emulator untuk mengevaluasi integrasi antara aplikasi, API, dan model machine learning [12]. Tim memastikan bahwa setiap modul, seperti otentikasi pengguna, pengolahan data, dan antarmuka pengguna, berfungsi tanpa hambatan. Pengujian ini juga mencakup pengujian stres dan kinerja untuk mengukur kemampuan sistem dalam menangani beban kerja yang tinggi.

2.2.3. Deployment

Proses deployment dilakukan melalui Google Cloud Console menggunakan layanan App Engine untuk memastikan bahwa aplikasi dapat diakses secara online oleh pengguna. Langkah pertama dalam tahap ini adalah melakukan konfigurasi file app.yaml, yang digunakan untuk menentukan parameter aplikasi, seperti lokasi region dan spesifikasi server. Setelah konfigurasi selesai, kode aplikasi diunggah ke App Engine menggunakan Google Cloud SDK. Proses ini mencakup pengaturan domain dan endpoint yang memungkinkan pengguna mengakses aplikasi secara langsung [13]. Deployment juga mencakup pengujian akhir untuk memastikan bahwa aplikasi yang diunggah berjalan dengan baik di lingkungan cloud. Dengan deployment yang berhasil, aplikasi FED dapat diakses oleh peternak secara efisien melalui perangkat mobile [14].

3. HASIL DAN PEMBAHASAN

Penelitian ini menghasilkan aplikasi mobile bernama Fertile Egg Detector (FED), sebuah solusi berbasis cloud untuk mendeteksi kesuburan telur secara otomatis. Proses pengembangan aplikasi ini



melibatkan serangkaian tahapan yang telah diselesaikan oleh tim multidisiplin. Berikut adalah hasil dari setiap tahap beserta pembahasannya.

3.1. Pengumpulan Data

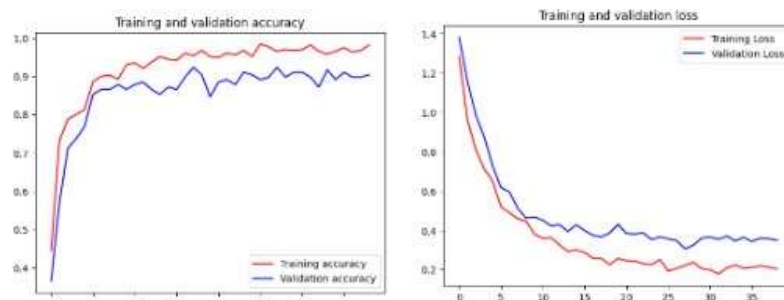
Dataset telur diperoleh dari Kaggle.com, yang telah dipilih berdasarkan relevansi dan kelengkapan data. Dataset ini terdiri dari sejumlah besar gambar telur yang digunakan untuk melatih model deteksi berbasis Convolutional Neural Network (CNN) [15]. Dataset ini mencakup gambar-gambar telur dengan berbagai kategori yang diklasifikasikan untuk mempermudah proses prediksi, seperti ditunjukkan pada Gambar 2.



Gambar 2 Contoh Gambar dalam Dataset Telur

3.2. Pemodelan

Pada tahap ini, model training mampu mendeteksi kesuburan telur dengan akurasi tinggi. Hasil evaluasi model ditunjukkan pada Gambar 3, yang memperlihatkan kurva Receiver Operating Characteristic (ROC). Kurva ROC ini menunjukkan kinerja model dalam memprediksi kesuburan telur, dengan area di bawah kurva (AUC) yang hampir sempurna. Hal ini membuktikan bahwa model memiliki kemampuan prediksi yang sangat baik.



Gambar 3 ROC Curve

3.3. Implementasi

Tahap implementasi melibatkan pengujian API yang dikembangkan untuk mendukung aplikasi. API untuk menambah pengguna ke dalam database dirancang agar setelah data pengguna berhasil ditambahkan, sistem dapat mengirimkan kode verifikasi melalui email untuk proses otentikasi, seperti yang ditunjukkan pada Gambar 4.

```

// Register user ke database
const register = async (req, res) => {
  try {
    // Validasi input
    const { email, password } = req.body;
    if (!email || !password) {
      return res.status(400).json({ message: 'Email dan password wajib diisi' });
    }

    // Cek apakah email sudah terdaftar
    const existingUser = await db.collection('users').findOne({ email });
    if (existingUser) {
      return res.status(400).json({ message: 'Email sudah terdaftar' });
    }

    // Hash password
    const hashedPassword = bcrypt.hashSync(password, 10);

    // Simpan user ke database
    const newUser = { email, password: hashedPassword };
    await db.collection('users').insertOne(newUser);

    // Login user
    const token = jwt.sign({ email }, process.env.JWT_SECRET, { expiresIn: '1h' });
    return res.status(200).json({ token });
  } catch (error) {
    console.error('Error register:', error);
    return res.status(500).json({ message: 'Gagal mendaftar pengguna atau email/username sudah terdaftar' });
  }
};

```

Gambar 4 API untuk Menambah Pengguna ke Database dengan Verifikasi Email

Selain itu, API untuk mengambil data telur berdasarkan ID pengguna dirancang untuk memungkinkan personalisasi data untuk setiap pengguna aplikasi, seperti terlihat pada Gambar 5.

```

// Endpoint API mengambil semua data users
server.route({
  method: 'GET',
  path: '/api/users',
  handler: async (request, h) => {
    try {
      // Validasi apakah user login
      // await server.plugins.auth.isAuthenticated();

      const snapshot = await db.collection('users').get();
      const data = snapshot.docs.map(doc => doc.data());

      return data;
    } catch (error) {
      return h.response({ error: 'Gagal mengambil data users' }).code(500);
    }
  }
});

try {
  await server.start();
  console.log('Server berjalan di', server.info.uri);
} catch (error) {
  console.error('Error starting server:', error);
}

```

Gambar 5 API untuk Mengambil Data Berdasarkan ID Pengguna

3.4. Pengujian

Pengujian API dilakukan menggunakan Postman untuk memastikan setiap endpoint berfungsi dengan baik. Hasil pengujian API register menunjukkan bahwa API berhasil dijalankan tanpa error, sehingga memastikan pengguna dapat mendaftar dengan lancar, seperti yang ditunjukkan pada Gambar 6.



Gambar 6 Pengujian API Register

3.5. Deployment



Proses deployment aplikasi dilakukan melalui Google Cloud Console menggunakan layanan App Engine. Langkah pertama dalam proses ini adalah mengunduh kode sumber dari GitHub ke Google Cloud, yang terlihat pada Gambar 7.



Gambar 7 Proses Cloning Repositori dari GitHub ke Google Cloud

Setelah itu, konfigurasi App Engine dilakukan dengan memilih region server untuk menjalankan aplikasi, seperti pada Gambar 8.

```
fatimachya69@cloudshell: (silent-turbine-410521) $ gcloud app create
You are creating an app for project [silent-turbine-410521].
WARNING: Creating an App Engine application for a project is irreversible and the region
cannot be changed. More information about regions is at
https://cloud.google.com/appengine/docs/locations/.
Please choose the region where you want your App Engine application located:

[1] asia-east1 (supports standard and flexible)
[2] asia-east2 (supports standard and flexible and search_api)
[3] asia-northeast1 (supports standard and flexible and search_api)
[4] asia-northeast2 (supports standard and flexible and search_api)
[5] asia-northeast3 (supports standard and flexible and search_api)
[6] asia-south1 (supports standard and flexible and search_api)
[7] asia-southeast1 (supports standard and flexible)
[8] asia-southeast2 (supports standard and flexible and search_api)
[9] australia-southeast1 (supports standard and flexible and search_api)
[10] europe-central2 (supports standard and flexible)
[11] europe-west (supports standard and flexible and search_api)
[12] europe-west2 (supports standard and flexible and search_api)
[13] europe-west3 (supports standard and flexible and search_api)
[14] europe-west4 (supports standard and flexible and search_api)
[15] northamerica-northeast1 (supports standard and flexible and search_api)
[16] southamerica-east1 (supports standard and flexible and search_api)
[17] us-central (supports standard and flexible and search_api)
[18] us-east1 (supports standard and flexible and search_api)
[19] us-east4 (supports standard and flexible and search_api)
[20] us-west1 (supports standard and flexible)
[21] us-west2 (supports standard and flexible and search_api)
[22] us-west3 (supports standard and flexible and search_api)
[23] us-west4 (supports standard and flexible and search_api)
[24] cancel
Please enter your numeric choice: 2
Creating App Engine application in project [silent-turbine-410521] and region [asia-east2]...done.
Success! The app is now created. Please use 'gcloud app deploy' to deploy your first app.
```

Gambar 8 Konfigurasi Region Server di App Engine

Proses upload aplikasi ke server App Engine ditunjukkan pada Gambar 9, yang memerlukan waktu sesuai dengan ukuran file dan koneksi jaringan.

```
fatimachya69@cloudshell: (silent-turbine-410521) $ cd Bangkit-CC
fatimachya69@cloudshell:~/Bangkit-CC [silent-turbine-410521] $ gcloud app deploy
Services to deploy:
description: [/home/fatimachya69/Bangkit-CC/app.yaml]
source: [/home/fatimachya69/Bangkit-CC]
target project: [silent-turbine-410521]
target service: [default]
target version: [20240130113107]
target url: https://[silent-turbine-410521].df.a.appspot.com/
target service account: [silent-turbine-410521@appspot.gserviceaccount.com]

Do you want to continue (Y/n)? y
Beginning deployment of service [default]...
Uploading 18 files to Google Cloud Storage
68
119
179
229
289
339
399
449
509
569
619
679
729
789
839
899
949
1009
1069
File upload done.
Updating service [default]...working...
```

Gambar 9 Proses Upload Aplikasi ke App Engine

Setelah deployment berhasil, aplikasi dapat diakses melalui alamat URL yang telah ditentukan, seperti pada Gambar 10

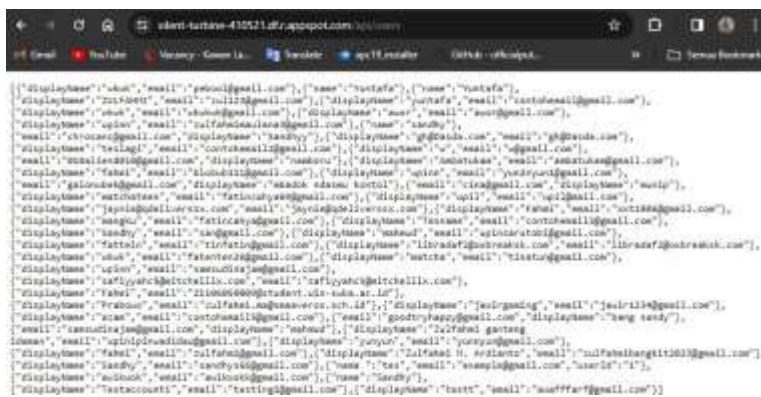
```
Setting traffic split for service [default]...done.
Deployed service [default] to [https://silent-turbine-410521.df.r.appspot.com]

You can stream logs from the command line by running:
$ gcloud app logs tail -a default

To view your application in the web browser run:
$ gcloud app browse
fatincarya6@cloudshell:~/Bangkit-CC [silent-turbine-410521]$ gcloud app browse
Did not detect your browser. Go to this link to view your app:
https://silent-turbine-410521.df.r.appspot.com
fatincarya6@cloudshell:~/Bangkit-CC [silent-turbine-410521]$
```

Gambar 10 Alamat URL Aplikasi yang Telah Berhasil Di-deploy

Tampilan akhir aplikasi yang berhasil di-deploy menggunakan Google App Engine ditunjukkan pada Gambar 11.



Gambar 11 Tampilan Akhir Aplikasi setelah Deployment

Hasil penelitian ini menunjukkan bahwa aplikasi FED dapat memberikan solusi yang efisien dan efektif bagi para peternak dalam mendeteksi kesuburan telur. Dengan memanfaatkan teknologi CNN dan GCP, aplikasi ini mampu meningkatkan produktivitas dan efisiensi operasional di sektor peternakan. Analisis ROC menunjukkan bahwa model yang dikembangkan memiliki akurasi tinggi, sementara pengujian API membuktikan bahwa integrasi sistem berjalan lancar. Deployment menggunakan App Engine memberikan fleksibilitas bagi pengguna untuk mengakses aplikasi secara online.

Keberhasilan penelitian ini tidak hanya memberikan manfaat praktis bagi peternak tetapi juga menunjukkan potensi teknologi berbasis cloud dalam mendukung inovasi di sektor pertanian. Namun, tantangan seperti kebutuhan dataset yang lebih besar dan optimisasi model dapat menjadi fokus penelitian selanjutnya.

4. KESIMPULAN

Berdasarkan hasil dan pembahasan yang telah dilakukan pada penelitian ini, dapat disimpulkan bahwa Penelitian dan pengembangan aplikasi Fertile Egg Detector (FED) mewakili langkah-langkah komprehensif untuk meningkatkan efisiensi dalam industri peternakan telur di Indonesia. Proyek ini menggabungkan keahlian dari divisi Machine Learning, Cloud Computing, dan Mobile Development untuk menciptakan solusi yang dapat memberikan dampak positif. Melalui teknologi Convolutional Neural Network (CNN) dan Google Cloud Platform (GCP), FED mampu mendeteksi keberhasilan telur secara otomatis, membantu peternak dalam pengelolaan produksi secara efisien.

Implementasi aplikasi FED melibatkan perancangan arsitektur aplikasi, otentikasi pengguna, pengelolaan dataset, dan pengujian fungsionalitas aplikasi. Selain itu, analisis biaya dan evaluasi dampak ekonomi dilakukan untuk memahami keefektifan solusi. Dengan keberhasilan deployment menggunakan Google App Engine, aplikasi ini dapat diakses melalui berbagai platform, memberikan kemudahan dalam monitoring dan pengambilan keputusan.

Dalam konteks industri peternakan telur di Indonesia pada tahun 2023, FED diharapkan dapat memberikan kontribusi positif terhadap peningkatan produktivitas, efisiensi, dan kesejahteraan peternak. Melalui penerapan teknologi terkini, penelitian ini tidak hanya menyajikan solusi teknis, tetapi juga memberikan dorongan terhadap inovasi dalam sektor pertanian, sesuai dengan kebutuhan dan perkembangan masyarakat.

REFERENSI

- [1] V. Goyal, A. Yadav, and R. Mukherjee, "A Literature Review on the Role of Internet of Things, Computer Vision, and Sound Analysis in a Smart Poultry Farm," *ACS Agricultural Science & Technology*, vol. 4, no. 4, pp. 368–388, 2024, doi: 10.1021/acsagscitech.3c00467.
- [2] G. Li, Y. Xu, Y. Zhao, Q. Du, and Y. Huang, "Evaluating convolutional neural networks for cage-free floor egg detection," *Sensors (Switzerland)*, vol. 20, no. 2, pp. 1–17, 2020, doi: 10.3390/s20020332.
- [3] J. Astill, R. A. Dara, E. D. G. Fraser, B. Roberts, and S. Sharif, "Smart poultry management: Smart sensors, big data, and the internet of things," *Comput Electron Agric*, vol. 170, p. 105291, 2020, doi: <https://doi.org/10.1016/j.compag.2020.105291>.
- [4] R. O. Ojo, A. O. Ajayi, H. A. Owolabi, L. O. Oyedele, and L. A. Akanbi, "Internet of Things and Machine Learning techniques in poultry health and welfare management: A systematic literature review," *Comput Electron Agric*, vol. 200, p. 107266, 2022, doi: <https://doi.org/10.1016/j.compag.2022.107266>.
- [5] N. Nurdyanto, P. S. Informatika, F. Komunikasi, D. A. N. Informatika, and U. M. Surakarta, "Aplikasi cloud computing untuk server berbasis django dengan google app engine," p. 2, 2018, [Online]. Available: <http://ejournal.umm.ac.id/index.php/jp2sd%0Ahari>
- [6] S. R. Vinta and R. Peeriga, "Rainfall Prediction using XGB Model with the Australian Dataset," *EAI Endorsed Transactions on Energy Web*, vol. 11, pp. 1–4, 2024, doi: 10.4108/ew.5386.
- [7] O. Badrakh and L. Choimaa, "Air quality predictions of Ulaanbaatar using machine learning approach," *Proceedings of Science*, vol. 378, 2021, doi: 10.22323/1.378.0012.
- [8] Y. Ashibani and Q. H. Mahmoud, "A Multi-Feature User Authentication Model Based on Mobile App Interactions," *IEEE Access*, vol. 8, pp. 96322–96339, 2020, doi: 10.1109/ACCESS.2020.2996233.
- [9] H. Al Bazar, A. Abdel-Wahab, M. Alshar'e, and A. Abualkishik, "An Adaptive Two-Factor Authentication Scheme Based on the Usage of Schnorr Signcryption Algorithm," *Informatica (Slovenia)*, vol. 47, no. 5, pp. 159–172, 2023, doi: 10.31449/inf.v47i5.4627.
- [10] S. Y. Yerima, M. K. Alzaylaee, A. Shajan, and P. Vinod, "Deep learning techniques for android botnet detection," *Electronics (Switzerland)*, vol. 10, no. 4, pp. 1–17, 2021, doi: 10.3390/electronics10040519.
- [11] A. R. Patta, N. Funabiki, M. Kuribayashi, and Y. W. Syaifudin, "An Implementation of Solution Progress Monitoring Function in Android Programming Learning Assistance System," *International Journal of Information and Education Technology*, vol. 13, no. 10, pp. 1597–1603, 2023, doi: 10.18178/ijiet.2023.13.10.1967.
- [12] A. Ehsan, M. A. M. E. Abuhaliqa, C. Catal, and D. Mishra, "RESTful API Testing Methodologies: Rationale, Challenges, and Solution Directions," *Applied Sciences (Switzerland)*, vol. 12, no. 9, 2022, doi: 10.3390/app12094369.
- [13] J. Yang, "Analyzing the Features , Usability , and Performance of Deploying a Containerized Mobile Web Application on Serverless Cloud Platforms," 2024.



- [14] M. A. Kamal*, H. W. Raza, M. M. Alam, and M. M. Su'ud*, "Highlight the Features of AWS, GCP and Microsoft Azure that Have an Impact when Choosing a Cloud Service Provider," *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 8, no. 5, pp. 4124–4132, 2020, doi: 10.35940/ijrte.d8573.018520.
- [15] Y. Wei, M. Gao, J. Xiao, C. Liu, Y. Tian, and Y. He, "Research and Implementation of Traffic Sign Recognition Algorithm Model Based on Machine Learning," *Journal of Software Engineering and Applications*, vol. 16, no. 06, pp. 193–210, 2023, doi: 10.4236/jsea.2023.166011.

